JIND: Joint Integration and Neural Discrimination for automated single-cell annotation

Mohit Goyal[*]
ECE, UIUC, USA

Guillermo Serrano
CIMA, University of Navarra, Spain

Ilan Shomorony
ECE, UIUC, USA

Mikel Hernaez[†]
IGB, UIUC, USA

Idoia Ochoa[‡]
EE, University of Navarra, Spain

Abstract

Single-cell RNA-seq is a powerful tool in the study of the cellular composition of different tissues and organisms. A key step in the analysis pipeline is the annotation of cell-types based on the expression of specific marker genes. Since manual annotation is labor-intensive and does not scale to large datasets, several methods for automated cell-type annotation have been proposed based on supervised learning. However, these methods generally require feature extraction and batch alignment prior to classification, and their performance may become unreliable in the presence of cell-types with very similar transcriptomic profiles, such as differentiating cells. We propose JIND, a framework for automated cell-type identification based on neural networks that directly learns a low-dimensional representation (latent code) in which cell-types can be reliably determined. To account for batch effects, JIND performs a novel asymmetric alignment in which the transcriptomic profile of unseen cells is mapped onto the previously learned latent space, hence avoiding the need of retraining the model whenever a new dataset becomes available. JIND also learns cell-type-specific confidence thresholds to identify and reject cells that cannot be reliably classified. We show on datasets with and without batch effects that JIND classifies cells more accurately than previously proposed methods while rejecting only a small proportion of cells. Moreover, JIND batch alignment is parallelizable, being more than five or six times faster than Seurat integration. Availability: `https://github.com/mohit1997/JIND`.

Introduction

Characterization of cell-types in a mixture of cells is an important step in single-cell genomic data analysis. This is often accomplished by using a clustering algorithm on the gene expression vectors, followed by manual labelling of clusters based on specific biological markers.

With the gain in popularity of single-cell RNA sequencing (scRNA-seq), carefully annotated large single-cell datasets [1–3] have been made public in recent years. These datasets, combined with supervised learning techiques, present a natural framework to transfer labels from an annotated scRNA-seq dataset (source batch) to an unannotated dataset (target batch) [4–11]. However, off-the-shelf classifiers do not perform well on this task because the source and target batches may exhibit technical variability, generally referred to as *batch effects*. These batch effects reduce the reliability

[*]email: mohit@illinois.edu

[†]also affiliated with the Center for Applied Medical Research (CIMA), University of Navarra, Spain

[‡]also affiliated with the ECE (Electrical and Computer Engineering) Department, UIUC (University of Illinois at Urbana Champaign), Urbana, IL 61801

Preprint. Under review.

of the prediction models resulting in poor classification performance. Moreover, since cells can exist in intermediate states during the process of differentiation [12], standard classification algorithms end up misclassifying cells that are in transitioning states or cells that are outliers [13] due to the inherent noise in the dataset.

To overcome these issues, we propose a novel framework based on neural netwroks (NNs) called JIND for cell-type identification. JIND automatically learns a low-dimensional representation (latent space) well suited for cell-type classification from the source batch itself. Then, to deal with batch effects, JIND projects the target batch onto the previously learned latent space without changing the source batch latent codes. This leads to an *asymmetric* alignment that eliminates the need to retrain the NN-based prediction model. In addition, JIND estimates cell-type-specific confidence levels during training, which capture the ease to distinguish each type from the rest. These confidence levels are then used to filter out (that is, label as unassigned) cells that cannot be classified with high confidence. Finally, the JIND framework allows the refinement of the parameters of the prediction model via self-training [14, 15], by treating the high confidence predictions on the target batch as new labeled data. We refer to this extension as JIND+.

We empirically show that JIND outperforms state-of-the-art methods on a variety of datasets, achieving approximately 97% classification accuracy on average. We also show that the proposed thresholding scheme is robust to datasets of varying difficulties, rejecting only about 4% of cells. Finally, we show that the misclassification rate can be meaningfully reduced with JIND+.

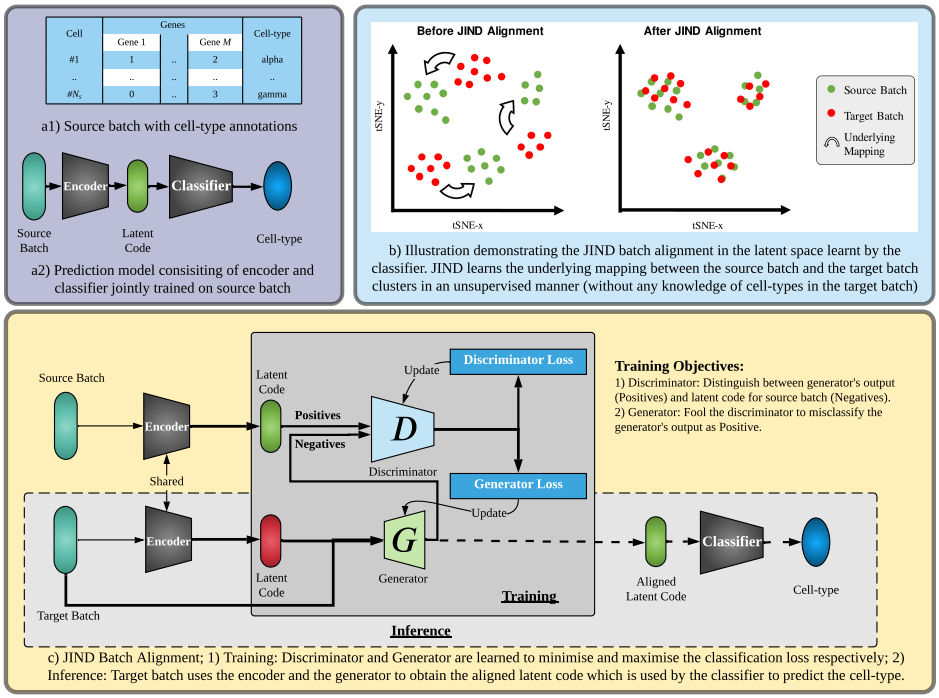

Figure 1: **Overview of JIND:** JIND uses a source batch (a1) with a gene expression matrix and corresponding cell-types to train a Neural Network-based prediction model (a2) which consists of an encoder and a classifier subnetwork. The low-dimensional representation output by the encoder subnetwork is denoted as the *latent code*. Note that this prediction model should not be directly used to annotate the target batch due to the presence of batch effects. b) To account for the technical variability across batches, batch alignment is required to align the source and target latent codes. c) JIND uses adversarial training via a generator and discriminator pair to align the source and target latent codes. The discriminator is trained to classify an input latent code either as a latent code produced by the generator (negative label) or as the source latent code produced by the encoder (positive label). In contrast, the generator is trained to fool the discriminator into misclassifying the generator's output as source latent code. Finally, the output of the trained generator (the aligned latent code) is used by the classifier subnetwork to infer the cell-types of the target batch.

Methods

JIND tackles the problem of supervised cell-type annotation of single-cell RNA sequencing data. The label information comes from a *source batch* dataset: a gene expression matrix (N_s cells $\times$ M genes), and the corresponding cell-type annotations (Figure 1(a1)). The goal is to label the target batch (N_t cells $\times$ M genes), where the cell-type information is absent. While existing methods require separate batch alignment techniques to be performed prior to classification, JIND trains a NN-based prediction model on the annotated source batch and then uses adversarial training to align the target batch onto the latent space learned by the NN. Thus, JIND is able to compensate for batch effects while avoiding the need for retraining the model when new data becomes available.

The NN used by JIND consists of two subnetworks, an encoder and a classifier (Figure 1(a2). For details on the architecture, refer to Supplementary Figure S1). First, the encoder network maps the input gene expression vector linearly onto a 256-dimensional latent space, referred to as latent code, which is then fed to the classifier subnetwork for predicting the cell-type. These two subnetworks are trained jointly on the source batch by minimizing a weighted categorical cross entropy loss. Since the target batch can have, in general, a different gene expression distribution than the source batch, the latent code (i.e., the encoder output) for both batches will likely have different distributions. Therefore, the latent code from the target batch needs to be modified so that the classifier subnetwork–which was trained on the source batch–can reliably predict the true cell-type (Figure 1(b)).

The proposed alignment technique, which aims at removing batch effects while maintaining useful biological variability for classification, is inspired by both Generative Adversarial Networks (GANs)[16] and methods developed for the Machine Learning problem known as domain adaptation[17]. More precisely, JIND uses adversarial training (Figure 1(c)) to correct the latent code from the target batch by learning a generator function to transform the distribution of the target latent code to that of the source latent code. To learn this generator function, a binary discriminator function is simultaneously learned. While the discriminator function aims at distinguishing between the generator's output and the source latent code, the generator function aims at fooling the discriminator into misclassifying the generator's output as source latent code. The output of the trained generator function is the aligned latent code, and it is later used for cell-type inference.

Since it is possible that some cells in the target batch might be undergoing cell differentiation, or that their gene expression might have abnormal patterns, JIND provides a structured way to reject (that is, label them as unassigned) some of the predictions made by the aforementioned prediction model. Specifically, JIND estimates cell-type-specific confidence thresholds from the source batch based on the outlier fraction input by user (set to 5% by default) such that the overall misclassification rate is minimized in a controlled manner. This is in comparison to other fixed-threshold-based rejection schemes used in existing methods such as scPred[5], SVM_{Rej}[10] and ACTINN[4]. which do not take into account the variability in ease of classification across different cell-types and datasets.

Finally, an extension to the JIND framework based on self-training [15, 14], coined JIND+, is proposed. In JIND+, additionally, the confident predictions made on the target batch post alignment are used to fine-tune the parameters of the encoder and classifier subnetworks.

Results

JIND can accurately annotate scRNA-seq datasets with batch effects: We compare JIND and JIND+ with SVM_{Rej} [10], scPred [5], Seurat-LT (Seurat Label Transfer) [9] and ACTINN [4]. These methods were selected as, in a recent study [10], SVM_{Rej} attained the best performance among existing automated cell identification methods, including methods incorporating prior knowledge in the form of marker genes. ACTINN and scPred were also among the best performing methods. We also include Seurat-LT which, unlike the other methods, does not have a rejection module.

Table 1: scRNA-seq datasets used for evaluation.

Name [Batches]	**# Cells $\times$ # Genes**	**#Cell Types**	**Batches**
PBMC [10x_v3, 10x_v5] [18]	15476 $\times$ 1199	9	2
Pancreas [Bar16 [19], Mur16 [20], Seg16 [21]]	14058 $\times$ 2448	22	3

For the performance evaluation, we consider datasets with batch effects described in Table 1 and specify, in each case, the source and the target batch. For example, *Pancreas* Bar16-Mur16 denotes that *Pancreas* Bar16 is the source batch and *Pancreas* Mur16 is the target batch.

We experiment with three pairs of source and target batches: *PBMC* 10x_v3-10x_v5, *Pancreas* Bar16-Mur16 and *Pancreas* Bar16-Seg16. Since SVM_{Rej}, scPred and ACTINN do not internally perform any batch alignment, these methods typically benefit from external integration tools [10]. Therefore, we also report their performance after aligning source and target batches using Seurat integration [9]. Table 2 summarizes the results for these experiments. We observe that JIND+ consistently achieves slightly better *raw* accuracy than JIND in all cases. Moreover, JIND+ reduces the rejection rates of JIND by a factor of 2 while keeping the effective accuracy almost identical. We also observe that JIND and JIND+ outperform previously proposed methods in raw accuracy in all cases, except for the *PBMC* dataset in which Seurat-LT achieves a raw accuracy 0.8% higher than JIND+. Nonetheless, on both *Pancreas* datasets, JIND+ outperforms Seurat-LT by 9% on average. When no external alignment is performed the rejection rates with SVM_{Rej}, scPred and ACTINN, for all three datasets, are significantly higher than with JIND+. Notably, SVM_{Rej} and ACTINN reject almost all cells in some cases. When SVM_{Rej}, scPred and ACTINN were evaluated after using Seurat batch alignment, we observe that their rejection rates are significantly reduced. However, scPred still rejects more than 10% of cells even after batch alignment in all three experiments. Our results are in agreement with the review conducted by Abdelaal et al. [10], which concludes that SVM_{Rej}, scPred and ACTINN benefit from batch alignment tools. However, JIND+ still outperforms ACTINN and SVM_{Rej}, achieving approximately 3% higher raw accuracy. In comparison to scPred, JIND+ achieves more than 7% higher raw accuracy on average. We also observe that using Seurat integration actually worsens the classification performance of both JIND and JIND+ (Supplementary Table S1).

Finally, to analyze the possible causes for misclassifications, we perform a differential expression analysis [22] (see Supplementary Figure S6) for specific cell-types between correct and false predictions made by JIND+ . Our analysis indicates that our misclassifications do not correspond to arbitrary mistakes made by the prediction model, but rather to potential annotation errors.

JIND aligns cell-type clusters in the latent space: To further analyze the benefits of JIND alignment, we selected four cell-types from *Pancreas* Bar16-Mur16 dataset namely, *Alpha, Beta, Delta*, and *Gamma*. We observe that the NN-based prediction model (used in JIND), after being trained on the source batch, rejects more than 50% of cells on the target batch. In contrast, after performing JIND alignment, only 5% of cells are rejected and more than 98% of the remaining cells are classified correctly. On comparing the distributions of the latent codes before and after alignment, we observe that JIND is able to effectively align the latent codes for the two batches (Supplementary Figure S7).

In summary, we demonstrate that JIND+ is highly accurate and more practical than existing cell annotation pipelines for transferring cell-type labels across different batches. JIND+ also provides a controlled way of rejecting low confidence predictions to avoid erroneous annotation.

Table 2: Comparison of different cell classification methods. *raw* is the inital accuracy of the classifier, *rej* is the percentage of cells rejected by the classifier and *eff* is the effective accuracy after rejecting unconfident predictions. For *SVM_{Rej}*, *scPred* and *ACTINN*, we report results on both Batched and Integrated data. Best raw accuracy rates are **bold faced** and rejection rates above 0.1 are colored red.

Datasets	*Metrics*	*JIND*	*JIND+*	*Seurat-LT*	*SVM_{Rej}*		*scPred*		*ACTINN*	
					Batched	Integrated	Batched	Integrated	Batched	Integrated
PBMC 10x_v3-10x_v5	*raw*	0.971	0.974	**0.981**	0.956	0.962	0.931	0.946	0.956	0.965
	rej	0.07	0.03	-	0.99	0.05	0.10	0.10	0.37	0.05
	eff	0.986	0.985	-	1.000	0.975	0.957	0.971	0.990	0.980
Pancreas Bar16-Mur16	*raw*	0.958	**0.959**	0.868	0.894	0.921	0.729	0.870	0.874	0.923
	rej	0.05	0.03	-	1.00	0.04	0.45	0.18	0.99	0.07
	eff	0.974	0.971	-	NA	0.939	0.726	0.931	1.000	0.953
Pancreas Bar16-Seg16	*raw*	0.987	**0.992**	0.923	0.925	0.953	0.819	0.898	0.930	0.952
	rej	0.05	0.02	-	0.99	0.04	0.41	0.18	0.99	0.08
	eff	0.997	0.997	-	1.000	0.963	0.868	0.951	1.000	0.971

References

[1] Regev, A. *et al.* Science forum: The human cell atlas. *eLife* **6** (2017).

[2] Han, X. *et al.* Mapping the mouse cell atlas by microwell-seq. *Cell* **172**, 1091 – 1107.e17 (2018). URL http://www.sciencedirect.com/science/article/pii/S0092867418301168.

[3] Schaum, N. *et al.* Single-cell transcriptomics of 20 mouse organs creates a tabula muris. *Nature* **562**, 367–372 (2018). URL https://doi.org/10.1038/s41586-018-0590-4.

[4] Ma, F. & Pellegrini, M. Automated identification of cell types in single cell rna sequencing. *bioRxiv* (2019). URL https://www.biorxiv.org/content/early/2019/01/28/532093. https://www.biorxiv.org/content/early/2019/01/28/532093.full.pdf.

[5] Alquicira-Hernandez, J., Sathe, A., Ji, H. P., Nguyen, Q. & Powell, J. E. scpred: accurate supervised method for cell-type classification from single-cell rna-seq data. *Genome Biology* **20**, 264 (2019). URL https://doi.org/10.1186/s13059-019-1862-5.

[6] Boufea, K., Seth, S. & Batada, N. N. scid: Identification of equivalent transcriptional cell populations across single cell rna-seq data using discriminant analysis. *bioRxiv* (2019). URL https://www.biorxiv.org/content/early/2019/01/31/470203. https://www.biorxiv.org/content/early/2019/01/31/470203.full.pdf.

[7] Li, C. *et al.* Scibet as a portable and fast single cell type identifier. *Nature Communications* **11**, 1818 (2020). URL https://doi.org/10.1038/s41467-020-15523-2.

[8] Kiselev, V. Y., Yiu, A. & Hemberg, M. scmap: projection of single-cell rna-seq data across data sets. *Nature Methods* **15**, 359–362 (2018). URL https://doi.org/10.1038/nmeth.4644.

[9] Stuart, T. *et al.* Comprehensive integration of single-cell data. *Cell* **177**, 1888–1902.e21 (2019).

[10] Abdelaal, T. *et al.* A comparison of automatic cell identification methods for single-cell rna sequencing data. *Genome Biology* **20**, 194 (2019). URL https://doi.org/10.1186/s13059-019-1795-z.

[11] Diaz-Mejia, J. J. *et al.* Evaluation of methods to assign cell type labels to cell clusters from single-cell rna-sequencing data. *F1000Research* **8**, ISCB Comm J–296 (2019). URL https://pubmed.ncbi.nlm.nih.gov/31508207. 31508207[pmid].

[12] Grün, D. Revealing routes of cellular differentiation by single-cell rna-seq. *Current Opinion in Systems Biology* **11**, 9 – 17 (2018). URL http://www.sciencedirect.com/science/article/pii/S2452310018300131. • Big data acquisition and analysis • Development and differentiation.

[13] Norton, S. S., Vaquero-Garcia, J., Lahens, N. F., Grant, G. R. & Barash, Y. Outlier detection for improved differential splicing quantification from RNA-Seq experiments with replicates. *Bioinformatics* **34**, 1488–1497 (2017). URL https://doi.org/10.1093/bioinformatics/btx790. https://academic.oup.com/bioinformatics/article-pdf/34/9/1488/25417002/btx790.pdf.

[14] Lee, D.-H. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)* (2013).

[15] Zou, Y., Yu, Z., Liu, X., Kumar, B. V. K. V. & Wang, J. Confidence regularized self-training (2019). 1908.09822.

[16] Goodfellow, I. J. *et al.* Generative adversarial networks (2014). 1406.2661.

[17] Zhu, J., Park, T., Isola, P. & Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR* **abs/1703.10593** (2017). URL http://arxiv.org/abs/1703.10593. 1703.10593.

[18] Park, J.-E., Polański, K., Meyer, K. & Teichmann, S. A. Fast batch alignment of single cell transcriptomes unifies multiple mouse cell atlases into an integrated landscape. *bioRxiv* (2018). URL https://www.biorxiv.org/content/early/2018/08/22/397042. https://www.biorxiv.org/content/early/2018/08/22/397042.full.pdf.

[19] Baron, M. *et al.* A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure. *Cell Systems* **3**, 346–360.e4 (2016). URL https://doi.org/10.1016/j.cels.2016.08.011.

[20] Muraro, M. J. *et al.* A single-cell transcriptome atlas of the human pancreas. *Cell Systems* **3**, 385–394.e3 (2016). URL https://doi.org/10.1016/j.cels.2016.09.002.

[21] Segerstolpe, Å. *et al.* Single-cell transcriptome profiling of human pancreatic islets in health and type 2 diabetes. *Cell Metabolism* **24**, 593–607 (2016). URL https://doi.org/10.1016/j.cmet.2016.08.020.

[22] Law, C. W., Chen, Y., Shi, W. & Smyth, G. K. voom: Precision weights unlock linear model analysis tools for rna-seq read counts. *Genome biology* **15**, R29 (2014).