Supplementary Material for "Binned multinomial logistic regression for integrative cell type annotation"

Keshav Motwani[1], Rhonda Bacher[2,3], and Aaron J. Molstad[3,4]
Department of Biostatistics, University of Washington[1]
Department of Biostatistics[2], Genetics Institute[3], and
Department of Statistics[4], University of Florida

1 Identifiability

In Section 3.1 of the main text, we discussed the case of non-identifiablity caused by the multinomial logistic link, and described the sum-to-zero constraint that alleviates this. There is another, rare, case of possible non-identifiability of $\boldsymbol{\gamma}_{(k)}$ which we describe here. If there exists some $k \in [K]$ for which there exists $j \in \mathcal{C}_k$ for which $|g_k(j)| > 1$ and there exists $l, l' \in g_k(j)$ with $l \neq l'$ and $\boldsymbol{\alpha}_l = \boldsymbol{\alpha}_{l'}$ and $\boldsymbol{\beta}_l = \boldsymbol{\beta}_{l'}$, then $\boldsymbol{\gamma}_{(k)}$ is not identifiable, as the l and l' columns of $\boldsymbol{\gamma}_{(k)}$ can be swapped with no changes to the probabilities.

2 Technical details

2.1 Gradient derivations

In this section we derive the gradients required in Algorithm 1. We begin by finding the gradient of $\boldsymbol{\beta} \mapsto \mathcal{F}_{0,0}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}, \boldsymbol{\gamma}^t)$, which we denote by $\nabla_{\boldsymbol{\beta}} \mathcal{F}_{0,0}(\boldsymbol{\alpha}^t, \cdot, \boldsymbol{\gamma}^t)$. We first focus on the partial derivative of $\boldsymbol{\beta} \mapsto L_{(k)i}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}, \boldsymbol{\gamma}^t_{(k)})$ with respect to $\boldsymbol{\beta}_{a,b}$, which we denote $\{\partial L_{(k)i} / \partial \boldsymbol{\beta}_{a,b}\}(\boldsymbol{\alpha}^t, \cdot, \boldsymbol{\gamma}^t)$. Recall

$$L_{(k)i}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}, \boldsymbol{\gamma}^t_{(k)}) = \sum_{j \in \mathcal{C}_k} \mathbb{1}(y_{(k)i} = j) \log \left(\frac{\sum_{l \in g_k(j)} \exp(\boldsymbol{\alpha}^t_l + \boldsymbol{x}^\top_{(k)i} \boldsymbol{\beta}_l + \boldsymbol{z}^\top_{(k)i} \boldsymbol{\gamma}^t_{(k)l})}{\sum_{v \in \mathcal{C}} \exp(\boldsymbol{\alpha}^t_v + \boldsymbol{x}^\top_{(k)i} \boldsymbol{\beta}_v + \boldsymbol{z}^\top_{(k)i} \boldsymbol{\gamma}^t_{(k)v})} \right),$$

and thus

$$\begin{aligned}
\frac{\partial L_{(k)i}}{\partial \boldsymbol{\beta}_{a,b}}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}, \boldsymbol{\gamma}^t) &= \sum_{j \in \mathcal{C}_k} \Bigg\{ \mathbb{1}(y_{(k)i} = j) \left(\frac{\sum_{v \in \mathcal{C}} \exp(\boldsymbol{\alpha}_v^t + \boldsymbol{x}_{(k)i}^\top \boldsymbol{\beta}_v + \boldsymbol{z}_{(k)i}^\top \boldsymbol{\gamma}_{(k)v}^t)}{\sum_{l \in g_k(j)} \exp(\boldsymbol{\alpha}_l^t + \boldsymbol{x}_{(k)i}^\top \boldsymbol{\beta}_l + \boldsymbol{z}_{(k)i}^\top \boldsymbol{\gamma}_{(k)l}^t)} \right) \\
&\qquad \Bigg(\frac{\sum_{l \in g_k(j)} \mathbb{1}(l = b) \boldsymbol{x}_{(k)i,a} \exp(\boldsymbol{\alpha}_l^t + \boldsymbol{x}_{(k)i}^\top \boldsymbol{\beta}_l + \boldsymbol{z}_{(k)i}^\top \boldsymbol{\gamma}_{(k)l}^t)}{\sum_{v \in \mathcal{C}} \exp(\boldsymbol{\alpha}_v^t + \boldsymbol{x}_{(k)i}^\top \boldsymbol{\beta}_v + \boldsymbol{z}_{(k)i}^\top \boldsymbol{\gamma}_{(k)v}^t)} \\
&\qquad - \frac{\sum_{l \in g_k(j)} \exp(\boldsymbol{\alpha}_l^t + \boldsymbol{x}_{(k)i}^\top \boldsymbol{\beta}_l + \boldsymbol{z}_{(k)i}^\top \boldsymbol{\gamma}_{(k)l}^t)}{\sum_{v \in \mathcal{C}} \exp(\boldsymbol{\alpha}_v^t + \boldsymbol{x}_{(k)i}^\top \boldsymbol{\beta}_v + \boldsymbol{z}_{(k)i}^\top \boldsymbol{\gamma}_{(k)v}^t)} \\
&\qquad\qquad \frac{\sum_{v \in \mathcal{C}} \mathbb{1}(v = b) \boldsymbol{x}_{(k)i,a} \exp(\boldsymbol{\alpha}_v^t + \boldsymbol{x}_{(k)i}^\top \boldsymbol{\beta}_v + \boldsymbol{z}_{(k)i}^\top \boldsymbol{\gamma}_{(k)v}^t)}{\sum_{v \in \mathcal{C}} \exp(\boldsymbol{\alpha}_v^t + \boldsymbol{x}_{(k)i}^\top \boldsymbol{\beta}_v + \boldsymbol{z}_{(k)i}^\top \boldsymbol{\gamma}_{(k)v}^t)} \Bigg) \Bigg\} \\
&= \frac{\sum_{l \in g_k(y_{(k)i})} \mathbb{1}(l = b) \boldsymbol{x}_{(k)i,a} \exp(\boldsymbol{\alpha}_l^t + \boldsymbol{x}_{(k)i}^\top \boldsymbol{\beta}_l + \boldsymbol{z}_{(k)i}^\top \boldsymbol{\gamma}_{(k)l}^t)}{\sum_{l \in g_k(y_{(k)i})} \exp(\boldsymbol{\alpha}_l^t + \boldsymbol{x}_{(k)i}^\top \boldsymbol{\beta}_l + \boldsymbol{z}_{(k)i}^\top \boldsymbol{\gamma}_{(k)l}^t)} \\
&\quad - \frac{\sum_{v \in \mathcal{C}} \mathbb{1}(v = b) \boldsymbol{x}_{(k)i,a} \exp(\boldsymbol{\alpha}_v^t + \boldsymbol{x}_{(k)i}^\top \boldsymbol{\beta}_v + \boldsymbol{z}_{(k)i}^\top \boldsymbol{\gamma}_{(k)v}^t)}{\sum_{v \in \mathcal{C}} \exp(\boldsymbol{\alpha}_v^t + \boldsymbol{x}_{(k)i}^\top \boldsymbol{\beta}_v + \boldsymbol{z}_{(k)i}^\top \boldsymbol{\gamma}_{(k)v}^t)} \\
&= \boldsymbol{x}_{(k)i,a} \left\{ [\widetilde{\boldsymbol{C}}_{(k)}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}, \boldsymbol{\gamma}_{(k)}^t)]_{i,b} - [\widetilde{\boldsymbol{P}}_{(k)}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}, \boldsymbol{\gamma}_{(k)}^t)]_{i,b} \right\}
\end{aligned}$$

where $\widetilde{\boldsymbol{P}}_{(k)}$ and $\widetilde{\boldsymbol{C}}_{(k)}$ are as in (5) and (6) from the main text. Hence

$$\begin{aligned}
\frac{\partial \mathcal{F}_{0,0}}{\partial \boldsymbol{\beta}_{a,b}}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}, \boldsymbol{\gamma}^t) &= -\frac{1}{N} \sum_{k=1}^K \sum_{i=1}^{n_k} \frac{\partial L_{(k)i}}{\partial \boldsymbol{\beta}_{a,b}}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}, \boldsymbol{\gamma}^t) \\
&= -\frac{1}{N} \sum_{k=1}^K \sum_{i=1}^{n_k} \boldsymbol{x}_{(k)i,a} \left\{ [\widetilde{\boldsymbol{C}}_{(k)}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}, \boldsymbol{\gamma}_{(k)}^t)]_{i,b} - [\widetilde{\boldsymbol{P}}_{(k)}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}, \boldsymbol{\gamma}_{(k)}^t)]_{i,b} \right\} \\
&= \frac{1}{N} \sum_{k=1}^K \boldsymbol{X}_{(k):,a}^\top \left\{ [\widetilde{\boldsymbol{P}}_{(k)}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}, \boldsymbol{\gamma}_{(k)}^t)]_{:,b} - [\widetilde{\boldsymbol{C}}_{(k)}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}, \boldsymbol{\gamma}_{(k)}^t)]_{:,b} \right\}.
\end{aligned}$$

From this, it is clear that

$$\nabla_{\boldsymbol{\beta}} \mathcal{F}_{0,0}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}, \boldsymbol{\gamma}^t) = \frac{1}{N} \sum_{k=1}^K \boldsymbol{X}_{(k)}^\top \left\{ \widetilde{\boldsymbol{P}}_{(k)}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}, \boldsymbol{\gamma}_{(k)}^t) - \widetilde{\boldsymbol{C}}_{(k)}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}, \boldsymbol{\gamma}_{(k)}^t) \right\}.$$

Applying identical arguments for $\boldsymbol{\gamma}$ and $\boldsymbol{\alpha}$, we also get

$$\begin{aligned}
\nabla_{\boldsymbol{\gamma}_{(k)}} \mathcal{F}_{0,0}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}^{t+1}, \boldsymbol{\gamma}) &= \frac{1}{N} \boldsymbol{Z}_{(k)}^\top \left\{ \widetilde{\boldsymbol{P}}_{(k)}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}^{t+1}, \boldsymbol{\gamma}_{(k)}) - \widetilde{\boldsymbol{C}}_{(k)}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}^{t+1}, \boldsymbol{\gamma}_{(k)}) \right\}, \quad k \in [K] \\
\nabla_{\boldsymbol{\alpha}} \mathcal{F}_{0,0}(\boldsymbol{\alpha}, \boldsymbol{\beta}^{t+1}, \boldsymbol{\gamma}^{t+1}) &= \frac{1}{N} \sum_{k=1}^K \left\{ \widetilde{\boldsymbol{P}}_{(k)}(\boldsymbol{\alpha}, \boldsymbol{\beta}^{t+1}, \boldsymbol{\gamma}_{(k)}^{t+1}) - \widetilde{\boldsymbol{C}}_{(k)}(\boldsymbol{\alpha}, \boldsymbol{\beta}^{t+1}, \boldsymbol{\gamma}_{(k)}^{t+1}) \right\}^\top \mathbf{1}_{n_k}.
\end{aligned}$$

2.2 Lipschitz continuity of the gradient

When constructing a majorizing function for the $\boldsymbol{\beta}$ update in (4) from the main text, we relied on the Lipschitz continuity/constant of $\nabla_{\boldsymbol{\beta}}\mathcal{F}_{0,0}(\boldsymbol{\alpha}^t, \cdot, \boldsymbol{\gamma}^t)$, thus we now prove this. Specifically, we want to show there exists $L_{\boldsymbol{\beta}}$ which satisfies that for fixed $\boldsymbol{\alpha}^t$ and $\boldsymbol{\gamma}^t$ and for all $\boldsymbol{\beta}', \boldsymbol{\beta}''$,

$$\left\|\nabla_{\boldsymbol{\beta}}\mathcal{F}_{0,0}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}', \boldsymbol{\gamma}^t) - \nabla_{\boldsymbol{\beta}}\mathcal{F}_{0,0}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}'', \boldsymbol{\gamma}^t)\right\|_F \le L_{\boldsymbol{\beta}} \left\|\boldsymbol{\beta}' - \boldsymbol{\beta}''\right\|_F . \tag{1}$$

We use an argument similar to that used by Powers et al. (2018). First we derive Lipschitz constants $L_{\boldsymbol{\beta}}^{\boldsymbol{P}}$ and $L_{\boldsymbol{\beta}}^{\boldsymbol{C}}$ such that

$$\left|\left[\widetilde{\boldsymbol{P}}_{(k)}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}', \boldsymbol{\gamma}^t_{(k)})\right]_{i,l} - \left[\widetilde{\boldsymbol{P}}_{(k)}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}'', \boldsymbol{\gamma}^t_{(k)})\right]_{i,l}\right| \le L_{\boldsymbol{\beta}}^{\boldsymbol{P}} \left\|\boldsymbol{\beta}' - \boldsymbol{\beta}''\right\|_F \tag{2}$$

and

$$\left|\left[\widetilde{\boldsymbol{C}}_{(k)}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}', \boldsymbol{\gamma}^t_{(k)})\right]_{i,l} - \left[\widetilde{\boldsymbol{C}}_{(k)}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}'', \boldsymbol{\gamma}^t_{(k)})\right]_{i,l}\right| \le L_{\boldsymbol{\beta}}^{\boldsymbol{C}} \left\|\boldsymbol{\beta}' - \boldsymbol{\beta}''\right\|_F \tag{3}$$

We obtain these Lipschitz constants by bounding norms of $\nabla_{\boldsymbol{\beta}}\widetilde{\boldsymbol{P}}_{(k)i,l}$ and $\nabla_{\boldsymbol{\beta}}\widetilde{\boldsymbol{C}}_{(k)i,l}$, where $\nabla_{\boldsymbol{\beta}}\widetilde{\boldsymbol{P}}_{(k)i,l}$ is the gradient of the function $\boldsymbol{\beta} \mapsto [\widetilde{\boldsymbol{P}}_{(k)}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}, \boldsymbol{\gamma}^t_{(k)})]_{i,l}$ and analogously for $\nabla_{\boldsymbol{\beta}}\widetilde{\boldsymbol{C}}_{(k)i,l}$. For sake of brevity, slightly abusing notation, we abbreviate $\widetilde{\boldsymbol{P}}_{(k)i,l} = [\widetilde{\boldsymbol{P}}_{(k)}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}, \boldsymbol{\gamma}^t_{(k)})]_{i,l}$ to mean the evaluated value instead of the function itself in this section, except when referring to gradients or partial derivatives of the function itself. We have that

$$\begin{aligned}
\frac{\partial \widetilde{\boldsymbol{P}}_{(k)i,l}}{\partial \boldsymbol{\beta}_{a,b}}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}, \boldsymbol{\gamma}^t_{(k)}) &= \frac{\mathbb{1}(b=l)\boldsymbol{x}_{(k)i,a}\exp(\boldsymbol{\alpha}^t_l + \boldsymbol{x}^\top_{(k)i}\boldsymbol{\beta}_l + \boldsymbol{z}^\top_{(k)i}\boldsymbol{\gamma}^t_{(k)l})}{\sum_{v\in\mathcal{C}}\exp(\boldsymbol{\alpha}^t_v + \boldsymbol{x}^\top_{(k)i}\boldsymbol{\beta}_v + \boldsymbol{z}^\top_{(k)i}\boldsymbol{\gamma}^t_{(k)v})} \\
&\quad - \frac{\exp(\boldsymbol{\alpha}^t_l + \boldsymbol{x}^\top_{(k)i}\boldsymbol{\beta}_l + \boldsymbol{z}^\top_{(k)i}\boldsymbol{\gamma}^t_{(k)l})}{\sum_{v\in\mathcal{C}}\exp(\boldsymbol{\alpha}^t_v + \boldsymbol{x}^\top_{(k)i}\boldsymbol{\beta}_v + \boldsymbol{z}^\top_{(k)i}\boldsymbol{\gamma}^t_{(k)v})} \\
&\qquad \frac{\sum_{v\in\mathcal{C}}\mathbb{1}(v=b)\boldsymbol{x}_{(k)i,a}\exp(\boldsymbol{\alpha}^t_v + \boldsymbol{x}^\top_{(k)i}\boldsymbol{\beta}_v + \boldsymbol{z}^\top_{(k)i}\boldsymbol{\gamma}^t_{(k)v})}{\sum_{v\in\mathcal{C}}\exp(\boldsymbol{\alpha}^t_v + \boldsymbol{x}^\top_{(k)i}\boldsymbol{\beta}_v + \boldsymbol{z}^\top_{(k)i}\boldsymbol{\gamma}^t_{(k)v})} \\
&= \boldsymbol{x}_{(k)i,a}\{\mathbb{1}(b=l)\widetilde{\boldsymbol{P}}_{(k)i,l} - \widetilde{\boldsymbol{P}}_{(k)i,l}\widetilde{\boldsymbol{P}}_{(k)i,b}\} \\
&= \boldsymbol{x}_{(k)i,a}\begin{cases} -\widetilde{\boldsymbol{P}}_{(k)i,l}\widetilde{\boldsymbol{P}}_{(k)i,b}, & b \ne l, \\ \widetilde{\boldsymbol{P}}_{(k)i,l}(1 - \widetilde{\boldsymbol{P}}_{(k)i,l}), & b = l. \end{cases}
\end{aligned}$$

Thus we have that

$$\nabla_{\boldsymbol{\beta}}\widetilde{\boldsymbol{P}}_{(k)i,l} = \boldsymbol{x}_{(k)i}\boldsymbol{v}^{i,l\top}_{(k)}$$

where

$$[\boldsymbol{v}^{i,l}_{(k)}]_b = \begin{cases} -\widetilde{\boldsymbol{P}}_{(k)i,l}\widetilde{\boldsymbol{P}}_{(k)i,b}, & b \ne l, \\ \widetilde{\boldsymbol{P}}_{(k)i,l}(1 - \widetilde{\boldsymbol{P}}_{(k)i,l}), & b = l. \end{cases}$$

Because $0 \le \widetilde{\boldsymbol{P}}_{(k)i,l} \le 1$ for any $l \in \mathcal{C}$, we have that

$$\left\|\boldsymbol{v}^{i,l}_{(k)}\right\|_2 \le \left\|\boldsymbol{v}^{i,l}_{(k)}\right\|_1 = \widetilde{\boldsymbol{P}}_{(k)i,l}(1 - \widetilde{\boldsymbol{P}}_{(k)i,l}) + \sum_{b\ne l}\widetilde{\boldsymbol{P}}_{(k)i,l}\widetilde{\boldsymbol{P}}_{(k)i,b} = 2\widetilde{\boldsymbol{P}}_{(k)i,l}(1 - \widetilde{\boldsymbol{P}}_{(k)i,l}) \le \frac{1}{2},$$

meaning

$$\left\|\nabla_{\boldsymbol{\beta}} \widetilde{\boldsymbol{P}}_{(k)i,l}\right\|_F \leq \left\|\boldsymbol{x}_{(k)i}\right\|_2 \left\|\boldsymbol{v}_{(k)}^{i,l}\right\|_2 \leq \frac{1}{2}\left\|\boldsymbol{x}_{(k)i}\right\|_2$$

Finally, using the mean value theorem, we can conclude that

$$L_{\boldsymbol{\beta}}^{\boldsymbol{P}} = \frac{1}{2}\left\|\boldsymbol{x}_{(k)i}\right\|_2$$

satisfies the desired inequality in (2).

We can also find $L_{\boldsymbol{\beta}}^{\boldsymbol{C}}$ which satisfies (3) by repeating the same argument. We have that

$$\begin{aligned}
\frac{\partial \widetilde{\boldsymbol{C}}_{(k)i,l}}{\partial \boldsymbol{\beta}_{a,b}}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}, \boldsymbol{\gamma}_{(k)}^t) &= \frac{\mathbb{1}\{l \in g_k(y_{(k)i})\}\mathbb{1}(b=l)\boldsymbol{x}_{(k)i,a}\exp(\boldsymbol{\alpha}_l + \boldsymbol{x}_{(k)i}^\top\boldsymbol{\beta}_l + \boldsymbol{z}_{(k)i}^\top\boldsymbol{\gamma}_{(k)l})}{\sum_{v\in g_k(y_{(k)i})}\exp(\boldsymbol{\alpha}_v + \boldsymbol{x}_{(k)i}^\top\boldsymbol{\beta}_v + \boldsymbol{z}_{(k)i}^\top\boldsymbol{\gamma}_{(k)v})} \\
&\quad - \frac{\mathbb{1}\{l \in g_k(y_{(k)i})\}\exp(\boldsymbol{\alpha}_l + \boldsymbol{x}_{(k)i}^\top\boldsymbol{\beta}_l + \boldsymbol{z}_{(k)i}^\top\boldsymbol{\gamma}_{(k)l})}{\sum_{v\in g_k(y_{(k)i})}\exp(\boldsymbol{\alpha}_v + \boldsymbol{x}_{(k)i}^\top\boldsymbol{\beta}_v + \boldsymbol{z}_{(k)i}^\top\boldsymbol{\gamma}_{(k)v})} \\
&\quad \frac{\sum_{v\in g_k(y_{(k)i})}\mathbb{1}(v=b)\boldsymbol{x}_{(k)i,a}\exp(\boldsymbol{\alpha}_v + \boldsymbol{x}_{(k)i}^\top\boldsymbol{\beta}_v + \boldsymbol{z}_{(k)i}^\top\boldsymbol{\gamma}_{(k)v})}{\sum_{v\in g_k(y_{(k)i})}\exp(\boldsymbol{\alpha}_v + \boldsymbol{x}_{(k)i}^\top\boldsymbol{\beta}_v + \boldsymbol{z}_{(k)i}^\top\boldsymbol{\gamma}_{(k)v})} \\
&= \frac{\mathbb{1}\{l \in g_k(y_{(k)i})\}\mathbb{1}(b=l)\boldsymbol{x}_{(k)i,a}\exp(\boldsymbol{\alpha}_l + \boldsymbol{x}_{(k)i}^\top\boldsymbol{\beta}_l + \boldsymbol{z}_{(k)i}^\top\boldsymbol{\gamma}_{(k)l})}{\sum_{v\in g_k(y_{(k)i})}\exp(\boldsymbol{\alpha}_v + \boldsymbol{x}_{(k)i}^\top\boldsymbol{\beta}_v + \boldsymbol{z}_{(k)i}^\top\boldsymbol{\gamma}_{(k)v})} \\
&\quad - \frac{\mathbb{1}\{l \in g_k(y_{(k)i})\}\exp(\boldsymbol{\alpha}_l + \boldsymbol{x}_{(k)i}^\top\boldsymbol{\beta}_l + \boldsymbol{z}_{(k)i}^\top\boldsymbol{\gamma}_{(k)l})}{\sum_{v\in g_k(y_{(k)i})}\exp(\boldsymbol{\alpha}_v + \boldsymbol{x}_{(k)i}^\top\boldsymbol{\beta}_v + \boldsymbol{z}_{(k)i}^\top\boldsymbol{\gamma}_{(k)v})} \\
&\quad \frac{\mathbb{1}\{b \in g_k(y_{(k)i})\}\boldsymbol{x}_{(k)i,a}\exp(\boldsymbol{\alpha}_b + \boldsymbol{x}_{(k)i}^\top\boldsymbol{\beta}_b + \boldsymbol{z}_{(k)i}^\top\boldsymbol{\gamma}_{(k)b})}{\sum_{v\in g_k(y_{(k)i})}\exp(\boldsymbol{\alpha}_v + \boldsymbol{x}_{(k)i}^\top\boldsymbol{\beta}_v + \boldsymbol{z}_{(k)i}^\top\boldsymbol{\gamma}_{(k)v})} \\
&= \boldsymbol{x}_{(k)i,a}\{\mathbb{1}(b=l)\widetilde{\boldsymbol{C}}_{(k)i,l} - \widetilde{\boldsymbol{C}}_{(k)i,l}\widetilde{\boldsymbol{C}}_{(k)i,b}\} \\
&= \boldsymbol{x}_{(k)i,a}\begin{cases} -\widetilde{\boldsymbol{C}}_{(k)i,l}\widetilde{\boldsymbol{C}}_{(k)i,b}, & b \neq l, \\ \widetilde{\boldsymbol{C}}_{(k)i,l}(1-\widetilde{\boldsymbol{C}}_{(k)i,l}), & b = l. \end{cases}
\end{aligned}$$

Now replacing $\widetilde{\boldsymbol{P}}_{(k)}$ with $\widetilde{\boldsymbol{C}}_{(k)}$ for the rest of the argument used in finding $L_{\boldsymbol{\beta}}^{\boldsymbol{P}}$, we get that

$$L_{\boldsymbol{\beta}}^{\boldsymbol{C}} = \frac{1}{2}\left\|\boldsymbol{x}_{(k)i}\right\|_2$$

satisfies the desired inequality in (3).

Now we use these results to find $L_{\boldsymbol{\beta}}$ that satisfies (1). For brevity, we abbreviate

$\widetilde{\boldsymbol{P}}_{(k)}(\boldsymbol{\beta}) = \widetilde{\boldsymbol{P}}_{(k)}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}, \boldsymbol{\gamma}^t_{(k)})$ and $\widetilde{\boldsymbol{C}}_{(k)}(\boldsymbol{\beta}) = \widetilde{\boldsymbol{C}}_{(k)}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}, \boldsymbol{\gamma}^t_{(k)})$. We have that

$$\begin{aligned}
&\left\|\nabla_{\boldsymbol{\beta}}\mathcal{F}_{0,0}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}', \boldsymbol{\gamma}^t) - \nabla_{\boldsymbol{\beta}}\mathcal{F}_{0,0}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}'', \boldsymbol{\gamma}^t)\right\|_F \\
&= \left\| \frac{1}{N}\sum_{k=1}^{K} \boldsymbol{X}_{(k)}^\top \left\{ \widetilde{\boldsymbol{P}}_{(k)}(\boldsymbol{\beta}') - \widetilde{\boldsymbol{C}}_{(k)}(\boldsymbol{\beta}') \right\} - \frac{1}{N}\sum_{k=1}^{K} \boldsymbol{X}_{(k)}^\top \left\{ \widetilde{\boldsymbol{P}}_{(k)}(\boldsymbol{\beta}'') - \widetilde{\boldsymbol{C}}_{(k)}(\boldsymbol{\beta}'') \right\} \right\|_F \\
&\le \frac{1}{N}\sum_{k=1}^{K} \left\| \boldsymbol{X}_{(k)}^\top \left[\left\{ \widetilde{\boldsymbol{P}}_{(k)}(\boldsymbol{\beta}') - \widetilde{\boldsymbol{C}}_{(k)}(\boldsymbol{\beta}') \right\} - \left\{ \widetilde{\boldsymbol{P}}_{(k)}(\boldsymbol{\beta}'') - \widetilde{\boldsymbol{C}}_{(k)}(\boldsymbol{\beta}'') \right\} \right] \right\|_F \\
&\le \frac{1}{N}\sum_{k=1}^{K} \left\|\boldsymbol{X}_{(k)}\right\|_F \left\{ \left\| \widetilde{\boldsymbol{P}}_{(k)}(\boldsymbol{\beta}') - \widetilde{\boldsymbol{P}}_{(k)}(\boldsymbol{\beta}'') \right\|_F + \left\| \widetilde{\boldsymbol{C}}_{(k)}(\boldsymbol{\beta}') - \widetilde{\boldsymbol{C}}_{(k)}(\boldsymbol{\beta}'') \right\|_F \right\} \\
&\le \frac{1}{N}\sum_{k=1}^{K} \left\|\boldsymbol{X}_{(k)}\right\|_F \left\{ \sqrt{\sum_{i=1}^{n_k}\sum_{l\in\mathcal{C}} ([\widetilde{\boldsymbol{P}}_{(k)}(\boldsymbol{\beta}')]_{i,l} - [\widetilde{\boldsymbol{P}}_{(k)}(\boldsymbol{\beta}'')]_{i,l})^2} + \sqrt{\sum_{i=1}^{n_k}\sum_{l\in\mathcal{C}} ([\widetilde{\boldsymbol{C}}_{(k)}(\boldsymbol{\beta}')]_{i,l} - [\widetilde{\boldsymbol{C}}_{(k)}(\boldsymbol{\beta}'')]_{i,l})^2} \right\} \\
&\le \frac{1}{N}\sum_{k=1}^{K} \left\|\boldsymbol{X}_{(k)}\right\|_F \left\{ \sqrt{\sum_{i=1}^{n_k}\sum_{l\in\mathcal{C}} \left(\frac{1}{2}\left\|\boldsymbol{x}_{(k)i}\right\|_2 \|\boldsymbol{\beta}' - \boldsymbol{\beta}''\|_F \right)^2} + \sqrt{\sum_{i=1}^{n_k}\sum_{l\in\mathcal{C}} \left(\frac{1}{2}\left\|\boldsymbol{x}_{(k)i}\right\|_2 \|\boldsymbol{\beta}' - \boldsymbol{\beta}''\|_F \right)^2} \right\} \\
&= \frac{1}{N}\sum_{k=1}^{K} \left\|\boldsymbol{X}_{(k)}\right\|_F \|\boldsymbol{\beta}' - \boldsymbol{\beta}''\|_F \sqrt{\sum_{i=1}^{n_k}\sum_{l\in\mathcal{C}} \left\|\boldsymbol{x}_{(k)i}\right\|_2^2} \\
&= \left(\frac{\sqrt{|\mathcal{C}|}}{N} \sum_{k=1}^{K} \left\|\boldsymbol{X}_{(k)}\right\|_F^2 \right) \|\boldsymbol{\beta}' - \boldsymbol{\beta}''\|_F .
\end{aligned}$$

Therefore

$$L_{\boldsymbol{\beta}} = \frac{\sqrt{|\mathcal{C}|}}{N} \sum_{k=1}^{K} \left\|\boldsymbol{X}_{(k)}\right\|_F^2$$

satisfies the desired inequality in (1), and in conclusion, (4) from the main text is satisfied for $s_{\boldsymbol{\beta}} \le \frac{1}{L_{\boldsymbol{\beta}}} = N/\{\sqrt{|\mathcal{C}|}\sum_{k=1}^{K}\|\boldsymbol{X}_{(k)}\|_F^2\}$.

Similarly, repeating the same argument, one could show that

$$L_{\boldsymbol{\gamma}_{(k)}} = \frac{\sqrt{|\mathcal{C}|}}{N} \left\|\boldsymbol{Z}_{(k)}\right\|_F^2$$

satisfies

$$\begin{aligned}
&\left\|\nabla_{\boldsymbol{\gamma}_{(k)}}\mathcal{F}_{0,0}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}^{t+1}, \boldsymbol{\gamma}^t_{(1)}, \ldots, \boldsymbol{\gamma}'_{(k)}, \ldots, \boldsymbol{\gamma}^t_{(K)}) - \nabla_{\boldsymbol{\gamma}_{(k)}}\mathcal{F}_{0,0}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}^{t+1}, \boldsymbol{\gamma}^t_{(1)}, \ldots, \boldsymbol{\gamma}''_{(k)}, \ldots, \boldsymbol{\gamma}^t_{(K)})\right\|_F \\
&\qquad\qquad \le L_{\boldsymbol{\gamma}_{(k)}} \left\|\boldsymbol{\gamma}'_{(k)} - \boldsymbol{\gamma}''_{(k)}\right\|_F
\end{aligned}$$

where $\nabla_{\boldsymbol{\gamma}_{(k)}}\mathcal{F}_{0,0}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}^{t+1}, \boldsymbol{\gamma}^t_{(1)}, \ldots, \cdot, \ldots, \boldsymbol{\gamma}^t_{(K)})$ denotes the gradient of

$$\boldsymbol{\gamma}_{(k)} \mapsto \mathcal{F}_{0,0}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}^{t+1}, \boldsymbol{\gamma}^t_{(1)}, \ldots, \boldsymbol{\gamma}_{(k)}, \ldots, \boldsymbol{\gamma}^t_{(K)})$$

and $L_{\boldsymbol{\alpha}} = \sqrt{|\mathcal{C}|}$ satisfies

$$\left\|\nabla_{\boldsymbol{\alpha}}\mathcal{F}_{0,0}(\boldsymbol{\alpha}', \boldsymbol{\beta}^{t+1}, \boldsymbol{\gamma}^{t+1}) - \nabla_{\boldsymbol{\alpha}}\mathcal{F}_{0,0}(\boldsymbol{\alpha}'', \boldsymbol{\beta}^{t+1}, \boldsymbol{\gamma}^{t+1})\right\|_F \le L_{\boldsymbol{\alpha}}\|\boldsymbol{\alpha}' - \boldsymbol{\alpha}''\|_F$$

where $\nabla_{\boldsymbol{\alpha}}\mathcal{F}_{0,0}(\cdot, \boldsymbol{\beta}^{t+1}, \boldsymbol{\gamma}^{t+1})$ denotes the gradient of $\boldsymbol{\alpha} \mapsto \mathcal{F}_{0,0}(\boldsymbol{\alpha}, \boldsymbol{\beta}^{t+1}, \boldsymbol{\gamma}^{t+1})$. Thus, for $s_{\boldsymbol{\gamma}_{(k)}} \le \frac{1}{L_{\boldsymbol{\gamma}_{(k)}}} = N/\{\sqrt{|\mathcal{C}|}\|\boldsymbol{Z}_{(k)}\|_F^2\}$ we have that

$$\begin{aligned}
&\mathcal{F}_{0,0}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}^{t+1}, \boldsymbol{\gamma}^t_{(1)}, \dots, \boldsymbol{\gamma}_{(k)}, \dots, \boldsymbol{\gamma}^t_{(K)}) \\
&\quad \le \mathcal{F}_{0,0}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}^{t+1}, \boldsymbol{\gamma}^t) + \operatorname{tr}\left\{\nabla_{\boldsymbol{\gamma}_{(k)}}\mathcal{F}_{0,0}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}^{t+1}, \boldsymbol{\gamma}^t)^\top(\boldsymbol{\gamma}_{(k)} - \boldsymbol{\gamma}^t_{(k)})\right\} + \frac{1}{2s_{\boldsymbol{\gamma}_{(k)}}}\|\boldsymbol{\gamma}_{(k)} - \boldsymbol{\gamma}^t_{(k)}\|_F^2
\end{aligned}$$

and for $s_{\boldsymbol{\alpha}} \le \frac{1}{L_{\boldsymbol{\alpha}}} = 1/\sqrt{|\mathcal{C}|}$, we have that

$$\mathcal{F}_{0,0}(\boldsymbol{\alpha}, \boldsymbol{\beta}^{t+1}, \boldsymbol{\gamma}^{t+1}) \le \mathcal{F}_{0,0}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}^{t+1}, \boldsymbol{\gamma}^{t+1}) + \operatorname{tr}\left\{\nabla_{\boldsymbol{\alpha}}\mathcal{F}_{0,0}(\boldsymbol{\alpha}^t, \boldsymbol{\beta}^{t+1}, \boldsymbol{\gamma}^{t+1})^\top(\boldsymbol{\alpha} - \boldsymbol{\alpha}^t)\right\} + \frac{1}{2s_{\boldsymbol{\alpha}}}\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^t\|_F^2.$$

3 Implementation details

3.1 IBMR

In this section, we describe how to select a reasonable set of candidate tuning parameters for both λ and ρ. We start by motivating our selection of ρ. If $\widehat{\boldsymbol{\beta}} = 0$ (e.g., if $\lambda = \infty$), then, we know that $\widehat{\boldsymbol{\gamma}}_{(k)}$ is optimal if

$$\frac{1}{N}\boldsymbol{Z}_{(k)}^\top\{\widetilde{\boldsymbol{P}}_{(k)}(\widehat{\boldsymbol{\alpha}}, \mathbf{0}, \widehat{\boldsymbol{\gamma}}_{(k)}) - \widetilde{\boldsymbol{C}}_{(k)}(\widehat{\boldsymbol{\alpha}}, \mathbf{0}, \widehat{\boldsymbol{\gamma}}_{(k)})\} + \rho\widehat{\boldsymbol{\gamma}}_{(k)} = \mathbf{0}.$$

However, there is no (finite) value of ρ that would lead to $\widehat{\boldsymbol{\gamma}}_{(k)} = \mathbf{0}$ being optimal. Thus, for a moment, consider the alternative optimization problem where we have an elastic-net penalty on $\boldsymbol{\gamma}_{(k)}$ rather than just a ridge penalty:

$$\underset{(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma})\in\mathcal{T}}{\arg\min}\left\{-\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) + \lambda\sum_{m=1}^p\|\boldsymbol{\beta}_{m,\cdot}\|_2 + (1-\phi)\frac{\rho}{2}\sum_{k=1}^K\|\boldsymbol{\gamma}_{(k)}\|_F^2 + \phi\rho\sum_{k=1}^K\|\boldsymbol{\gamma}_{(k)}\|_{1,1}\right\}.$$

With $\phi = 0$, this is identical to (3) from the main text. Now if $\widehat{\boldsymbol{\beta}} = 0$ (i.e., $\lambda = \infty$), then, we know that $\widehat{\boldsymbol{\gamma}}_{(k)} = 0$ is optimal if

$$\frac{1}{N}\boldsymbol{Z}_{(k)}^\top\{\widetilde{\boldsymbol{P}}_{(k)}(\widehat{\boldsymbol{\alpha}}, \mathbf{0}, \mathbf{0}) - \widetilde{\boldsymbol{C}}_{(k)}(\widehat{\boldsymbol{\alpha}}, \mathbf{0}, \mathbf{0})\} + \phi\rho\boldsymbol{S} = \mathbf{0}.$$

for some $\boldsymbol{S} \in \mathbb{R}^{r\times|\mathcal{C}|}$ with $|\boldsymbol{S}_{jk}| \le 1$ for all (j, k). This equality will hold if

$$\phi\rho \ge \left\|\frac{1}{N}\boldsymbol{Z}_{(k)}^\top\{\widetilde{\boldsymbol{P}}_{(k)}(\widehat{\boldsymbol{\alpha}}, \mathbf{0}, \mathbf{0}) - \widetilde{\boldsymbol{C}}_{(k)}(\widehat{\boldsymbol{\alpha}}, \mathbf{0}, \mathbf{0})\}\right\|_\infty$$

with $\widehat{\boldsymbol{\alpha}}$ being fit with $\widehat{\boldsymbol{\beta}} = \mathbf{0}$ and $\widehat{\boldsymbol{\gamma}}_{(k)} = \mathbf{0}$ fixed.

Since $\phi = 0$ corresponds to the optimization we want to solve, we can set ϕ to be a small number ($\phi = 10^{-3}$ by default in our implementation), and use this to get $\rho_{\max}$, the largest value of ρ we consider in tuning. For this value, $\widehat{\boldsymbol{\gamma}}_{(k)}$ will intuitively be very close to $\mathbf{0}$, since the two optimization problems are nearly identical. This gives

$$\rho_{\max} = \max_{k \in [K]} \left\{ \frac{\left\| \frac{1}{N} \boldsymbol{Z}_{(k)}^{\top} \{ \widetilde{\boldsymbol{P}}_{(k)}(\widehat{\boldsymbol{\alpha}}, \mathbf{0}, \mathbf{0}) - \widetilde{\boldsymbol{C}}_{(k)}(\widehat{\boldsymbol{\alpha}}, \mathbf{0}, \mathbf{0}) \} \right\|_{\infty}}{\phi} \right\}.$$

We then consider `n_rho` candidate tuning parameters $\bar{\rho} \in [\delta_\rho \rho_{\max}, \rho_{\max}]$ ($\delta_\rho < 1$) equally spaced on the log-scale where $\delta_\rho = 10^{-4}$ and `n_rho` $= 5$ by default. This procedure is similar to how `glmnet` chooses the tuning parameter sequence for ridge regression problems.

For each ρ in our candidate set, we construct a unique set of candidate λ. Let $\bar{\rho}$ be the fixed value of ρ coming from the candidate set and let $\widehat{\boldsymbol{\alpha}}^{\bar{\rho}}$ and $\widehat{\boldsymbol{\gamma}}_{(k)}^{\bar{\rho}}$ denote the optimal values of $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}^{(k)}$ with $\widehat{\boldsymbol{\beta}} = \mathbf{0}$ fixed. Considering the optimization with respect to $\boldsymbol{\beta}$ with $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}^{(k)}$ fixed at $\widehat{\boldsymbol{\alpha}}^{\bar{\rho}}$ and $\widehat{\boldsymbol{\gamma}}_{(k)}^{\bar{\rho}}$, we see that $\widehat{\boldsymbol{\beta}} = \mathbf{0}$ if

$$\frac{1}{N} \sum_{k=1}^{K} \boldsymbol{X}_{(k)}^{\top} \{ \widetilde{\boldsymbol{P}}_{(k)}(\widehat{\boldsymbol{\alpha}}^{\bar{\rho}}, \mathbf{0}, \widehat{\boldsymbol{\gamma}}_{(k)}^{\bar{\rho}}) - \widetilde{\boldsymbol{C}}_{(k)}(\widehat{\boldsymbol{\alpha}}^{\bar{\rho}}, \mathbf{0}, \widehat{\boldsymbol{\gamma}}_{(k)}^{\bar{\rho}}) \} + \lambda \boldsymbol{S} = 0$$

where $\boldsymbol{S} \in \mathbb{R}^{p \times |\mathcal{C}|}$ with $\|\boldsymbol{S}_{j,\cdot}\|_2 \leq 1$ for all $j \in [p]$. This equality will hold if

$$\lambda \geq \max_{j \in [p]} \left\{ \left\| \left[\frac{1}{N} \sum_{k=1}^{K} \boldsymbol{X}_{(k)}^{\top} \{ \widetilde{\boldsymbol{P}}_{(k)}(\widehat{\boldsymbol{\alpha}}^{\bar{\rho}}, \mathbf{0}, \widehat{\boldsymbol{\gamma}}_{(k)}^{\bar{\rho}}) - \widetilde{\boldsymbol{C}}_{(k)}(\widehat{\boldsymbol{\alpha}}^{\bar{\rho}}, \mathbf{0}, \widehat{\boldsymbol{\gamma}}_{(k)}^{\bar{\rho}}) \} \right]_{j,\cdot} \right\|_2 \right\} = \lambda_{\max}$$

Therefore, for each $\bar{\rho}$ in the candidate set, we consider `n_lambda` candidate tuning parameters $\lambda(\bar{\rho}) \in [\delta_\lambda \lambda_{\max}, \lambda_{\max}]$ equally spaced on the log-scale where $\delta_\lambda = 10^{-4}$ and `n_lambda` $= 25$ by default.

We choose the best tuning parameter combination by minimizing the negative log-likelihood on the validation data.

3.2 `IBMR-NG`, `subset`, and `relabel`

For `IBMR-NG`, we choose tuning parameters in the same way as for `IBMR`, setting $\bar{\rho} = \infty$ (i.e. $\widehat{\gamma}_{(k)}^{\bar{\rho}} = \mathbf{0}$).

Regarding our implementation of `subset` and `relabel`: as mentioned in the text, while these models could be fit using `glmnet`, for consistency in the algorithm and convergence criterion (which can have a very slight affect on performance), we fit these models using the our software for `IBMR-NG` with the inputs as described in Section 5.2 and `n_lambda` $= 25$ as with `IBMR-NG`.

3.3 Seurat

We follow the "Mapping and annotating query datasets" Seurat vignette to first integrate the training datasets and then use this integrated reference for prediction on the validation and test datasets. Note that we use the same normalized gene expression data matrix (including same genes) as for all other methods, described in Section 4.

However, note that Seurat cannot handle the different resolution labels across training datasets (which is the primary motivation behind our method). Therefore, we subset the training data to only the cells which are annotated at the finest resolution (up to renaming), similarly to as described for the subset method in Section 5.2 of the main text, and then integrate the subsetted data.

Additionally, while the TransferData function in Seurat provides "scores" per category for prediction on a new cell, these cannot be interpreted as probabilities since they do not sum to 1 (or any constant) in general. Therefore, we cannot evaluate the likelihood function, and thus only report error rates for this method. Moreover, to make "coarse predictions" (as defined in Section 6.1 of the main text) on the validation dataset (for tuning parameter selection) and test dataset (for performance evaluation), we use the prediction from the TransferData function as the "fine prediction", and define the "coarse prediction" as the coarse category to which the "fine prediction" belongs to (unlike how we defined a "coarse prediction" in Section 6.1 of the main text where probabilities per fine category were available). This then allows us to define the error rate based on the observed labels in the validation and test datasets.

We choose the tuning parameters n.dim (number of CCA and PCA components used by this method) and k.anchor (number of nearest neighbors for data integration and prediction) by minimizing the error rate on the validation dataset over the 25 tuning parameter combinations of $\texttt{n.dim} \in \{10, 20, 30, 40, 50\}$ and $\texttt{k.anchor} \in \{3, 5, 10, 15, 20\}$, where this particular choice of values was chosen because the vignette used a value of $\texttt{n.dim} = 30$ and $\texttt{k.anchor} = 5$, but other vignettes (e.g. "Multimodal reference mapping" and "Tips for integrating large datasets") vary n.dim to be as large as 50 and k.anchor as large as 20. We also chose a grid of size 25 to be fair with the other methods we compare to, which all have at least 25 tuning parameter combinations considered (only IBMR-int has more, at 125).

3.4 SingleR

We follow the tutorial in the "Pseudo-bulk aggregation" section of the "SingleR Book" which allows us to use single-cell data as training data for use in the SingleR method.

Once again, just like Seurat, SingleR cannot handle the different resolution labels across training datasets, so we subset the training data to only the cells which are annotated at the finest resolution (up to renaming), similarly to as described for the subset method in Section 5.2 of the main text. SingleR provides "scores" per category (which are based on a summary statistic of correlation of the cell with training data) for prediction on a new cell, but just like Seurat, these cannot be interpreted as probabilities, and thus the likelihood function cannot be evaluated, nor can "coarse predictions" be defined as done in Section 6.1

of the main text. For this reason, we define "coarse prediction" for `SingleR` as we did for `Seurat`.

We choose the tuning parameters `de.n` (number of differentially expressed genes used by this method) and `quantile` (the quantile to use to summarize distribution of correlation of new cell with training data) by minimizing the error rate on the validation dataset over the 25 tuning parameter combinations of `de.n` $\in \{20, 40, 60, 80, 100\}$ and `quantile` $\in \{0.6, 0.7, 0.8, 0.9, 1\}$, where this particular choice of values was chosen because the default value of `de.n` for 28 labels is `de.n` $= 71$, and the default value of `quantile` is `quantile` $= 0.8$. Once again, also chose a grid of size 25 to be fair with the other methods we compare to, which all have at least 25 tuning parameter combinations considered (only `IBMR-int` has more than 25, at 125).

4 Data processing

4.1 Filtering cells

We removed some cells from datasets with certain labels in order to create binning functions which treat the labels in `hao_2020` as the finest resolution categories. The cell type labels which we removed from each dataset are listed in Table 1. For `blish_2020`, `haniffa_2021`, `tsang_2020`, and `su_2020` we also removed cells originating from patients with COVID-19 and only kept cells originating from healthy patients.

For all datasets, we also removed low-quality cells based on the percentage of mitochondrial reads and number of genes expressed (with nonzero counts) in each cell. Specifically, let $\boldsymbol{X}^c_{(k)}$ be the full $\ddot{n}_k \times \ddot{p}$ gene expression count matrix for the kth dataset. Define $s_{(k)i} = \sum_{g=1}^{\ddot{p}} \boldsymbol{X}^c_{(k)i,g}$. Also, let $\mathcal{M} \subset \{1, \ldots, \ddot{p}\}$ be the set of mitochondrial genes (the genes whose names start with "MT-"). Define the percentage of mitochondrial reads to be

$$m_{(k)i} = 100 \cdot \sum_{g \in \mathcal{M}} \frac{\boldsymbol{X}^c_{(k)i,g}}{s_{(k)i}}.$$

Furthermore, define the number of expressed genes to be $e_{(k)i} = \sum_{g=1}^{\ddot{p}} \mathbb{1}(\boldsymbol{X}^c_{(k)i,g} > 0)$. Let $\mathcal{I}_{(k)}$ be the set of cells with no more than 5 percent mitochondrial reads and at least 200 genes expressed

$$\mathcal{I}_{(k)} = \left\{i : m_{(k)i} < 5\right\} \cap \left\{i : e_{(k)i} > 200\right\}$$

and define $\dot{n}_k = |\mathcal{I}_{(k)}|$ and reassign $\boldsymbol{X}^c_{(k)} = \boldsymbol{X}^c_{(k)\mathcal{I}_{(k)},\cdot}$ to be the filtered count matrix.

4.2 Data normalization

Let $\boldsymbol{X}^c_{(k)}$ and $s_{(k)i}$ be defined as in the last section. Then the normalized matrix $\boldsymbol{X}_{(k)}$ is defined by

$$\boldsymbol{X}_{(k)i,g} = \log\left(\frac{10^4 \cdot \boldsymbol{X}^c_{(k)i,g}}{s_{(k)i}} + 1\right), \quad i \in [n_k], \quad g \in [p], \quad k \in [K].$$

	Dataset	Labels	Removed labels
1	`hao_2020`	ASDC, B intermediate, B memory, B naive, CD14 Mono, CD16 Mono, CD4 CTL, CD4 Naive, CD4 TCM, CD4 TEM, CD8 Naive, CD8 TCM, CD8 TEM, cDC1, cDC2, dnT, Eryth, gdT, HSPC, ILC, MAIT, NK, NK˙CD56bright, pDC, Plasmablast, Platelet, Treg Memory, Treg Naive	*Proliferating*
2	`tsang_2021`	CD4-positive, alpha-beta memory T cell, CD8-positive, alpha-beta memory T cell, double negative T cell (DNT), gamma-delta T cell, memory B cell, mucosal invariant T cell (MAIT), naive B cell, naive CD4+ T cell, naive CD8+ T cell, plasmablast, regulatory T cell, NK˙CD16hi, NK˙CD56hiCD16lo, classical monocyte, conventional dendritic cell, non-classical monocyte, plasmacytoid dendritic cell, platelet	double-positive T cell (DPT), granulocyte, intermediate monocyte, NK˙CD56loCD16lo, TCRVbeta13.1pos, TissueResMemT
3	`haniffa_2021`	B˙cell, CD4, CD8, CD14, CD16, DCs, HSC, MAIT, NK˙16hi, NK˙56hi, Plasmablast, Platelets, RBC, Treg, gdT, pDC	*prolif*
4	`su_2020`	CD16- NK, CD16+ NK, classical monocyte, memory B, memory CD4, memory CD8, myeloid DC, naive B, naive CD4, naive CD8, non-classical CD16+ monocyte, plasmacytoid DC, Treg	intermediate monocyte
5	`10x_pbmc_5k_v3`	CD16- NK, CD16+ NK, classical monocyte, DCs, memory B, memory CD4, memory CD8, naive B, naive CD4, naive CD8, non-classical CD16+ monocyte, Treg	intermediate monocyte
6	`blish_2020`	B, CD14 Monocyte, CD16 Monocyte, CD4 T, CD8 T, DC, gd T, NK, PB, pDC, Platelet, RBC	Granulocyte
7	`kotliarov_2020`	B, CD4+ memory T, CD4+ naive T/DNT, CD8+ memory T, CD8+ naive T, Monocyte/mDC, NK, Non-classical monocyte, pDC	Unconv T
8	`10x_pbmc_10k`	B, CD16- NK, CD16+ NK, classical monocyte, memory CD4, memory CD8, naive CD4, naive CD8, Treg	intermediate monocyte
9	`10x_sorted`	CD14+ Monocytes, CD19+ B cells, CD34+ Cells, CD4+/CD25+ Regulatory T Cells, CD4+/CD45RA+/CD25- Naive T cells, CD4+/CD45RO+ Memory T Cells, CD56+ Natural Killer Cells, CD8+/CD45RA+ Naive Cytotoxic T Cells	
10	`ding_2019`	B cell, CD14+ monocyte, CD16+ monocyte, CD4+ T cell, Cytotoxic T cell, Dendritic cell, Natural killer cell, Plasmacytoid dendritic cell	Megakaryocyte

Table 1: Description of labels in each dataset and labels which were removed during preprocessing in order to construct binning functions with respect to the labels used by `hao_2020`. Label names are given exactly as named by the original data source. Asterisks in the removed labels column indicate wildcard expressions.

This is the standard log-normalization used in `Seurat`.

4.3 Ranking of genes

First, we obtain the intersection of genes available across all datasets. We subset all datasets to include these intersection genes only. To rank genes for screening purposes we use the FindVariableFeatures function in Seurat with selection.method = "vst" on the normalized matrix and rank genes according to the vst.variance.standardized column in descending order for each dataset. We refer the reader to Stuart et al. (2019) for the details, but briefly, the mean-variance relationship is estimated by fitting a smooth function to the estimated mean and estimated variance of each gene, and compares the expected value of the variance for a gene, given its estimated mean, to the estimated variance of the gene. We then take the average of ranks across all dataset-patient combinations, and use these ranks to order the genes and reassign $\mathcal{G}$ to be the ordered set of genes. When varying the number of genes p, we take the first p genes from this ordered list, and reassign $\boldsymbol{X}_{(k)} = \boldsymbol{X}_{(k)\cdot,\mathcal{G}_{1:p}}$.

5 Performance metrics

In the simulation studies, we compared methods using Kullback-Leibler (KL) divergence, Hellinger distance, and the error rate. In this section, we give explicit forms for each of these metrics and provide a brief description of why we chose them.

Both KL divergence and Hellinger distance quantify the similarity between two probability distributions. Specifically, suppose we are given a probability mass function P and an estimate of the probability mass function $\widehat{P}$. Then, $P(\boldsymbol{x}_i) = \{\pi_1^*(\boldsymbol{x}_i), \dots, \pi_{|\mathcal{C}|}^*(\boldsymbol{x}_i)\}$ is a vector of probabilities (which are nonnegative and sum to one) and similarly for $\widehat{P}(\boldsymbol{x}_i)$. In our simulations, we define KL divergence as

$$n_{\text{test}}^{-1} \sum_{i=1}^{n_{\text{test}}} \sum_{\ell=1}^{|\mathcal{C}|} \log\left(\frac{\widehat{\pi}_\ell(\boldsymbol{x}_i)}{\pi_\ell^*(\boldsymbol{x}_i)}\right) \widehat{\pi}_\ell(\boldsymbol{x}_i)$$

and Hellinger distance as

$$n_{\text{test}}^{-1} \sum_{i=1}^{n_{\text{test}}} \sqrt{\frac{1}{2} \sum_{\ell=1}^{|\mathcal{C}|} \left(\sqrt{\widehat{\pi}_\ell(\boldsymbol{x}_i)} - \sqrt{\pi_\ell^*(\boldsymbol{x}_i)}\right)^2}.$$

We include both since these are distinct measures of the distance between two probability distributions.

Error rate, in contrast, is simply the classification accuracy. Given a set of testing set responses $\widetilde{y}_1, \dots, \widetilde{y}_{n_{\text{test}}}$ and their corresponding predictors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_{n_{\text{test}}}$, the error rate is simply

$$n_{\text{test}}^{-1} \sum_{i=1}^{n_{\text{test}}} \sum_{\ell=1}^{|\mathcal{C}|} \mathbb{1}(\widetilde{y}_i = \ell) \cdot \mathbb{1}(\arg\max_{k \in \mathcal{C}} \widehat{\pi}_k(\boldsymbol{x}_i) = \ell),$$

where as before, $\mathbb{1}$ is the indicator function. While error rate is of course a useful performance metric, it does not quantify how well the entire probability mass function is estimated. For example, suppose both $\widehat{\pi}_k(\boldsymbol{x}_i) = \pi^*_k(\boldsymbol{x}_i)$ for each ℓ, but $\widetilde{y}_i \neq \arg\max_k \widehat{\pi}_k(\boldsymbol{x}_i)$. Then, even though our probability mass perfectly matches the truth, this testing point would lower the error rate. Thus, it is important to measure both classification accuracy (error rate) and the degree of similarity between both the true and estimated probability mass function.

6 Simulation studies with mislabeled cells

In Supplementary Figure 5, we display simulation studies results under a scenario in which we randomly mislabel fine cell types within coarse cell types. Specifically, we mislabel fine categories within a coarse category (e.g. swap a label which was simulated to be A_1 with A_2, or B_1 with B_2, etc.). If cells are incorrectly annotated, we believe this is the most likely scenario since cells from different coarse categories are generally very distinct. We only mislabel observations in the two datasets (Datasets 5 and 6 from Section 5.1) and vary the mislabeling rate from 5% to 25%. All other simulation details are as in Section 5.1, and we keep $N = 4800$, $p = 500$, $s = 40$, $b = 0.1$ fixed.

7 Supplementary Figures

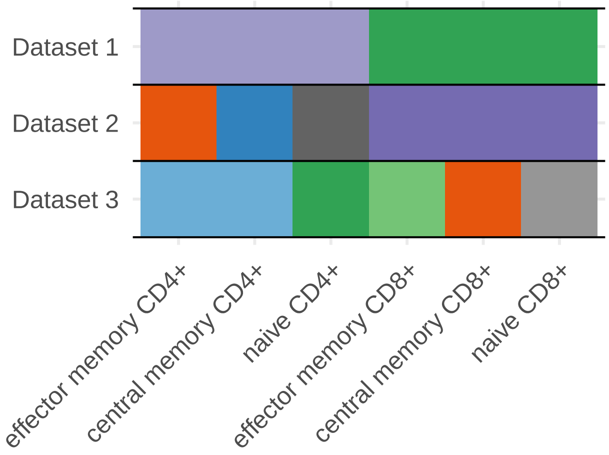

Figure 1: Graphical representation of binning functions corresponding to the example in Figure 1 of the main text (see Section 2.1 of the main text for a description of binning functions), where within each row, a unique color represents a label in that dataset which is a bin of finest resolution categories.

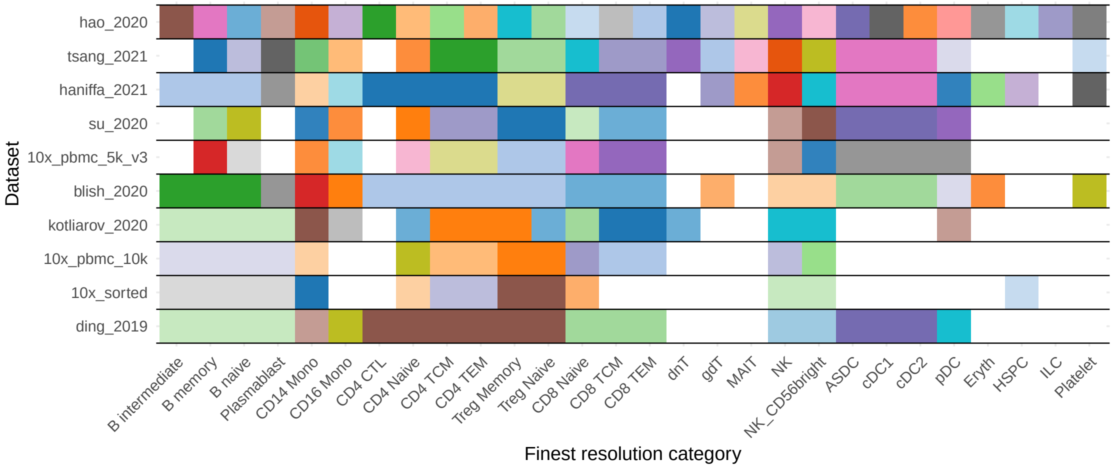

Figure 2: Graphical representation of the relationship between observed (annotated) labels and the finest resolution categories for each of the ten datasets from our integrative analysis in Section 6 of the main text. Within each row, when a color spans multiple finest resolution categories (columns), this indicates cells of these fine resolution categories were "binned" into a broader annotation label (coarse category) represented by the color. For example, in the `ding_2019` dataset (bottom row), each cell was annotated with one of eight distinct labels. One of these labels was "B cell" (represented by a pastel green color), which consists of fine labels "B intermediate", "B memory", "B naive", and "Plasmablast". White spaces denote finest resolution categories which were not represented by the observed labels in a particular dataset.

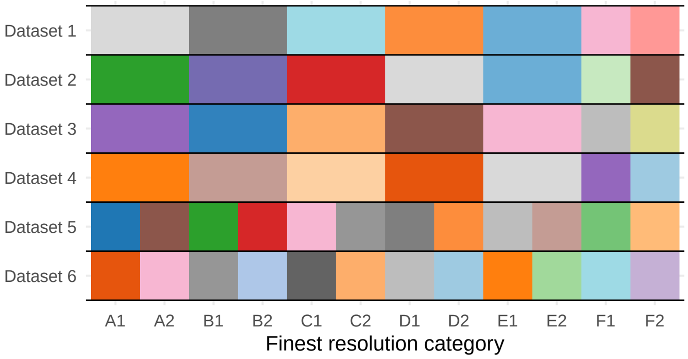

Figure 3: Graphical representation of binning functions used for our simulation study in Section 5 of the main text, where within each row, a unique color represents a label in that dataset which is a bin of the finest resolution categories with that color.

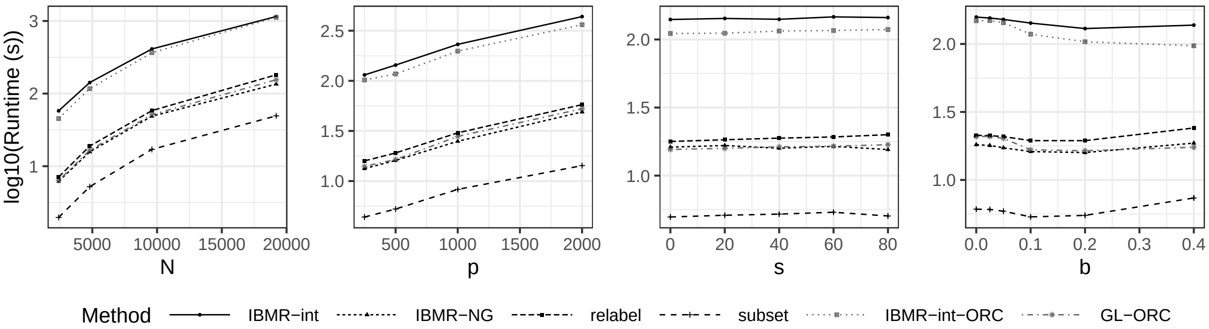

Figure 4: Timing results for simulation study, comparing six competing methods with varying (left) N, the total sample size; (middle left) p, the total number of features; (middle right) s, the number of nonzero features which have shared coefficients for fine categories within a coarse label; and (right) b, the ratio of the norm of the batch effect and norm of the true predictors. Points denote the average and error bars denote the standard error for each method across 50 replicates. Throughout, the defaults are $N = 4800$, $p = 500$, $s = 40$, $b = 0.1$.

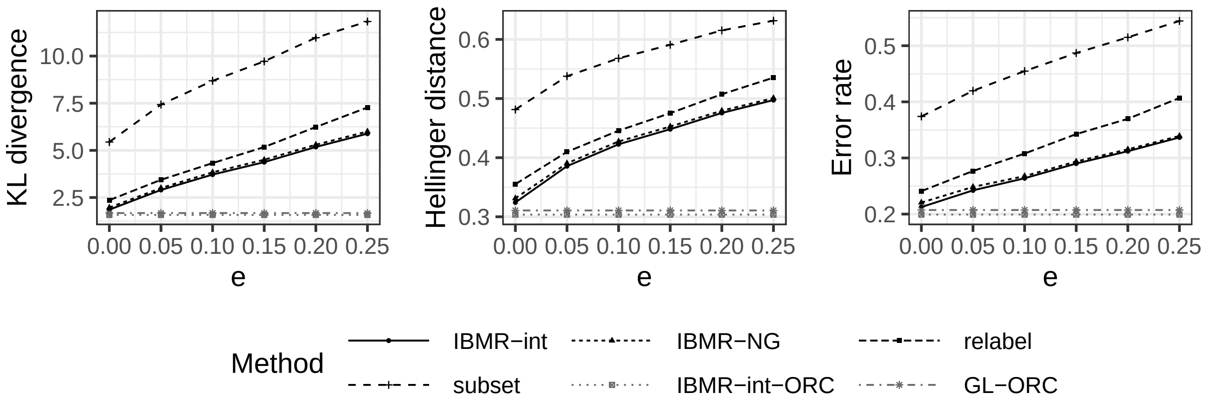

Figure 5: (left) Kullback-Leibler divergence, (middle) Hellinger distance, and (right) error rate for six competing methods with varying e, the percentage of observations in datasets 5 and 6 which were mislabeled within a coarse category and $N = 4800$, $p = 500$, $s = 40$, $b = 0.1$ fixed.

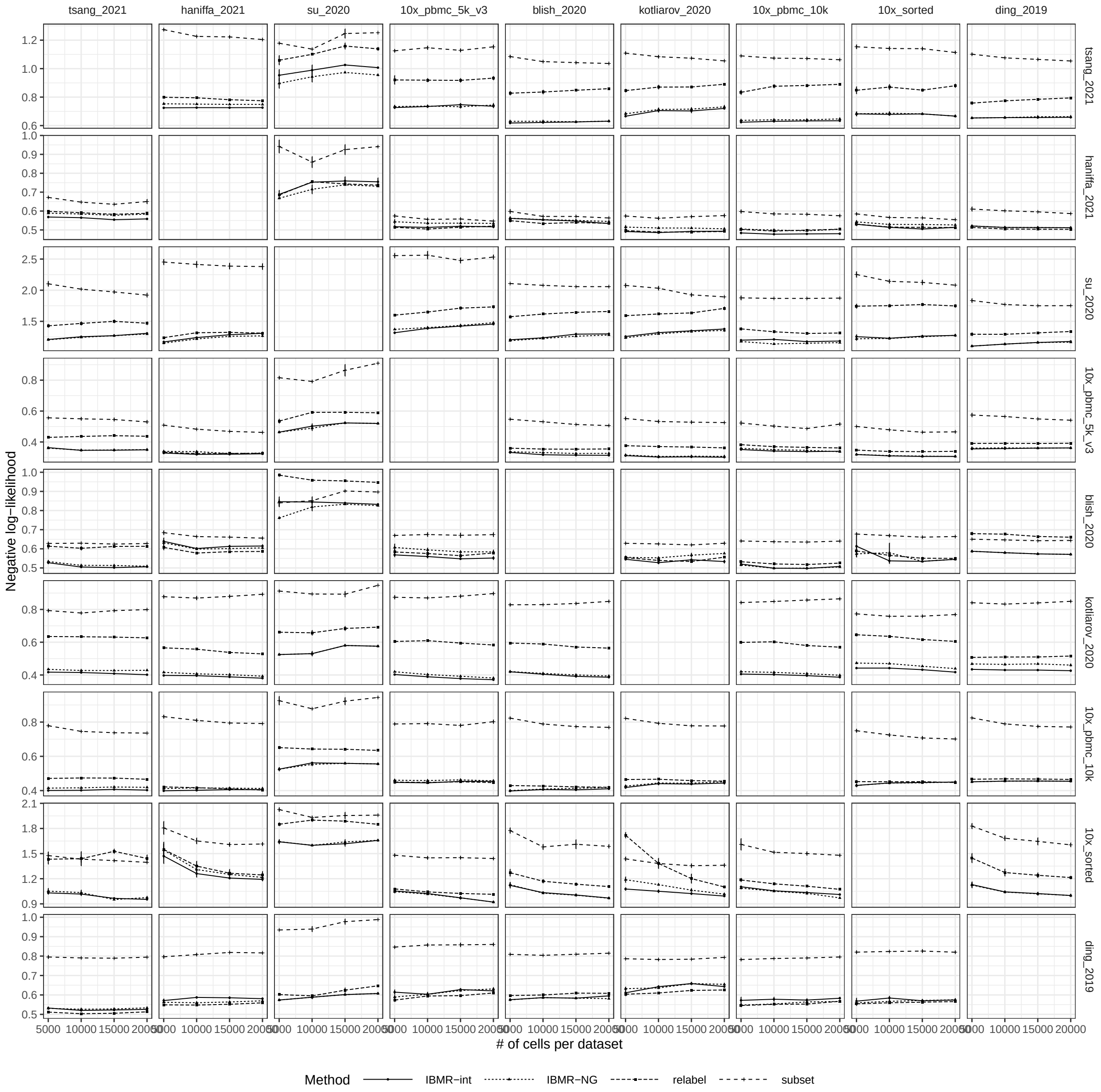

Figure 6: Negative log-likelihood for each method considered, for varying numbers of cells per dataset used for fitting the model with the number of genes $p = 1000$ fixed, for each training/validation/test dataset combination (subplots). The column denotes the validation dataset, the row denotes the test dataset, and the remaining 8 datasets were used for training. Points denote the average and error bars denote the standard error of the negative log-likelihood across 5 replicates of different subsampled training datasets.

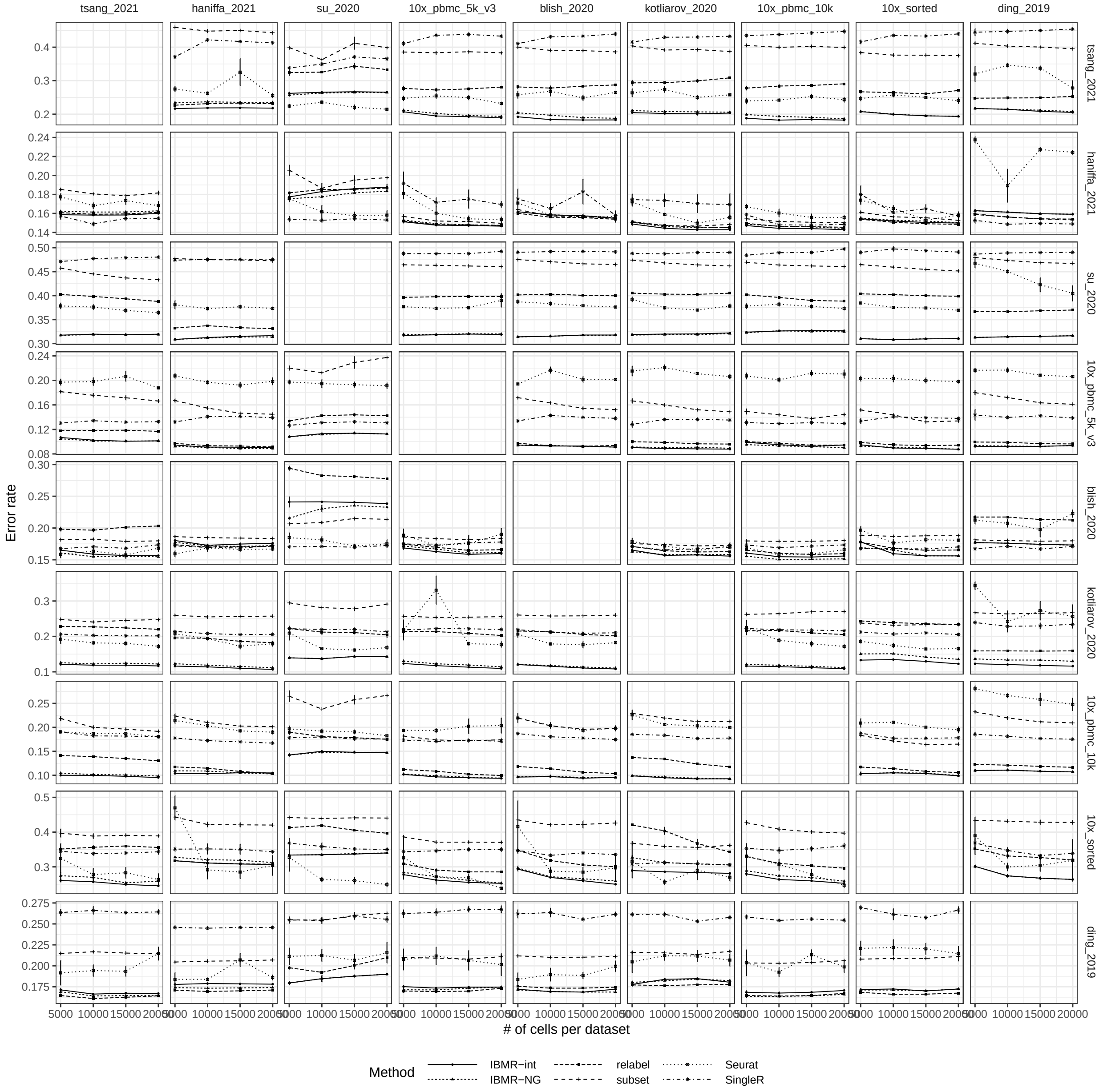

Figure 7: Error rate for each method considered, for varying numbers of cells per dataset used for fitting the model with the number of genes $p = 1000$ fixed, for each training/validation/test dataset combination (subplots). The column denotes the validation dataset, the row denotes the test dataset, and the remaining 8 datasets were used for training. Points denote the average and error bars denote the standard error of the error rate across 5 replicates of different subsampled training datasets.

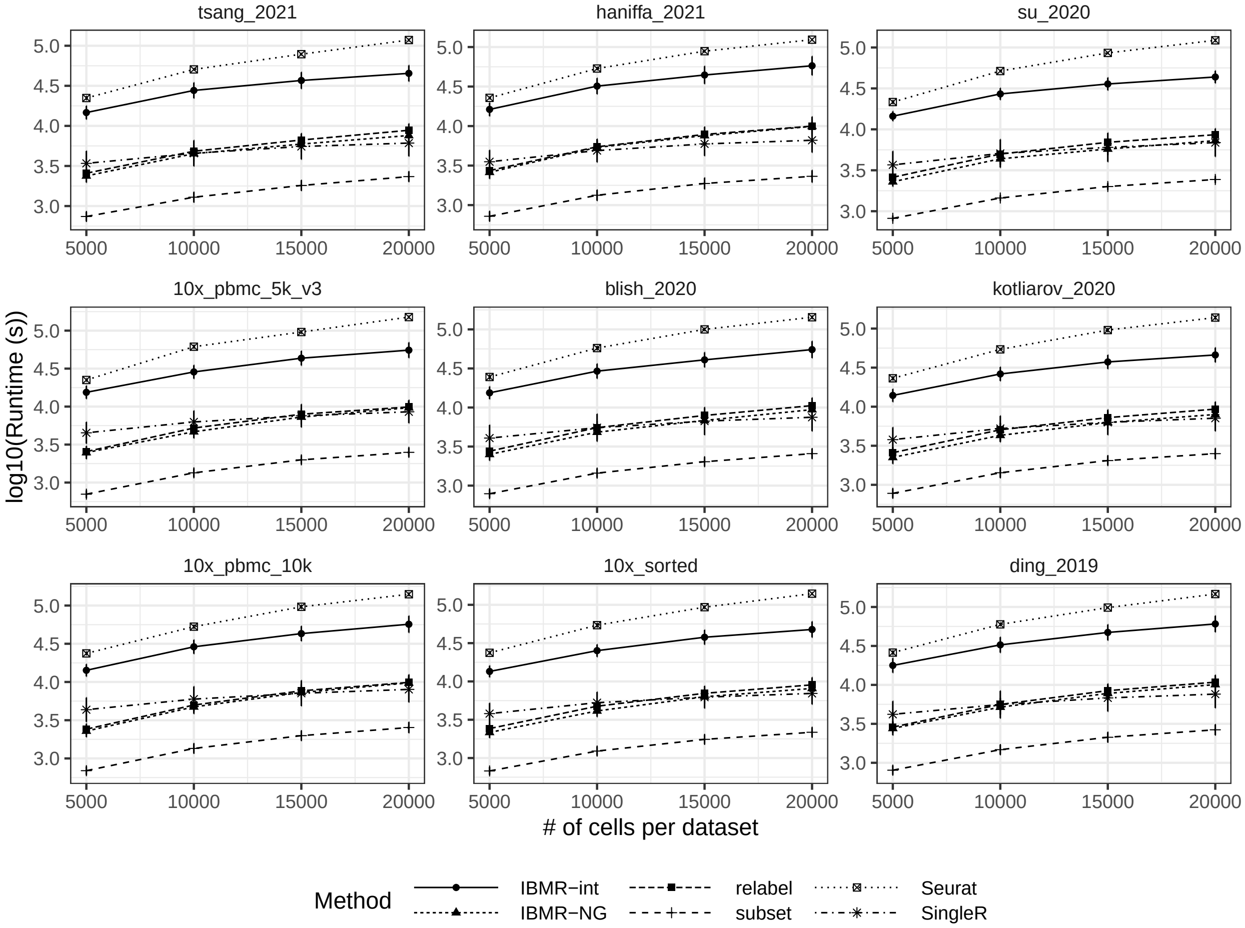

Figure 8: Timing results in real data application rate for each method considered, for each test dataset (subplots), for varying numbers of cells per dataset used for fitting the model with the number of genes $p = 1000$ fixed. Points denote the average and error bars denote the standard error of the average runtime for a test dataset across training/validation dataset combinations, for which the average for each training/validation dataset combination was over five replicates of different subsampled training datasets.

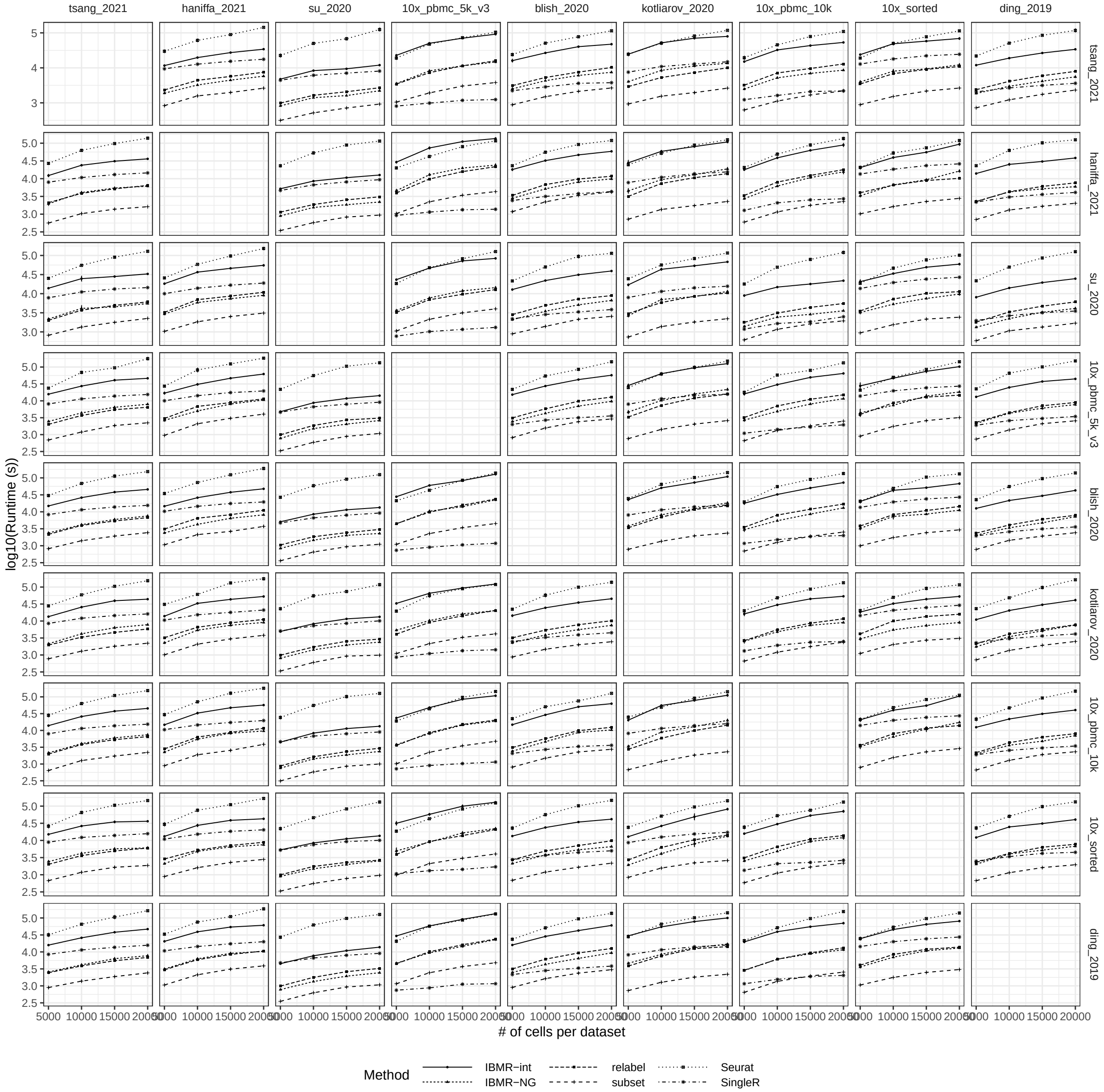

Figure 9: Timing results in real data application for each method considered, for varying numbers of cells per dataset used for fitting the model with the number of genes $p = 1000$ fixed, for each training/validation/test dataset combination (subplots). The column denotes the validation dataset, the row denotes the test dataset, and the remaining 8 datasets were used for training. Points denote the average and error bars denote the standard error of the runtime across 5 replicates of different subsampled training datasets.

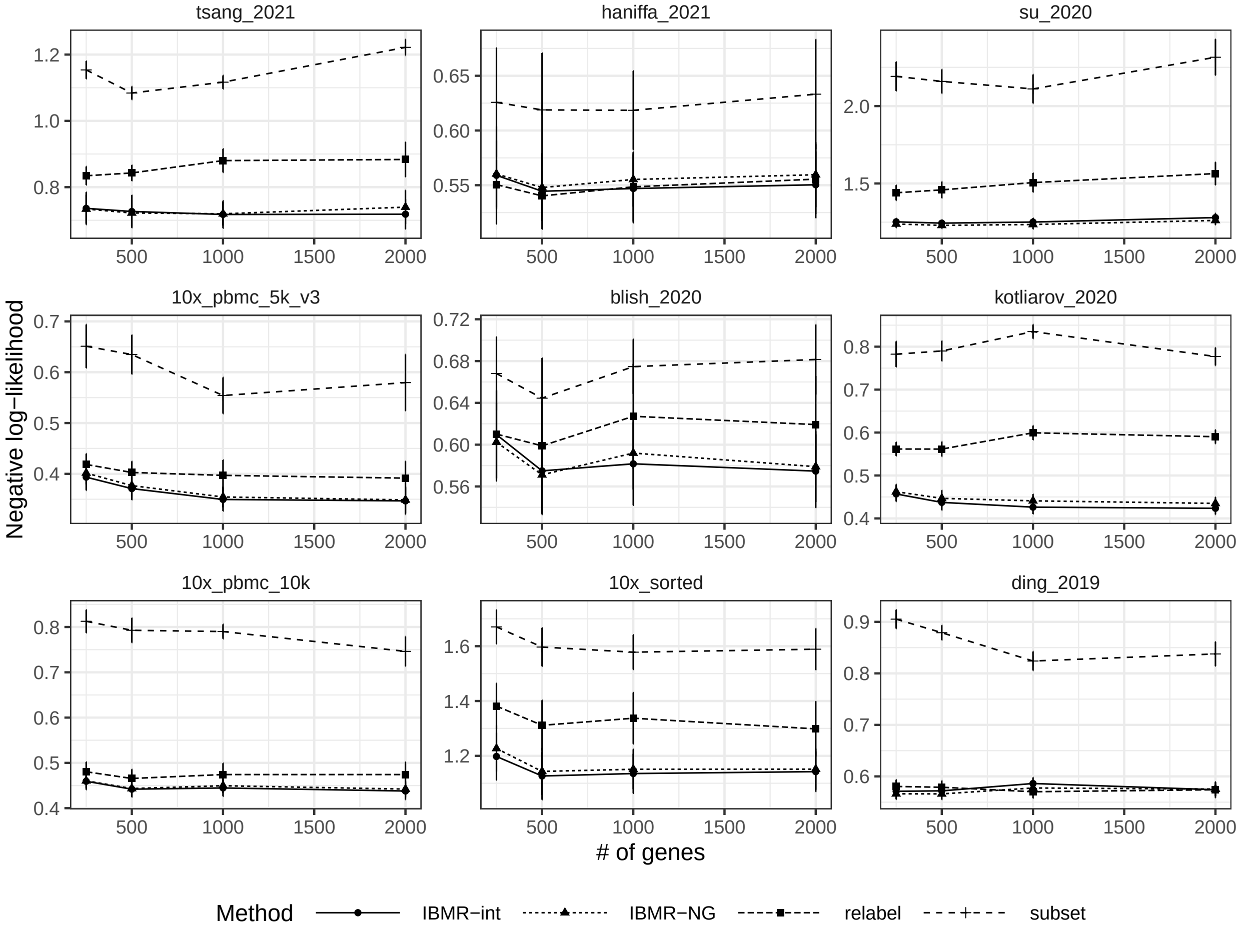

Figure 10: Negative log-likelihood for each method considered, for each test dataset (subplots), for varying numbers of genes used for fitting the model with the number of cells per dataset $n_k = 10000$ fixed. Points denote the average and error bars denote the standard error of the average negative log-likelihood for a test dataset across training/validation dataset combinations, for which the average for each training/validation dataset combination was over five replicates of different subsampled training datasets.

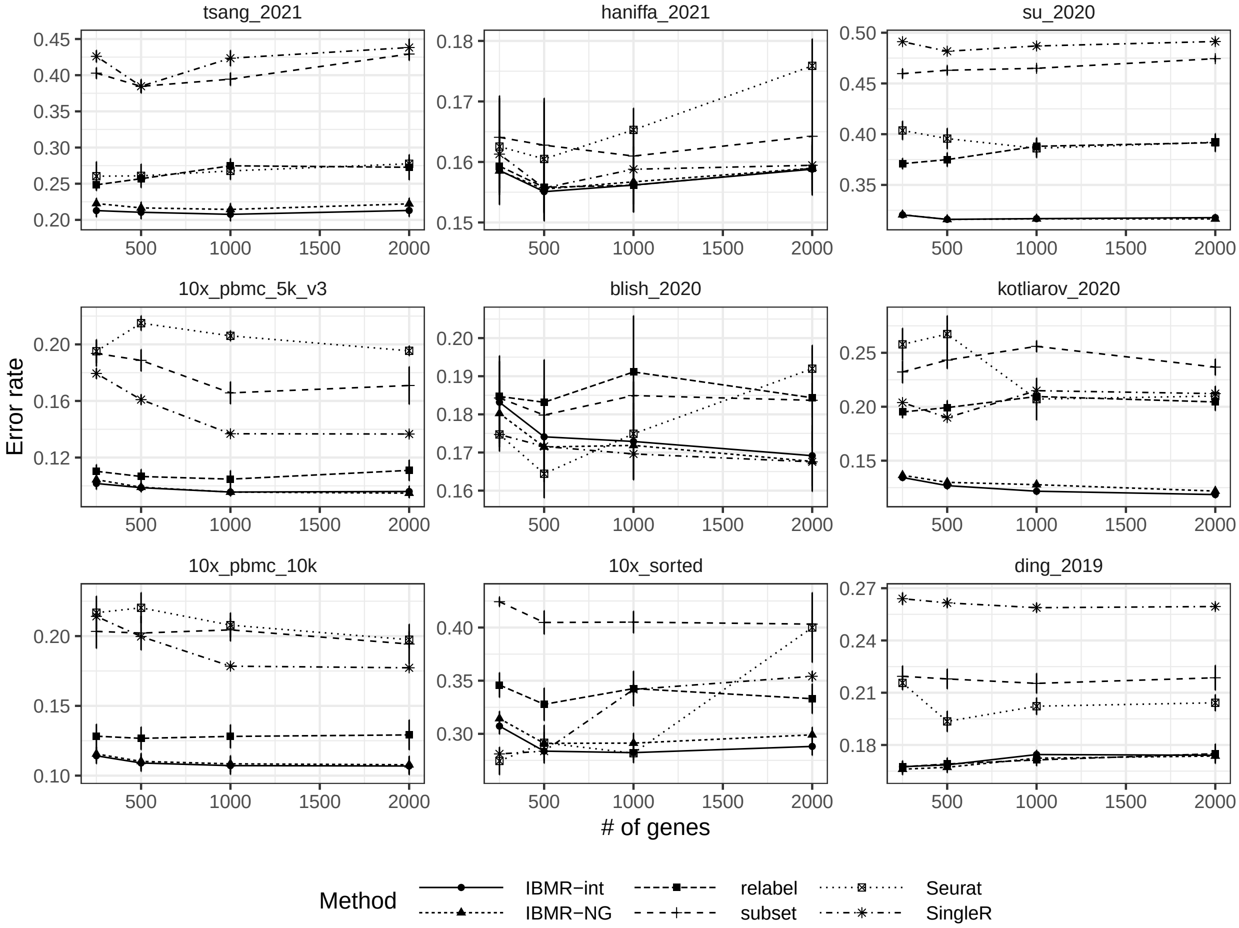

Figure 11: Error rate for each method considered, for each test dataset (subplots), for varying numbers of genes used for fitting the model with the number of cells per dataset $n_k = 10000$ fixed. Points denote the average and error bars denote the standard error of the average error rate for a test dataset across training/validation dataset combinations, for which the average for each training/validation dataset combination was over five replicates of different subsampled training datasets.

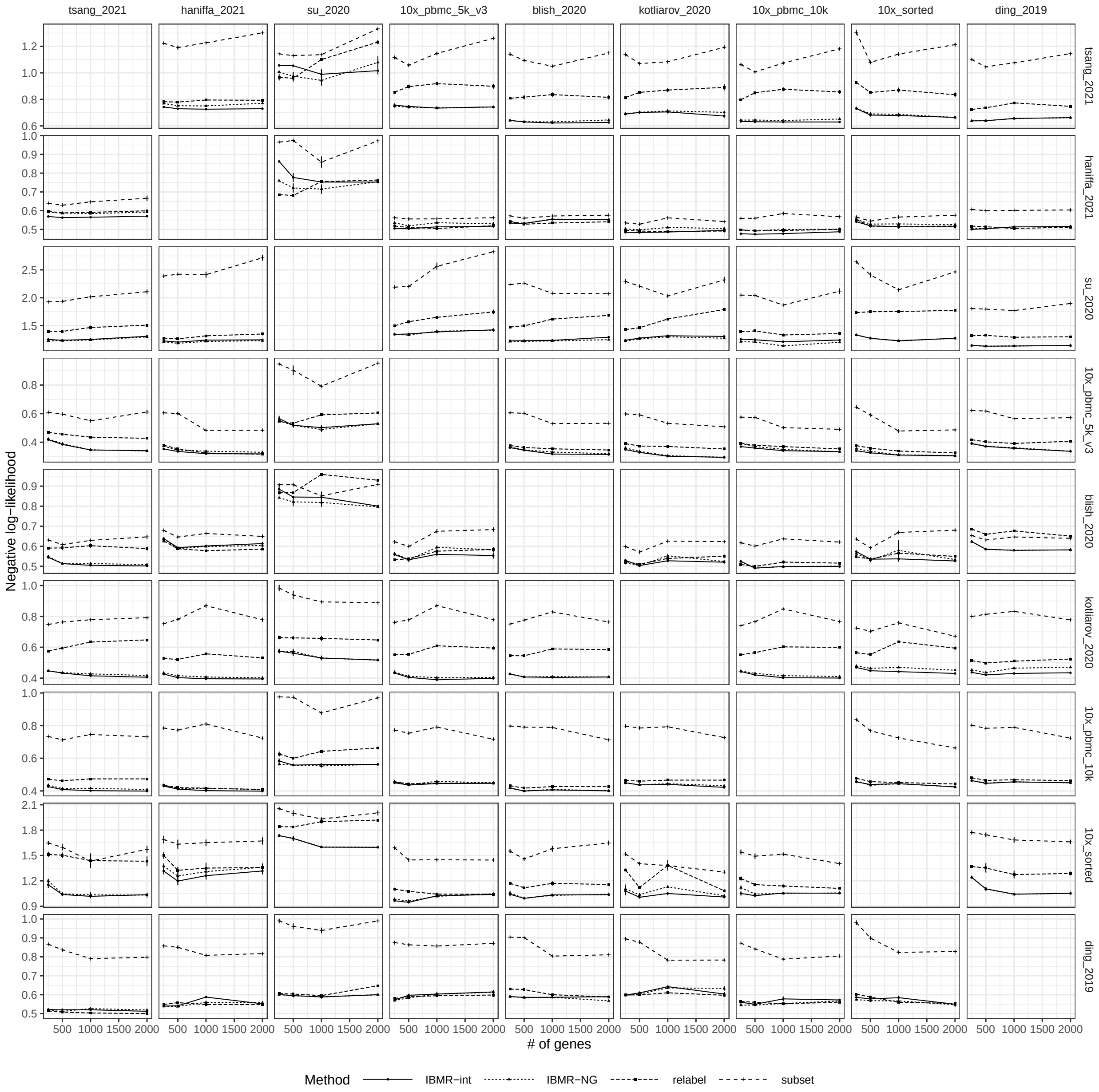

Figure 12: Negative log-likelihood for each method considered, for varying numbers of genes used for fitting the model with the number of cells per dataset $n_k = 10000$ fixed, for each training/validation/test dataset combination (subplots). The column denotes the validation dataset, the row denotes the test dataset, and the remaining 8 datasets were used for training. Points denote the average and error bars denote the standard error of the negative log-likelihood across 5 replicates of different subsampled training datasets.

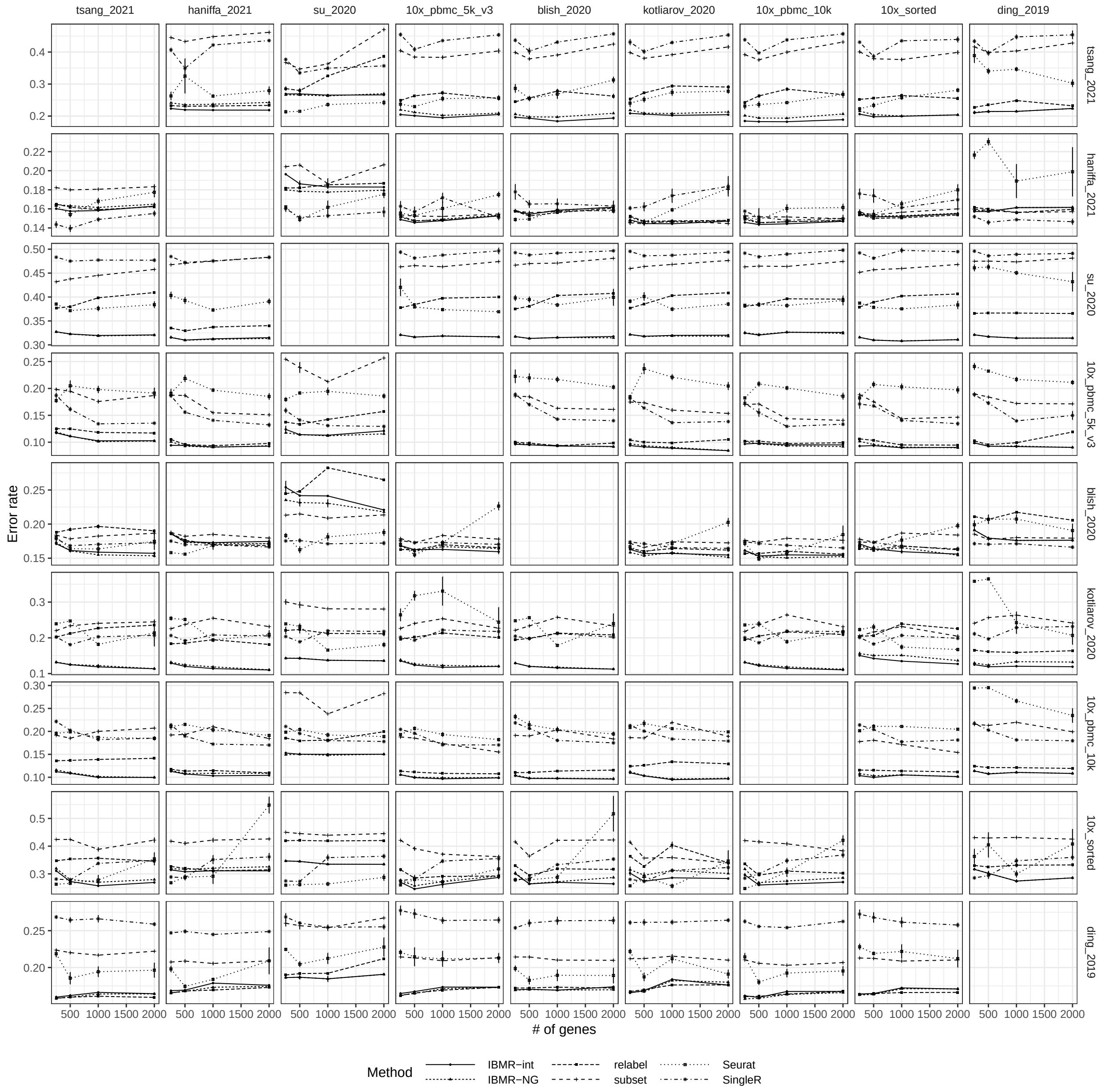

Figure 13: Error rate for each method considered, for varying numbers of genes used for fitting the model with the number of cells per dataset $n_k = 10000$ fixed, for each training/validation/test dataset combination (subplots). The column denotes the validation dataset, the row denotes the test dataset, and the remaining 8 datasets were used for training. Points denote the average and error bars denote the standard error of the error rate across 5 replicates of different subsampled training datasets.

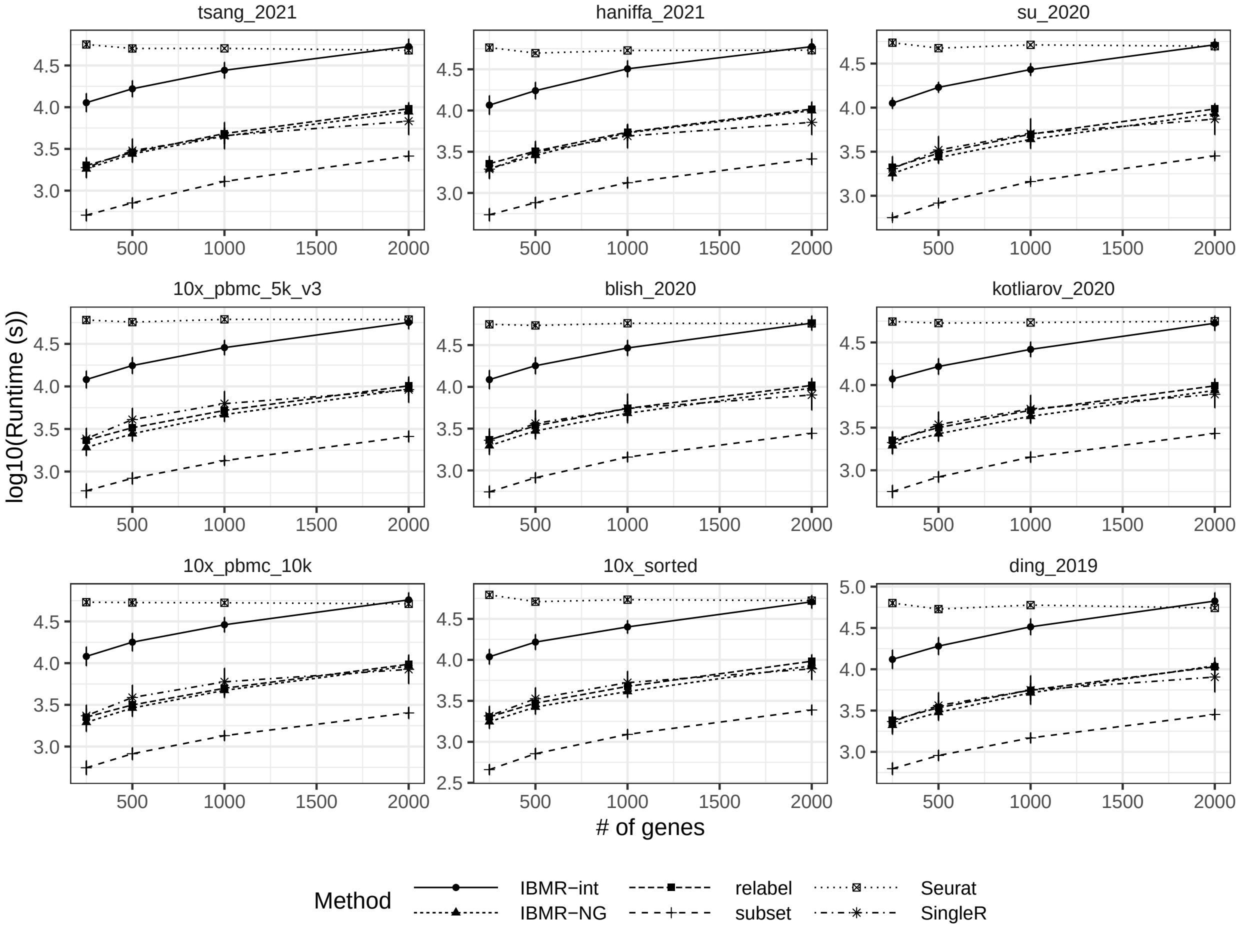

Figure 14: Timing results in real data application for each method considered, for each test dataset (subplots), for varying numbers of genes used for fitting the model with the number of cells per dataset $n_k = 10000$ fixed. Points denote the average and error bars denote the standard error of the average runtime for a test dataset across training/validation dataset combinations, for which the average for each training/validation dataset combination was over five replicates of different subsampled training datasets.

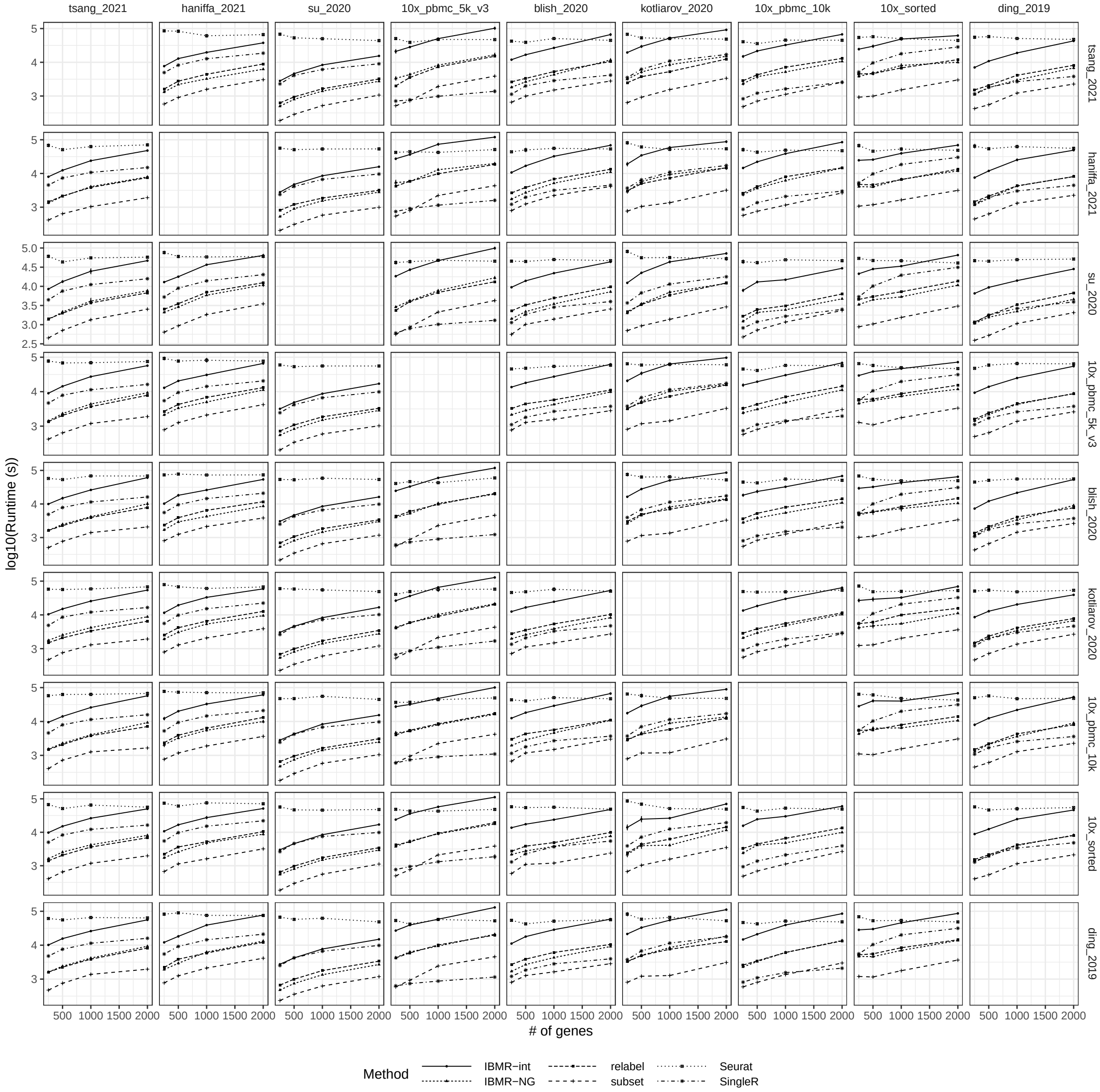

Figure 15: Timing results for each method considered, for varying numbers of genes used for fitting the model with the number of cells per dataset $n_k = 10000$ fixed, for each training/validation/test dataset combination (subplots). The column denotes the validation dataset, the row denotes the test dataset, and the remaining 8 datasets were used for training. Points denote the average and error bars denote the standard error of the runtime across 5 replicates of different subsampled training datasets.

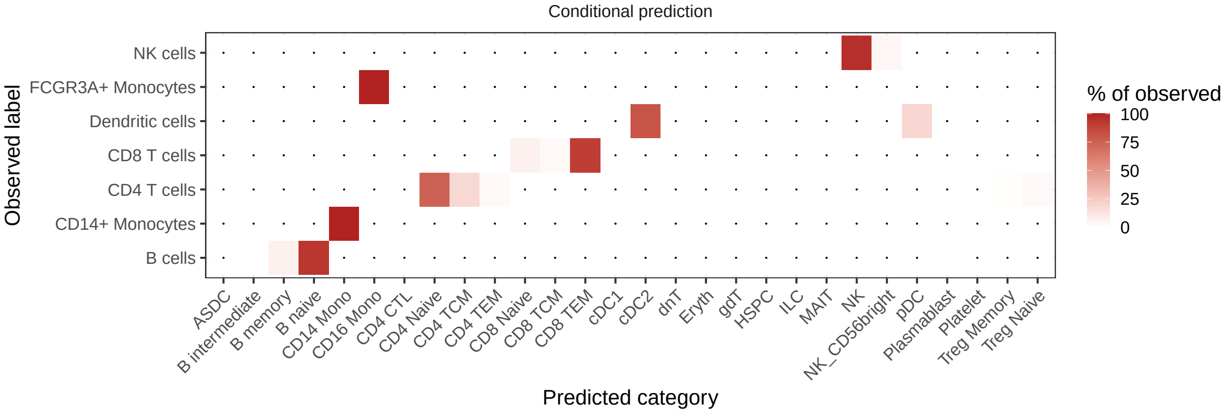

Figure 16: Heatmap showing the percentage of cells in (left) coarse and (right) conditional predicted categories for each observed label in the pre- and post-treatment dataset from Kang et al. (2018). Dots indicate that exactly zero cells are in that combination of observed label and prediction.

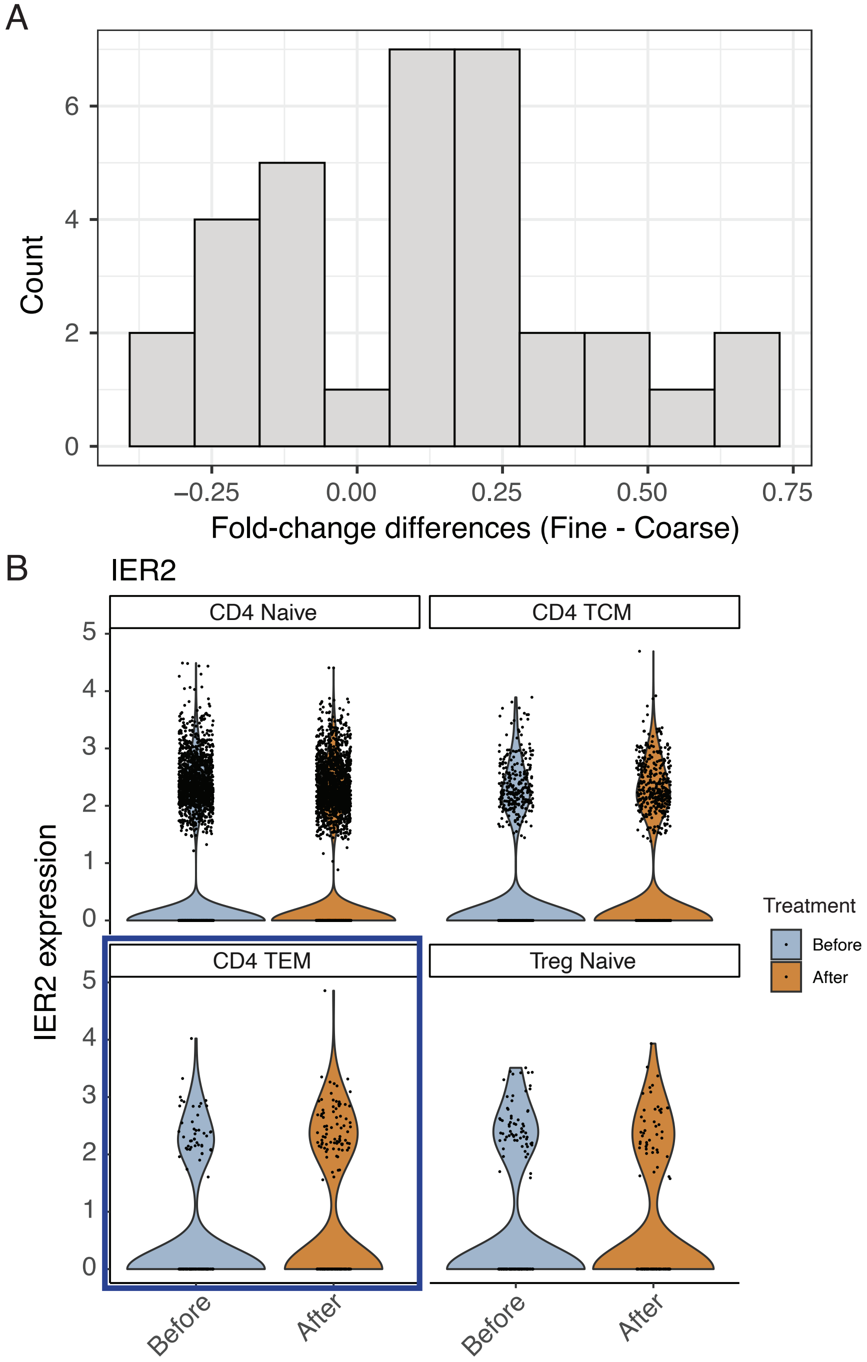

Figure 17: (A) Histogram of log fold change differences between the fine-level DE analysis and the coarse-level analysis for all 31 genes which were only identified in the fine-level analysis. (B) Gene expression of IER2 in the four fine categories. IER2 is significantly DE only in CD4 TEM (highlighted panel), evident in the increase non-zero expression after treatment.

References

Kang, H. M., Subramaniam, M., Targ, S., Nguyen, M., Maliskova, L., McCarthy, E., Wan, E., Wong, S., Byrnes, L., Lanata, C. M., et al. (2018). Multiplexed droplet single-cell rna-sequencing using natural genetic variation. *Nature biotechnology*, 36(1):89–94.

Powers, S., Hastie, T., and Tibshirani, R. (2018). Nuclear penalized multinomial regression with an application to predicting at bat outcomes in baseball. *Statistical Modelling*, 18(5-6):388–410.

Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck III, W. M., Hao, Y., Stoeckius, M., Smibert, P., and Satija, R. (2019). Comprehensive integration of single-cell data. *Cell*, 177(7):1888–1902.