Single-cell RNAseq integration: methods and challenges

Ming ‘Tommy’ Tang

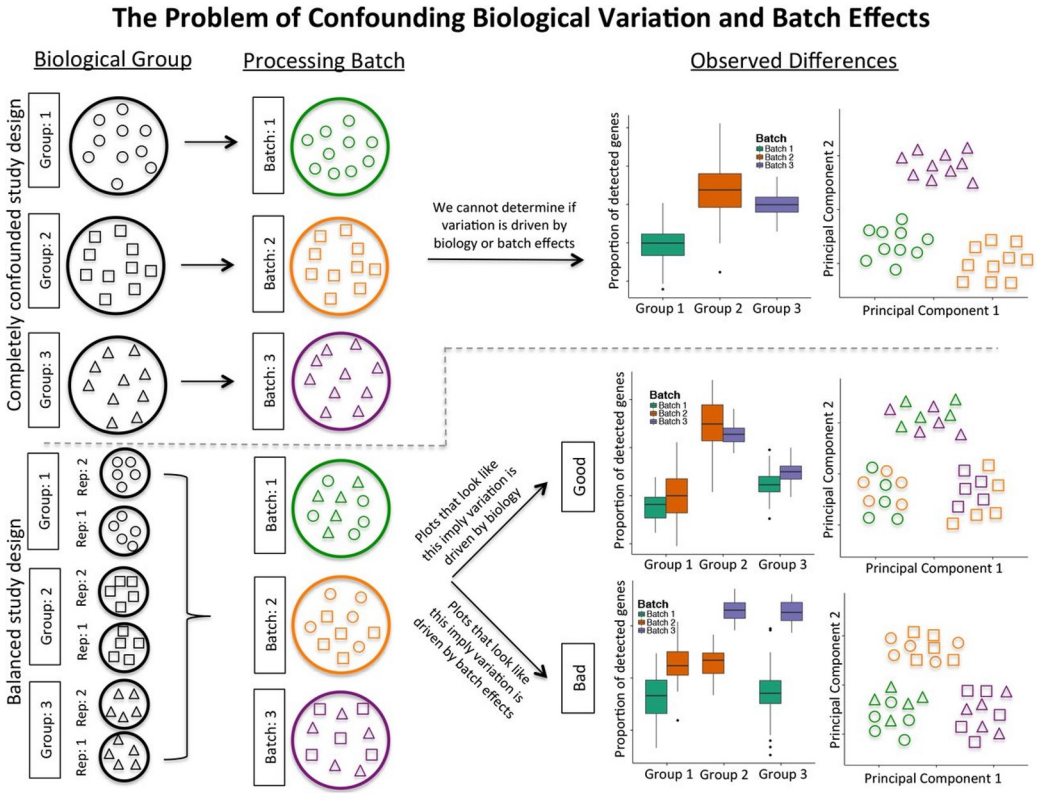
Director of Computational Biology at Immunitas

Divingintogeneticsandgenomics.com

Twitter: tangming2005

01/24/2023

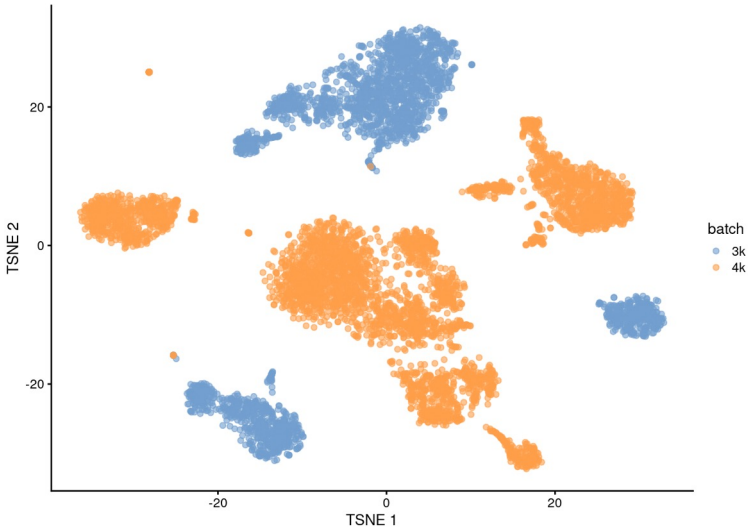
Avoid batch and confounding effects: experimental design

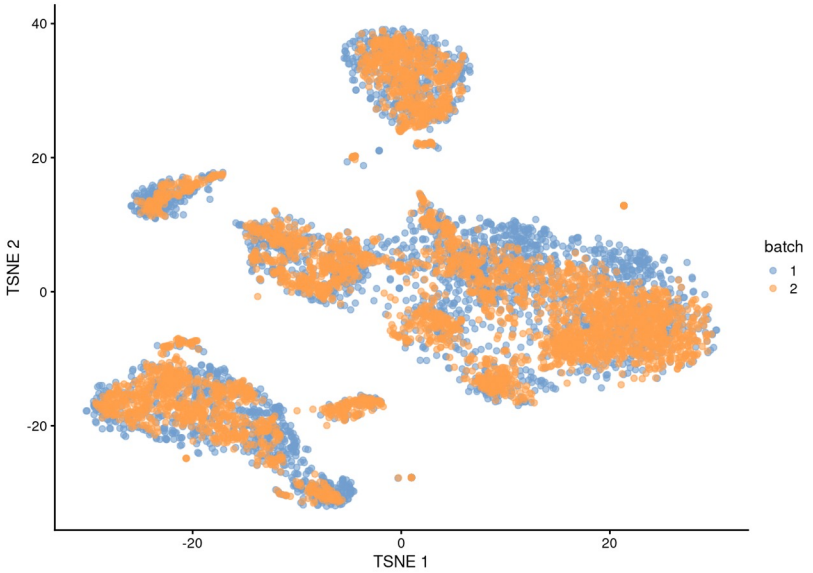


Luciano Martelotto

Hicks *et al.*, *Biostatistics*. 2018

Data integration/batch correction

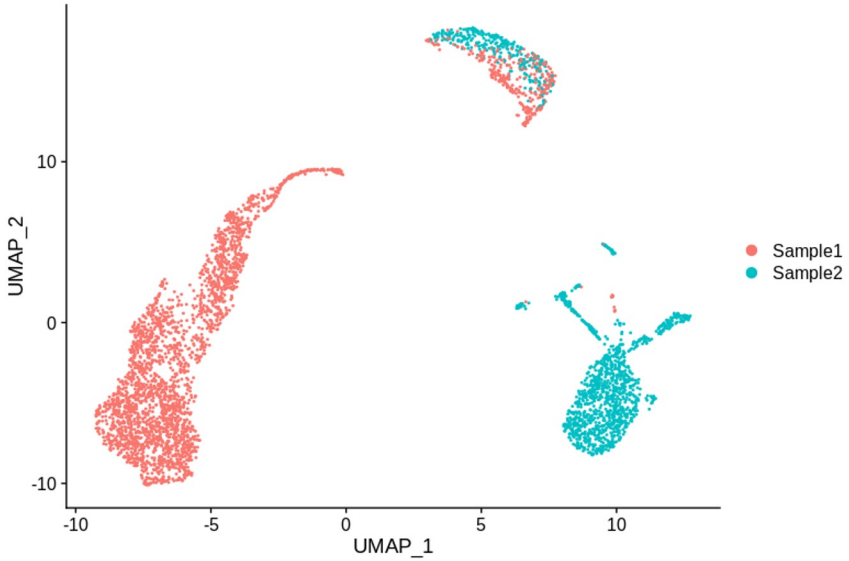


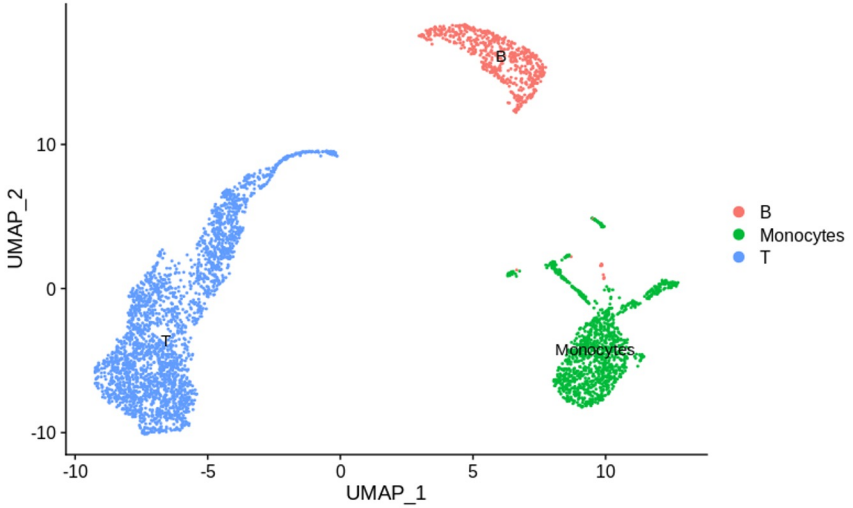


http://bioconductor.org/books/3.14/OSCA.multisample/integrating-datasets.html#motivation

Data integration

- Batch effect or not? Correct or not





https://constantamateur.github.io/2020-06-09-scBatch1/

Sacrificing biology by integration

6.4.2 Sacrificing biology by integration

Earlier in this chapter, we defined clusters from corrected values after applying `fastMNN()` to cells from all samples in the chimera dataset. Alert readers may realize that this would result in the removal of biological differences between our conditions. Any systematic difference in expression caused by injection would be treated as a batch effect and lost when cells from different samples are aligned to the same coordinate space. Now, one may not consider injection to be an interesting biological effect, but the same reasoning applies for other conditions, e.g., integration of wild-type and knock-out samples (Section 5) would result in the loss of any knock-out effect in the corrected values.

This loss is both expected and desirable. As we mentioned in Section 3, the main motivation for performing batch correction is to enable us to characterize population heterogeneity in a consistent manner across samples. This remains true in situations with multiple conditions where we would like one set of clusters and annotations that can be used as common labels for the DE or DA analyses described above. The alternative would be to cluster each condition separately and to attempt to identify matching clusters across conditions - not straightforward for poorly separated clusters in contexts like differentiation.

It may seem distressing to some that a (potentially very interesting) biological difference between conditions is lost during correction. However, this concern is largely misplaced as the correction is only ever used for defining common clusters and annotations. The DE analysis itself is performed on pseudo-bulk samples created from the uncorrected counts, preserving the biological difference and ensuring that it manifests in the list of DE genes for affected cell types. Of course, if the DE is strong enough, it may result in a new condition-specific cluster that would be captured by a DA analysis as discussed in Section 6.4.1.

New Results

Follow this preprint

PMD Uncovers Widespread Cell-State Erasure by scRNAseq Batch Correction Methods

Scott R Tyler, Supinda Bunyavanich, Eric E Schadt

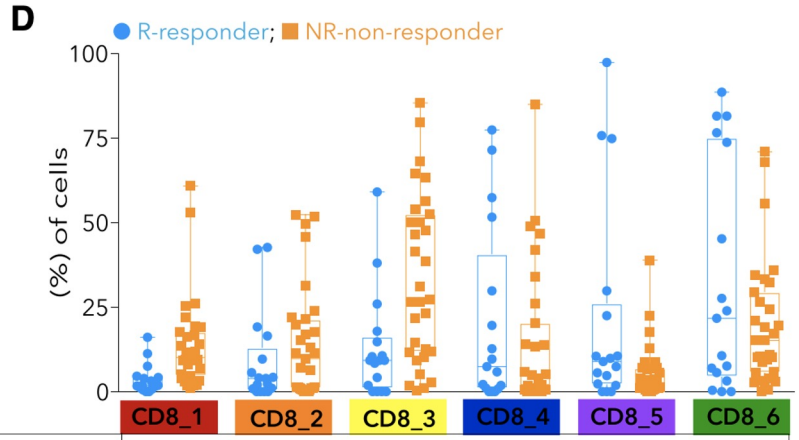
doi: https://doi.org/10.1101/2021.11.15.468733

This article is a preprint and has not been certified by peer review [what does this mean?].

0 0 0 0 2 0 127

http://bioconductor.org/books/3.14/OSCA.multisample/differential-abundance.html#sacrificing-differences

Differential cell abundance (DA) analysis



	CD8_1	CD8_2	CD8_3	CD8_4	CD8_5	CD8_6
All samples	$6.7x10^{-5}$	n.s	0.001	n.s	n.s	n.s
Ag presentation	$1x10^{-5}$	n.s	$6.6x10^{-5}$	0.01	0.02	n.s

```
##
##                          5  6   7   8   9  10
##   Allantois             97 15 139 127 318 259
##   Blood progenitors 1    6  3  16   6   8  17
##   Blood progenitors 2   31  8  28  21  43 114
##   Cardiomyocytes        85 21  79  31 174 211
##   Caudal Mesoderm       10 10   9   3  10  29
##   Caudal epiblast        2  2   0   0  22  45
```

6.2 Performing the DA analysis

Our DA analysis will again be performed with the *edgeR* package. This allows us to take advantage of the NB GLM methods to model overdispersed count data in the presence of limited replication - except that the counts are not of reads per gene, but of cells per label (Lun, Richard, and Marioni 2017). The aim is to share information across labels to improve our estimates of the biological variability in cell abundance between replicates.

```
library(edgeR)
# Attaching some column metadata.
extra.info <- colData(merged)[match(colnames(abundances), merged$sample),]
y.ab <- DGEList(abundances, samples=extra.info)
y.ab
```

http://bioconductor.org/books/3.14/OSCA.multisample/differential-abundance.html#overview

Multi-sample Differential expression: pseudo-bulk for the win

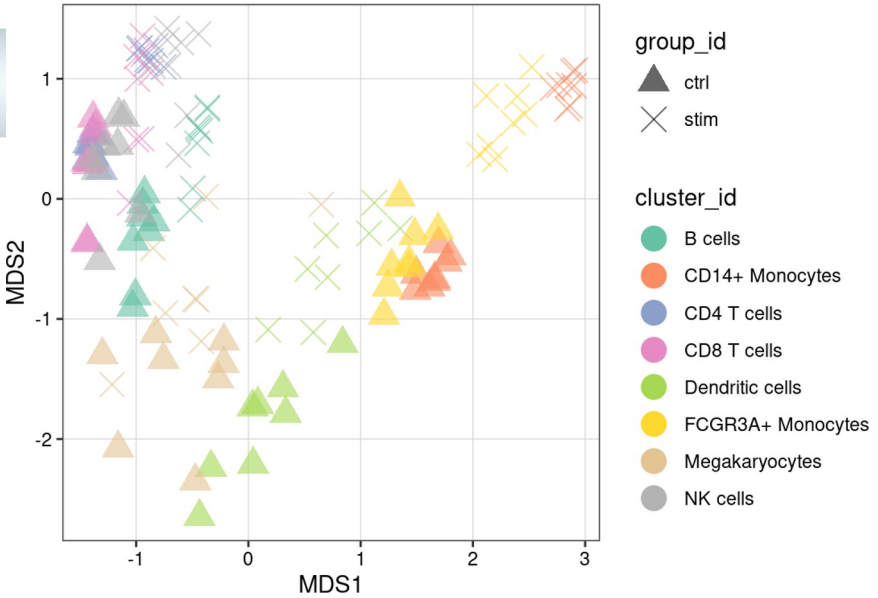
nature COMMUNICATIONS

ARTICLE

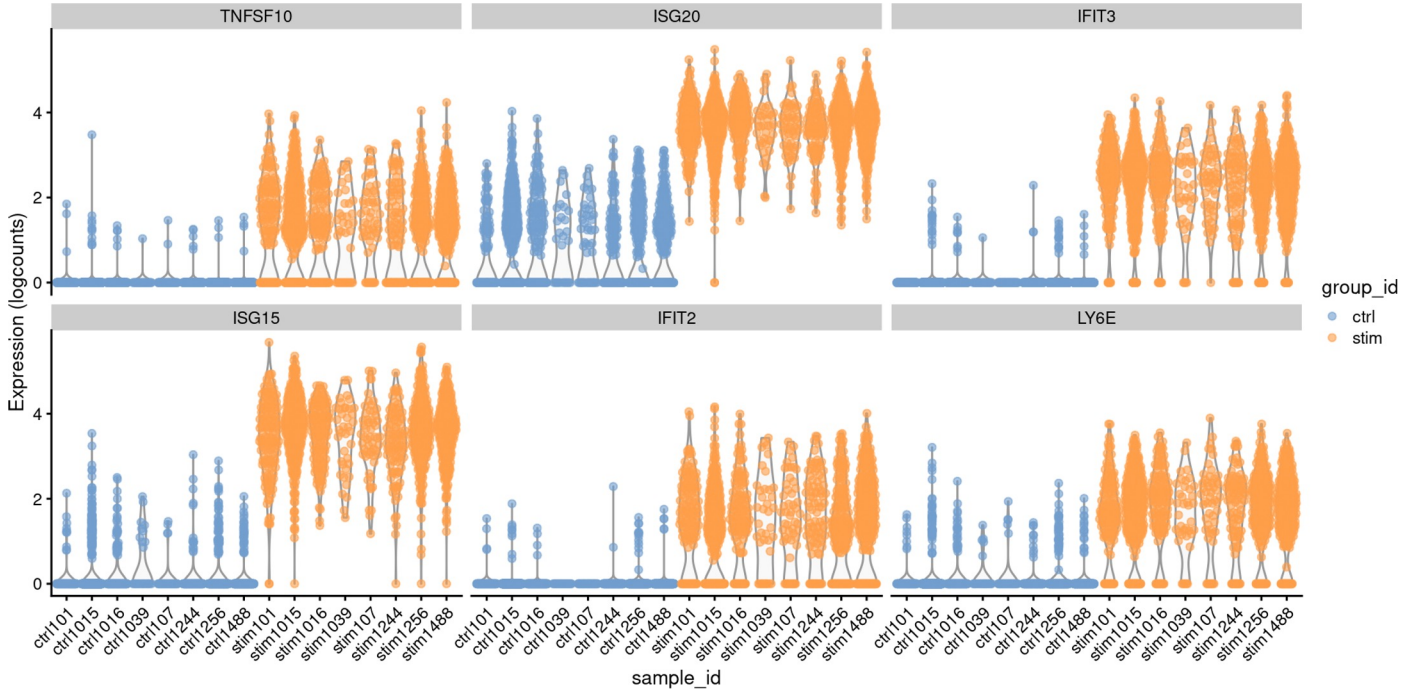
https://doi.org/10.1038/s41467-021-25960-2 OPEN

Confronting false discoveries in single-cell differential expression

Jordan W. Squair[1,2,3], Matthieu Gautier[1,2], Claudia Kathe[1,2], Mark A. Anderson[1,2], Nicholas D. James[1,2], Thomas H. Hutson[1,2], Rémi Hudelle[1,2], Taha Qaiser[3], Kaya J. E. Matson[4], Quentin Barraud[1,2], Ariel J. Levine[4], Gioele La Manno[1], Michael A. Skinnider[1,2,5,6✉] & Grégoire Courtine[1,2,6✉]



Muscat::pbDS() or Scran::pseudoBulkDEG



https://www.nature.com/articles/s41467-020-19894-4

Differential expression (DE) vs Differential abundance (DA)

14.6.1 DE or DA? Two sides of the same coin

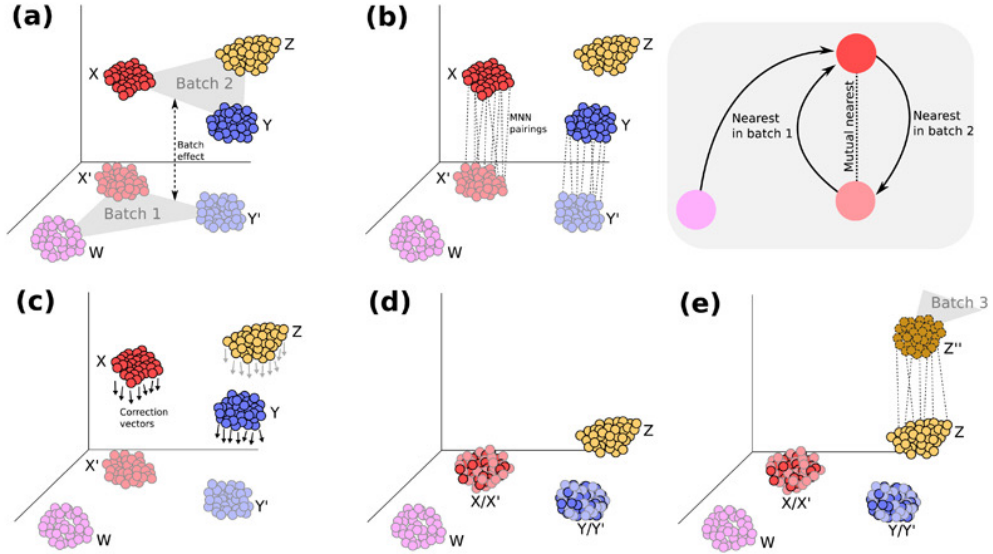
While useful, the distinction between DA and DE analyses is inherently artificial for scRNA-seq data. This is because the labels used in the former are defined based on the genes to be tested in the latter. To illustrate, consider a scRNA-seq experiment involving two biological conditions with several shared cell types. We focus on a cell type X that is present in both conditions but contains some DEGs between conditions. This leads to two possible outcomes:

1. The DE between conditions causes X to form two separate clusters (say, X_1 and X_2) in expression space. This manifests as DA where X_1 is enriched in one condition and X_2 is enriched in the other condition.
2. The DE between conditions is not sufficient to split X into two separate clusters, e.g., because the data integration procedure identifies them as corresponding cell types and merges them together. This means that the differences between conditions manifest as DE within the single cluster corresponding to X.

We have described the example above in terms of clustering, but the same arguments apply for any labelling strategy based on the expression profiles, e.g., automated cell type assignment (Chapter 12). Moreover, the choice between outcomes 1 and 2 is made implicitly by the combined effect of the data merging, clustering and label assignment procedures. For example, differences between conditions are more likely to manifest as DE for coarser clusters and as DA for finer clusters, but this is difficult to predict reliably.

https://www.biorxiv.org/content/10.1101/2022.03.15.484475v1

Linear embedding integration using Mutual Nearest Neighbors (MNN)



fastMNN()

https://www.bioconductor.org/packages/release/bioc/vignettes/batchelor/inst/doc/correction.html

Seurat: CCA (canonical correlation analysis) + MNN

Harmony for batch correction

nature methods

Explore content ▾ About the journal ▾ Publish with us ▾ Subscribe

nature › nature methods › articles › article

Article | Published: 18 November 2019

Fast, sensitive and accurate integration of single-cell data with Harmony

Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-ru Loh & Soumya Raychaudhuri ✉

Nature Methods **16**, 1289–1296 (2019) | Cite this article

73k Accesses | **1562** Citations | **149** Altmetric | Metrics

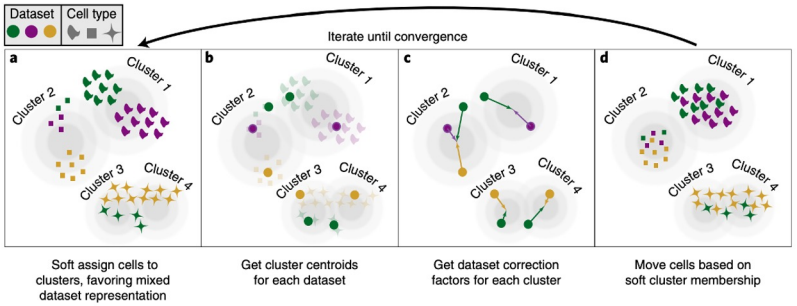


Fig. 1 | Overview of Harmony algorithm. PCA embeds cells into a space with reduced dimensionality. Harmony accepts the cell coordinates in this reduced space and runs an iterative algorithm to adjust for dataset specific effects. **a**, Harmony uses fuzzy clustering to assign each cell to multiple clusters, while a penalty term ensures that the diversity of datasets within each cluster is maximized. **b**, Harmony calculates a global centroid for each cluster, as well as dataset-specific centroids for each cluster. **c**, Within each cluster, Harmony calculates a correction factor for each dataset based on the centroids. **d**, Finally, Harmony corrects each cell with a cell-specific factor: a linear combination of dataset correction factors weighted by the cell's soft cluster assignments made in step **a**. Harmony repeats steps **a** to **d** until convergence. The dependence between cluster assignment and dataset diminishes with each round. Datasets are represented with colors, cell types with different shapes.

Harmony only corrects the PCA coordinates, it does not give you the batch corrected values like Seurat CCA based method

iNMF in Liger

Iterative single-cell multi-omic integration using online learning

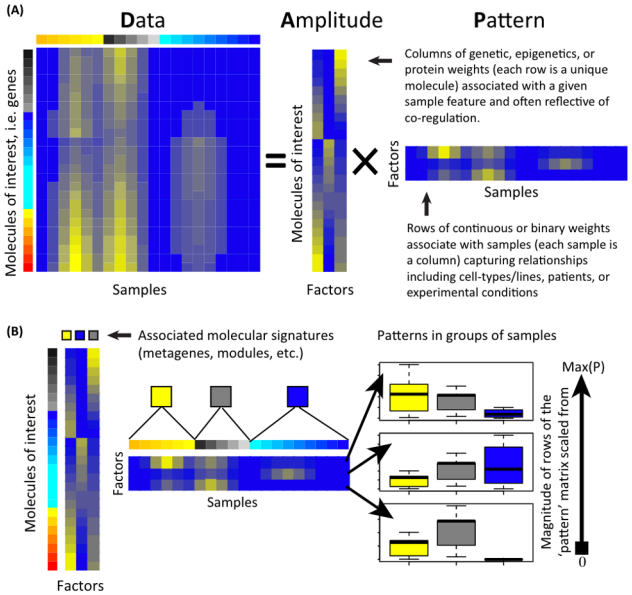
Chao Gao[1], Jialin Liu[1], April R. Kriebel[1], Sebastian Preissl[2], Chongyuan Luo[3,4,7], Rosa Castanon[3], Justin Sandoval[3], Angeline Rivkin[3], Joseph R. Nery[3], Margarita M. Behrens[5], Joseph R. Ecker[3,4], Bing Ren[2] and Joshua D. Welch[1,6] ✉

Integrating large single-cell gene expression, chromatin accessibility and DNA methylation datasets requires general and scalable computational approaches. Here we describe online integrative non-negative matrix factorization (iNMF), an algorithm for integrating large, diverse and continually arriving single-cell datasets. Our approach scales to arbitrarily large numbers of cells using fixed memory, iteratively incorporates new datasets as they are generated and allows many users to simultaneously analyze a single copy of a large dataset by streaming it over the internet. Iterative data addition can also be used to map new data to a reference dataset. Comparisons with previous methods indicate that the improvements in efficiency do not sacrifice dataset alignment and cluster preservation performance. We demonstrate the effectiveness of online iNMF by integrating more than 1 million cells on a standard laptop, integrating large single-cell RNA sequencing and spatial transcriptomic datasets, and iteratively constructing a single-cell multi-omic atlas of the mouse motor cortex.

on a dataset of 1.3 million cells from the mouse embryo, online iNMF finishes dimension reduction in 25min using 1.9 GB of RAM on a laptop, whereas Harmony requires 98min and 109 GB of RAM on a large-memory server

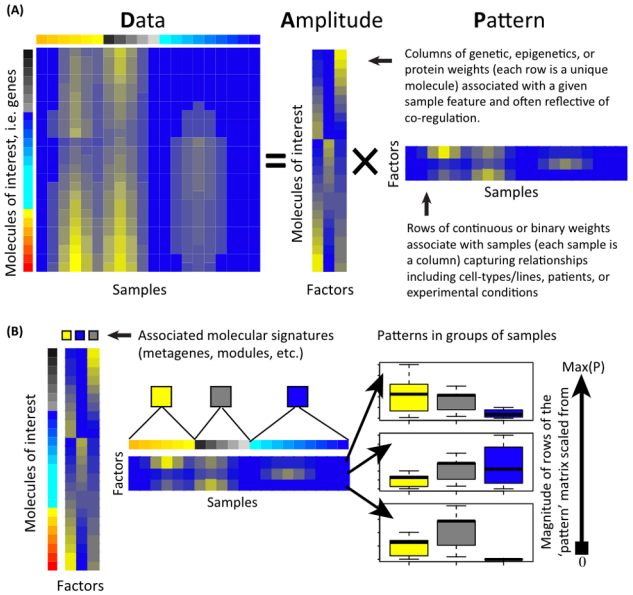
https://www.nature.com/articles/s41587-021-00867-x

Integrative NMF (iNMF)



https://divingintogeneticsandgenomics.com/post/matrix-factorization-for-single-cell-rnaseq-data/

Integrative NMF (iNMF)



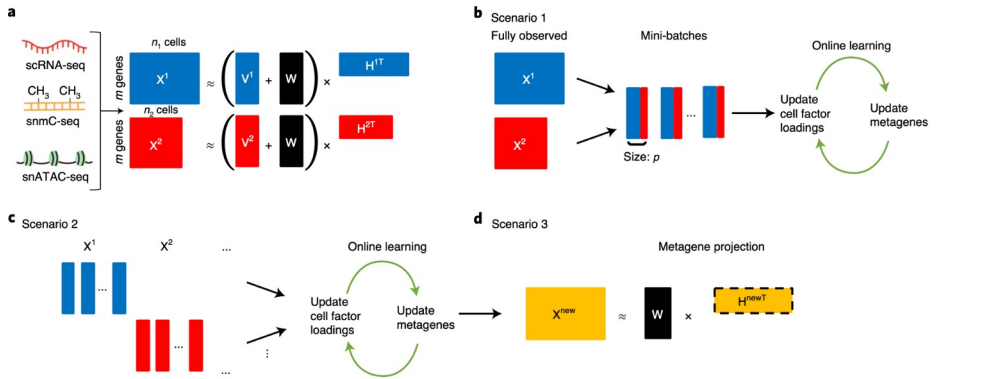


Fig. 1 | Overview of the online iNMF algorithm. a, Schematic of iNMF: the input single-cell datasets are jointly decomposed into shared (*W*) and dataset-specific (*V*^{i}) metagenes and corresponding 'metagene expression levels' or cell factor loadings (*H*^{i}). These metagenes and cell factor loadings provide a quantitative definition of cell identity and how it varies across biological settings. **b**–**d**, Three different scenarios in which online learning can be used for single-cell data integration. **b**, Scenario 1: the single-cell datasets are large but fully observed. Online iNMF processes the data in random mini-batches, enabling memory usage independent of dataset size. Each cell may be used multiple times in different epochs of training to update the metagenes. **c**, Scenario 2: the datasets arrive sequentially, and online iNMF processes the datasets as they arrive, using each cell to update the metagenes exactly once. **d**, Scenario 3: online iNMF is performed as in Scenario 1 or Scenario 2 to learn *W* and *V*^{i}. Then cell factor loadings for the newly arriving dataset are calculated using the shared metagenes (*W*) learned from previously processed datasets. The new dataset is not used to update the metagenes.

https://divingintogeneticsandgenomics.com/post/matrix-factorization-for-single-cell-rnaseq-data/

https://github.com/welch-lab/liger

Variational autoencoder (VAE) based integration method

Probabilistic models for single-cell omics data

scvi-tools accelerates data analysis and model development, powered by PyTorch and AnnData.

```
pip install scvi-tools
```

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature › nature communications › articles › article

Article | Open access | Published: 17 October 2022

Online single-cell data integration through projecting heterogeneous datasets into a common cell-embedding space

Lei Xiong, Kang Tian, Yuzhe Li, Weixi Ning, Xin Gao & Qiangfeng Cliff Zhang

https://github.com/jsxlei/SCALEX

nature methods

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature › nature methods › articles › article

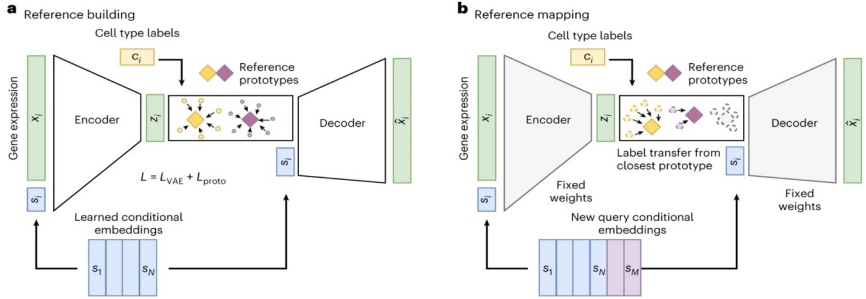
Article | Open access | Published: 09 October 2023

Population-level integration of single-cell datasets enables multi-scale analysis across samples

Carlo De Donno, Soroor Hediyeh-Zadeh, Amir Ali Moinfar, Marco Wagenstetter, Luke Zappia, Mohammad Lotfollahi & Fabian J. Theis

Fig. 1: scPoli enables learning cell-level and sample-level representations.

From: Population-level integration of single-cell datasets enables multi-scale analysis across samples



Benchmarking of single-cell data integration

nature methods

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature › nature methods › analyses › article

Analysis | Open access | Published: 23 December 2021

Benchmarking atlas-level data integration in single-cell genomics

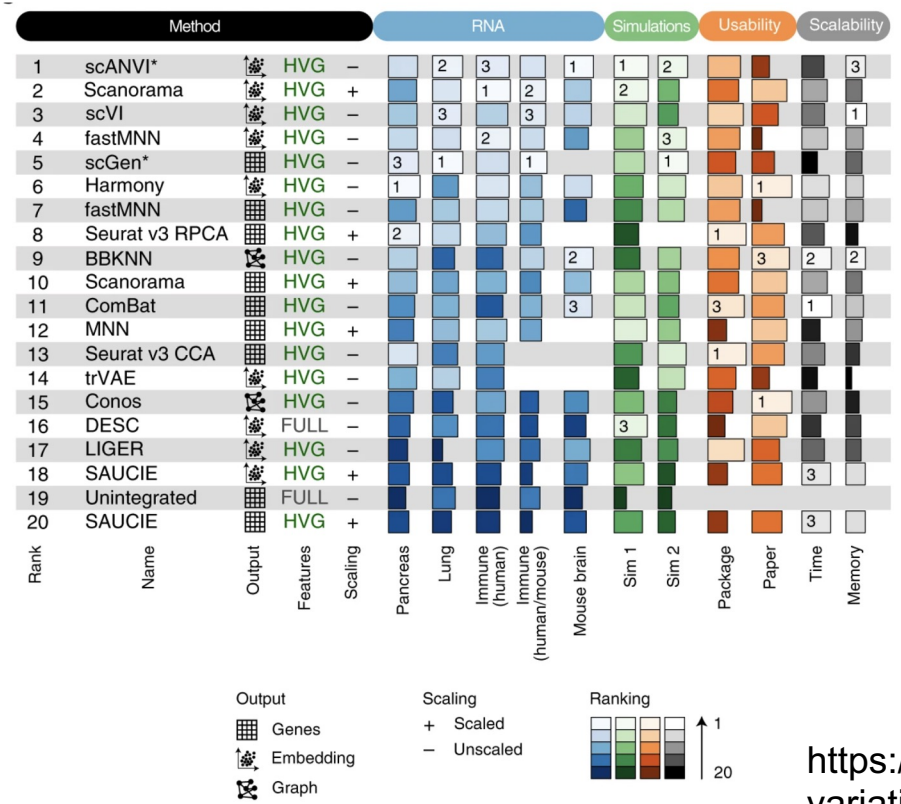
Malte D. Luecken, M. Büttner, K. Chaichoompu, A. Danese, M. Interlandi, M. F. Mueller, D. C. Strobl, L. Zappia, M. Dugas, M. Colomé-Tatché ✉ & Fabian J. Theis ✉

Nature Methods **19**, 41–50 (2022) | Cite this article

Abstract

Single-cell atlases often include samples that span locations, laboratories and conditions, leading to complex, nested batch effects in data. Thus, joint analysis of atlas datasets requires reliable data integration. To guide integration method choice, we benchmarked 68 method and preprocessing combinations on 85 batches of gene expression, chromatin accessibility and simulation data from 23 publications, altogether representing >1.2 million cells distributed in 13 atlas-level integration tasks. We evaluated methods according to scalability, usability and their ability to remove batch effects while retaining biological variation using 14 evaluation metrics. We show that highly variable gene selection improves the performance of data integration methods, whereas scaling pushes methods to prioritize batch removal over conservation of biological variation. Overall, scANVI, Scanorama, scVI and scGen perform well, particularly on complex integration tasks, while single-cell ATAC-sequencing integration performance is strongly affected by choice of feature space. Our freely available Python module and benchmarking pipeline can identify optimal data integration methods for new data, benchmark new methods and improve method development.

Benchmarking of single-cell data integration



scVI/scANVI performs the best. It uses variational auto-encoder

Harmony is pretty good, but less scalable. I use the R version a lot, Scanpy also has a built-in function.

Seurat CCA is slow.

BBKNN looks very scalable. scanpy has an implementation

https://divingintogeneticsandgenomics.com/post/how-to-code-a-variational-autoencoder-vae-in-r-using-mnist-dataset/

Challenges

- Lacking of gold-standard dataset.
- Batch correction and preservation of biological difference.
- Time and memory consumption
- Choose a good method for future use on your own data (Harmony, iNMF, fastMNN, scVI, scPoli, SCALEX). Different methods may outperform for different datasets