DIMENSIONALITY REDUCTION AND DATA INTEGRATION FOR scRNA-SEQ DATA BASED ON INTEGRATIVE HIERARCHICAL POISSON FACTORISATION *

THOMAS WONG† AND MAURICIO BARAHONA‡

Abstract. Single-cell RNA sequencing (scRNA-seq) data sets consist of high-dimensional, sparse and noisy feature vectors, and pose a challenge for classic methods for dimensionality reduction. We show that application of Hierarchical Poisson Factorisation (HPF) to scRNA-seq data produces robust factors, and outperforms other popular methods. To account for batch variability in composite data sets, we introduce Integrative Hierarchical Poisson Factorisation (IHPF), an extension of HPF that makes use of a noise ratio hyper-parameter to tune the variability attributed to technical (batches) *vs.* biological (cell phenotypes) sources. We exemplify the advantageous application of IHPF under data integration scenarios with varying alignments of technical noise and cell diversity, and show that IHPF produces latent factors with a dual block structure in both cell and gene spaces for enhanced biological interpretability.

Key words. Single cell RNA-seq data, matrix factorisation, latent factors, data integration, dimension reduction

1. Introduction.

Recent advances in sequencing technologies make it possible to profile biological processes at single-cell resolution, allowing the characterisation of the cell-to-cell variability that underpins numerous phenomena in biology and medicine. Single-cell RNA sequencing (scRNA-seq) has a wide range of applications measuring cell heterogeneity, from cancerous tumours to antibiotic resistance in bacterial populations [1, 2]. In scRNA-seq data sets, each sample is a cell described by a high-dimensional feature vector containing the transcription counts of tens of thousands of genes. Hence the *sample space* correspond to cells, and the *feature space* refers to genes. The feature vectors are discrete, noisy and sparse, with many zeros and low counts.

Due to the large number of genes (features), dimensionality reduction is a key ingredient in the analysis workflow of scRNA-seq data sets. Many computational pipelines extract a reduced set of informative and interpretable *latent factors* that capture the distinguishing characteristics of cell types present in the data set. However, the sparse and discrete nature of scRNA-seq data poses challenges to classic methods for dimensionality reduction, e.g., Principal Component Analysis (PCA) or Non-negative Matrix Factorisation (NMF).

An additional challenge is posed by data integration. It is common to integrate data sets sequenced with different technologies and across different laboratories to validate hypotheses against previous findings, or to uncover rare biological signals that are difficult to observe in individual experiments,. For example, researchers might combine a newly sequenced data set with existing reference data sets to identify cell types. Under such data integration scenarios, biologically meaningful signals must be separated from technical effects, such as differences in sequencing depth, quality control, or population effects (e.g., demographics of patients providing the

*Funding: This work is supported by the Wellcome Trust under Grant 108908/B/15/Z and by the EPSRC under grant EP/N014529/1 funding the EPSRC Centre for Mathematics of Precision Healthcare at Imperial.

†Department of Mathematics, Imperial College London, London SW7 2AZ, U.K (minghei.wong15@imperial.ac.uk).

‡Department of Mathematics, Imperial College London, London SW7 2AZ, U.K (m.barahona@imperial.ac.uk,).

samples) [3]. Such batch effects are often non-linear, and hence problematic for batch correction methods based on, e.g., PCA [4].

There are two major data integration tasks in single-cell genomics. One type of task is multi-omics integration, which aims to combine different modalities of data (e.g., transcriptomics and proteomics) through *a shared cell space*, as done by tools such as MOFA+ [5] and DIABLO [6]. A second integration task aims to combine data sets of the same modality but collected from different experiments, and correct batch differences *on a shared gene space*. Examples of these methods include LIGER [7] and Scanorama [8]. Here, we focus on this latter task.

Several tools have been recently deployed for batch correction of scRNA-seq data ranging from traditional matrix factorisation, such as PCA or NMF, to deep learning methods, such as variational autoencoders. For a survey, see [4,9]. A common feature of these methods is that they involve dimension reduction of the data followed by graph-based clustering using, e.g., k-nearest neighbour graphs. Batch correction is applied on the distance between cluster centroids under a suitable metric. However, these methods typically require long computational times, thus limiting their use for large-scale single-cell studies [10], and can lack robustness, with performance highly dependent on data pre-processing and gene filtering; on specific choices of hyper-parameters; and on fine-tuning for particular sequencing technologies [9].

An additional, desirable requirement in dimensionality reduction of scRNA-seq data sets is that latent factors are interpretable in both the cell and gene spaces, so that they can be used to generate biologically meaningful data-driven hypotheses linking gene expression and cell phenotypes. However, many of the current methods lack interpretability due to the signed nature and lack of sparsity of the latent factor scores in the gene or cell spaces. Several methods have been proposed to improve the interpretability of PCA-based methods. In particular, NMF ensures that the latent factors learn a part-based structure, with non-negative scores, but with no guarantee of sparsity. These problems are compounded and especially relevant when dealing with data integration of cell types across technical batches.

In this paper, we focus on dimensionality reduction of scRNA-seq data sets and study the capability of different matrix factorisation methods to induce robust and interpretable latent factors that can simplify downstream analyses. We first carry out a computational examination of dimensionality reduction based on Hierarchical Poisson Factorisation (HPF) [11] for single-batch data sets in comparison to other methods. We then introduce Integrative Hierarchical Poisson Factorisation (IHPF), an extension to HPF that allows data integration and provides a robust, flexible and interpretable method for the analysis of multi-batch scRNA-seq data sets under different data integration scenarios.

The paper is organised as follows. In Section 2, we provide background on existing factorisation methods for dimensionality reduction pertaining to scRNA-seq analyses. In Section 3, we describe our algorithm for Integrative Hierarchical Poisson Factorisation (IHPF), our numerical experiments and data sets. In Section 4, we first demonstrate the advantages of HPF for single-batch scRNA-seq data, and then how IHPF can be used to carry out dimensionality reduction under different data integration scenarios with improved performance, robustness and interpretability. We conclude with a brief Discussion in Section 5.

2. Background.

2.1. Matrix factorisation and dimensionality reduction. Consider a data set consisting of N samples, each of them described by $M \gg 1$ features. In the case of

scRNA-seq, the N samples are cells of different types, and each of the M features is the transcription count of a particular gene. The data is compiled into a data matrix X of dimensions $N \times M$.

Matrix factorisation methods attempt to find $\widehat{X}$, an approximation of the data set X in terms of two matrices W and V with reduced dimension (rank) $K \ll M$ such that their product is close to the original matrix:

$$X_{N\times M} \approx W_{N\times K} V_{K\times M} =: \widehat{X}. \tag{2.1}$$

The matrix W contains the cell scores and the matrix V contains the gene scores with respect to the K *latent factors* that recapitulate (approximately) the information contained in the M features.

PCA and NMF are two classic methods that produce such a matrix factorisation, which differ in how the approximation (2.1) is computed. PCA minimises the Frobenius norm

$$\min_{W,V} \|X - WV\|_F, \tag{2.2}$$

and it is well known that the PCA approximations are obtained through the singular value decomposition (SVD) of the matrix X, i.e., the K factors are obtained from the top K singular vectors of X ordered in decreasing order of their singular values [12]. PCA-based methods have been widely used with scRNA-seq data sets, typically after applying a log-transformation to the data counts. However, the cell and gene scores (W and V) are signed and are not sparse which, together with the log-transformation, reduce the interpretability of the latent space.

Because in scRNA-seq the data matrix is non-negative element-wise ($X \geq 0$), a factorisation that captures this property is desirable. NMF proposes such a factorisation; it minimises the Frobenius norm but imposing non-negativity (element-wise) of the factorisation matrices [13]:

$$\min_{W,V} \|X - WV\|_F \quad \text{subject to} \quad W \geq 0,\, V \geq 0. \tag{2.3}$$

It has been shown that NMF learned part-based representations, which represent positive contributions of the latent factors to the samples [14]. However, there is no guarantee of sparsity in the NMF approximation.

2.2. Probabilistic viewpoint of factorisation. Dimensionality reduction (and associated matrix factorisations) also has probabilistic interpretations, specifically in terms of probabilistic factor analysis (FA) and hierarchical models.

It has been shown [15] that PCA can be understood from a probabilistic perspective as a generative model where the samples are noisy observations of a combination of latent factors that are multivariate Gaussian [16, 17]:

$$\begin{aligned} X_{N\times M} &\sim \text{Gaussian}(WV, \sigma^2\mathbb{I}) \\ W_{N\times K} &\sim \text{Gaussian}(0, \mathbb{I}) \\ V_{K\times M} &\sim \text{Gaussian}(0, \mathbb{I}) \\ \sigma &\sim \text{log-Normal}(0, 1). \end{aligned}$$

NMF also has a probabilistic interpretation related to Poisson factor analysis (PFA) [18]. It can be shown that under NMF, X is assumed to be a matrix of Poisson

random variables with rates given by a factorisation into two matrices Θ and Δ of Gamma distributed random variables:

$$X_{N\times M} \sim \text{Poisson}(\Theta\,\Delta) \tag{2.4}$$

$$\Theta_{N\times K} \sim \text{Gamma}(1,\infty) \tag{2.5}$$

$$\Delta_{K\times M} \sim \text{Gamma}(1,\infty). \tag{2.6}$$

Here, Θ and Δ correspond, respectively, to cell and gene scores onto the K latent factors. Note that Poisson-Gamma random variables are related to the Negative Binomial distribution, which is widely used to model scRNA-seq counts.

This probabilistic interpretation has been extended recently into the Hierarchical Poisson Factorisation (HPF) [11], which uses a hierarchical Gamma prior:

$$X_{N\times M} \sim \text{Poisson}\,(\Theta\,\Delta) \tag{2.7}$$

$$\Theta_{N\times K} \sim \text{Gamma}(a,\Xi) \tag{2.8}$$

$$\Xi_{N\times K} \sim \text{Gamma}(a',b') \tag{2.9}$$

$$\Delta_{K\times M} \sim \text{Gamma}(c,\mathrm{Z}) \tag{2.10}$$

$$\mathrm{Z}_{K\times M} \sim \text{Gamma}(c',d') \tag{2.11}$$

where the cell scores Θ follow a Gamma distribution with shape parameter a and a Gamma distributed rate Ξ. Similarly, the gene scores Δ are Gamma distributed with shape parameter c and a Gamma distributed rate Z. HPF has been shown to perform better than NMF when applied to sparse data sets since the hierarchical Gamma prior is more flexible in modelling zero-inflated data counts [11].

2.3. Data integration. The integration of data sets obtained with different technologies under diverse experimental settings poses additional challenges, as we aim to capture biologically meaningful signals in feature (gene) space under technical batch effects in sample (cell) space. Traditionally, integrative PCA (IPCA) was developed for batch correction in bulk RNA transcriptomics data, but it does not perform satisfactorily for scRNA-seq data due to the high sparsity of single cell data [2].

Integrative NMF (INMF) was introduced as an extension of NMF to harmonise data sets that share their feature space [19]. Let us consider S batches (i.e., data sets collected from different sources) with non-negative data matrices $X_s \in \mathbb{R}_+^{N_s\times M}$, $s = 1,\ldots,S$, so that $N = \sum_{s=1}^{S} N_s$.

As above, we want to find a description of the data set in terms of K latent factors, by obtaining non-negative matrices $W \in \mathbb{R}_+^{N\times K}$, $V_s \in \mathbb{R}_+^{K\times M}$, $s = 1,\ldots,S$, and $B \in \mathbb{R}_+^{K\times M}$ containing, respectively, the cell scores, the gene scores specific to each data set, and the gene scores common to all the data sets. The INMF factorisation is achieved by minimising the following loss function:

$$\sum_{s=1}^{S} \|X_s - W(\mathbf{1}_s,:)(V_s+B)\|_F + \frac{1}{\alpha}\sum_{s=1}^{S} \|W(\mathbf{1}_s,:)V_s\|_F\ . \tag{2.12}$$

where $\mathbf{1}_s$ is an indicator function for the cells in batch s and therefore $W(\mathbf{1}_s,:) \in \mathbb{R}_+^{N_s\times K}$ is the submatrix containing the cell scores of cells in batch s. The hyper-parameter α, denoted as the *noise ratio*, regularises the magnitude of the components of the different data sets and serves to assign the variability in the data between cell types and batches.

3. Methods.

3.1. Integrative Hierarchical Poisson Factorisation (IHPF). As shown in our numerical study in Section 4.1 below, HPF presents several advantages for dimensionality reduction of scRNA-seq data, especially with regard to robustness, sparsity and interpretability. However, HPF does not perform well when integrating data sets of different provenance (e.g., with batch or other sample effects) [20]. In such scenarios, a series of *ad hoc* methods (e.g., clustering of biologically related latent factors) are applied to ameliorate batch effects.

Here we introduce an algorithm for Integrative Hierarchical Poisson Factorisation (IHPF), which extends the applicability of HPF to allow for data integration in a systematic way.

Given data matrices $X_s \in \mathbb{R}_+^{N_s \times M}$, $s = 1, \ldots, S$ corresponding to S batches, IHPF assumes hierarchical Gamma priors on the cell scores Θ and, inspired by (2.12), it decomposes the gene scores into two components: $\Delta^{(s)}$ accounting for batch-specific effects, and B capturing shared effects. Both $\Delta^{(s)}$ and B are described by a hierarchical Gamma prior:

$$
\begin{aligned}
X^{(s)}_{N_s \times M} &\sim \text{Poisson}\left(\Theta(\mathbf{1}_s, :)\left(\Delta^{(s)} + B\right)\right),\ s = 1, \ldots, S \\
\Theta_{N \times K} &\sim \text{Gamma}(a, \Xi) \\
\Xi_{N \times K} &\sim \text{Gamma}(a', b') \\
\Delta^{(s)}_{K \times M} &\sim \text{Gamma}(c, \mathrm{Z}^{(s)}), \quad s = 1, \ldots, S \\
\mathrm{Z}^{(s)}_{K \times M} &\sim \text{Gamma}(c', d'), \quad s = 1, \ldots, S \\
\mathrm{B}_{K \times M} &\sim \text{Gamma}(c, \mathrm{H}) \\
\mathrm{H}_{K \times M} &\sim \text{Gamma}(c', d'),
\end{aligned}
$$

where, again, $\Theta(\mathbf{1}_s, :) \in \mathbb{R}_+^{N_s \times K}$ is the submatrix containing the cell scores of cells in batch s. The pseudo-code for the IHPF generative model is presented in Algorithm 1.

Algorithm 2 describes the coordinate ascent algorithm [11] used to infer the shape and rate parameters of the six Gamma posteriors for the gene and cell scores:

$$\xi_i|X,\ \theta_{ik}|X,\ \eta_g|X,\ \zeta_{gs}|X,\ \beta_{gk}|X,\ \delta_{gsk}|X,$$

where each is Gamma distributed and the shape and rate parameters are defined accordingly for each posterior, e.g.,

$$\xi_i|X \sim \text{Gamma}(\xi_i^{shp}, \xi_i^{rte}).$$

The multinomial distributions used in Algorithm 2 for the allocation of gene and cell counts to each latent factor are initialised with a Dirichlet prior with weights given by the noise ratio $\alpha \in [0, 1]$. This hyper-parameter represents the relative importance of data set specific gene latent factors towards the shared factors, i.e., how much variance in the data to be attributed to technical noise between data sets. Taking the limit $\alpha \to 0$ recovers the original HPF.

ALGORITHM 1. *Generative model of IHPF (integrative hierarchical Poisson factorisation) for single-cell data.*

Algorithm 1: *Generative model of IHPF*

Set hyper-parameters a, a', b', c, c', d'
for each cell i **do**
  Sample rate of cell score: $\xi_i \sim \text{Gamma}(a', b')$
  for each factor k **do**
    Sample cell score: $\theta_{ik} \sim \text{Gamma}(a, \xi_i)$
  end for
end for
for each gene g and batch s **do**
  Sample rate of shared gene score:
    $\eta_g \sim \text{Gamma}(c', d')$
  Sample rate of batch-specific gene score:
    $\zeta_{gs} \sim \text{Gamma}(c', d')$
  for each factor k **do**
    Sample shared gene score:
      $\beta_{gk} \sim \text{Gamma}(c, \eta_g)$
    Sample batch-specific gene score:
      $\delta_{gsk} \sim \text{Gamma}(c, \zeta_{gs})$
  end for
end for
for cell i and gene g **do**
  Denote batch label of cell i to be s
  Sample observed expression level:
    $x_{ig} \sim \text{Poisson}\left(\sum_{k=1}^{K} \theta_{ik}(\beta_{gk} + \delta_{gsk})\right)$
end for

ALGORITHM 2. *Coordinate Ascent Algorithm for likelihood inference in IHPF.*

Algorithm 2: *Coordinate Ascent for IHPF inference*

Set the number of factors K, the noise ratio α, and the hyper-parameters a, a', b', c, c', d'.
Initialise allocations to each factor with Dirichlet prior:

$$(\phi_{ig1}, \ldots, \phi_{igK}, \omega_{ig1}, \ldots, \omega_{igK}) \sim \mathrm{Dir}(1, \ldots, 1, \alpha, \ldots, \alpha)$$

repeat
  for each gene g **do**
    Compute shapes and rates of Gamma posterior of shared gene scores:

$$\beta_{gk}^{shp} = c + \sum_i x_{ig}\phi_{igk}; \qquad \beta_{gk}^{rte} = \frac{\eta_g^{shp}}{\eta_g^{rte}} + \sum_i \frac{\theta_{ik}^{shp}}{\theta_{ik}^{rte}}$$

$$\eta_g^{shp} = c' + Kc; \qquad \eta_g^{rte} = d' + \sum_k \frac{\beta_{gk}^{shp}}{\beta_{gk}^{rte}}$$

    for each batch s **do**
      Compute shapes and rates of Gamma posterior of batch-specific gene scores:

$$\delta_{gsk}^{shp} = c + \sum_{i \in s} x_{ig}\omega_{igk} \qquad \delta_{gsk}^{rte} = \frac{\zeta_g^{shp}}{\zeta_g^{rte}} + \sum_{i \in s} \frac{\theta_{ik}^{shp}}{\theta_{ik}^{rte}}$$

$$\zeta_{gs}^{shp} = c' + Kc \qquad \zeta_{gs}^{rte} = d' + \sum_k \frac{\delta_{gsk}^{shp}}{\delta_{gsk}^{rte}}$$

    end for
  end for
  for each cell i **do**
    Let s denote the batch label of cell i
    Compute shapes and rates of Gamma posterior of cell scores:

$$\theta_{ik}^{shp} = a + \sum_g x_{ig}(\phi_{igk} + \omega_{igk}) \qquad \theta_{ik}^{rte} = \frac{\xi_i^{shp}}{\xi_i^{rte}} + \sum_g \left(\frac{\beta_{gk}^{shp}}{\beta_{gk}^{rte}} + \frac{\delta_{gsk}^{shp}}{\delta_{gsk}^{rte}} \right)$$

$$\xi_i^{shp} = a' + Ka \qquad \xi_i^{rte} = b' + \sum_k \frac{\theta_{ik}^{shp}}{\theta_{ik}^{rte}}$$

  end for
  for each cell i and gene g such that $x_{ig} > 0$ **do**
    Use the digamma function $\Psi(\cdot)$ to compute the multinomial distributions:

$$\phi_{igk} \propto \exp\left(\Psi(\theta_{ik}^{shp}) - \log \theta_{ik}^{rte} + \Psi(\beta_{gk}^{shp}) - \log \beta_{gk}^{rte} \right)$$

$$\omega_{igk} \propto \exp\left(\Psi(\theta_{ik}^{shp}) - \log \theta_{ik}^{rte} + \Psi(\delta_{gsk}^{shp}) - \log \delta_{gsk}^{rte} \right)$$

  end for
until convergence

3.2. Data sets and data integration scenarios. We have collected 16 scRNA-seq data sets of different size, sequencing technology, and scope from the literature (Table 1), and we have used them to benchmark our method in two ways.

TABLE 1

Data sets used to benchmark various methods for data integration. Data sets are combined to produce integrated data sets to conform a specific scenario reflecting different assumptions on the biological relationship between data sets.

ID	Description	Technology	Cells (N)	Genes (M)	Scenario	Reference
1	Mouse Jurkat cells	10X Genomics	2885	32378	A	Zheng 1 [21]
2	Mouse 293t cells	10X Genomics	3257	32378	A	Zheng 2 [21]
3	Mouse 50% Jurkat/50% 293t cells	10X Genomics	3388	32378	A	Zheng 3 [21]
4	Human pancreas	InDrop	8569	20126	B	Baron [22]
5	Human pancreas	Cel-seq2	2449	19140	B	Muraro [23]
6	Human pancreas	Fluidigm C1	638	25463	B	Lawlor [24]
7	Human pancreas	Smart-seq2	2989	26179	B	Segerstolpe [25]
8	Human pancreas	Celseq	1276	20148	B	Grün [26]
9	Human CD19+ B cells	10X Genomics	2261	32378	C	Zheng 5 [21]
10	Human CD14+ monocytes	10X Genomics	295	32378	C	Zheng 6 [21]
11	Human CD4+ helper T cells	10X Genomics	3713	32378	C	Zheng 7 [21]
12	Human CD56+ natural killer cells	10X Genomics	6657	32378	C	Zheng 8 [21]
13	Human CD8+ cytotoxic T cells	10X Genomics	3990	32378	C	Zheng 9 [21]
14	Human CD4+CD45RO+ memory T cells	10X Genomics	3628	32378	C	Zheng 10 [21]
15	Human CD4+CD25+ regulatory T cells	10X Genomics	3365	32378	C	Zheng 11 [21]
16	Human PBMC	10X Genomics	2293	32378	C	Zheng 4 [21]

Firstly, and before considering data integration, we apply HPF separately to data sets ID 3-8 in Table 1 obtained with different sequencing technologies.

Secondly, we consider data integration and apply IHPF to combinations of the data sets in Table 1 grouped to conform three biological scenarios:

- *Scenario A:* a collection of data sets sequenced with the same technology and containing different abundance of the same cell types;
- *Scenario B:* a collection of data sets containing the several cell types sequenced with different technologies;
- *Scenario C:* a collection of data sets sequenced with the same technology in which there is little overlap in cell types between batches.

Table 2 summarises the three data scenarios. In our case, Scenario A consists of 3 batches of mouse cells (ID 1-3) sequenced with the 10X Genomics technology and comprising two cell types (293t and jurkat) mixed in different proportions. Scenario B consists of 5 batches of human pancreas cells (ID 4-8) of 10 different types obtained with different sequencing technologies. Scenario C consists of 8 batches (ID 9-16) sequenced with with the same technology (10X Genomics) and comprising 6 different cell types highly concentrated in particular batches.

TABLE 2

Summary of the three scenarios for data integration constructed from individual data sets from Table 1.

	Scenario A (ID 1-3)	Scenario B (ID 4-8)	Scenario C (ID 9-16)
Total number of cells, N	9530	15921	26202
Number of common genes, M	13130	5057	8234
Number of batches, S	3	5	8
Number of cell types, C	2	10	6
AMI cell types *vs.* batches	0.498	0.032	0.842

The scenarios reflect experimental situations with different alignment between cell types and batch labels. In Scenario A, batches and cell types have some alignment and, importantly, there is good overlap of cell types across batches. In Scenario B, the situation is complex, with many batches and cell types and with differing overlap of cell types across batches. In Scenario C, batches and cell types are highly aligned with little overlap of cell types across batches. The Adjusted Mutual Information (AMI) between cell types *vs.* batches in Table 2 measures the different alignment.

3.3. Numerical Experiments. Our numerical experiments fall into two broad categories:

1. *Dimensionality reduction of single-batch data sets using HPF:* Before considering data integration, we study the performance of HPF for dimensionality reduction of single-batch scRNA-seq data sets by analysing 6 data sets (ID 3-8). We evaluate the quality of data reconstruction and the robustness with respect to hyper-parameters and data pre-processing and compare the results to NMF and PCA.
2. *Data integration of multi-batch data sets under different scenarios:* To study data integration under Scenarios A-C. we merge the corresponding data sets, keeping the intersection of genes across data sets, and evaluate the quality of clusterings and the output of visualisations induced by latent factors extracted with four dimension reduction methods (IHPF, INMF, HPF, PCA). We also evaluate the impact of the noise ratio in data integration across scenarios, and investigate the block-structure of latent factors in both sample (cell) and feature (gene) spaces for the different methods.

3.3.1. Evaluating the quality of models. To measure the quality of the reconstructed data $\widehat{X}$ (obtained by any model) relative to the observed data X we use:

- *Average explained deviance:* Average of the explained deviance computed from the Kullback-Leibler (KL) divergence and the Frobenius norm:

$$\overline{ED} = \frac{1}{2}\left(\frac{\mathbb{KL}(X||\overline{X}) - \mathbb{KL}(X||\widehat{X})}{\mathbb{KL}(X||\overline{X})} + \frac{||X-\overline{X}||_F - ||X-\widehat{X}||_F}{||X-\overline{X}||_F}\right) \tag{3.1}$$

  where $\overline{X} = \frac{1}{M}X\mathbf{1}\mathbf{1}^T$ is the matrix of cell averages, $||\cdot||_F$ denotes the Frobenius norm, and the KL divergence between matrices X and Y is:

$$\mathbb{KL}(X||Y) = \sum_{i,j} X_{ij} \log \frac{X_{ij}}{Y_{ij}}.$$

  A large average explained deviance ($\overline{ED} \to 1$) reflects the fact the reconstruction $\widehat{X}$ provides a better approximation than the average $\overline{X}$.
- *Kolmogorov-Smirnov (KS) statistics:* Given two samples of sizes n_1 and n_2 taken from distributions with empirical cumulative distribution functions (eCDFs) F_{1,n_1} and F_{2,n_2}, respectively, the KS statistic is defined as:

$$D_{n_1,n_2} = \sup_x |F_{1,n_1}(x) - F_{2,n_2}(x)|\,. \tag{3.2}$$

A low value of the KS statistic indicates a small difference between the two samples [27].

Here, we compute cell and gene KS statistics:

- *Cell KS statistic:* For each cell $i = 1, \ldots, N$, compute the coefficient of variation (CV) of the observed and reconstructed counts across genes, i.e., $CV_{\text{cell},i} = \sigma(X(i,:))/\mu(X(i,:))$ and $\widehat{CV}_{\text{cell},i} = \sigma(\widehat{X}(i,:))/\mu(\widehat{X}(i,:))$, where $\sigma(\cdot)$ and $\mu(\cdot)$ denote the standard deviation and mean. Construct the empirical cumulative distribution function (eCDF) $F_{\text{cell},N}$ of $\{CV_{\text{cell},i}\}_{i=1}^{N}$. Similarly, construct $\widehat{F}_{\text{cell},N}$, the eCDF of $\{\widehat{CV}_{\text{cell},i}\}_{i=1}^{N}$. The cell KS statistic is given by:

$$D_{\text{KS,cell}} := \sup_x \left| F_{\text{cell},N}(x) - \widehat{F}_{\text{cell},N}(x) \right|. \tag{3.3}$$

- *Gene KS statistic:* This is built similarly to the cell KS statistic. Construct $F_{\text{gene},M}$, the eCDF of $CV_{\text{gene},j} = \sigma(X(:,j))/\mu(X(:,j))$, i.e., the CV across cells of each gene $j = 1, \ldots, M$ from the observed counts. Construct $\widehat{F}_{\text{gene},M}$, the eCDF of $\widehat{CV}_{\text{gene},j} = \sigma(\widehat{X}(:,j))/\mu(\widehat{X}(:,j))$, i.e., the CV across cells for each gene $j = 1, \ldots, M$ from the reconstructed counts. The gene KS statistic is:

$$D_{\text{KS,gene}} := \sup_x \left| F_{\text{gene},M}(x) - \widehat{F}_{\text{gene},M}(x) \right|. \tag{3.4}$$

We also use the summary KS statistic $\overline{D}_{\text{KS}}$ defined as the average of gene and cell KS statistics:

$$\overline{D}_{\text{KS}} = \frac{1}{2}\left(D_{\text{KS,cell}} + D_{\text{KS,gene}}\right). \tag{3.5}$$

3.3.2. Implementation of algorithms. Our computational implementation of IHPF builds upon the HPF Python code provided by the authors of [11], by extending the model to contain batch-specific factors (Algorithms 1 and 2) regularised by the noise ratio α.

The hyper-parameters of HPF and IHPF are set to the defaults in Ref. [11], unless otherwise indicated. The noise ratio of IHPF is set to 0.1, except when studying the importance of the noise ratio for clustering performance under data scenarios (Section 4.2.2).

For PCA and NMF we use the implementations in scikit-learn. For INMF we provide an implementation in Python compatible with scikit-learn. The noise ratio of INMF is set to the default value of 1.0, as suggested in [19]. Pre-processing of the single-cell data sets is done using Scanorama [8] and Scanpy [28]. KS statistics and other statistical tests are computed using Scipy.

Computer code for all methods used in this paper is available at https://github.com/barahona-research-group/scIHPF.

4. Results.

4.1. HPF-based dimensionality reduction of single-batch scRNA-seq data. To characterise the performance and robustness of HPF when applied to single data sets, we use six of the data sets (ID 3-8) in Table 1 which were all obtained using different sequencing technologies. We show that HPF provides consistent and improved performance for scRNA-seq data across sequencing technologies compared to PCA and NMF. Furthermore, HPF displays low sensitivity to model hyper-parameters and to data pre-processing (e.g., number of pre-selected genes).

4.1.1. Robustness of HPF to hyper-parameters across sequencing technologies. We first study the robustness of HPF (2.7)–(2.11) under changes of two (of its six) hyper-parameters: the cell shape parameter a and the gene shape parameter c. The other hyper-parameters (a', b', c', d') are fixed to the values recommended in the scHPF package [11].

We infer HPF models for the six data sets (ID 3-8) with $K = 2, 4, 6, 8, 10$ latent factors varying the hyper-parameters a and c between $[0.1, 2.1]$. We then evaluate the performance of the models using both the cell KS statistic (3.3) and the gene KS statistic (3.4).

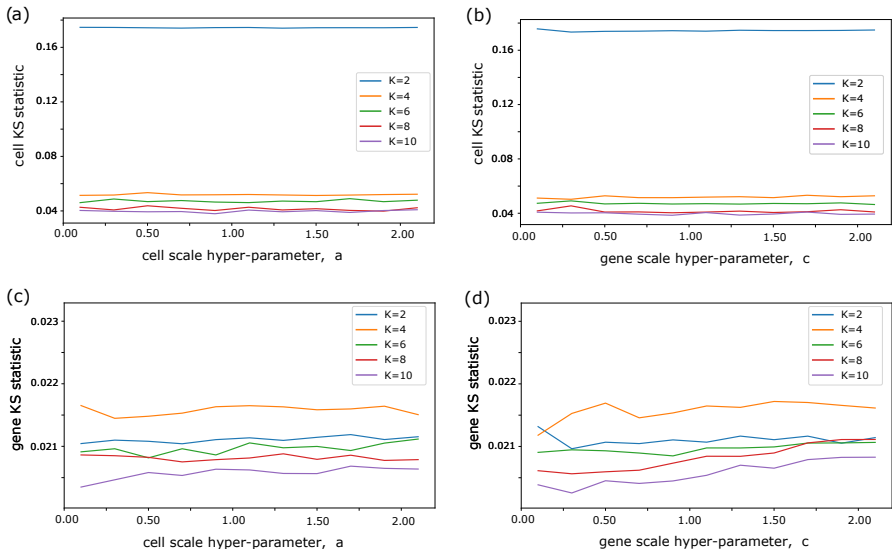


FIG. 3. *(a)-(b) Cell KS statistic (3.3) and (c)-(d) gene KS statistic (3.4) for HPF models of data set ID 3 obtained with varying cell and gene scale hyper-parameters a and c and different number of latent factors K. For all values of K, changing the hyper-parameters a, c has almost no effect on the model performance (as measured by the KS statistics).*

As an illustration of the results, Figure 3 shows the performance of HPF on data set ID 3. We see that HPF is robust to changes in the hyper-parameters a and c, with little change in the outcome for all values of K. The other five data sets (ID 4-8), which were obtained experimentally with other sequencing technologies, show similar behaviour (see Figs. 11-15 in the SI). Hence HPF is shown to be robust to the choice of hyper-parameters across different sequencing technologies.

Once established the robustness of HPF to the hyper-parameters, we set $a = c = 0.3$ as suggested in the scHPF package [11] and keep these parameters fixed throughout the rest of the paper.

4.1.2. Robustness of HPF to data pre-processing. Another source of variability in dimensionality reduction for scRNA-seq data is the dependence on pre-processing steps (e.g., normalisation, imputation, filtering) which are present in most scRNA-seq analysis pipelines. A common pre-processing step is *feature selection*, i.e., reducing the number of genes under consideration by filtering and selecting genes that are relevant to the biological process of interest. Such feature selection is intended to remove noisy features whilst reducing computational time and improving algorithmic convergence. This gene filtering step is typically based on a mixture of

statistical criteria (e.g., choosing genes that display high variability) together with highly specific criteria that incorporate *ad hoc* biological knowledge particular to the different single cell studies (see e.g., the two HPF studies [11, 20] for an example of such specific biological criteria). Such gene selection can have a large impact in the ensuing dimensionality reduction.

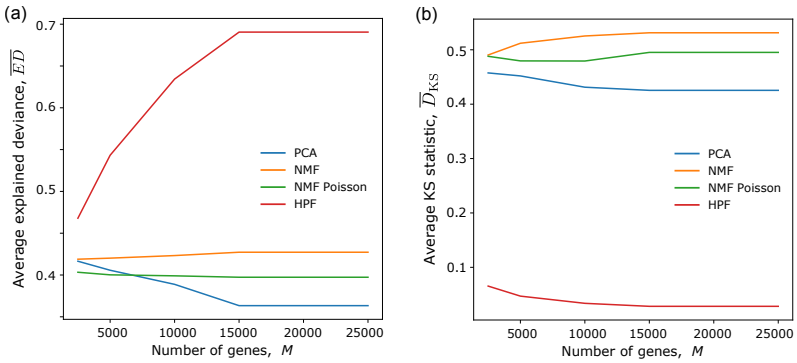


FIG. 4. *(a) Average explained deviance (3.1) and (b) average KS statistic (3.5) of different matrix factorisation methods (PCA, NMF, NMF-Poisson, HPF) applied to data set set ID 3 computed using the top M genes with highest variability, where M is increased from 2500 to 25000. The number of latent factors is $K = 10$ for all methods. HPF performs best across methods, as indicated by the highest $\overline{ED}$ and lowest $\overline{D}_{KS}$.*

To evaluate the effect of feature (gene) selection on dimensionality reduction methods, we order the genes in decreasing order of variability, and infer models using the top M genes with highest coefficient of variation, with varying $M = 2500, 5000, 10000, 15000, 17500, 25000$. Figure 4 shows the numerical experiments on data set ID 3. We find that HPF performs better compared to other dimensionality reduction methods, namely, PCA, NMF with Frobenius norm as loss function (NMF), and NMF with KL divergence as loss function (NMF-Poisson). HPF has the best performance (both higher average explained deviance $\overline{ED}$ (3.1) and lower average KS statistic $\overline{D}_{\mathrm{KS}}$ (3.5)), and is the only method that performs consistently better as more genes are included. The results for the other five data sets (ID 4-8) can be found in the SI (Figs. 16-20). In general, HPF is the best performing method whereas PCA has the lowest explained deviance and is the least robust method under addition of more genes. NMF and NMF-Poisson perform slightly better than PCA but their performance drops as the number of genes increases.

4.1.3. Summary of comparative performance. Finally, in Table 3 we summarise the performance of HPF compared to PCA, NMF and NMF-Poisson (NMF-P) on the six data sets (ID 3-8), which were obtained experimentally with different sequencing technologies. HPF displays the best performance across the six data sets obtained with different technologies according to both the average explained variance $\overline{ED}$ and the average KS statistic $\overline{D}_{\mathrm{Ks}}$. The performance of HPF is substantially better across all technologies and especially so for data sets sequenced with the newer 10X genomics, inDrop and celseq2/celseq technologies (ID 3,4,5,8) which deliver sparser measurements. PCA has the worst performance in $\overline{ED}$ but only marginally below that of NMF-Poisson and NMF, whereas NMF is the worst performing as measured by $\overline{D}_{\mathrm{KS}}$.

TABLE 3

(a) Average explained deviance and (b) average KS statistic of different matrix factorisation methods performed on data sets obtained experimentally using six different sequencing technologies (10X genomics, inDrop, celseq2, fluidigmc1, smartseq2 and celseq, see Table 1). For all methods, the number of selected genes is fixed at $M = 25000$ and the number of latent factors is taken as $K = 10$. HPF performs best in all cases; the larger the explained variance and the smaller the KS statistic, the better the reconstruction. The best performing method is shown in boldface.

(a) Average explained deviance, $\overline{ED}$

Method	ID 3	ID 4	ID 5	ID 6	ID 7	ID 8	*Average*
PCA	0.363	0.399	0.317	0.385	0.340	0.206	0.335
NMF	0.427	0.436	0.380	0.453	0.430	0.315	0.407
NMF-P	0.397	0.414	0.310	0.429	0.399	0.239	0.365
HPF	**0.691**	**0.743**	**0.616**	**0.516**	**0.584**	**0.647**	**0.633**

(b) Average KS statistic, $\overline{D}_{\mathrm{KS}}$

Method	ID 3	ID 4	ID 5	ID 6	ID 7	ID 8	*Average*
PCA	0.425	0.489	0.408	0.423	0.355	0.409	0.418
NMF	0.531	0.537	0.463	0.471	0.419	0.420	0.473
NMF-P	0.495	0.532	0.461	0.448	0.440	0.409	0.464
HPF	**0.028**	**0.088**	**0.124**	**0.383**	**0.332**	**0.084**	**0.173**

4.2. Data integration of scRNA-seq data sets using IHPF. In the previous section, we showed that HPF performs well for dimensionality reduction of single data sets of scRNA-seq counts. We now consider the question of data integration. We will first study dimensionality reduction under integrative approaches (INMF and IHPF) and non-integrative approaches (HPF and PCA) under the three biological scenarios delineated in Section 3.2, and compare the quality of the extracted latent factors for clustering of cell types and visualisation. We will then consider the effect of the noise ratio in integrative schemes, specifically under different data scenarios with different alignment between cell types and batches. Finally, we show that the sparsity intrinsic to IHPF has the effect of inducing a dual block structure in the latent factors with respect to both gene and cell spaces, thus enhancing interpretability.

4.2.1. Latent factors for clustering and visualisation. Dimensionality reduction is a key step in computational pipelines for (unsupervised) clustering of scRNA-seq data sets. Ideally, one aims to extract a few latent factors that recapitulate the cell labels as fully as possible, and use the factors as data-driven coordinates on which to base classification, clustering, visualisations or coarse-grained descriptions that relate cell phenotypes to coherent groups of genes. As discussed in Section 4.1, PCA- or NMF-based factors have had only limited success with scRNA-seq data due to the high levels of noise and zeroes in the data [9]. To circumvent such limitations, complex pipelines have been developed including the use of similarity graphs and graph-based partitioning [8, 29, 30]. However, such methods encounter problems under data integration scenarios since the factors could capture mostly batch information, so that the clusters can fail to recapitulate the cell types.

Rather than multi-step pipelines, we focus here on extracting latent factors that capture cell type variability in the presence of batch effects under data integration Scenarios A-C (Table 2). We evaluate the extracted factors obtained with four dimensionality reduction methods: two data integration methods (INMF, IHPF) that take into account the presence of batches, and two non-integrative methods (HPF,

PCA) that ignore batch information. In all cases, we extract $K = C$ factors, where C is the number of cell types present in the data, and we used the row-normalised cell scores (from matrices W or Θ) to cluster the cells using k-means clustering with $k = C$.

TABLE 4

Silhouette coefficient of the clusterings from latent factors by applying k-means clustering to K normalised cell latent factors obtained from four matrix factorisation models. In the clustering, the number of latent factors K and the number of clusters k in k-means are set to equal the actual number of cell types C, i.e., $K = k = C$ (see Table 2).

	Method	Scenario A (ID 1-3)	Scenario B (ID 4-8)	Scenario C (ID 9-16)
Data integration	INMF	0.577	-0.431	0.140
	IHPF	0.870	0.410	0.768
No integration	HPF	0.999	0.890	0.994
	PCA	0.564	-0.347	0.695

Firstly, we use the silhouette coefficient (SC) to evaluate the suitability of the cell latent factor scores to induce well-separated clusters. SC measures the ratio of intra- and inter-cluster distances and is bounded by $-1 \leq \text{SC} \leq 1$. Table 4 shows that, for Scenario A, both HPF and IHPF cell scores lead to well-separated clusters (with SC close to 1), and PCA and INMF give clusters with reasonable separation. For Scenarios B-C, HPF and IHPF cell scores lead to reasonably well-separated clusters, whereas PCA leads to reasonable separability only in Scenario C. In general, INMF cell scores have poorer cluster separability.

Whereas the SC in Table 4 measures intrinsic cluster separability, it does not evaluate cluster quality with respect to the underlying cell types or, alternatively, to the batch labels. To quantify this, we compute in Table 5 the Adjusted Mutual Information (AMI) between the k-means clusterings obtained from the cell scores of four dimensionality methods with respect to: (a) batch labels and (b) cell types.

Table 5A shows that the cell clusterings from both integrative methods (IHPF and INMF) have smaller AMI with respect to the batch clusters than non-integrative methods (PCA and HPF). This is a direct confirmation that both integrative methods correct batch effects. Importantly, however, IHPF reduces batch effects in an adaptive manner, i.e., it only eliminates genuine batch effects that are not aligned with the cell types. This can be seen by comparing the AMI of IHPF clustering *vs.* batches in Table 5A and the AMI of cell types *vs.* batches in Table 2. In contrast, INMF over-corrects batch effects when, as in Scenario C, batch variability is aligned with the biological signal. This over-correction in INMF leads to degraded performance (as seen in Table 5B).

Table 5B shows that, among the integrative methods, IHPF has better performance than INMF in recovering the biological signal, with consistently higher AMI of clustering *vs.* cell types across all scenarios. Overall, IHPF has the best performance of all methods in Scenarios A and B, but not in Scenario C. This is expected, and a direct consequence of the structure of the composite data sets. In Scenarios A-B, batch correction is required, since the cell types and batch labels do not align and it is advantageous to disentangle both sources of variability. Specifically, in Scenario A, whereas HPF remains batch-aligned and fails to integrate the mixed-cell batch, IHPF achieves a near perfect performance correcting batch differences while keeping the biological signal.

On the other hand, in Scenario C, where cell types and batches are highly aligned

TABLE 5

Adjusted Mutual Information (AMI) between the clustering obtained from latent factors versus (a) batches and (b) cell types. The clustering is obtained by applying k-means clustering to K normalised cell latent factors obtained from four matrix factorisation models. Two of the models use batch labels for data integration (INMF, IHPF) and the other two (HPF, PCA) do not attempt data integration. In the k-means clustering, the number of latent factors K and the number of clusters k are set equal to the actual number of cell types C, i.e., $K = k = C$ (see Table 2). We use standard values of the noise ratio for INMF ($\alpha = 1$) and IHPF ($\alpha = 0.1$).

(a) AMI of cell clustering *vs.* batch labels

	Method	Scenario A (ID 1-3)	Scenario B (ID 4-8)	Scenario C (ID 9-16)
Data integration	INMF	0.294	0.253	0.065
	IHPF	0.498	0.097	0.866
No integration	HPF	0.739	0.824	0.956
	PCA	0.739	0.493	0.935

(b) AMI of cell clustering *vs.* cell types

	Method	Scenario A (ID 1-3)	Scenario B (ID 4-8)	Scenario C (ID 9-16)
Data integration	INMF	0.200	0.199	0.043
	IHPF	**0.975**	**0.368**	0.699
No integration	HPF	0.448	0.144	0.787
	PCA	0.448	0.211	**0.875**

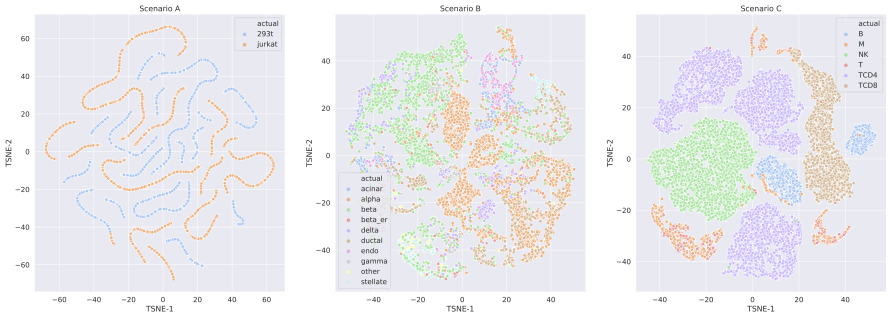
(see AMI in Table 2), integrative schemes remove some of the biological signal corresponding to cell types along with batch effects. Non-integrative models like HPF or PCA perform best in this case, although this is due to learning the cell types partly by picking up the implicit information that links batch labels and cell types in the data. Note, however, that IHPF factors can reduce batch effects without washing out the cell labels, unlike INMF. Hence IHPF performs considerably better than INMF also in Scenario C because IHPF is able to reduce batch effects in an adaptive manner, respecting batch effects that are aligned with the biological signal. Further, as shown below, the intensity of data integration through IHPF can be tuned using the noise ratio, thus allowing IHPF to ignore batch labels in cases where data integration is not advantageous.

Another important use of latent factors is as part of visualisation algorithms (e.g., TSNE) for biological exploration of data sets. Figure 5 shows the projections obtained from both IHPF and HPF factors using TSNE (a standard tool to project high-dimensional data sets) with default perplexity parameter [31]. Our results show that IHPF leads to projections with more clearly separated cell groups and less separation by batches (see Fig. 21 in SI).

In summary, IHPF factors provide robust performance across the different scenarios with improved performance over INMF, and outperforming the non-integrative methods (HPF and PCA) when the data set matches the assumptions underpinning batch correction.

4.2.2. The effect of the noise ratio in data integration schemes . The noise ratio α in INMF (2.12) and IHPF (see Algorithm 2) encodes how much of the observed data variance is attributed to the shared biological signal (i.e., cell types) and to batch-specific differences. When the noise ratio becomes zero, the batch labels are

(a) TSNE from IHPF cell scores



(b) TSNE from HPF cell scores

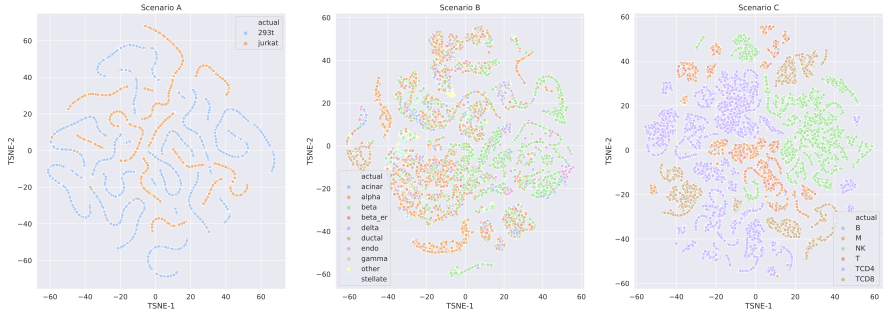


FIG. 5. *TSNE projections of Scenarios A-C obtained from cell latent factor scores from (a) IHFP and (b) HPF, and coloured by cell type.*

then ignored and IHPF and INMF recover the corresponding non-integrative results.

The noise ratio is an adjustable hyper-parameter in both integrative schemes. Figure 3 shows that IHPF consistently outperforms INMF for a range of noise ratios, as measured by the quality of the clustering compared to the cell labels. Note that IHPF recovers HPF in the limit of zero noise ratio, as expected. However, the change in the performance of IHPF under variation of the noise ratio hyper-parameter also reveals insights about the data structure of the three scenarios. For Scenario A, data integration is appropriate since cell types and batch labels are not aligned and there is a large shared common gene space; hence IHPF disentangles batch effects from biological variability (cell types) and provides a consistent improvement with respect to HPF across noise ratios. For Scenario B, IHPF still provides an advantage over HPF but the complexity of the data, with many more cell types and a reduced number of common genes (Table 2), makes the task more difficult. Finally, for Scenario C, data integration does not improve the performance. This is expected since cell types and batches are highly aligned and data integrative methods suffer from over-correction. Note that although data integration is not advantageous, IHPF still performs well, and much better than the other integrative method, INMF. Scanning the noise ratio hyper-parameter thus provides additional insight into the relationship of batch effects and the biological signal (cell types) to be detected.

4.2.3. Block structure of gene and cell latent factors. In biological applications, it is often desirable that the latent factors are both descriptive and interpretable, i.e., ideally, a few, sparse factors recapitulate the original data. In this

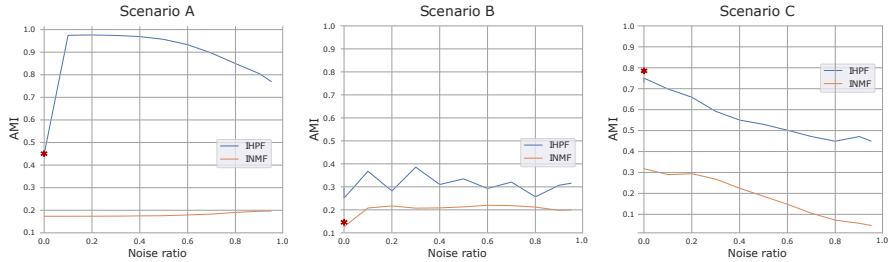


FIG. 6. *Performance of the data integration algorithms INMF and IHPF for Scenarios A–C as a function of the noise ratio hyper-parameter. For each scenario, the number of latent factors is fixed to the number of cell types ($K = C$). The noise ratio α is varied from 0.1 to 0.9 at intervals of 0.1, and two additional points are computed at $\alpha = 0.0001$ and 0.95. To alleviate local minima in the non-convex optimisation, the algorithms are run a minimum of 20 times, and the maximum across all runs is reported. As in Table 5A, we measure the performance with the AMI of the k-means clusters vs. cell labels. The red asterisks mark the HPF value (run independently) confirming that IHPF recovers HPF as the noise ratio becomes zero.*

context, a sparse factor is one that has high scores concentrated in a few variables and low (or zero) scores for the remaining variables; hence the score matrices would have a well defined *block structure*. Sparse factors can be the basis for clusters with improved interpretability that link cell types with a reduced set of genes.

Most dimensionality reduction methods (e.g., PCA) are not designed to induce sparsity in the latent factors, yet the intrinsic sparsity of IHPF makes it a good candidate for improved interpretability. To explore this idea, we examine the block structure in the matrices of cell and gene latent factor scores obtained with different dimensionality reduction algorithms. Figure 7 shows the *cell scores* for Scenarios A-C obtained with PCA, INMF and IHPF. The block-structure present in these matrices is a reflection of the fact that the cell latent factors capture information about cell types. Indeed, these cell score matrices are the basis for the clustering and visualisation presented in Section 4.2.1, and the silhouette coefficients (SC) in Figure 8 confirm that IHPF cell latent factor scores lead to well separated clusters in all Scenarios, and PCA and INMF can also produce reasonable separability (especially good for PCA in Scenario C).

In contrast, Figure 9 shows that only the IHPF *gene scores* exhibit a well defined block structure across all scenarios. Consequently, as shown by the SC in Figure 10, only IHPF gene latent factors induce well separated gene clusters associated with the extracted latent factors. Therefore IHPF learns latent factors that have a dual block-structure in both cell and gene spaces with the potential for enhanced explainability and biological interpretability by linking cell types to gene clusters.

5. Discussion. Here we have presented computational studies of dimension reduction and data integration for single-cell RNAseq data based on hierarchical Poisson factorisation (HPF). For data sets with no batch effects, we show that HPF leads to dimension reduction with improved performance compared to other factorisation methods (Table 3). In contrast to other methods, HPF performs increasingly well as more genes are included in the analysis. This feature of HPF reduces the need for pre-filtering of genes based on additional criteria (e.g., high variability), and can thus allow the use of larger sets of genes for discovery science.

We then introduced IHPF, an algorithm to perform data integration that extends HPF by modelling the technical noise between data sets with the use of latent fac-

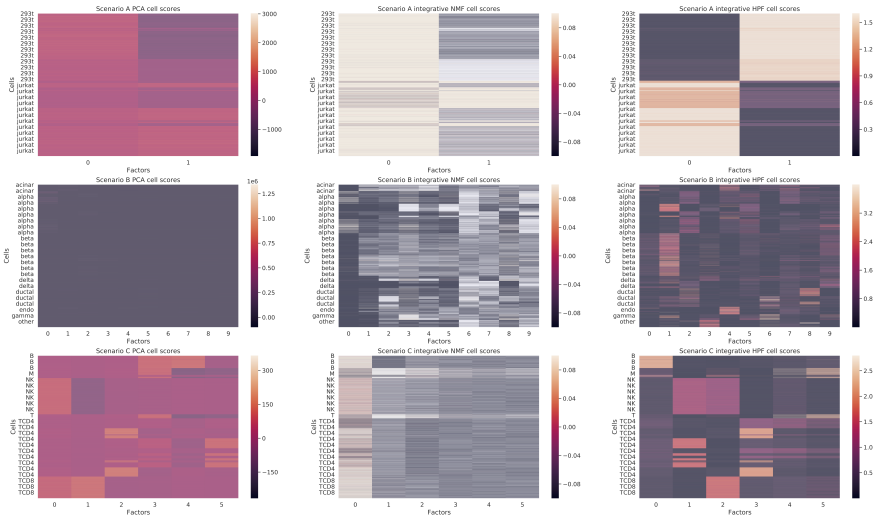


FIG. 7. *Cell latent factor scores ($N \times K$ matrices) obtained from PCA, INMF and IHPF for Scenarios A–C. The rows of the matrices correspond to cells sorted by cell type and the columns correspond to the latent factors. For each scenario the number of latent factors is equal to the number of cell types ($K = C$). The block structure of the matrices is well aligned with the cell types, especially for IHPF.*

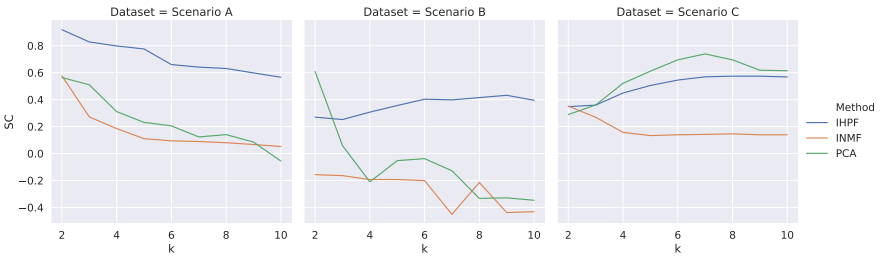


FIG. 8. *Silhouette coefficient (SC) of k-means clusterings (as a function of k) obtained from the cell scores of PCA, INMF and IHPF for Scenarios A-C. Well separated clusterings, indicated by high SC values, are obtained from IHPF cell scores in all Scenarios, and from PCA cell scores for Scenario C.*

tors adapted to batch labels. As in other data integration methods (e.g., INMF), a hyper-parameter denoted the noise ratio can be tuned to accommodate the amount of variability ascribed to batch effects *vs.* biological effects (i.e., cell types). When the noise ratio becomes negligible, the batch labels are ignored and IHPF recovers the results of HPF. The noise ratio can be adjusted depending on the different data integration scenarios. When there is good overlap of cell types across batches and a large set of common genes (Scenario A above), IHPF achieves improved performance over HPF and other matrix factorisation methods. At the other extreme, if batches and cell types are highly aligned with little overlap of cell types across batches (Scenario C above), the best performance is achieved for a zero noise ratio (i.e., it is better not to perform data integration). Therefore IHPF offers a flexible and controlled way to correct the effect of batch effects. We note that IHPF has no requirements on library sizes of the cells in the data sets, unlike PCA-based methods such as Scanorama which

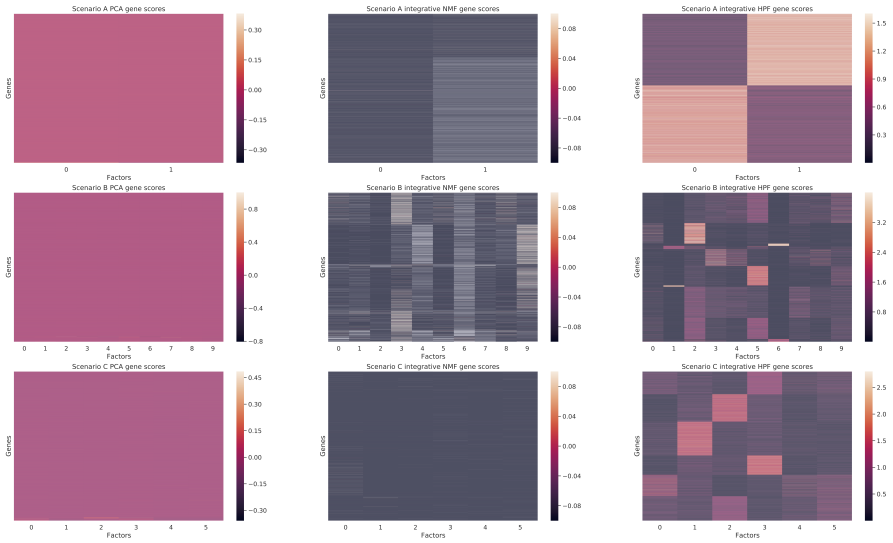


FIG. 9. *Gene latent factor scores ($M \times K$ matrices) obtained from PCA, INMF and IHPF for Scenarios A–C. As above, $K = C$. The rows (genes) are ordered using spectral co-clustering for ease of visualisation. There is only well-aligned block structure for the IHPF gene scores.*

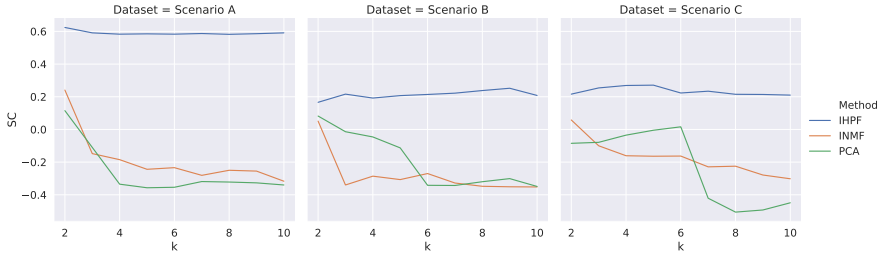


FIG. 10. *Silhouette coefficient (SC) of k-means clusterings (as a function of k) obtained from the gene scores of PCA, INMF and IHPF for Scenarios A-C. Only IHPF gene factors lead to well separated clusters with high SC values whereas INMF and PCA gene factors lead to poor clusters with negative values of SC.*

requires pre-processing of cells to have normalised sizes. IHPF is shown to recover factors with dual block structure in both the cell and gene spaces, thus enhancing the explainability of the latent factors in terms of groups of genes with biological content. We have made IHPF available as a Python package for use by the community.

There are several directions for extensions and improvement of this work. Currently, IHPF can only correct differences between data sets that are categorical in nature, such as batch effects. When the differences are continuous factors, such as population effects, a different model is needed. Further, the latent factors obtained with IHPF could also serve as the seed for more sophisticated classification and clustering methods, including variational autoencoders or graph-based multi-scale clustering methods [30, 32].

Acknowledgments. We thank Dr Zijing Liu for helpful discussions.

REFERENCES

[1] X. Tang, Y. Huang, J. Lei, H. Luo, and X. Zhu, "The single-cell sequencing: new developments and medical applications," *Cell & Bioscience*, vol. 9, no. 1, p. 53, 2019. [Online]. Available: https://doi.org/10.1186/s13578-019-0314-y

[2] B. Hwang, J. H. Lee, and D. Bang, "Single-cell rna sequencing technologies and bioinformatics pipelines," *Experimental & Molecular Medicine*, vol. 50, no. 8, p. 96, Aug 2018. [Online]. Available: https://doi.org/10.1038/s12276-018-0071-8

[3] A. Butler, P. Hoffman, P. Smibert, E. Papalexi, and R. Satija, "Integrating single-cell transcriptomic data across different conditions, technologies, and species," *Nature biotechnology*, vol. 36, no. 5, pp. 411–420, 2018.

[4] H. T. N. Tran, K. S. Ang, M. Chevrier, X. Zhang, N. Y. S. Lee, M. Goh, and J. Chen, "A benchmark of batch-effect correction methods for single-cell rna sequencing data," *Genome Biology*, vol. 21, no. 1, p. 12, Jan 2020.

[5] R. Argelaguet, D. Arnol, D. Bredikhin, Y. Deloro, B. Velten, J. C. Marioni, and O. Stegle, "Mofa+: a statistical framework for comprehensive integration of multi-modal single-cell data," *Genome Biology*, vol. 21, no. 1, p. 111, May 2020.

[6] A. Singh, C. P. Shannon, B. Gautier, F. Rohart, M. Vacher, S. J. Tebbutt, and K.-A. LÃª Cao, "DIABLO: an integrative approach for identifying key molecular drivers from multi-omics assays," *Bioinformatics*, vol. 35, no. 17, pp. 3055–3062, 01 2019.

[7] J. D. Welch, V. Kozareva, A. Ferreira, C. Vanderburg, C. Martin, and E. Z. Macosko, "Single-cell multi-omic integration compares and contrasts features of brain cell identity," *Cell*, vol. 177, no. 7, pp. 1873 – 1887.e17, 2019.

[8] B. Hie, B. Bryson, and B. Berger, "Efficient integration of heterogeneous single-cell transcriptomes using scanorama," *Nature Biotechnology*, vol. 37, no. 6, pp. 685–691, 2019.

[9] S. Sun, J. Zhu, Y. Ma, and X. Zhou, "Accuracy, robustness and scalability of dimensionality reduction methods for single-cell rna-seq analysis," *Genome Biology*, vol. 20, no. 1, p. 269, 2019.

[10] R. Petegrosso, Z. Li, and R. Kuang, "Machine learning and statistical methods for clustering single-cell RNA-sequencing data," *Briefings in Bioinformatics*, vol. 21, no. 4, pp. 1209–1223, 06 2019. [Online]. Available: https://doi.org/10.1093/bib/bbz063

[11] H. M. Levitin, J. Yuan, Y. L. Cheng, F. J. Ruiz, E. C. Bush, J. N. Bruce, P. Canoll, A. Iavarone, A. Lasorella, D. M. Blei, and P. A. Sims, "De novo gene signature identification from single-cell rna-seq with hierarchical poisson factorization," *Molecular Systems Biology*, vol. 15, no. 2, p. e8557, 2019.

[12] G. H. Golub, *Matrix computations*, fourth edition. ed., ser. Johns Hopkins studies in the mathematical sciences. Baltimore, Maryland: The Johns Hopkins University Press, 2013.

[13] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proceedings of the 13th International Conference on Neural Information Processing Systems*, ser. NIPSâ00. Cambridge, MA, USA: MIT Press, 2000, p. 535â541.

[14] ——, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999. [Online]. Available: https://doi.org/10.1038/44565

[15] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[16] C. M. Bishop, *Pattern recognition and machine learning*, ser. Information science and statistics. New York, NY, USA: Springer, 2006 - 2006.

[17] A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman, and D. M. Blei, "Automatic differentiation variational inference," *Journal of Machine Learning Research*, vol. 18, no. 14, pp. 1–45, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-107.html

[18] M. Zhou and L. Carin, "Negative binomial process count and mixture modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, p. 307â320, Feb 2015.

[19] Z. Yang and G. Michailidis, "A non-negative matrix factorization method for detecting modules in heterogeneous omics multi-modal data," *Bioinformatics (Oxford, England)*, vol. 32, no. 1, pp. 1–8, 2016.

[20] P. A. Szabo, H. M. Levitin, M. Miron, M. E. Snyder, T. Senda, J. Yuan, Y. L. Cheng, E. C. Bush, P. Dogra, P. Thapa, D. L. Farber, and P. A. Sims, "Single-cell transcriptomics of human t cells reveals tissue and activation signatures in health and disease," *Nature Communications*, vol. 10, no. 1, p. 4706, 2019.

[21] G. X. Y. Zheng, J. M. Terry, P. Belgrader, P. Ryvkin, Z. W. Bent, R. Wilson, S. B. Ziraldo, T. D. Wheeler, G. P. McDermott, J. Zhu, M. T. Gregory, J. Shuga, L. Montesclaros, J. G.

Underwood, D. A. Masquelier, S. Y. Nishimura, M. Schnall-Levin, P. W. Wyatt, C. M. Hindson, R. Bharadwaj, A. Wong, K. D. Ness, L. W. Beppu, H. J. Deeg, C. McFarland, K. R. Loeb, W. J. Valente, N. G. Ericson, E. A. Stevens, J. P. Radich, T. S. Mikkelsen, B. J. Hindson, and J. H. Bielas, "Massively parallel digital transcriptional profiling of single cells," *Nature Communications*, vol. 8, no. 1, p. 14049, Jan 2017. [Online]. Available: https://doi.org/10.1038/ncomms14049

[22] M. Baron, A. Veres, S. Wolock, A. Faust, R. Gaujoux, A. Vetere, J. Ryu, B. Wagner, S. Shen-Orr, A. Klein, D. Melton, and I. Yanai, "A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure," *Cell Systems*, vol. 3, no. 4, pp. 346–360.e4, Oct 2016. [Online]. Available: https://doi.org/10.1016/j.cels.2016.08.011

[23] M. J. Muraro, G. Dharmadhikari, D. Grün, N. Groen, T. Dielen, E. Jansen, L. van Gurp, M. A. Engelse, F. Carlotti, E. J. P. de Koning, and A. van Oudenaarden, "A single-cell transcriptome atlas of the human pancreas," *Cell systems*, vol. 3, no. 4, pp. 385–394.e3, Oct 2016.

[24] N. Lawlor, J. George, M. Bolisetty, R. Kursawe, L. Sun, V. Sivakamasundari, I. Kycia, P. Robson, and M. L. Stitzel, "Single-cell transcriptomes identify human islet cell signatures and reveal cell-type-specific expression changes in type 2 diabetes," *Genome research*, vol. 27, no. 2, pp. 208–222, Feb 2017.

[25] Å. Segerstolpe, A. Palasantza, P. Eliasson, E.-M. Andersson, A.-C. Andréasson, X. Sun, S. Picelli, A. Sabirsh, M. Clausen, M. K. Bjursell, D. M. Smith, M. Kasper, C. Ämmälä, and R. Sandberg, "Single-cell transcriptome profiling of human pancreatic islets in health and type 2 diabetes," *Cell metabolism*, vol. 24, no. 4, pp. 593–607, Oct 2016.

[26] D. Grün, M. J. Muraro, J.-C. Boisset, K. Wiebrands, A. Lyubimova, G. Dharmadhikari, M. van den Born, J. van Es, E. Jansen, H. Clevers, E. J. P. de Koning, and A. van Oudenaarden, "De novo prediction of stem cell identity using single-cell transcriptome data," *Cell stem cell*, vol. 19, no. 2, pp. 266–277, Aug 2016.

[27] O. Thas, *Comparing distributions*. Springer, 2010.

[28] F. A. Wolf, P. Angerer, and F. J. Theis, "Scanpy: large-scale single-cell gene expression data analysis," *Genome Biology*, vol. 19, no. 1, p. 15, Feb 2018. [Online]. Available: https://doi.org/10.1186/s13059-017-1382-0

[29] Y. Qian, P. Expert, P. Panzarasa, and M. Barahona, "Geometric graphs from data to aid classification tasks with graph convolutional networks," *Patterns*, vol. 2, no. 4, p. 100237, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S266638992100057X

[30] Z. Liu and M. Barahona, "Graph-based data clustering via multiscale community detection," *Applied Network Science*, vol. 5, no. 1, p. 3, 2020. [Online]. Available: https://doi.org/10.1007/s41109-019-0248-7

[31] L. van der Maaten, "Accelerating t-sne using tree-based algorithms," *Journal of Machine Learning Research*, vol. 15, pp. 3221–3245, 2014. [Online]. Available: http://jmlr.org/papers/v15/vandermaaten14a.html

[32] C. Xu, R. Lopez, E. Mehlman, J. Regier, M. I. Jordan, and N. Yosef, "Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models," *Molecular Systems Biology*, vol. 17, no. 1, p. e9620, 2021. [Online]. Available: https://www.embopress.org/doi/abs/10.15252/msb.20209620

Supplementary Information.

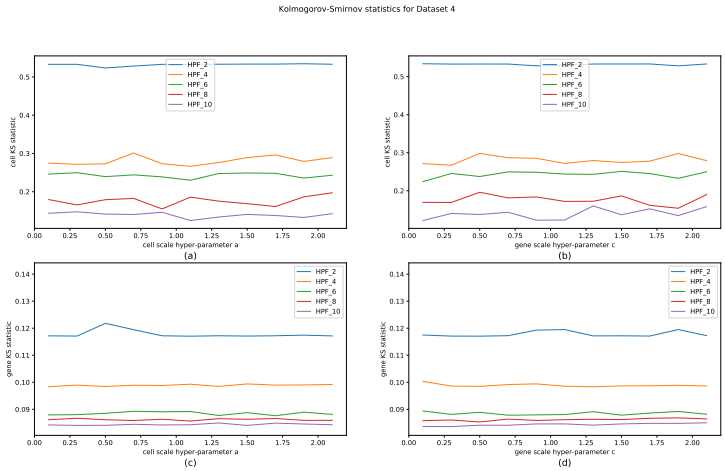


FIG. 11. *(a)-(b) Cell KS statistic (3.3) and (c)-(d) gene KS statistic (3.4) for HPF models of data set ID 4 obtained with varying cell and gene scale hyper-parameters a and c and different number of latent factors K. For all values of K, changing the hyper-parameters a, c has almost no effect on the model performance (as measured by the KS statistics).*

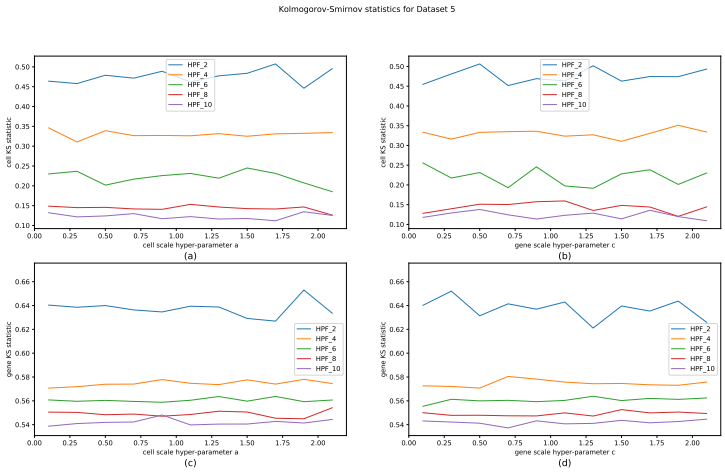


FIG. 12. *(a)-(b) Cell KS statistic (3.3) and (c)-(d) gene KS statistic (3.4) for HPF models of data set ID 5 obtained with varying cell and gene scale hyper-parameters a and c and different number of latent factors K. For all values of K, changing the hyper-parameters a, c has almost no effect on the model performance (as measured by the KS statistics).*

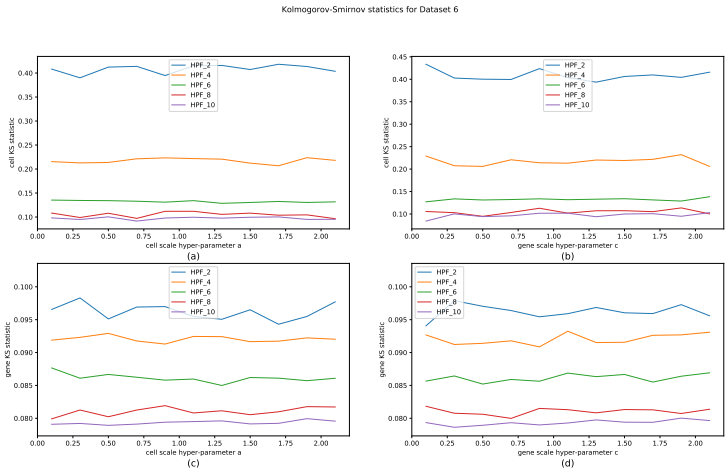


FIG. 13. *(a)-(b) Cell KS statistic (3.3) and (c)-(d) gene KS statistic (3.4) for HPF models of data set ID 6 obtained with varying cell and gene scale hyper-parameters a and c and different number of latent factors K. For all values of K, changing the hyper-parameters a, c has almost no effect on the model performance (as measured by the KS statistics).*

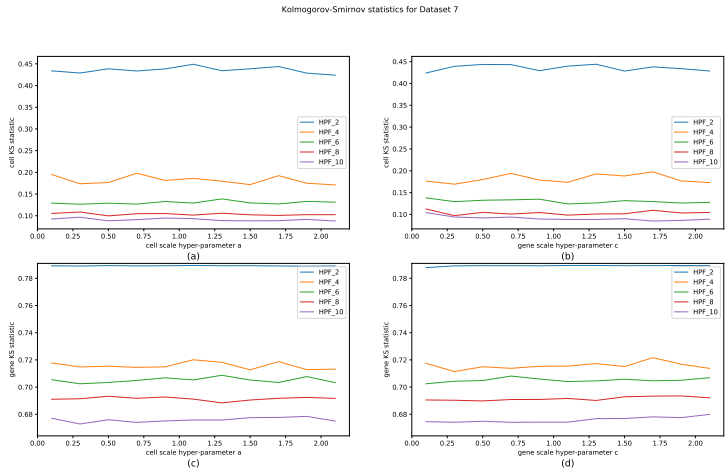


FIG. 14. *(a)-(b) Cell KS statistic (3.3) and (c)-(d) gene KS statistic (3.4) for HPF models of data set ID 7 obtained with varying cell and gene scale hyper-parameters a and c and different number of latent factors K. For all values of K, changing the hyper-parameters a, c has almost no effect on the model performance (as measured by the KS statistics).*

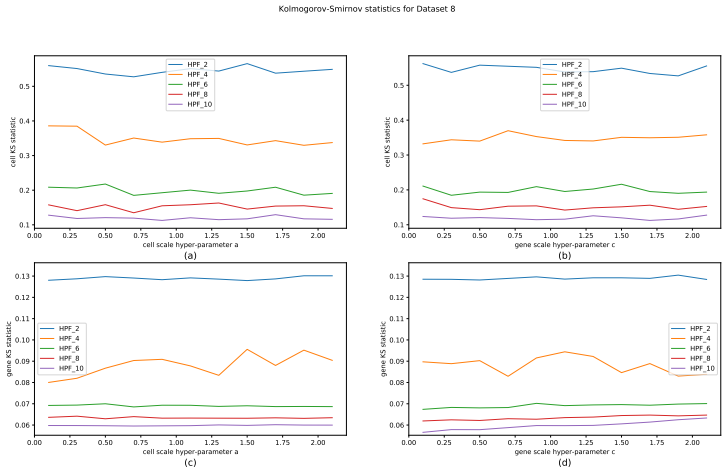


FIG. 15. *(a)-(b) Cell KS statistic (3.3) and (c)-(d) gene KS statistic (3.4) for HPF models of data set ID 8 obtained with varying cell and gene scale hyper-parameters a and c and different number of latent factors K. For all values of K, changing the hyper-parameters a, c has almost no effect on the model performance (as measured by the KS statistics).*

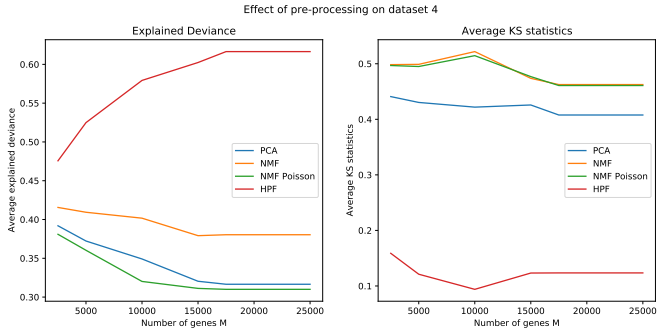


FIG. 16. *(a) Average explained deviance (3.1) and (b) average KS statistic (3.5) of different matrix factorisation methods (PCA, NMF, NMF-Poisson, HPF) applied to data set set ID 4 computed using the top M genes with highest variability, where M is increased from 2500 to 25000. The number of latent factors is $K = 10$ for all methods.*

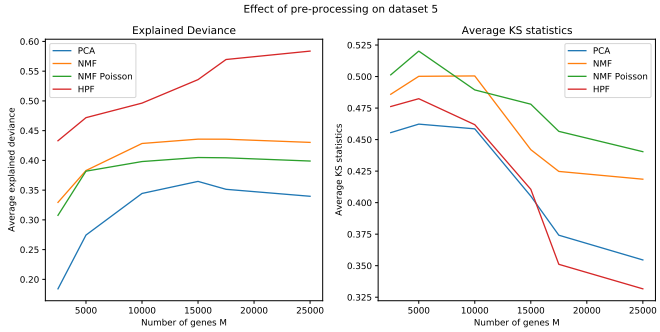


FIG. 17. *(a) Average explained deviance (3.1) and (b) average KS statistic (3.5) of different matrix factorisation methods (PCA, NMF, NMF-Poisson, HPF) applied to data set set ID 5 computed using the top M genes with highest variability, where M is increased from 2500 to 25000. The number of latent factors is $K = 10$ for all methods.*

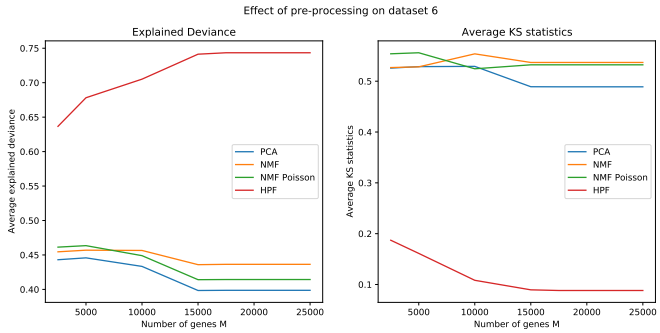


FIG. 18. *(a) Average explained deviance (3.1) and (b) average KS statistic (3.5) of different matrix factorisation methods (PCA, NMF, NMF-Poisson, HPF) applied to data set set ID 6 computed using the top M genes with highest variability, where M is increased from 2500 to 25000. The number of latent factors is $K = 10$ for all methods.*

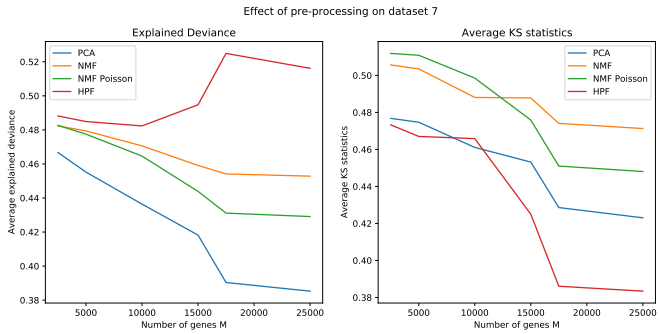


FIG. 19. *(a) Average explained deviance (3.1) and (b) average KS statistic (3.5) of different matrix factorisation methods (PCA, NMF, NMF-Poisson, HPF) applied to data set set ID 7 computed using the top M genes with highest variability, where M is increased from 2500 to 25000. The number of latent factors is $K = 10$ for all methods.*

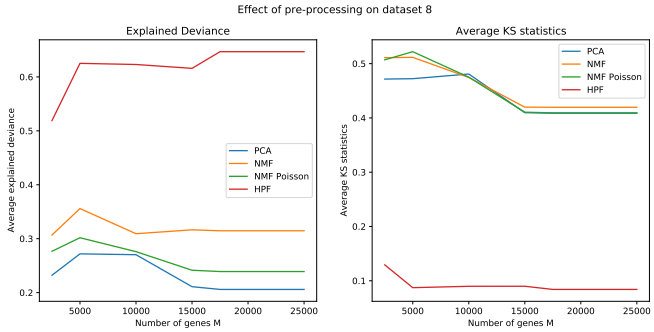
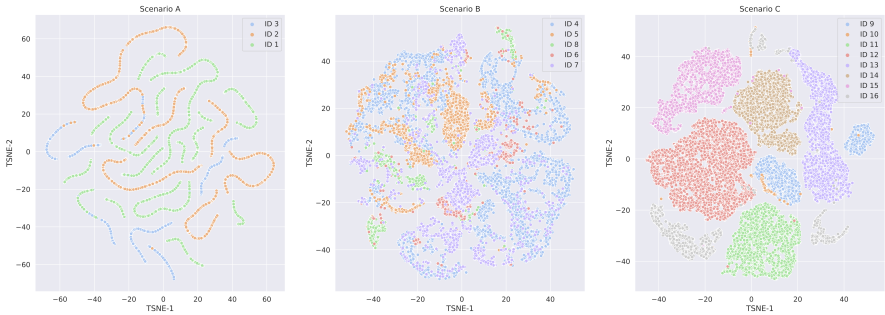


FIG. 20. *(a) Average explained deviance (3.1) and (b) average KS statistic (3.5) of different matrix factorisation methods (PCA, NMF, NMF-Poisson, HPF) applied to data set set ID 8 computed using the top M genes with highest variability, where M is increased from 2500 to 25000. The number of latent factors is $K = 10$ for all methods.*

(a) TSNE from IHPF cell scores



(b) TSNE from HPF cell scores

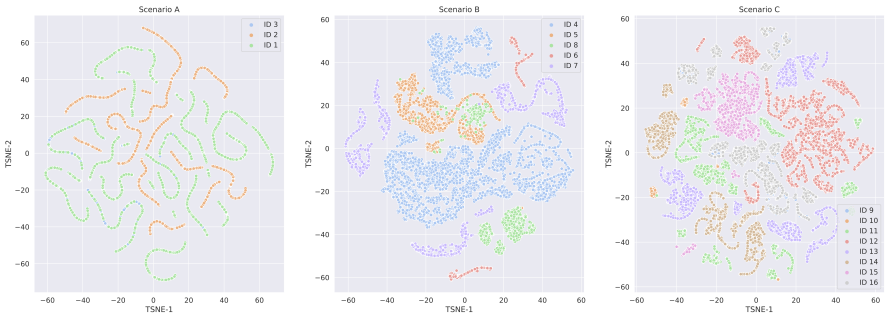


FIG. 21. *TSNE projections of Scenarios A-C obtained from cell latent factor scores from (a) IHFP and (b) HPF, and coloured by batch label.*