A Microservice-oriented AI Automation Framework for Supporting Single-cell Downstream Analysis

Hong Qing Yu
School of Computer Science and Engineering
University of Derby
Derby, United Kingdom
Email: h.yu@derby.ac.uk

Ali Kermanizadeh
School of Human Sciences
University of Derby
Derby, United Kingdom
Email: a.kermanizadeh@derby.ac.uk

Sam O'Neill
School of Computer Science and Engineering
University of Derby
Derby, United Kingdom
Email: s.oneill@derby.ac.uk

Oyetola Florence Idowu
School of Computer Science and Engineering
University of Derby
Derby, United Kingdom
Email: o.idowu4@unimail.derby.ac.uk

***Abstract*—Single-cell analysis has real potential for reshaping the future of biomedical research allowing for a better understanding of the natural properties of both healthy and diseased tissues that, in turn, allow for better opportunities for overcoming current challenges in drug discovery, diagnostics and prognostics. A large body of research in this field produces large quantities of data. Merging with fast-developed Machine learning (ML) and Artificial Intelligent (AI) algorithms would allow single-cell analysis research to be conducted more efficiently and accurately than currently possible. Therefore, there has been a surge of ML and AI developments for the whole life cycle of the downstream single-cell analysis process. However, there is a limitation to reusing, exchanging, sharing, applying the most advanced technologies, and automating the experimental environments and outcomes in cross-disciplinary collaborative research. This paper presents an automation framework to address these limitations and shows how AI and ML research can contribute to biomedical automation and control. Moreover, the real-world case will be evaluated to demonstrate the prototype implementation at the end of the paper.**

***Index Terms*—Single-cell analysis, RNA-Seq, Machine Learning, AI Automation, Downstream Analysis, Knowledge Graph**

I. Introduction

Since the turn of the century, bio-informatics research underwent a technological revolution allowing the generation of single-cell genomics data. One of the most common and fast-developed techniques is single-cell Ribonucleic acid sequencing (RNA-seq) which represents quantification and profiling of the changing gene expressions in single cells and how they differ across thousands of cells within a heterogeneous sample [1]. Single-cell genomics offers a unique opportunity to allow joint multidisciplinary research across multiple types of datasets [2]. The generation of large datasets has its own challenges that include but are not limited to sharing and reusing computational methods, algorithms, pipelines, and other resources. In addition, the automatic creation of a new analysis process composing existing research outcomes to solve a given task is crucial to fast-track the research output, and dissemination [3], [4]. Our research aims to provide an automatic AI framework that could address the challenges in single-cell downstream data analysis including data annotation, data quality control, data normalisation, data dimensional reduction, and data analysis [5]. In data analysis, cutting-edge methods such as Machine learning algorithms can be applied for more efficient and meaningful classification, clustering, segmentation, and prediction [6]. With the development of Deep Neural Networks (DNN), models have been developed for single-cell analysis which creates further technology burdens to reuse and deploy the solutions by non-programming-focused researchers [7].

The paper has two main contributions to make the downstream data analysis more shareable and reusable and automatically provide new solutions creatively:

1) The developed analysis methods can be shared and reused as semantic microservices registered in the framework with annotations. Therefore, the microservices can be dynamically allocated and composed later to perform new and suitable tasks.
2) A semantic knowledge representation and learning framework is developed. The framework can learn the context knowledge of the tasks and dynamically search, compose, and run the registered microservices to complete future (similar) requested tasks.

In Section 2, current single-cell downstream analysis methods, pipelines and tools will be discussed. In Section 3, the proposed automatic AI framework will be introduced and explained. In Section 4, the use case will be evaluated with visualisation results. In Section 5, the conclusion will be provided with future work discussions.

 ISBN: 978-1-68558-058-2

II. Single-cell Downstream Analysis Methods, Pipeline and Tools

Single-cell RNA sequencing (scRNA-seq) data can be analysed using big data processing and machine learning technologies. These technologies enable the investigation of complex biological questions at the single-cell level that can result in more direct and physiologically relevant and urgently needed observations and understandings. [8] Briefly, the single-cell RNA sequencing process can be broken down into several sequential stages [9], [10]:

- Single-cell isolation can occur from healthy or diseased tissue from any organ of interest. The cell is lysed, which is followed by RNA isolation, purification and quantification, RNA fragmentation, and cDNA generation (reverse transcription uses primers are used to initiate binding to its complementary sequences on the RNA template and serves as a starting point for the synthesis of a new strand ensuring the preservation of original cellular information [11].
- Library-based amplified and tagged cDNA from each cell can be pooled and sequenced [12]. At the end of this stage, large quantities of raw data are prepared.
- Computation algorithms focused on downstream data analysis. The analysis can be used for data reprocessing (quality control), normalisation, feature extraction to clustering, sub-population identification, and understanding gene expression differences across different contexts [13].

In the downstream analysis, quality control refers to the identification of low-quality cells (culture of single cells in droplets, plates, or microfluidic devices can be technically challenging, which can result in the cell undergoing biological stress or even death. On occasion, data from more than one cell can be captured, or even data is recorded where no cell is present at all these undesired variances from the experimental norm are referred to as "low-quality"). This is generally achieved by analyzing the raw data, which is the most critical step for downstream analysis and results in interpretation [14]. Numerous tools (i.e., FASTQC, Kraken, and RNA-SeQC) [15] with different metrics have been developed for quality control, such as the total number of reads detected per cell and the total number of unique genes detected in each sample. These tools provide interfaces allowing researchers to upload the raw data to visualise and process the data with specified metrics and thresholds. For example, "External RNA Controls Consortium Spike-In Controls" can be used to provide information on the sensitivity, specificity, and dynamic range of the datasets by measuring abundances and ratios between spike-in RNAs and endogenous RNAs. This ratio can estimate the total amount of RNA in the captured cells.

The normalisation step is also essential to ensure accurate and reliable downstream analysis. Normalisation approaches not only account for sequencing depth but also account for library sizes. Library sizes vary for many reasons, including natural differences in cell size, variation of RNA capture, and variation in the efficiency of PCR amplification used to generate enough RNA to create the sequencing library. There are two main approaches to this correction. Many methods use a simple linear scaling to adjust counts such that each cell (row) has about the same total library size. Examples include converting to counts per million (CPM) and closely related methods such as scran. While simple, these approaches do a reasonable job of correcting for differences in library size. Other methods are more complex and helpful in dealing with complex sources of unwanted variation (e.g., for highly heterogeneous populations of cells with different sizes) [18]. The extra function of Removing Unwanted Variation (RUV) is proposed by [19], which adjusts for nuisance technical effects by performing factor analysis on suitable sets of control genes (e.g., ERCC spike-ins) or samples (e.g., replicate libraries).

The feature extraction step focuses on dimensionality reduction which will increase the analysis interoperability (a large set of variables and return a smaller set of components that still contain most of the information in the original dataset) and decrease the analysis complexity. The most popular algorithms are tSNE (t-Distributed Stochastic Neighbour Embedding) [16] and UMAP (Uniform Approximation and Projection) [17]. tSNE combines dimensionality reduction (e.g., PCA) with random walks on the nearest-neighbour network to map high-dimensional data to a 2-dimensional space. UMAP is a non-linear and nondeterministic dimensionality reduction method that requires the random seed to ensure reproducibility. While tSNE optimises for local structure, UMAP tries to balance the preservation of local and global structure [18].

The final step is clustering or classification analysis to understand differences in gene expression. Machine learning algorithms such as K-means, logistic regression, support vector machines, random forests, and neural networks can be applied [20]. Asking bioinformatics scientists and researchers to code solutions step-by-step is not helpful or efficient. Therefore, many pipeline tools are developed such as Scater [13] (a pipeline R library to support researchers to have a full programming package on the downstream analysis). Most recently the devCellPy [21] (a Python tool that enables automated prediction of cell types across complex annotation hierarchies) and the R code pipeline [22] and scWizard (a web application tool for specifying the template of the downstream analysis) [23] have been utilised.

However, these pipelines or the full stack of development packages still require high-level knowledge of coding with a specific programming language. Therefore, reusing, exchanging, and sharing the experimental environments and research outcomes in cross-disciplinary collaboration are very challenging [24]. In addition, comparing many different algorithms to find the best one for processing data and analysis is extremely time-consuming and requires very specialised knowledge [4], [25]. For instance, independent research groups have not extensively used Deep Learning (DL) algorithms in their biological studies due to a lack of expertise and robust computation resources [26]. Finally, cutting-edge technologies will be delayed in application because of the high bar of programming. Thus, an automated framework will be the key

 ISBN: 978-1-68558-058-2

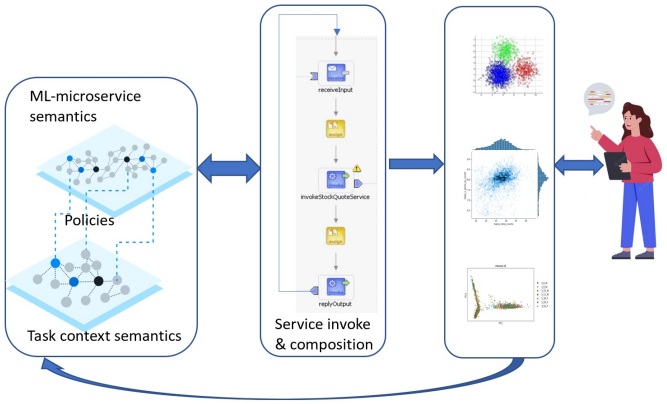

Fig. 1. Platform Architecture

to addressing these challenges [27]. The paper proposes a framework based on advanced knowledge graphs, microservices, and knowledge-based AI technologies. This framework aims to separate algorithm development and analysis tasks for different disciplinary researchers. In the end, the platform will be able to automatically create a downstream analysis pipeline for bio-scientists.

III. The Proposed Framework Architecture and Prototype Development

The proposed framework consists of three layers as shown in Figure 1:

- Semantic layer that provides metadata descriptions of the analysis task context, policies, and microservices registered in the platform. The task context simply specifies the inputs, analysis task, domain, and desired output data. Each microservice only has one function to do a specific task that can be performed in a different stage of the analysis. The stages are normally data translation and loading, data normalisation, data processing, and data analysis.
- Automation layer that selects and coordinates microservices to create a pipeline dynamically for completing the task. The task can be a simple task handled by one single microservice. However, most of the functions need to compose several services together dynamically. The automation performs selections and coordination through semantic reasoning and context-based reinforcement learning.
- The output and interaction layer that releases the results of the analysis process which can be single microservice outputs or a pipeline's outcomes produced at each stage of the analysis. The researchers can interact with the system at any stage to suggest ratings and provide further context. The interaction data will be fed back to the semantic layers to enhance the policies.

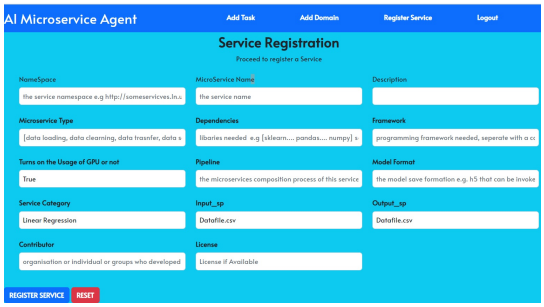

Fig. 2. Microservice registration interface

A. Semantic Knowledge Graph schema

There are four types of schema defined using OWL (Web Ontology Language) [28]. OWL is a knowledge graph ontology design standard.

- The microservice ontology contains namespace, identity, input data, output data, domain, dependency, purpose, and ML category, as well as other properties to able, create the dynamic invocation settings (see Figure 2).
- The task context ontology includes task identity, task input data, task desired output data, and domain.
- The solution pipeline ontology defines a workflow created by having one or composing multiple microservices to achieve the desired output.
- The policies ontology defines scores for each service against each task context with a default score of 0. This means that the platform will learn the procedures and try to remember the best solution by evaluating all possible microservices or combinations.

B. Dynamic microservices selection and composition

We implemented five engines to deal with microservices selection, invocation, optimisation, composition, and policy learning. The process of dynamically creating a solution is represented in Figure 3.

The analysis task context is the input to trigger the automatic process and the search engine then starts to create a SPARQL (a knowledge graph query language) [29] query to match semantic compatible microservices that can produce the desired output. The outcomes from the search engine can be either, a single result (a compatible single microservice or pipeline found), multiple results, or no result. The first situation is simply to complete the task and feed the results to the task requester, then the requester can provide the score as feedback to the policy engine. For the second situation, a queue containing all possible compatible results is created to allow the invocation engine to invoke them one by one to feed the output to the optimisation engine that can select the best solution through quality and performance evaluation as well as run-time feedback from the task requester. For the no-matching condition, the composition engine starts to try to compose a sequence of microservice services to complete the task by relaxing some context-searching criteria. The composition is

 ISBN: 978-1-68558-058-2

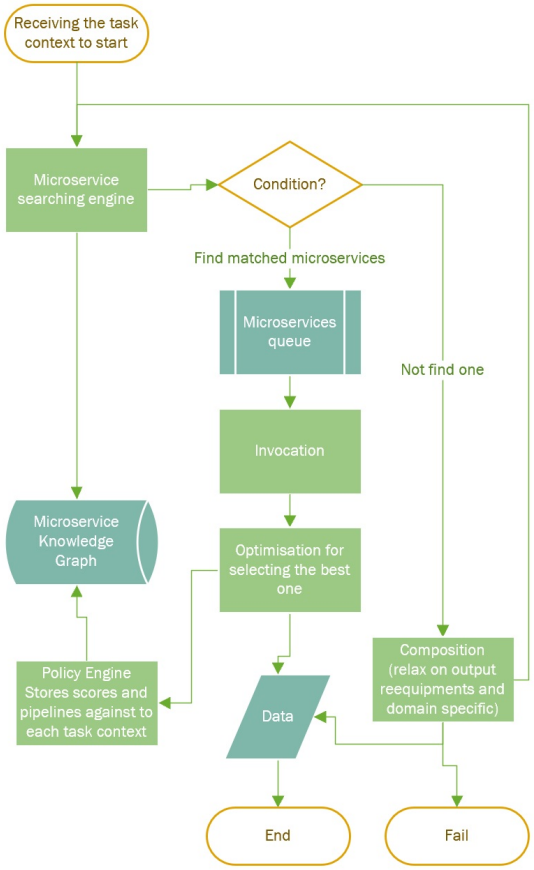

Fig. 3. Automatic data processing and analysis engines

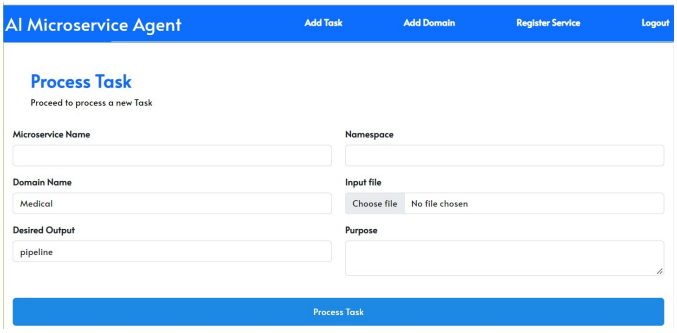

Fig. 4. Task specification interface

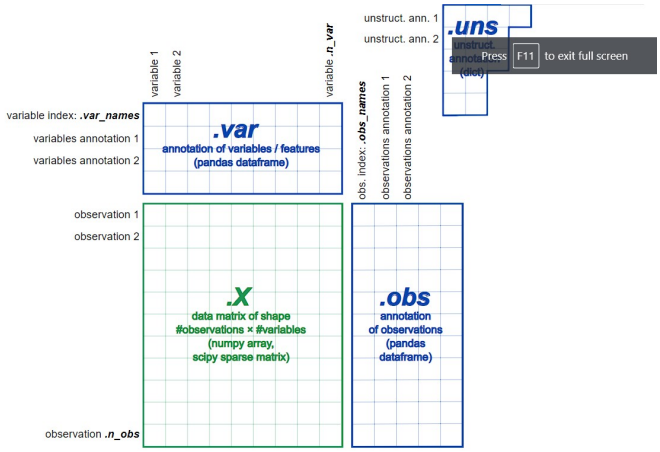

Fig. 5. AnnData Structure

completed when the relaxed criteria are no longer required after one or some other microservices produced a mediation outcome to fill the gap. The composition process may also succeed or fail. The results will be recorded through the policy engine to remember the successful microservice, pipeline of them, or no solution outcomes against a task context. In the future, a similar task will be performed faster by reusing the whole solution or part of the solution.

C. Interfaces for interaction

The web interfaces are developed in the framework to support interactions. AI microservices can be registered and shared by researchers or AI engineers (see Figure 2). Researchers can then use the task interface to specify the analysis task for asking the framework to provide the best solution based on the knowledge about the registered microservices (see Figure 4). Whenever a step is completed in the process towards the goal, the output can be presented to the researcher for immediate feedback to support the next steps or overall input to the solution (see Figure 7). All the feedback will contribute to the task context policies that improve the output of the framework.

IV. Use Case Demonstration

In this section, we use a clustering analysis case study to highlight how the proposed framework can solve a real-world downstream single-cell data analysis task. The clustering analysis task works on a mouse brain single-cell RNASeq dataset. The dataset is publicly available through a workshop tutorial at [30]. There are five sequential processing and analysis steps:

1) **Data semantic transforming and loading:** For instance, applying AnnData structure [31], where AnnData stores observations (samples) of variables/features in the

```
@prefix ns1: <http://aimicroservice.derby.ac.uk/> .
ns1:genQualityControl a ns1:Bioinfomatic_genQualityControl ;
    ns1:category ns1:Bioinformatics ;
    ns1:contributor <https://www.derby.ac.uk/staff/hongqing-yu/> ;
    ns1:dependency "matplotlib"@en,
        "pandas"@en,
        "scanpy"@en,
        "seaborn"@en ;
    ns1:description "https://scanpy.readthedocs.io/en/stable/"@en ;
    ns1:formate "py"@en ;
    ns1:framwork "annData_qualityControl"@en ;
    ns1:input [ ns1:paramter [ ns1:iocategory ns1:brain_raw ;
                    ns1:iodatatype ns1:h5ad ;
                    ns1:pid "0"@en ] ] ;
    ns1:licence <https://en.wikipedia.org/wiki/Free-software_license> ;
    ns1:output [ ns1:paramter [ ns1:iocategory ns1:brain_qc ;
                    ns1:iodatatype ns1:h5ad ;
                    ns1:pid "0"@en ] ] ;
    ns1:uri ns1:genQualityControl .
```

Fig. 6. Quality control microservice semantic description

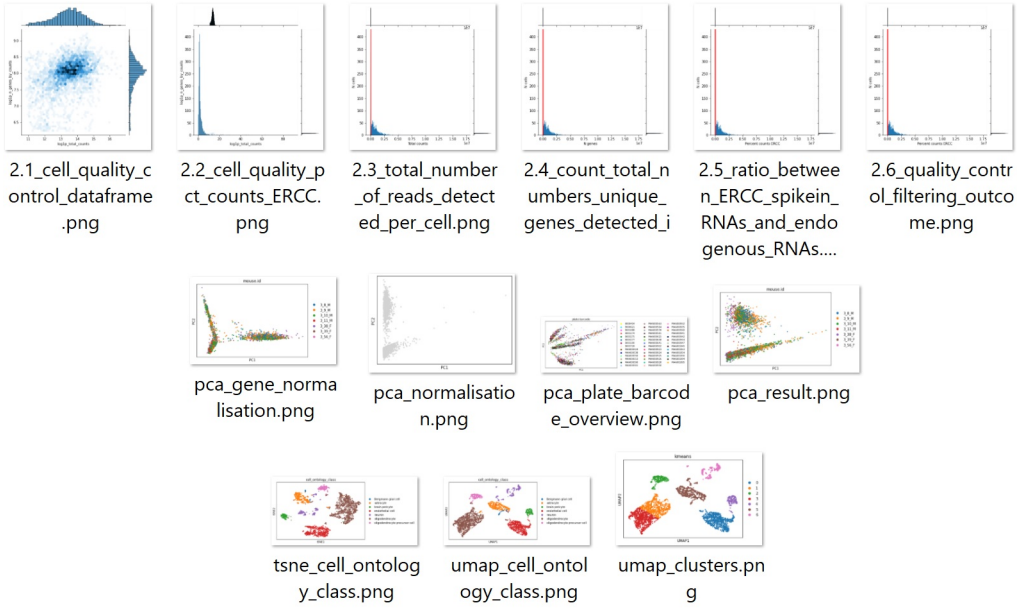

Fig. 7. Visualisations of analysis steps

rows of a matrix (see Figure 5).

2) **Data quality control:** This aims to find and remove the poor quality cell observation data which were not detected in the previous processing of the raw data. The low-quality cell data may potentially introduce analysis noise and obscure the biological signals of interest in the downstream analysis.
3) **Data normalisation:** Dimensionality reduction and scaling of the data. Biologically, dimensional reduction is valuable and appropriate since cells respond to their environment by turning on regulatory programs that result in the expression of modules of genes. As a result, gene expression displays structured co-expression, and dimensionality reduction by the algorithm such as principle component analysis can group those co-varying genes into principle components, ordered by how much variation they explained.
4) **Data feature embedding:** Further dimensionality reduction using advanced algorithms, such as t-SNE and UMAP. They are powerful tools for visualising and understanding big and high-dimensional datasets.
5) **Clustering analysis:** Group cells into different clusters based on the embedded features.

Based on the above five steps, we developed seven microservices which include AnnData loading, data quality control, normalisation services (PCA+CPM algorithm), two feature embedding services (t-SNE and UMAP), and clustering services (K-mean clustering and Louvain graphical clustering algorithms).

The microservices were semantically registered into the framework through the interface. Figure 6 depicts an example of quality control microservice semantic description in the knowledge graph repository.

With all the microservices registered, researchers can start expressing the analysis task to stop, interact and provide feedback at any stage during the process of automatically creating the solution. The researchers can also see visualisations of outputs produced by different steps (see Figure 7). Therefore, researchers can provide preferences for selecting microservices if there are options.

A realistic example is that a researcher can specify a clustering task applied to the mouse brain single-cell RNASeq dataset. The framework will first try to see if a single microservice can complete this task. The answer is 'no' because no semantic-matched microservice can take the RNASeq CSV input and provide the clustering output. At this juncture, the microservice that can take the RNASeq CSV will be invoked to process the data into the next step with the output of AnnData. If there are multiple choices in the composition sequence, all possibilities will be invoked to run unless the previous knowledge in the policies has a priority. The possibilities have multiple solutions at the end for researchers to analyse for giving professional feedback to the system. The feedback will help greatly with the knowledge graph policies. For example, suppose the researcher gives feedback to the system that UMAP is the better embedding method than t-SNE but has no priority on the clustering methods. In that case, the framework will produce two possible clustering results shown in Figure 8.

V. Conclusion and Future Work

The potential of single-cell research, along with ML and AI technologies, to address critical biomedical and disease classification and clustering issues and facilitate comprehension in the near future is substantial [32], [33]. Our research identified the current limitations of reusing, exchanging, sharing,

 ISBN: 978-1-68558-058-2

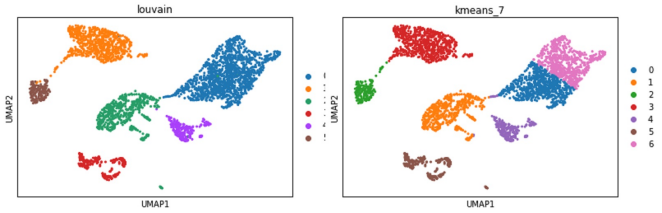

Fig. 8. Two clustering outcomes from automatic process

applying the most advanced technologies and automating the experimental environments and outcomes in cross-disciplinary research collaboration. Therefore, we proposed an AI automation framework that can semantically share implemented ML algorithms or AI models for a general purpose. Possible solutions can be automatically generated through interactions with researchers.

Our future work will increase the general informatics purposed ML algorithms and AI model developments and registration. More downstream analysis tasks can be tested and evaluated.

REFERENCES

[1] Perkel, J., (2021). Single-cell analysis enters the multiomics age. Nature. 595. 614-616. https://doi.org/10.1038/d41586-021-01994-w.

[2] Stuart, T. and Satija, R., Integrative single-cell analysis. Nat Rev Genet 20, 257–272 (2019). https://doi.org/10.1038/s41576-019-0093-7

[3] Pasquini, G. Eduardo Rojo Arias, J., Schäfer, P. and Busskamp, V., Automated methods for cell type annotation on scRNA-seq data, Computational and Structural Biotechnology Journal, Volume 19, 2021, Pages 961-969, ISSN 2001-0370, https://doi.org/10.1016/j.csbj.2021.01.015.

[4] Yuan, G. C., Cai, L., Elowitz, M. et al., Challenges and emerging directions in single-cell analysis. Genome Biol 18, 84 (2017). https://doi.org/10.1186/s13059-017-1218-y

[5] Chen, G., Ning, B. and Shi, T., Single-Cell RNA-Seq Technologies and Related Computational Data Analysis. Front Genet. 2019 Apr 5;10:317. doi: 10.3389/fgene.2019.00317. PMID: 31024627; PMCID: PMC6460256.

[6] Raimundo, F., Meng-Papaxanthos, L., Vallot, C. and Vert, J., Machine learning for single-cell genomics data analysis, Current Opinion in Systems Biology, Volume 26, 2021, Pages 64-71, ISSN 2452-3100, https://doi.org/10.1016/j.coisb.2021.04.006.

[7] Ma, Q., Xu, D., Deep learning shapes single-cell data analysis. Nat Rev Mol Cell Biol 23, 303–304 (2022). https://doi.org/10.1038/s41580-022-00466-x

[8] Zhang, Z. and et al., Critical downstream analysis steps for single-cell RNA sequencing data, Briefings in bioinformatics (2021): n. pag.

[9] Haque, A., Engel, J., Teichmann and S. A. et al. A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications. Genome Med 9, 75 (2017). https://doi.org/10.1186/s13073-017-0467-4

[10] Slovin, S., Carissimo, A., Panariello, F., Grimaldi, A., Bouché, V., Gambardella, G. and Cacchiarelli, D., Single-Cell RNA Sequencing Analysis: A Step-by-Step Overview. Methods Mol Biol. 2021, 2284:343-365.

[11] Kivioja, T., Vähärautio, A., Karlsson, K., Bonke, M., Enge, M., Linnarsson, S. and Taipale, J., Counting absolute numbers of molecules using unique molecular identifiers. Nat Methods. 2011 Nov 20;9(1):72-4. doi: 10.1038/nmeth.1778. PMID: 22101854.

[12] van Dijk, E. L., Auger, H., Jaszczyszyn, Y. and Thermes, C. Ten years of next-generation sequencing technology. Trends Genet. 2014;30:418–26.

[13] McCarthy, D. J., Campbell, K. R., Lun, A. T. L. and Wills, Q. F., Scater: pre-processing, quality control, normalisation and visualization of single-cell RNA-seq data in R, Bioinformatics, Volume 33, Issue 8, 15 April 2017, Pages 1179–1186, https://doi.org/10.1093/bioinformatics/btw777

[14] Li, X., Nair, A., Wang, S. and Wang, L., (2015). Quality Control of RNA-Seq Experiments. In: Picardi, E. (eds) RNA Bioinformatics. Methods in Molecular Biology, vol 1269. Humana Press, New York, NY. https://doi.org/10.1007/978-1-4939-2291-8:8

[15] Bacher, R. and Kendziorski, C. Design and computational analysis of single-cell RNA-sequencing experiments. Genome Biol 17, 63 (2016). https://doi.org/10.1186/s13059-016-0927-y

[16] N. Rogovschi, J. Kitazono, N. Grozavu, T. Omori and S. Ozawa, t-Distributed stochastic neighbor embedding spectral clustering, 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 1628-1632, doi: 10.1109/IJCNN.2017.7966046.

[17] McInnes, L. and Healy, J. (2018). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. ArXiv, abs/1802.03426.

[18] Sharma, A., Single-cell RNASeq data from Mouse Brain, 2021, https://www.kaggle.com/datasets/aayush9753/singlecell-rnaseq-data-from-mouse-brain/discussionSingle-cell

[19] Risso, D., Ngai, J., Speed, T. and et al., Normalisation of RNA-seq data using factor analysis of control genes or samples. Nat Biotechnol 32, 896–902 (2014).

[20] Le, H., Peng, B., Uy, J., Carrillo, D., Zhang, Y. and Aevermann, B. D., Scheuermann RH. Machine learning for cell type classification from single nucleus RNA sequencing data. PLoS One. 2022 Sep 23;17(9):e0275070. doi: 10.1371/journal.pone.0275070. PMID: 36149937; PMCID: PMC9506651.

[21] Galdos, F. X., Xu, S., Goodyer, W. R. and et al., devCellPy is a machine learning-enabled pipeline for automated annotation of complex multilayered single-cell transcriptomic data. Nat Commun 13, 5271 (2022). https://doi.org/10.1038/s41467-022-33045-x

[22] Chen, Y., Pal, B., Lindeman, G.J. and et al. R code and downstream analysis objects for the scRNA-seq atlas of normal and tumorigenic human breast tissue. Sci Data 9, 96 (2022). https://doi.org/10.1038/s41597-022-01236-2

[23] Wei, J., Xie, Q., Qu, Y., Huang, G., Chen, Z. and Du, H., scWizard: A web-based automated tool for classifying and annotating Single-cells and downstream analysis of single-cell RNA-seq data in cancers, Computational and Structural Biotechnology Journal, Volume 20, 2022, Pages 4902-4909, ISSN 2001-0370, https://doi.org/10.1016/j.csbj.2022.08.028.

[24] Hodzic, E., Single-cell analysis: Advances and future perspectives. Bosn J Basic Med Sci. 2016 Nov 10;16(4):313-314. doi: 10.17305/bjbms.2016.1371. PMID: 27320288; PMCID: PMC5136769.

[25] Menon, S., Lui, V.C.H. and Tam, P.K.H., 2021. Bioinformatics tools and methods to analyze Single-cell RNA sequencing data. International Journal of Innovative Science and Research Technology,(IJISRT), 6(8), pp.282-288.

[26] Ma, Q., Xu, D. Deep learning shapes single-cell data analysis. Nat Rev Mol Cell Biol 23, 303–304 (2022). https://doi.org/10.1038/s41580-022-00466-x

[27] Saliba A. E., Westermann A. J., Gorski S. A., Vogel J., Single-cell RNA-seq: advances and future challenges. Nucleic Acids Res. 2014 Aug;42(14):8845-60. doi: 10.1093/nar/gku555. Epub 2014 Jul 22. PMID: 25053837; PMCID: PMC4132710.

[28] Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D., Patel-Schneijder, P. and Stein, L. A., (2004). OWL Web Ontology Language Reference (Recommendation). World Wide Web Consortium (W3C).

[29] Prud'hommeaux, E. and Seaborne, A., SPARQL Query Language for RDF , W3C Recommendation, 2008.

[30] Luecken, Malte D. and Theis, Fabian J., Current best practices in single-cell RNA-seq analysis: a tutorial, Journal of Molecular Systems Biology, Volume 15, 2019, https://doi.org/10.15252/msb.20188746. Dataset download url: https://www.singlecellcourse.org/

[31] Cannoodt, R., (2022). anndata: 'anndata' for R. https://anndata.dynverse.org, https://github.com/dynverse/anndata.

[32] Chen, Y., Pal, B., Lindeman, G.J. and et al., R code and downstream analysis objects for the scRNA-seq atlas of normal and tumorigenic human breast tissue. Sci Data 9, 96 (2022). https://doi.org/10.1038/s41597-022-01236-2

[33] Dohmen, J., Baranovskii, A., Ronen, J. and et al., Identifying tumor cells at the single-cell level using machine learning. Genome Biol 23, 123 (2022). https://doi.org/10.1186/s13059-022-02683-1