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. Such problems are compounded when dealing with composite data sets formed by different batches. 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 batches *vs.* biological sources (cell phenotypes). We exemplify the application of IHPF under different data integration scenarios with varying alignments of batches and cell diversity, and show that IHPF produces latent factors that can be advantageously applied for cell clustering and visualisation. In addition, the extracted factors have a dual block structure in both cell and gene spaces with 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 features (i.e., genes), dimensionality reduction is a key ingredient in the analysis workflow of scRNA-seq data sets. Hence many computational pipelines are designed to extract a reduced set of informative and interpretable *latent factors* that capture the distinguishing characteristics of the cell types in the data set. However, the sparse and discrete nature of scRNA-seq data poses challenges to classic methods of 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 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

*Department of Mathematics, Imperial College London, London SW7 2AZ, United Kingdom (ming-hei.wong15@imperial.ac.uk).

†Department of Mathematics, Imperial College London, London SW7 2AZ, United Kingdom (m.barahona@imperial.ac.uk,).

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 scVI [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 characteristic 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 (e.g., PCA) 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 interpretability. In particular, NMF ensures latent factors that learn a part-based structure, with non-negative scores, but with no guarantee of sparsity. Lack of interpretability is compounded, and especially relevant, when dealing with data integration of cell types across technical batches.

In this paper, we focus on data integration and dimensionality reduction of scRNA-seq data sets comprised of several batches, and consider this problem in the context of matrix factorisation methods that induce robust and interpretable latent factors to simplify downstream analyses.

To do so, we 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. Section 2 provides background on existing factorisation methods for dimensionality reduction pertaining to scRNA-seq analyses. Section 3 describes our algorithm for Integrative Hierarchical Poisson Factorisation (IHPF), the data sets, our numerical experiments, and we demonstrate how IHPF can be used to carry out dimensionality reduction under different data integration scenarios. In Section 4, we show that the latent factors obtained by IHPF provide a good basis for cell clustering and visualisation, and are also sparse and interpretable due to their dual block-structure in both cell and gene spaces. 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 matrix 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:

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

Here, the matrix W contains the cell scores and the matrix V contains the gene scores with respect to K *latent factors*. The latent factors thus recapitulate (approximately) the information contained in the M features, and each sample (cell) can be described by K coordinates.

PCA and NMF are two classic methods that produce such a matrix factorisation, differing 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 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 [11]. PCA-based methods have been widely used with scRNA-seq data sets, typically after applying a log-transformation to the raw data counts. However, the fact that the cell and gene scores (W and V) are signed and not sparse, together with the log-transformation, reduces the interpretability of the latent space.

The data matrix in scRNA-seq is non-negative element-wise ($X \geq 0$); hence 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 [12]:

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

NMF learns part-based representations, which represent positive contributions of the latent factors to the samples [12], yet with no guarantee of sparsity.

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

From a probabilistic perspective, it has been shown [13] that PCA can be understood as a generative model where the samples are noisy observations of a combination of latent factors that are multivariate Gaussian [14, 15]:

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

NMF also has a probabilistic interpretation related to Poisson factor analysis (PFA) [16]. 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:

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

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

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

Here, Θ and Δ correspond, respectively, to cell and gene scores onto the K latent factors Θ and are Gamma distributed random variables with shape parameter 1 and infinite rate. Note that Poisson-Gamma random variables are related to the Negative Binomial distribution, which is widely used to model scRNA-seq counts.

Recently, this probabilistic interpretation inspired the Hierarchical Poisson Factorisation (HPF) [17], which uses a hierarchical Gamma prior:

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

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

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

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

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

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 [17].

2.3. Data integration. The problem of integrating data sets obtained with different technologies under diverse experimental settings poses additional challenges, as one is usually interested in capturing biologically meaningful signals in feature (gene) space under technical batch effects in sample (cell) space.

The classic matrix factorisation methods introduced in Section 2.1 have been extended for data integration. Traditionally, PCA has been used for batch correction in bulk RNA transcriptomics data, but it does not perform satisfactorily on scRNA-seq data due to the high sparsity of single cell data [2]. To circumvent these limitations, Scanorama [8] and BBKNN [18] apply graph-based algorithms on the principal components extracted by PCA to remove batch effects and improve alignment between cell types. BBKNN uses a k-nearest neighbour algorithm to build a graph to capture cell-to-cell similarity, attempting to balance the contribution from each batch. Scanorama identifies mutual nearest neighbours between batches and then finds the optimal order of batch alignment for panorama stitching, followed by a non-linear transformation to create the integrated latent space and batch-corrected gene expression matrices.

Integrative NMF (INMF) extennds NMF to harmonise data sets that share their gene space [19]. For S batches with non-negative data matrices $X_s \in \mathbb{R}_+^{N_s\times M}$, $s = 1,\ldots,S$, with $N = \sum_{s=1}^{S} N_s$, INMF finds a description 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, cell scores, gene scores specific to each data set, and gene scores common to all data sets. The INMF factorisation minimises the 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 the indicator function for cells in batch s and $W(\mathbf{1}_s,:) \in \mathbb{R}_+^{N_s\times K}$ is the submatrix containing the cell scores for batch s. The hyper-parameter α, denoted as the *noise ratio*, regularises the magnitude of the components of the batches and assigns the variability in the data between cell types and batches.

Recently, deep learning has been used to perform dimensionality reduction and batch correction at the same time. scVI [7] uses a Variational Auto-Encoder (VAE) to learn a latent space representation of cells, modelling both library size and batch effects using a zero-inflated negative binomial distribution for gene expression counts. A significant drawback is that the latent space representation is not interpretable.

3. Methods.

3.1. Integrative Hierarchical Poisson Factorisation (IHPF). It has been shown that the performance of HPF can suffer when integrating data sets of different provenance (e.g., batches or other sample effects); hence various *ad hoc* methods (e.g., clustering of biologically related latent factors) have been applied to ameliorate batch effects [20].

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

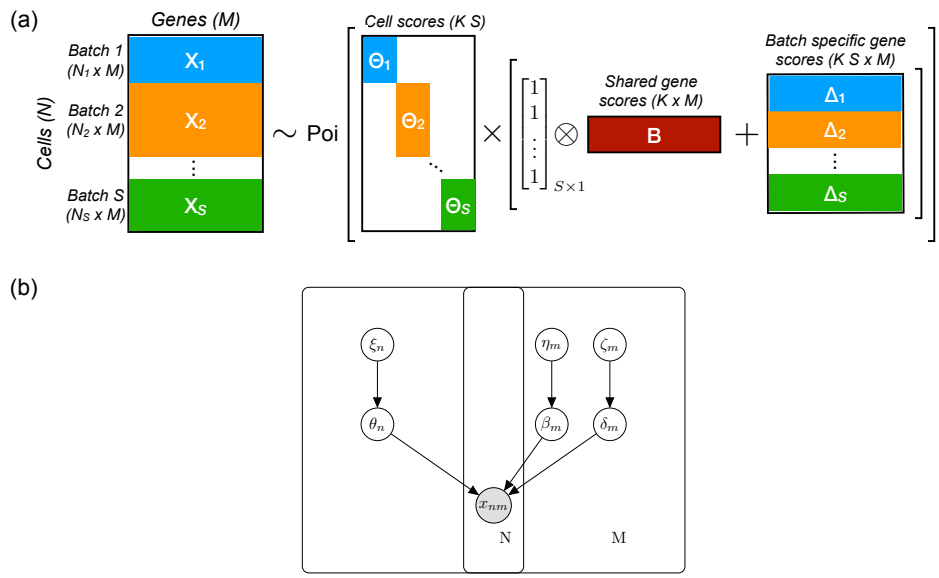


FIG. 1. *Two representations of the IHPF model. (a) IHPF can be viewed as a probabilistic matrix factorisation where the count matrix X, formed by the X_i of the S batches, is modelled as a Poisson distribution with weights given by a matrix factorisation with cell scores Θ_i together with batch-specific gene scores Δ_i and common gene scores B; (b) Graphical representation of the hierarchical IHPF model with N cells and M genes.*

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

hierarchical Gamma prior. Then the IHPF model is given by:

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

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 3.1 and the IHPF algorithm is summarised in Figure 1.

Algorithm 3.1 IHPF model for single-cell data.

```
Set hyper-parameters a, a', b', c, c', d'
for each cell i do
    Sample rate of cell score:   ξ_i ~ Gamma(a', b')
    for each factor k do
        Sample cell score:   θ_ik ~ Gamma(a, ξ_i)
    end for
end for
for each gene g and batch s do
    Sample rate of shared gene score:
        η_g ~ Gamma(c', d')
    Sample rate of batch-specific gene score:
        ζ_gs ~ Gamma(c', d')
    for each factor k do
        Sample shared gene score:
            β_gk ~ Gamma(c, η_g)
        Sample batch-specific gene score:
            δ_gsk ~ Gamma(c, ζ_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 ~ Poisson(Σ_{k=1}^{K} θ_ik (β_gk + δ_gsk))
end for
```

We use mean-field variational inference [21] to infer the IHPF model. Algorithm 3.2 describes the coordinate ascent algorithm [17] used to infer the shape and rate parameters of the six Gamma posteriors for gene and cell scores in Algorithm 3.1:

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

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

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

Algorithm 3.2 Coordinate Ascent Algorithm for likelihood inference in IHPF.

Set the number of factors K, noise ratio α, and hyper-parameters a, a', b', c, c', d'.
Initialise allocations to gene latent factors 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:

$$\beta_{gk}^{shp} = c + \sum_i x_{ig}\phi_{igk}$$
$$\beta_{gk}^{rte} = \frac{\eta_g^{shp}}{\eta_g^{rte}} + \sum_i \frac{\theta_{ik}^{shp}}{\theta_{ik}^{rte}}$$
$$\eta_g^{shp} = c' + Kc$$
$$\eta_g^{rte} = d' + \sum_k \frac{\beta_{gk}^{shp}}{\beta_{gk}^{rte}}$$

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

$$\delta_{gsk}^{shp} = c + \sum_{i \in s} x_{ig}\omega_{igk}$$
$$\delta_{gsk}^{rte} = \frac{\zeta_g^{shp}}{\zeta_g^{rte}} + \sum_{i \in s} \frac{\theta_{ik}^{shp}}{\theta_{ik}^{rte}}$$
$$\zeta_{gs}^{shp} = c' + Kc$$
$$\zeta_{gs}^{rte} = d' + \sum_k \frac{\delta_{gsk}^{shp}}{\delta_{gsk}^{rte}}$$

  end for
 end for
 for each cell i **do**
  Denote batch label of cell i to be s
  Compute shapes and rates of Gamma posterior of cell scores:

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

 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

The multinomial distributions used in Algorithm 3.2 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 shared and batch specific gene latent factors, i.e., how much variance in the data is to be attributed to technical noise between data sets. Taking the limit $\alpha \to 0$ recovers the original HPF.

3.2. Data sets and data integration scenarios. We have collated from the published literature 16 scRNA-seq data sets of varying sizes and scope, and collected with different sequencing technologies (Table 1). Quality control is applied on the data sets by keeping only cells with at least 600 transcripts.

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)	Sequecing Depth	Scenario	Reference
1	Mouse Jurkat cells	10X Genomics	2885	32378	3060	A	Zheng 1 [22]
2	Mouse 293t cells	10X Genomics	3257	32378	3168	A	Zheng 2 [22]
3	Mouse 50% Jurkat/50% 293t cells	10X Genomics	3388	32378	3249	A	Zheng 3 [22]
4	Human pancreas	InDrop	8569	20126	2622	B	Baron [23]
5	Human pancreas	Cel-seq2	2449	19140	3908	B	Muraro [24]
6	Human pancreas	Fluidigm C1	638	25463	3294	B	Lawlor [25]
7	Human pancreas	Smart-seq2	2989	26179	3342	B	Segerstolpe [26]
8	Human pancreas	Celseq	1276	20148	3196	B	Grün [27]
9	Human CD19+ B cells	10X Genomics	2261	32378	1598	C	Zheng 5 [22]
10	Human CD14+ monocytes	10X Genomics	295	32378	1824	C	Zheng 6 [22]
11	Human CD4+ helper T cells	10X Genomics	3713	32378	1581	C	Zheng 7 [22]
12	Human CD56+ natural killer cells	10X Genomics	6657	32378	1714	C	Zheng 8 [22]
13	Human CD8+ cytotoxic T cells	10X Genomics	3990	32378	1509	C	Zheng 9 [22]
14	Human CD4+CD45RO+ memory T cells	10X Genomics	3628	32378	1543	C	Zheng 10 [22]
15	Human CD4+CD25+ regulatory T cells	10X Genomics	3365	32378	1609	C	Zheng 11 [22]
16	Human PBMC	10X Genomics	2293	32378	1743	C	Zheng 4 [22]

We group data sets in Table 1 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; here marker genes can easily tell the difference between cell types.
- *Scenario B:* a collection of data sets sequenced with different technologies and containing several cell types;
- *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 the same technology (10X Genomics) and comprising 6 different cell types highly concentrated in particular batches.

These three 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. To measure the alignment between cell types and batches, we use the Adjusted Mutual Information (AMI), as shown in Table 2.

TABLE 2
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	32643	15369	32643
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

3.3. Numerical Experiments and Code. To study data integration, we merge the corresponding data sets (keeping the intersection of genes across data sets) according to Scenarios A-C. Pre-processing of the single-cell data sets is done using Scanpy [28] and Scanorama [8], with details described in Section A.1. Our computational implementation of IHPF builds upon the original Python code for HPF in Ref. [17], and we use default hyper-parameters set in that reference. The data sets for the three data integration scenarios and the code implementing IHPF is available at https://github.com/barahona-research-group/scIHPF.

3.4. Data integration of scRNA-seq data sets using IHPF.

Robustness of IHPF to hyper-parameters. First, we check the robustness of IHPF 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 [17]. The IHPF noise ratio α is set to 0.1.

We run IHPF for the three data integration scenarios varying the hyper-parameters a and c between $[0.1, 1.1]$. In all cases, we extract $K = C$ factors, where C is the number of cell types present in the data. To evaluate the performance of model, we compute the negative mean log-likelihood of counts, under the Poisson assumption. Throughout this paper, we will always be dealing with Poisson log-likelihoods. Figure 2 shows that the performance of IHPF in all the data integration scenarios is highly robust to changes in the hyper-parameters a and c.

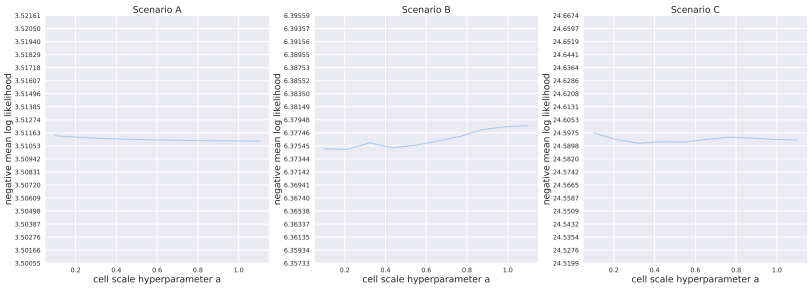
Once the robustness of IHPF to the hyper-parameters is established, we set $a = c = 0.3$ as suggested in the scHPF package [17] and keep these parameters fixed throughout the rest of the paper.

Selecting the IHPF noise ratio. The IHPF noise ratio α (see Algorithm 3.2) encodes how much of the observed data variance is attributed to the shared biological signal (i.e., cell types) or to batch-specific differences. When the noise ratio becomes zero, the batch labels are ignored and IHPF recovers the corresponding non-integrative result (i.e., HPF).

The noise ratio needs to be tuned for the specific data integration scenarios: a noise ratio that is too large will result in over-correction, so that the cell scores will not capture the biological difference between cell types; on the other hand, a noise ratio that is too small will result in extra cell clusters whereby cells with similar biology could be separated into smaller groups by batch.

While prior knowledge can be used to select the noise ratio (e.g., experimental design or specific biological hypotheses for an experimental scenario), we propose a statistical procedure that can be used to guide the choice of noise ratio. To do so, we employ a *leave-one-batch-out* cross-validation procedure, as follows. For each scenario

(a) Robustness under change of cell scale hyper-parameter, a



(b) Robustness under change of gene scale hyper-parameter, c

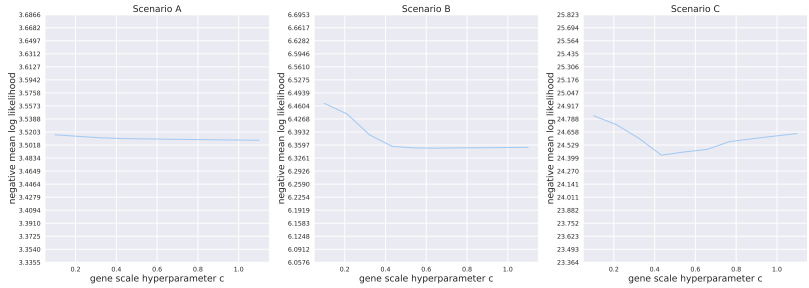


FIG. 2. *Negative Mean log-likelihood of IHPF for the three scenarios with varying hyper-parameters: (a) cell scale hyper-parameter, a; (b) gene scale hyper-parameter, c. For all scenarios, varying the hyper-parameters has almost no effect on the model performance.*

and noise ratio, we apply IHPF on all but one of the batches. A new HPF model is trained using the learn shared gene space as initialisation, and we then project the leave-out batch on the shared gene space, and compute the reconstruction loss (negative mean Poisson likelihood) for that batch. We then repeat this procedure for each batch and compute the weighted average of the likelihood. The results are shown in Figure 3. (See Section A.2 for the results of all batches before averaging.) We then use the elbow method to identify the optimal noise ratio for each scenario, by taking the largest noise ratio until we experience a steep increase in reconstruction loss.

For Scenario A, there is a sharp increase of the negative likelihood even for small values of the noise ratio, hence there is no observed improvement with data integration and the optimal noise ratio selected is 0.001. For Scenario B, the noise ratio can be increased up to values of around 0.5 before the likelihood sharply deteriorates; hence the optimal noise ratio selected in this case is 0.5. For Scenario C, the optimal noise ratio selected is 0.01; hence in this case only mild data integration across batches is beneficial. Henceforth in the paper, we will fix these noise ratios for Scenarios A, B and C, unless otherwise indicated.

The optimal noise ratios selected are broadly consistent with our prior biological knowledge of the data sets. Indeed, batches in Scenario A similar to each other, batches in Scenario C are highly aligned with the cell types, and batches in Scenario B are both technically distinct and containing cell mixtures; hence data integration is more important in this last case. The noise ratio thus serves as an indication of

the amount of data integration necessary for the particular collection of data sets, with the aim of taking into account batch variability yet still capturing the biological variability in the data.

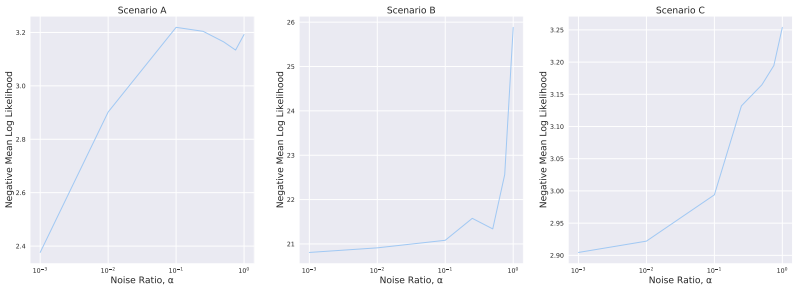


FIG. 3. *Cross-validation leave-one-batch-out likelihood for each data scenario, computed as the weighted mean of likelihoods for each batch, weighted by batch size. For every batch in each scenario, the IHPF model is fitted using the rest of the batches. This is then repeated for different noise ratios.*

4. Results. In this section, we first compare the quality of the latent factors obtained with different methods for the purpose of clustering cell types and visualisation, and consider the effect of the noise ratio for data scenarios with different alignment between cell types and batches. Secondly, we show that IHPF extracts sparse latent factors with a dual block structure with respect to both gene and cell spaces, thus enhancing their interpretability.

4.1. Using IHPF latent factors for cell 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 latent 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. 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 latent factors could capture mostly batch information, and in that case the clusters would fail to recapitulate the cell types. Rather than considering only complex multi-step computational pipelines with many parameters, we examine here if relatively simple data integration methods can extract latent factors that capture cell type variability in the presence of batch effects.

We evaluate five widely-used methods for dimensionality reduction and data integration based on different principles: Scanorama, BBKNN, scVI and INMF, along with our proposed method IHPF. The code for scVI [7], Scanorama [8] and BBKNN [18] were all downloaded from their corresponding Github pages, and we use recommended parameters and apply pre-processing steps on the count matrices following the recommendations by the authors. For INMF [19] we have coded a Python implementation compatible with scikit-learn, and set the noise ratio to the default

value of 1.0, as suggested by the authors. For INMF and IHPF, we do not apply any pre-processing, other than the necessary quality control measures. Details of the pre-processing steps for each method can be found in the Supplementary Information.

Cell clustering from cell latent factors. We use each of the dimensionality reduction methods to extract $K = C$ factors, where C is the number of cell types present in the data, and the row-normalised cell scores (e.g, from matrices W or Θ) are used to cluster the cells using k-means with $k = C$.

We first evaluate if the extracted cell latent factor scores induce well-separated clusters using the silhouette coefficient (SC), the (bounded) ratio of intra- and inter-cluster distances ($-1 \leq \text{SC} \leq 1$). For all methods other than BBKNN, SC is computed based on the cell latent factors (with latent dimension equal to the number of cell types in the scenario). For BBKNN, SC is computed based on the distance graph generated by the algorithm. Table 3 shows that, for all scenarios, IHPF gives the best separated clusters, as measured by the SC. On the other hand, methods that rely on Gaussian latent space and PCA, such as scVI and BBKNN give the least well-separated clusters, due to the mapping to non-sparse continuous variables. Scanorama and INMF provide an intermediate regime as a result of the graph-induced sparsity and the part-based property, respectively.

TABLE 3

Silhouette coefficient of the clusterings obtained by applying k-means clustering to K normalised cell latent factors obtained from five matrix factorisation models. Here, $k = K = C$.

Method	Scenario A (ID 1-3)	Scenario B (ID 4-8)	Scenario C (ID 9-16)
Scanorama	0.447	0.292	0.335
BBKNN	0.007	0.079	0.094
scVI	0.107	0.185	0.075
INMF	0.458	-0.229	0.446
IHPF	0.963	0.520	0.390

The SC computed above is an intrinsic measure of cluster separability, but does not evaluate the quality of the clusters with respect to the underlying cell types or, alternatively, to the batch labels. Therefore, to quantify the quality of the clusterings, we compute the Adjusted Mutual Information (AMI) between the cell clustering (i.e., k-means clusterings obtained from the cell scores) with respect to batch labels and cell types (Table 4).

The AMI of the cell clusterings with respect to the batch labels (Table 4A) varies substantially for the different clustering methods and scenarios, but in general we observe that only IHPF is able to reduce 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 to the AMI of true cell types *vs.* batches. The other methods either over-correct or under-correct batch effects, and fail to adapt to the underlying variation in the alignment of batches and cell types.

The objective of our clustering is to obtain a clustering from the latent factors that compares well with the cell labels (i.e., it is able to extract the biological signal in the presence of batch effects). The AMI of the cell clusterings *vs.* cell types (Table 4B) show that IHPF has the overall best performance. IHPF consistently outperforms INMF, BBKNN and scVI across all scenarios. Although their performance is comparable in Scenarios A and C, where integration is not crucial, IHPF

TABLE 4

Adjusted Mutual Information (AMI) between the clustering obtained from latent factors versus (a) batches and (b) cell types. All the methods are applied without any gene selection. k-means is applied to K normalised cell latent factors obtained from each method, where the number of clusters k is set equal to the number of cell types C, i.e., $K = k = C$ (see Table 2).

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

Method	Scenario A (ID 1-3)	Scenario B (ID 4-8)	Scenario C (ID 9-16)
Scanorama	0.492	0.365	0.515
BBKNN	0.070	0.241	0.081
scVI	0.195	0.029	0.133
INMF	0.170	0.260	0.297
IHPF	0.498	0.037	0.536
AMI true cell types vs. batches	*0.498*	*0.032*	*0.842*

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

Method	Scenario A (ID 1-3)	Scenario B (ID 4-8)	Scenario C (ID 9-16)
Scanorama	0.965	0.371	0.594
BBKNN	0.158	0.489	0.084
scVI	0.395	0.589	0.139
INMF	0.407	0.276	0.323
IHPF	0.980	0.661	0.539

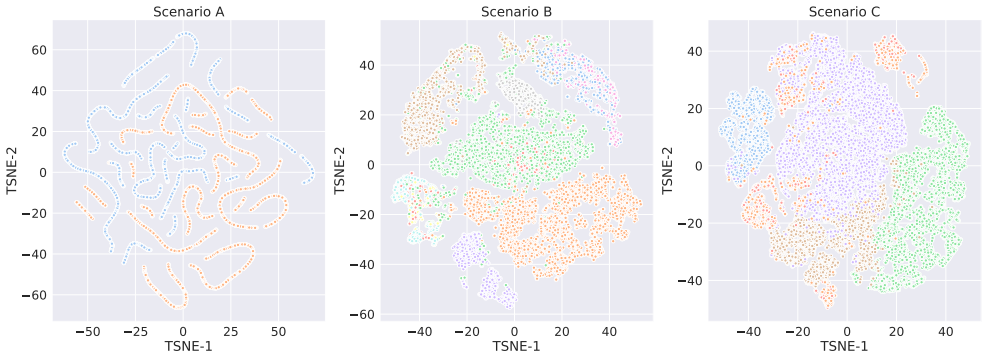
outperforms Scanorama in Scenario B, where batch correction through adaptable data integration is beneficial. Therefore IHPF achieves good performance across different data integration scenarios with a simple model specification with a small number of parameters (especially compared to complex pipelines or deep learning methods such as Scanorama or scVI).

Visualisation based on cell latent factors. Another important use of latent factors is as part of visualisation pipelines used for biological exploration of data sets. Typically, the latent factors extracted with a dimensionality reduction method are used as inputs to nonlinear mapping algorithms (such as TSNE or UMAP) to obtain representations of the data on the plane. Such planar representations are broadly used to display high dimensional data and to guide biological hypotheses. The quality of the latent factors is thus an important factor to obtain useful planar representations.

As a brief illustration of the benefits of data integration, we compare the visualisations obtained with TSNE starting from PcA factors, as a simple baseline of dimensionality reduction without batch correction, compared to TSNE mappings using IHPF factors. Figure 4 shows the projections obtained from both IHPF and PCA factors using TSNE with default perplexity parameter [31]. Our results show that IHPF-based TNSE projections have better separated cell groups, which are less affected by potentially conflicting grouping by batches. This can be observed in the same projections coloured by batch label (Fig. 10 in SI), where we observe how the IHPF-based TSNE projections induce spatial groupings where cells of the same type but of different batches lie in close proximity, in contrast with the PCA group-

ings that are highly concomitant with batch labels. To see the same visualisations coloured according to batch type, see Section A.3 in the Supplementary Information.

(a) TSNE from IHPF factors (coloured by cell type)



(b) TSNE from PCA components (coloured by cell type)

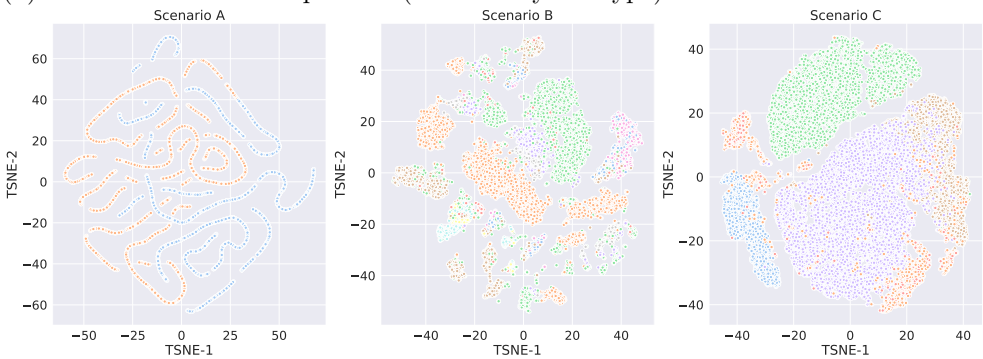


FIG. 4. *TSNE projections of Scenarios A-C obtained from latent factors from (a) IHPF and (b) PCA, and coloured by cell type.*

4.2. IHPF induces block-structured factors in both gene and cell spaces. In biological applications, it is often desirable that the latent factors are both descriptive and interpretable. Ideally, a few, sparse factors should be able to recapitulate the original data. In this 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*. Such sparse factors can be the basis for clusters with improved biological interpretability, because they can link cell types to a reduced set of characteristic genes.

Most dimensionality reduction methods (e.g., PCA) are not designed to induce sparsity in the latent factors. On the other hand, IHPF is characterised by an intrinsic sparsity that makes its factors good candidates for improved interpretability. To explore this idea in more detail, we examine the block structure in the matrices of both cell and gene latent factor scores obtained with different methods.

Block-structure in cell space. Figure 5 shows the *cell scores* for Scenarios A-C obtained with PCA, INMF and IHPF. The block-structure present in all these matrices is a reflection of the fact that the cell latent factors capture information about

cell types; indeed, these cell score matrices were the starting point for the clustering and visualisation results presented in Section 4.1.

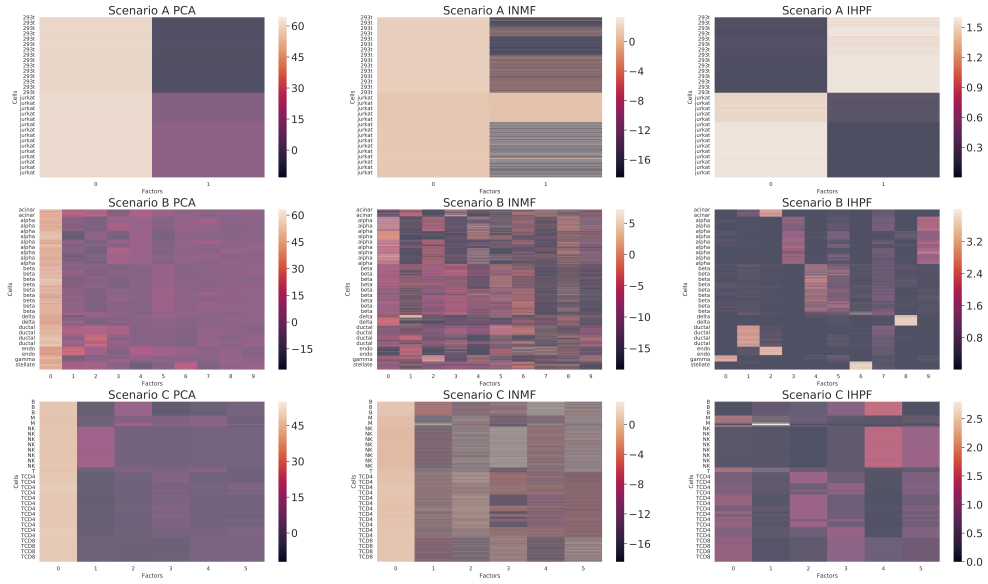


FIG. 5. *Cell latent factor scores ($N \times K$ matrices) obtained from PCA, INMF and IHPF for Scenarios A-C. The rows of the matrices correspond to cells sorted by cell type and the columns correspond to the latent factors. For each scenario, the number of latent factors is equal to the number of cell types ($K = C$). The block structure of the matrices is well aligned with the cell types, especially for IHPF.*

Indeed, the silhouette coefficients (SC) for the k-means clusterings obtained from the cell scores of these three methods (Figure 6) confirm that IHPF cell scores lead to well separated clusters in all Scenarios, and PCA and INMF can also produce reasonable separability (especially for Scenario C, where data integration is not crucial).

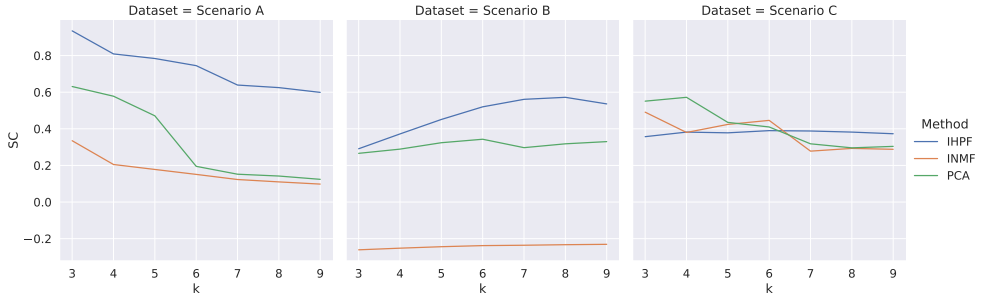


FIG. 6. *Silhouette coefficient (SC) of k-means clusterings (as a function of k) obtained from the $N \times K$ cell scores of PCA, INMF and IHPF for Scenarios A-C. Well separated clusterings, indicated by high SC values, are obtained from IHPF cell scores in all Scenarios.*

Block-structure in gene space. The results above examine the structure of the latent factor scores in cell space, but in many cases it is advantageous to induce

clusterings that also display high concentration and explainability in gene space. We have thus analysed the k-means clusterings obtained from the $K \times M$ matrices of gene scores (e.g., the matrix of common gene scores B in the case of IHPF). In contrast with the situation observed for the cell scores in Figure 5, we find that only the IHPF *gene scores* exhibit a well defined block structure across all three data scenarios (Figure 7). Consequently, the SC in Figure 8 shows that only IHPF gene latent factors induce well-separated gene clusters associated with the extracted latent factors. In summary, 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, as explored below.

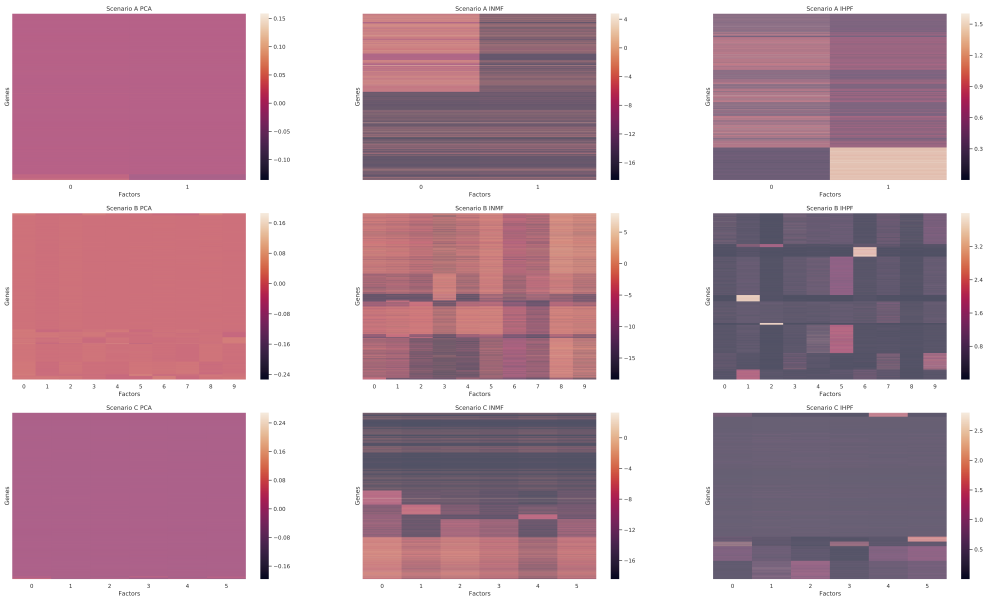


FIG. 7. *Gene latent factor scores ($K \times M$ matrices) obtained from PCA, INMF and IHPF for Scenarios A-C. As above, $K = C$. The rows (genes) are ordered using spectral co-clustering for ease of visualisation. There is only well-aligned block structure for the IHPF gene scores.*

Biological homogeneity of extracted gene clusters . To characterise further the biological interpretability of the gene clusters extracted with IHPF, we proceed as follows. We start by selecting the number of clusters that gives the highest Silhouette coefficient (SC) of the gene scores (Figure 8). For Scenario A, the optimal number of gene clusters is 3. For Scenario B, the optimal number of gene clusters is 8. For Scenario C, the optimal number of gene clusters is 10. Note that the number of gene clusters is different from the number of cell types, as there is no one-to-one mapping between cell types and gene groups—a cell type can be linked to multiple gene groups and *vice versa*.

We then compute the biological homogeneity index (BHI) [32,33] of the obtained gene clusterings. The BHI is a measure of within-cluster biological similarity of the Gene Ontology (GO) terms associated with the genes present in each cluster. Specif-

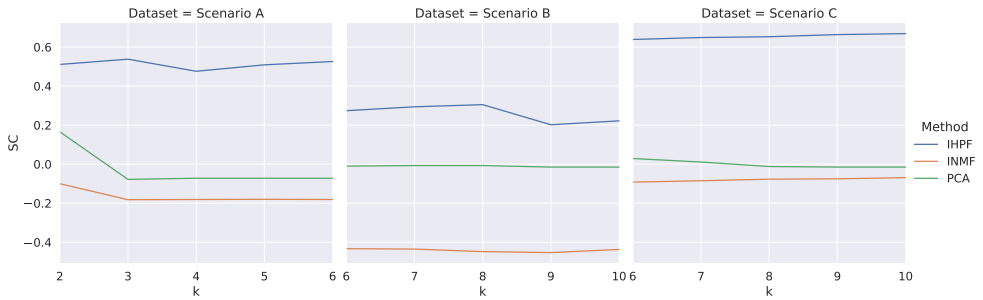


FIG. 8. *Silhouette coefficient (SC) of k-means clusterings (as a function of k) obtained from the gene scores of PCA, INMF and IHPF for Scenarios A-C. Only IHPF gene factors lead to well separated clusters with high SC values whereas INMF and PCA gene factors lead to poor clusters with negative values of SC.*

ically, for a clustering $\{C_k\}_{k=1}^K$ with K clusters C_k, the BHI is given by:

$$\mathrm{BHI}\left(\{C_k\}_{k=1}^K\right) = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k(n_k-1)}\sum_{\substack{i,j\in C_k\\ i\neq j}} S(i,j)$$

where n_k is the size of cluster C_k, and $S(i,j)$ is the biological similarity between gene i and gene j, obtained using an information-theoretic semantic similarity based on GO terms. The BHI gives a score between 0 and 1 [34]. For further details on the calculation of the BHI, see Section A.4 in the Supplementary Information.

For each gene clustering, we also compute the BHI of 100 randomised surrogate clusterings with the same number and sizes of clusters. We then compute the z-score and p-value of the BHI of the IHPF clustering against the ensemble of random clusterings to obtain a level of significance for the obtained clustering [35].

The results in Table 5 show that the clusterings obtained from the IHPF gene scores for Scenarios B and C are significant, with small p-values. Hence the gene clusterings correspond to meaningful biological groupings of genes.

TABLE 5

The z-scores and p-values for the BHI of the gene clusterings obtained from IHPF gene factors. The gene clusterings were obtained using k-means with $k = 3, 8, 10$ for Scenarios A, B and C, respectively.

	Scenario A	Scenario B	Scenario C
z-score	1.982	2.506	4.356
p-value	0.024	0.0061	0.0000066

5. Discussion. Here we have presented a computational method to carry out dimensionality reduction and data integration for single-cell RNAseq data based on integrative hierarchical Poisson factorisation (IHPF). The proposed algorithm extends HPF and performs data integration by modelling the technical noise between data sets with the use of latent factors adapted to batch labels.

As in other data integration methods (e.g., INMF), the noise ratio hyper-parameter 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 experimental data integration scenarios. In cases when there is good overlap of cell types across batches and a large set of common genes (Scenario A), or the other extreme when batches and cell types are highly aligned with little overlap of cell types across batches (Scenario C), data integration brings little advantage. In that case, IHPF indicates that a close to zero noise ratio is appropriate, and still provides very good performance. On the other hand, in more complex situations (where batches are technically different and contain a varied mixture of cells), IHPF offers a flexible and controlled way to correct the effect of batch effects and can outperform other methods. 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 requires pre-processing of cells to have normalised sizes. In addition, IHPF is shown to recover factors with dual block structure in both the cell and gene spaces, thus enhancing the explainability of 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, 36, 37].

Acknowledgments. We thank Dr Zijing Liu for helpful discussions. We acknowledge funding 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.

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.

[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.

[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] R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef, "Deep generative modeling for single-cell transcriptomics," *Nature methods*, vol. 15, no. 12, pp. 1053–1058, 2018.

[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.

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

[12] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

[18] K. PolaÅski, M. D. Young, Z. Miao, K. B. Meyer, S. A. Teichmann, and J.-E. Park, "BBKNN: fast batch alignment of single cell transcriptomes," *Bioinformatics*, vol. 36, no. 3, pp. 964–965, 08 2019.

[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] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, pp. 859–877, 01 2016.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[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.

[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.

[30] Z. Liu and M. Barahona, "Graph-based data clustering via multiscale community detection," *Applied Network Science*, vol. 5, no. 1, p. 3, 2020.

[31] L. van der Maaten, "Accelerating t-sne using tree-based algorithms," *Journal of Machine Learning Research*, vol. 15, pp. 3221–3245, 2014.

[32] G. Brock, V. Pihur, S. Datta, and S. Datta, "clvalid: An r package for cluster validation," *Journal of Statistical Software*, vol. 25, no. 4, p. 1â22, 2008. [Online]. Available: https://www.jstatsoft.org/index.php/jss/article/view/v025i04

[33] S. Datta and S. Datta, "Methods for evaluating clustering algorithms for gene expression data using a reference set of functional classes," *BMC Bioinformatics*, vol. 7, no. 1, p. 397, Aug 2006.

[34] G. Yu, F. Li, Y. Qin, X. Bo, Y. Wu, and S. Wang, "GOSemSim: an R package for measuring semantic similarity among GO terms and gene products," *Bioinformatics*, vol. 26, no. 7, pp. 976–978, 02 2010.

[35] Z. Liu and M. Barahona, "Similarity measure for sparse time course data based on gaussian processes," in *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, ser. Proceedings of Machine Learning Research, C. de Campos and M. H. Maathuis, Eds., vol. 161. PMLR, 27–30 Jul 2021, pp. 1332–1341.

[36] R. Lambiotte, J.-C. Delvenne, and M. Barahona, "Random walks, markov processes and the multiscale modular organization of complex networks," *IEEE Transactions on Network Science and Engineering*, vol. 1, no. 2, pp. 76–90, 2014.

[37] 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.

Appendix A. Supplementary Information.

A.1. Data pre-processing for the different scRNA-seq methods. Our pre-processing starts with quality control on the data sets, whereby we filter cells with a low transcript count. For each of the computational methods, we follow the pre-processing steps recommended by the original authors using Scanpy [28]. Specifically, we use the following functions:

- Scanorama: sc.pp.recipe_zheng17
- scVI: sc.pp.normalize_total, sc.pp.log1p
- BBKNN: sc.pp.recipe_zheng17
- INMF: None
- IHPF: None

A.2. Leave-One-Batch-Out cross-validation. Detailed results of the leave-one-batch-out cross-validation likelihood for each of the batches are shown in Figure 9. The weighted average of these are presented in Figure 3 in the main text.

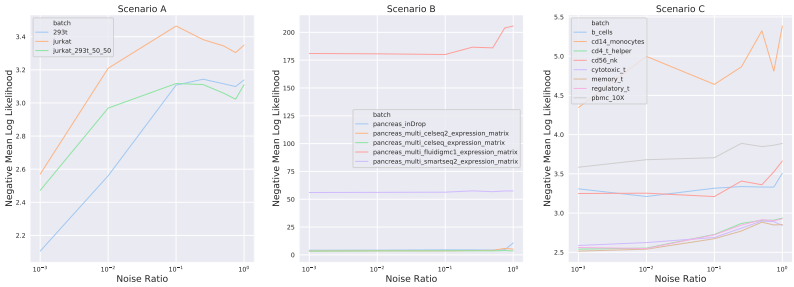


FIG. 9. *Cross-validation negative mean Poisson likelihood for each scenario. For every batch in each scenario, IHPF model is fitted using the rest of the batches for different noise ratios.*

A.3. Visualisation. Additional results on the visualisation of data sets using TSNE and PCA with cells colored by batch labels are shown in Figure 10.

(a) TSNE from IHPF factors (coloured by batch labels)

(b) TSNE from PCA components (coloured by batch labels)

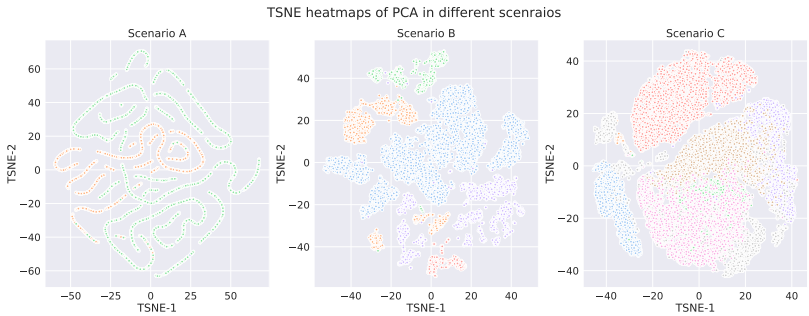


Fig. 10. *TSNE projections of Scenarios A-C obtained from latent factors from (a) IHPF and (b) PCA, and coloured by batch label.*

A.4. Computation of BHI. The genes in Scenarios A, B and C are mapped to GO terms using the org.Hs.eg.db annotation package. A total of 60.88 % of genes in Scenario A, 98.55 % of genes in Scenario B and 60.88 % of genes in Scenario C are mapped to a valid GO term. After that, the BHI is computed using the R package clValid [32].