Journal of Machine Learning Research 27 (2026) 1-64 Submitted 1/26; Revised 10/26; Published 10/26

A New Family of Poisson Non-negative Matrix Factorization Methods Using the Shifted Log Link

Eric Weine ERICW456@MIT.EDU
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA, USA

Peter Carbonetto PCARBO@UCHICAGO.EDU
Department of Human Genetics
University of Chicago
Chicago, IL, USA

Rafael A. Irizarry RAFAEL_IRIZARRY@DFCI.HARVARD.EDU
Department of Data Science
Dana-Farber Cancer Institute
Boston, MA, USA

Matthew Stephens MSTEPHENS@UCHICAGO.EDU
Department of Statistics and Department of Human Genetics
University of Chicago
Chicago, IL, USA

Editor: Samuel Kaski

Abstract

Poisson non-negative matrix factorization (NMF) is a widely used method to find interpretable "parts-based" decompositions of count data. While many variants of Poisson NMF exist, they all assume that the "parts" in the decomposition combine additively. This assumption may be natural in some settings, but not in others. Here, we introduce Poisson NMF with the shifted-log link function to relax this assumption. The shifted-log link function has a single tuning parameter, and as this parameter varies the model changes from assuming that parts combine additively to assuming that parts combine more multiplicatively. We provide an algorithm to fit this model by maximum likelihood, and also an approximation that substantially reduces computation time for large, sparse datasets. We illustrate these new methods on a variety of real and simulated data. Our examples show that while standard Poisson NMF often leads to sparse, cluster-like representations of data, the shifted-log link can produce more modular, layered results.

Keywords: non-negative matrix factorization, topic modeling, single-cell RNA sequencing, count data, approximate inference.

1. Introduction

Non-negative Matrix Factorization (NMF) (Lee and Seung, 1999) is a widely used method for dimensionality reduction of non-negative data matrices. Given a non-negative data matrix $\mathbf{Y}$, NMF methods attempt to find low-rank, non-negative matrices $\mathbf{L}$ and $\mathbf{F}$ such

©2026 Eric Weine and Peter Carbonetto and Rafael A. Irizarry and Matthew Stephens.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v27/26-0085.html.

that

$$\mathbf{Y} \approx \mathbf{L}\mathbf{F}^\top = \sum_{k=1}^{K} \mathbf{l}_{\cdot k}\mathbf{f}_{\cdot k}^\top. \tag{1}$$

Here $\mathbf{l}_{\cdot k}$ (respectively $\mathbf{f}_{\cdot k}$) denotes the k^{th} column of $\mathbf{L}$ (respectively $\mathbf{F}$). Thus, NMF decomposes the data into a sum of K components; if each of these K components, *individually*, has a physical or scientific interpretation then the decomposition is said to provide a "parts-based" representation of the data (Lee and Seung, 1999). The emphasis on *individually* is crucial here, because it distinguishes the idea of a parts-based representation from other low-dimensional representations, or embeddings. While other matrix factorization methods, like principal components analysis (PCA), also provide a decomposition of the form (1), Lee and Seung (1999) argue that the decompositions provided by NMF tend to produce more individually-interpretable components. That is, NMF more often provides a parts-based representation. This feature has led to widespread adoption of NMF in practical applications (Pritchard et al., 2000; Blei et al., 2003; Luce et al., 2016; Dey et al., 2017; Mackevicius et al., 2019; Alexandrov et al., 2020).

Count data, which consist of integer counts (y_{ij}), represent a common type of non-negative data frequently analyzed by NMF. Examples include word counts in documents, transcript counts in RNA-seq, and mutation counts in tumors. Most NMF methods for count data assume a Poisson model, where the elements y_{ij} are independent and Poisson distributed, and where the expected values of the count variables are modeled directly as $\mathbb{E}[y_{ij}] = (\mathbf{L}\mathbf{F}^\top)_{ij}$ (e.g. Lee and Seung, 1999; Gopalan et al., 2015; Zito and Miller, 2024; Landy et al., 2025). This corresponds to assuming that the parts in the decomposition (1) contribute additively to the expectation. We refer to such approaches as *standard Poisson NMF*.

While the additive assumption of standard Poisson NMF may be natural in some settings, it may be less appropriate in others where components may combine multiplicatively (e.g., gene expression, see Sanford et al. (2020); Zhou et al. (2024)). Here we introduce a more flexible approach to Poisson NMF that incorporates a shifted-log link function to relate the expected counts to the elements of $(\mathbf{L}\mathbf{F}^\top)_{ij}$. Depending on the value of a single hyper-parameter, the shifted log link can capture a range of behaviors from additive to more multiplicative.

Our work makes three main contributions. First, we provide an algorithm to fit Poisson NMF with the shifted-log link function by maximum likelihood. Second, because the MLE for this model is computationally impractical for very large datasets, we develop an efficient approximation to the log-likelihood whose computational complexity scales only with the number of non-zero entries in the data, allowing efficient (approximate) maximum likelihood estimation for sparse datasets common in text and biological applications. Third, we demonstrate our methods on real and simulated data, showing that the choice of link function can substantially impact results, and that the shifted-log link can produce more interpretable parts-based representations than standard Poisson NMF in some settings.

Our new approach to Poisson NMF is a special case of a *generalized bi-linear model* (GBM) (Choulakian, 1996) (see also Collins et al. (2001)) which, analogous to a generalized linear model (McCullagh and Nelder, 1989), uses a link function to relate the expected value of a distribution in the exponential family to a bi-linear term (i.e., $\mathbf{L}\mathbf{F}^\top$). Previous

work has provided methods to fit Poisson GBMs with a variety of link functions, including the canonical log link (Townes et al. (2019); Weine et al. (2024)) and the $\log(1 + \exp(y))$ link (Seeger and Bouchard, 2012). However these link functions are not bijections on the non-negative real line, and so they do not naturally lead to NMF methods; rather, they are more like versions of PCA for Poisson data. As far as we are aware, our paper is the first to provide a version of Poisson NMF with non-identity link function, and, furthermore, one that is practical for large, sparse count data.

2. Poisson NMF with the shifted log link function, and connections to existing models

Given a data matrix of counts $\mathbf{Y} \in \mathbb{N}_0^{n \times p}$ we assume the following Poisson NMF model:

$$y_{ij} \overset{\text{indep.}}{\sim} \text{Poisson}(\lambda_{ij}) \tag{2}$$

$$g(\lambda_{ij}; c) = b_{ij} \tag{3}$$

$$\mathbf{B} = \mathbf{L}\mathbf{F}^\top, \tag{4}$$

where g denotes the following shifted-logarithm link function:

$$g(\lambda; c) = \alpha_c \log\left(1 + \lambda/c\right). \tag{5}$$

Here $\mathbf{L} \in \mathbb{R}_{\geq 0}^{n \times K}$, $\mathbf{F} \in \mathbb{R}_{\geq 0}^{p \times K}$, $c \in \mathbb{R}_{>0}$, and $\alpha_c := \max(1, c)$ is a scaling constant that we introduce for convenience to make the scale of $\mathbf{B}$ more comparable across values of c.

Importantly, the shifted-log link, g, is a bijection on the non-negative real line, making it suited to NMF (unlike, say, the log link used in Townes et al. (2019) and Weine et al. (2024)). Its inverse is $g^{-1}(b; c) = c \cdot (\exp(b/\alpha_c) - 1)$. For brevity we refer to this link function as the "log1p" ("log 1 plus") link, noting that it is really a family of link functions indexed by the choice of c. We similarly refer to the model (2)-(4) as the "log1p NMF model", or simply "log1p NMF", and denote its log-likelihood as $\ell_{\text{log1p}}(\mathbf{L}, \mathbf{F}, c; \mathbf{Y})$.

The behavior of the link function g depends on the value of c (Figure 1). If c is large, then $g(\lambda; c) \approx \lambda$, in which case log1p NMF becomes standard Poisson NMF where parts combine additively. If c is small (i.e., near 0), for a fixed λ the quantity λ/c gets large and thus $\log\left(1 + \frac{\lambda}{c}\right) \approx \log\left(\frac{\lambda}{c}\right) = \log(\lambda) - \log(c)$. Thus for small c the link function acts more like the log link and the parts combine more multiplicatively. That is, log1p NMF provides a bridge between standard Poisson NMF, in which parts combine additively, and a new set of NMF models in which parts combine more multiplicatively. The next subsections provide more formal statements of these ideas.

2.1 Connections with existing models and methods

2.1.1 Standard Poisson NMF

The standard Poisson NMF model (Lee and Seung, 1999) is given by (2)-(4) but replacing $g(\lambda_{ij}; c)$ with λ_{ij}. That is, standard Poisson NMF is equivalent to using the identity link function instead of a log1p link function. To reflect this we use $\ell_{\text{id}}(\mathbf{L}, \mathbf{F}; \mathbf{Y})$ to denote the log-likelihood for standard Poisson NMF. As $c \to \infty$, the log1p link converges to the identity

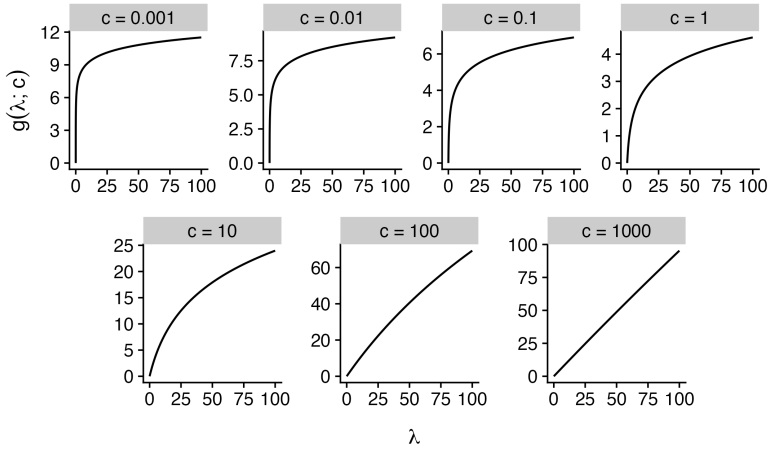


Figure 1: Plots of the link function $g(\lambda; c) = \alpha_c \times \log(1 + \lambda/c)$ for various values of c.

link ($g(\lambda; c) \to \lambda$) and so the log1p NMF model converges to the standard Poisson NMF model, as formalized in the following Theorem.

Theorem 1 *For any fixed* $\mathbf{Y} \in \mathbb{N}_0^{n \times p}$*, for all* $\mathbf{L} \in \mathbb{R}_{\geq 0}^{n \times K}$ *and* $\mathbf{F} \in \mathbb{R}_{\geq 0}^{p \times K}$*,*

$$\lim_{c \to \infty} \ell_{log1p}(\mathbf{L}, \mathbf{F}, c; \mathbf{Y}) = \ell_{id}(\mathbf{L}, \mathbf{F}; \mathbf{Y}).$$

Proof *See Appendix A.1.* ■

Note that the standard Poisson NMF model is also essentially equivalent to the multinomial factor model often referred to as the "topic model" (Carbonetto et al., 2021). Thus, the log1p NMF model with large c is also essentially equivalent to the standard topic model.

2.1.2 Poisson GLM-PCA

Another factor model for count data is the *Poisson GLM-PCA* model (Townes et al., 2019; Nicol and Miller, 2025; Weine et al., 2024). This model uses the log link instead of the log1p link, and dispenses with the non-negative assumption:

$$\begin{aligned} y_{ij} &\overset{\text{indep.}}{\sim} \text{Poisson}(\lambda_{ij}) \\ \log(\lambda_{ij}) &= b_{ij} \\ \mathbf{B} &= \mathbf{L}\mathbf{F}^\top, \end{aligned} \tag{6}$$

where $\mathbf{L} \in \mathbb{R}^{n \times K}$ and $\mathbf{F} \in \mathbb{R}^{p \times K}$. (Typically, additional orthogonality constraints are placed on $\mathbf{L}$ and $\mathbf{F}$ for identifiability.) Note that under this model we have

$$\lambda_{ij} = \prod_{k=1}^{K} \exp(l_{ik} f_{jk}),$$

so Poisson GLM-PCA assumes that the latent factors combine *multiplicatively* to influence λ_{ij}. The k^{th} factor can then be interpreted as capturing a log-fold change in the Poisson mean due to inclusion of that factor. Indeed, if λ_{ij} and $\lambda_{ij}^{(-k)}$ denote, respectively, the values of λ including and excluding the k^{th} factor, then in GLM-PCA

$$\log \frac{\lambda_{ij}}{\lambda_{ij}^{(-k)}} = l_{ik} f_{jk}. \tag{7}$$

A fundamental difference between GLM-PCA and log1p NMF is that GLM-PCA does not impose non-negative constraints on $\mathbf{L}$ and $\mathbf{F}$. Thus, as its name suggests, GLM-PCA is much more like PCA than like NMF. Indeed, non-negative constraints would usually not make sense in (6), as they would imply $\lambda_{ij} \geq 1$. However, the two models do have a connection: for values of $c \ll \lambda_{ij}^{(-k)}$, the interpretation of the k^{th} factor is approximately (7) (up to a constant that is equal to 1 for $c \leq 1$). To see this, we first establish the following theorem:

Theorem 2 *Let $\lambda_{ij} > 0$ and $\lambda_{ij}^{(-k)} > 0$ denote, respectively, the values of λ including and excluding the k^{th} factor in the log1p NMF model. Then,*

$$\log \frac{\lambda_{ij}}{\lambda_{ij}^{(-k)}} = (1/\alpha_c) l_{ik} f_{jk} + \varepsilon_{ij}, \tag{8}$$

where

$$\varepsilon_{ij} = \log\left(1 + c/\lambda_{ij}^{(-k)}\right) - \log(1 + c/\lambda_{ij}). \tag{9}$$

Proof *See Appendix A.2.* ■

Thus, in cases where $c \ll \lambda_{ij}^{(-k)}$ (and therefore $c \ll \lambda_{ij}$ due to non-negativity constraints), both logarithms in (9) are close to zero, and so $\varepsilon_{ij} \approx 0$; indeed, $0 \leq \varepsilon_{ij} \leq c/\lambda_{ij}^{(-k)}$. In this regime, $\log(\lambda_{ij}/\lambda_{ij}^{(-k)}) \approx (1/\alpha_c) l_{ik} f_{jk}$, so the k^{th} factor can be interpreted, just as in GLM-PCA (7), as capturing a log-fold change in the Poisson mean, scaled by the constant $1/\alpha_c$ (which is equal to 1 when $c \leq 1$). This is what we mean when we say that factors combine "more multiplicatively" in log1p NMF with small c; throughout, we refer to this behavior, and to models exhibiting it, as "quasi-multiplicative".

2.1.3 Frobenius-norm NMF applied to shifted log counts

Another approach to dealing with count data, which is particularly common in the analysis of single cell RNA-seq data, is to transform the counts and then apply methods designed for Gaussian data to the transformed counts (Ahlmann-Eltze and Huber, 2023). The shifted log *transformation* is commonly used in this context, especially when performing NMF (Willwerscheid, 2021; Johnson et al., 2023). Specifically, for some fixed $c > 0$, these methods find the solution to

$$\min_{\mathbf{L} \in \mathbb{R}_{\geq 0}^{n \times K}, \mathbf{F} \in \mathbb{R}_{\geq 0}^{p \times K}} ||\tilde{\mathbf{Y}} - \mathbf{L}\mathbf{F}^\top||_F \tag{10}$$

where

$$\tilde{y}_{ij} = \log(1 + y_{ij}/c). \tag{11}$$

This is equivalent to maximum likelihood estimation of a Gaussian NMF model on the transformed counts, assuming the same residual variance for each element of the matrix.

While there are both theoretical and practical concerns with applying Gaussian methods to count data, especially in the context of single-cell RNA sequencing (Nicol and Miller, 2025; Townes et al., 2019), in practice it can sometimes lead to reasonable results with good performance in downstream tasks (Ahlmann-Eltze and Huber, 2023). In Appendix G, we investigate empirically how closely results from fitting (10)-(11) match those from log1p link Poisson NMF. One theoretical disadvantage of the transformation approach is that the parameter c controls both (a) the relationship between the latent factors and the underlying mean and (b) the variance stabilization properties of the transformation (Ahlmann-Eltze and Huber, 2023). It is possible that, in the transformation approach, the value of c that best stabilizes variance does not correspond with the desired value of c for downstream interpretation of the factors, and thus there may be a trade-off between these two goals. Indeed, for very large settings of c in equation (11), $\tilde{\mathbf{Y}} \approx \frac{1}{c}\mathbf{Y}$, which when solving equation (10) is equivalent to assuming that each element of $\mathbf{Y}$ is Gaussian with approximately the *same* variance. In our log1p model c simply controls the relationship between the factors and the underlying mean structure; the variance stabilization issue is avoided by direct use of the Poisson likelihood for the count data.

3. Algorithms for fitting log1p NMF

3.1 Fitting model parameters with block coordinate ascent

We take a maximum likelihood approach to fitting the log1p NMF model described in equations (2)-(4). The log-likelihood of the log1p NMF model is

$$\ell_{\text{log1p}}(\mathbf{L}, \mathbf{F}, c; \mathbf{Y}) = \sum_{i=1}^{n}\sum_{j=1}^{p}\left(y_{ij}\log\left[\exp\left\{\frac{1}{\alpha_c}\sum_{k=1}^{K} l_{ik}f_{jk}\right\} - 1\right] - c\cdot\exp\left\{\frac{1}{\alpha_c}\sum_{k=1}^{K} l_{ik}f_{jk}\right\}\right), \tag{12}$$

where we have omitted terms that do not involve $\mathbf{L}$ or $\mathbf{F}$.

Maximizing $\ell_{\text{log1p}}(\mathbf{L}, \mathbf{F}, c; \mathbf{Y})$ with respect to $\mathbf{L}$ and $\mathbf{F}$ is a high-dimensional, non-convex optimization problem. However, as shown in Theorem 3 (see Appendix A.3), $\ell_{\text{log1p}}(\mathbf{L}, \mathbf{F}, c; \mathbf{Y})$ with c fixed is a bi-concave function of $\mathbf{L}$ and $\mathbf{F}$ (so $-\ell_{\text{log1p}}(\mathbf{L}, \mathbf{F}, c; \mathbf{Y})$ is bi-convex). This motivates an alternating optimization approach, which alternates repeatedly between optimizing for $\mathbf{L}$ with $\mathbf{F}$ fixed, and optimizing for $\mathbf{F}$ with $\mathbf{L}$ fixed. Conveniently, these sub-problems break down into a series of simpler tasks that can be performed in an embarrassingly parallel way, as we now describe.

To begin this description, consider the following Poisson regression model with log1p link:

$$\begin{aligned} y_i &\overset{\text{indep.}}{\sim} \text{Poisson}(\lambda_i) \\ g(\lambda_i; c) &= \mathbf{x}_i^\top \boldsymbol{\beta}, \end{aligned}$$

Algorithm 1 Alternating Optimization Method for Fitting Poisson log1p NMF Model. Row i and column j of $\mathbf{Y}$ are denoted, respectively, by $\mathbf{y}_{i\cdot}$ and $\mathbf{y}_{\cdot j}$.

Require: Count data $\boldsymbol{Y} \in \mathbb{N}_0^{n \times p}$, initial estimates $\mathbf{L} \in \mathbb{R}_{\geq 0}^{n \times K}$, $\mathbf{F} \in \mathbb{R}_{\geq 0}^{p \times K}$, constant $c > 0$, and a function Pois-reg-log1p$(\mathbf{X}, \mathbf{y}, c)$ that returns the constrained MLE of $\boldsymbol{\beta}$ in a non-negative Poisson regression with the log1p link.

while not converged **do**
  for $i = 1, \ldots, n$ **do** ▷ These can be performed in parallel.
    $\mathbf{l}_{i\cdot} \leftarrow$ Pois-reg-log1p$(\mathbf{F}, \mathbf{y}_{i\cdot}, c)$
    Store $\mathbf{l}_{i\cdot}$ in the ith row of $\mathbf{L}$.
  end for
  for $j = 1, \ldots, p$ **do** ▷ These can be performed in parallel.
    $\mathbf{f}_{j\cdot} \leftarrow$ Pois-reg-log1p$(\mathbf{L}, \mathbf{y}_{\cdot j}, c)$
    Store $\mathbf{f}_{j\cdot}$ in the jth row of $\mathbf{F}$.
  end for
end while
return $\mathbf{L}, \mathbf{F}$

where $\mathbf{y} \in \mathbb{N}_0^N$ is a vector of counts, $\mathbf{X} \in \mathbb{R}_{\geq 0}^{N \times q}$ is a fixed matrix of non-negative "covariates" with ith row $\mathbf{x}_i^\top$, and $\boldsymbol{\beta} \in \mathbb{R}_{\geq 0}^q$ is an unknown vector of non-negative regression coefficients. This regression model has log-likelihood ℓ_{log1pReg}, given by

$$\ell_{\text{log1pReg}}(\boldsymbol{\beta}, c; \boldsymbol{y}, \mathbf{X}) = \sum_{i=1}^{N} \left(y_i \log \left\{ \exp \left\{ \mathbf{x}_i^\top \boldsymbol{\beta} / \alpha_c \right\} - 1 \right\} - c \cdot \exp \left\{ \mathbf{x}_i^\top \boldsymbol{\beta} / \alpha_c \right\} \right), \tag{13}$$

which can also be shown to be concave in $\boldsymbol{\beta}$ by a similar argument to Theorem 3.

The Poisson NMF model (2) can be considered as a series of regressions in two different ways: each column of $\mathbf{Y}$ is a regression on the columns of $\mathbf{L}$ (with elements of $\mathbf{F}$ as the regression coefficients), or each row of $\mathbf{Y}$ is a regression on the columns of $\mathbf{F}$ (with elements of $\mathbf{L}$ as the regression coefficients). More algebraically, the log1p NMF log-likelihood (12) can be written as a sum of regression log-likelihoods in either of two ways:

$$\ell_{\text{log1p}}(\mathbf{L}, \mathbf{F}, c; \mathbf{Y}) = \sum_{j=1}^{p} \ell_{\text{log1pReg}}(\mathbf{f}_{j\cdot}, c; \mathbf{y}_{\cdot j}, \mathbf{L}) = \sum_{i=1}^{n} \ell_{\text{log1pReg}}(\mathbf{l}_{i\cdot}, c; \mathbf{y}_{i\cdot}, \mathbf{F}), \tag{14}$$

where $\mathbf{y}_{\cdot j}$ (respectively $\mathbf{y}_{i\cdot}$) denotes the column vector containing the j^{th} column (respectively i^{th} row) of the matrix $\mathbf{Y}$. Thus, with $\mathbf{L}$ fixed, optimizing over $\mathbf{F}$ involves solving p independent non-negative Poisson regression problems. Similarly, with $\mathbf{F}$ fixed, optimizing over $\mathbf{L}$ involves solving n independent non-negative Poisson regression problems. Because these steps involve solving independent regression problems, they can be done in parallel (see Algorithm 1). Fitting each regression problem requires a numerical solver, for which we use coordinate ascent as described in Appendix C. The resulting algorithm essentially extends the "Alternating Poisson Regression" approach (Carbonetto et al., 2021; Weine et al., 2024) used for other Poisson factor models to accommodate the shifted-log link function.

3.2 Computational complexity and an approximation for sparse data

The computational complexity of each outer-loop iteration of Algorithm 1 is $\mathcal{O}(npK)$, which scales linearly with the size, np, of the data matrix $\mathbf{Y}$. For standard Poisson NMF, with sparse data matrices $\mathbf{Y}$, computation can be reduced to $\mathcal{O}(\omega K + (n+p)K)$ where ω denotes the number of non-zero entries in $\mathbf{Y}$; see Carbonetto et al. (2021). Large sparse data matrices are ubiquitous in count data (e.g., single cell RNA-seq, document-term matrices), and np may be orders of magnitude larger than ω. In such cases, despite its parallel nature, Algorithm 1 may become computationally impractical, even though similar algorithms for standard Poisson NMF are practical. To address this, we now introduce an approximate maximum likelihood algorithm for log1p NMF that scales with ω instead of np, while remaining accurate.

To highlight the role of data sparsity in the log1p NMF log-likelihood, we re-write the log-likelihood (12) as

$$\ell_{\mathrm{log1p}}(\mathbf{L}, \mathbf{F}, c; \mathbf{Y}) = \sum_{(i,j) \notin \mathcal{I}_0} y_{ij} \log \left(\exp \left\{ \frac{1}{\alpha_c} \sum_{k=1}^{K} l_{ik} f_{jk} \right\} - 1 \right) - c \sum_{i=1}^{n} \sum_{j=1}^{p} \exp \left(\frac{1}{\alpha_c} \sum_{k=1}^{K} l_{ik} f_{jk} \right),$$

where $\mathcal{I}_0 = \{(i,j) : y_{ij} = 0\}$ is the index set of 0 counts in the matrix $\mathbf{Y}$. Computing the first term scales with $\omega = np - |\mathcal{I}_0|$, while computing the second term is the computational bottleneck: it sums np exponential terms, each of which requires K operations to compute.

The reason that standard Poisson NMF can be made efficient for sparse data matrices is that this problematic "sum of exponentials" term does not occur; instead there is a sum of linear terms, which can be computed efficiently in $\mathcal{O}((n+p)K)$. A simple way to make the log1p NMF computationally tractable would be to approximate the exponential terms with linear terms. Unfortunately, in general, this would yield a very bad approximation: $\exp(x)$ is accurately approximated by a linear function only for x very close to 0. Therefore we improve this naive idea in two important ways. First, following previous work in approximate GLM / GBM inference (Huggins et al., 2017; Zoltowski and Pillow, 2018; Keeley et al., 2020), we use a quadratic approximation $\exp(x) \approx \eta_0 + \eta_1 x + \eta_2 x^2$, where η_0, η_1, and η_2 are chosen either by Taylor approximation of $\exp(x)$ about some x_0, or by Chebyshev approximation over some interval $[x_L, x_U]$. Second, we approximate *only the terms in the sum corresponding to* $y_{ij} = 0$ (i.e. $(i,j) \in \mathcal{I}_0$). The intuition is that for such terms the corresponding estimates of λ_{ij} will typically be small (for $\mathbf{L}, \mathbf{F}$ consistent with the data), and so the quadratic approximation will be accurate in the parts of the space that matter.

These two ideas, when combined, give the following approximate log-likelihood (omitting an additive constant):

$$\begin{aligned} \ell_{\mathrm{log1p}}(\mathbf{L}, \mathbf{F}, c; \mathbf{Y}) \approx \sum_{(i,j) \notin \mathcal{I}_0} y_{ij} \log \left(\exp \left\{ \frac{1}{\alpha_c} \sum_{k=1}^{K} l_{ik} f_{jk} \right\} - 1 \right) - c \sum_{(i,j) \notin \mathcal{I}_0} \exp \left(\frac{1}{\alpha_c} \sum_{k=1}^{K} l_{ik} f_{jk} \right) \\ - \frac{\eta_1 c}{\alpha_c} \sum_{(i,j) \in \mathcal{I}_0} \sum_{k=1}^{K} l_{ik} f_{jk} - \frac{\eta_2 c}{\alpha_c^2} \sum_{(i,j) \in \mathcal{I}_0} \left(\sum_{k=1}^{K} l_{ik} f_{jk} \right)^2. \end{aligned} \tag{15}$$

Computing the first two terms of equation (15) requires $\mathcal{O}(\omega K)$ operations. While naively it appears that the subsequent linear and quadratic terms require $\mathcal{O}(|\mathcal{I}_0| K)$ operations, both

Model	**Log-likelihood Complexity**
Standard Poisson NMF	$\mathcal{O}((\omega + n + p)K)$ (Carbonetto et al., 2021)
Poisson GLM-PCA	$\mathcal{O}(npK)$ (Weine et al., 2024)
Log1p Poisson NMF Exact	$\mathcal{O}(npK)$
Log1p Poisson NMF Approximate	$\mathcal{O}\big((\omega + n + p)K + (n + p)K^2\big)$

Table 1: Computational complexity of log-likelihood for Poisson matrix factorization variants on sparse data matrices (size $n \times p$ with ω non-zero entries).

terms can actually be computed much more efficiently by simple algebraic rearrangements (see Appendix B for more details). The total computational complexity of this approximate log-likelihood becomes

$$\mathcal{O}\left((\omega + n + p)K + (n + p)K^2\right).$$

Table 1 summarizes and compares the computational complexity of computing the log-likelihood for different Poisson matrix factorization models on sparse data. As shown in Figure 2, for large, sparse data, especially with relatively small settings of K, our approximation can be orders of magnitude faster to compute than is the exact log-likelihood (as well as GLM-PCA, which has the same computational complexity as (12); Weine et al. (2024)). Moreover, for large, sparse data, our approximate log-likelihood is nearly as fast to compute as the log-likelihood of standard Poisson NMF.

Using the approximate log-likelihood of equation (15), we can use the same computational approach as Algorithm 1. That is, we repeatedly optimize equation (15) for $\mathbf{L}$ with $\mathbf{F}$ fixed, and for $\mathbf{F}$ with $\mathbf{L}$ fixed. Just as with the exact log-likelihood, these subproblems are concave and embarrassingly parallel.

3.3 Accuracy of the sparse computational approximation

To assess the accuracy of the approximate log-likelihood, we simulated data and examined likelihood ratios between the fitted models (using the exact and approximate optimization schemes). Specifically, we generated data with $n = p = 500$ and $K = 5$ from a log1p NMF model with values of $c = 10^{-3}$, $c = 1$, or $c = \infty$ (i.e., standard Poisson NMF), following the procedure of Simulation 2 (Section 5.1.2). We generated 30 replicate datasets for $c \in \{1, \infty\}$ and 100 replicate datasets for $c = 0.001$ due to large variation across replicates in that setting. Then, for each dataset, we fit log1p NMF to convergence using both the approximate and exact objective functions for a grid of settings of c, ranging between 10^{-4} and 10^4. We set η_0, η_1, and η_2 using either i) a second order Taylor approximation of $\exp(x)$ about $x = 0$ or ii) a Chebyshev approximation over the interval $[0, \log(1 + 1/c)]$ for the setting of c used to fit the model.

The results are shown in Figure 3. Overall, the Chebyshev approximation method performs very accurately, especially when the model is fit with $c \geq 0.01$. The Taylor approximation method performs poorly for small values of c, likely because the range of the optimal value for b_{ij} can still be quite large when $y_{ij} = 0$. More concretely, if for some

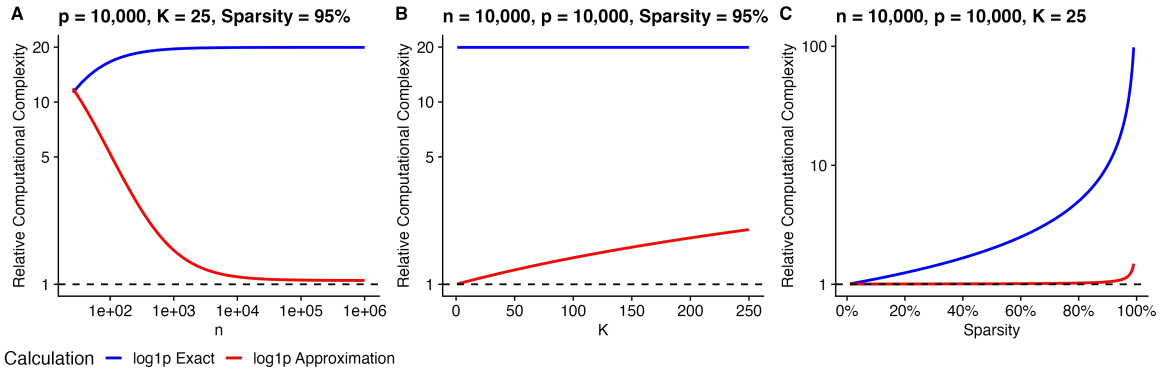


Figure 2: Scaling of computational complexity of equations (12) and (15) with data characteristics and choice of K. The y-axes are the ratio of the computational complexity of the specified calculation relative to the complexity of computing the log-likelihood of the standard Poisson NMF model, where 1 indicates the two calculations have the same complexity. (A) Scaling with respect to n with all other variables fixed (note that this is equivalent to scaling with p). (B) Scaling with K with all other variables fixed. (C) Scaling with sparsity $\left(\left[1 - \frac{\omega}{np}\right] \cdot 100\%\right)$ with all other variables fixed.

$y_{ij} = 0$ the constrained MLE $\hat{\lambda}_{ij} = \varepsilon$, then the corresponding MLE of b_{ij} is

$$\hat{b}_{ij} = \alpha_c \log\left(1 + \frac{\varepsilon}{c}\right).$$

Even when ε is small, if c is small relative to ε, then $\hat{b}_{ij}$ can be relatively large. This will make the Taylor approximation of $\exp(b_{ij})$ about $b_{ij} = 0$ inaccurate near $\hat{b}_{ij}$. In principle, the Chebyshev approximation approach will also become less accurate when c becomes very small, but at least in our simulations using the adaptive approximation interval of $[0, \log(1 + 1/c)]$ greatly improved performance.

In addition to being less accurate for fitting the log1p NMF model with small c, the Taylor approximation approach seems to result in an objective that is much more difficult to optimize. For the smallest values of c used to fit the model, optimizing the log-likelihood with a Taylor approximation requires an order of magnitude more iterations to converge than the Chebyshev approach, which appears to have similar convergence properties to the exact log-likelihood (Figure K.1). Interestingly, we notice that across all simulation scenarios and with both the exact and approximate methods, optimizing the log1p NMF model with large c requires far fewer iterations to converge than does optimizing with small c.

We present the results of an additional benchmarking study of the Chebyshev approximation approach on real data in Appendix F.

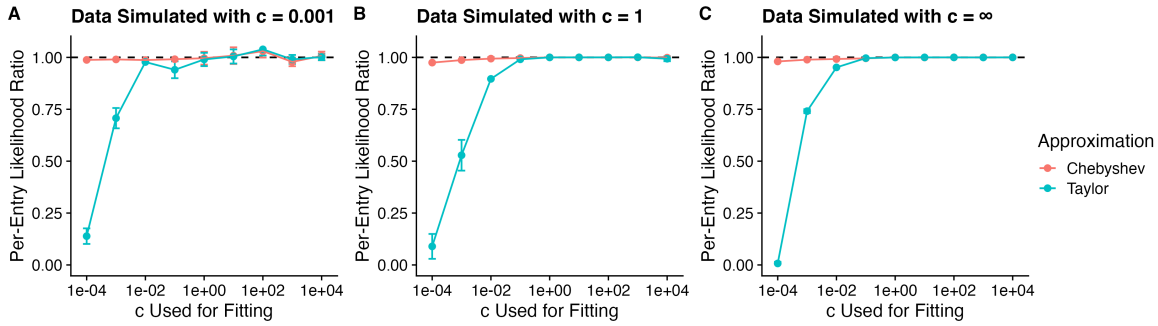


Figure 3: Per-entry likelihood ratio (the likelihood of the data when optimized with the approximate objective divided by the likelihood when optimized with the exact objective, raised to the power $1/np$) of factor models fit with $K = 5$ and varying settings of c. Points show means over simulation replicates (30 per generative regime; 100 for $c = 10^{-3}$) and error bars show 95% confidence intervals. All data are generated with $n = p = 500$ and $K = 5$. (A) Likelihood ratios when data are generated from the log1p model with $c = 10^{-3}$. (B) Likelihood ratios when data are generated from the log1p model with $c = 1$. (C) Likelihood ratios when data are generated from the log1p model with $c = \infty$.

3.4 Adding row-specific scaling constants

In practical applications, different rows of the data matrix may have somewhat different scales. For example, in single-cell RNA sequencing, the total number of mRNA molecules captured in each cell (sometimes referred to as "library size") is thought to mostly be a result of technical randomness in the measurement process (Sarkar and Stephens, 2021). Or, in text data, the total number of words in a document is typically less interesting than the *relative* term usage in a document. To capture this kind of effect we adapt the model (2)–(4) to incorporate a fixed "size factor" s_i for each row, replacing the assumption $y_{ij} \sim \text{Poisson}(\lambda_{ij})$ with $y_{ij} \sim \text{Poisson}(s_i\lambda_{ij})$ for fixed s_i values. The resulting model is equivalent (in terms of estimating $\mathbf{L}, \mathbf{F}$) to using a different value of c for each row; specifically, it is equivalent to setting the link hyper-parameter to $c \cdot s_i$ in row i while keeping the scaling constant $\alpha_c = \max(1, c)$ unchanged:

$$\begin{aligned} y_{ij} &\sim \text{Poisson}(\lambda_{ij}) \\ \alpha_c \log\left(1 + \frac{\lambda_{ij}}{c \cdot s_i}\right) &= b_{ij} \\ \mathbf{B} &= \mathbf{L}\mathbf{F}^\top. \end{aligned} \tag{16}$$

Since $c \to 0$ implies that $c \cdot s_i \to 0$ and $c \to \infty$ implies that $c \cdot s_i \to \infty$, all results regarding the role of c in the model (2)–(4) also hold for the modified model (16). In our applications we set the size factors on the order of 1 so that the role of c is not obfuscated by the scale

of s_i. In particular, we set

$$s_i = \frac{\sum_{j=1}^{p} y_{ij}}{\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{p} y_{ij}},$$

so that $s_1, \ldots, s_n$ have mean 1.

Note that using size factors $s_i = \sum_j y_{ij}$ is closely connected to fitting a multinomial model to the data. This is because, with these size factors, the Poisson log-likelihood $y_{ij} \sim \text{Poisson}(s_i \lambda_{ij})$ is equivalent to the multinomial log-likelihood

$$y_{i1}, \ldots, y_{ip} | s_i \sim \text{Multinomial}(s_i; \lambda_{i1}, \ldots, \lambda_{ip}),$$

provided $\sum_j \lambda_{ij} = 1$ (Baker, 1994). Furthermore, for standard Poisson NMF it turns out that the constraint $\sum_j \lambda_{ij} = 1$ is automatically satisfied by the maximum likelihood estimates under the Poisson model, and so the multinomial model can be fit by simply fitting the Poisson model, without imposing the constraint directly (Carbonetto et al., 2021). In the log1p model this constraint is no longer guaranteed to be satisfied, so fitting the modified log1p Poisson model is not exactly equivalent to fitting a multinomial model with log1p link (although it could be viewed as an approximation to this). We leave investigation of methods for exactly fitting the multinomial model with log1p link to future work.

3.5 Initialization

To initialize the log1p model with K factors, we first optimize a rank-1 model to the data and then we initialize other entries to small positive numbers. That is, we fit each model in two steps:

1. Fit the log1p NMF with $K = 1$ to $\mathbf{Y}$ using Algorithm 1 with random initializations of $\mathbf{L}$ and $\mathbf{F}$. This yields initial estimates $\hat{\boldsymbol{l}}$ and $\hat{\boldsymbol{f}}$.

2. Fit the log1p NMF with K factors to $\mathbf{Y}$ using Algorithm 1 with initializations of $\mathbf{L} = [\, \hat{\boldsymbol{l}}\ \boldsymbol{u}_1\ \ldots\ \boldsymbol{u}_{K-1}\,]$ and $\mathbf{F} = [\, \hat{\boldsymbol{f}}\ \boldsymbol{v}_1\ \ldots\ \boldsymbol{v}_{K-1}\,]$, where $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{K-1}$ and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{K-1}$ are column vectors with very small positive numbers in each entry.

This initialization procedure was used to encourage the fitted models to find a "baseline" factor (i.e., a factor that is loaded on all samples), which could improve interpretability for quasi-multiplicative models. Such a baseline is especially natural for small values of c (i.e., $c \ll 1$), as in this case $g(\lambda; c) \approx \log(\lambda) - \log(c)$, and so regardless of the data each sample intrinsically has a loading of approximately 1 on a factor that is uniformly $-\log(c)$. In fact, as described in Appendix J, as $c \to 0^+$, the log1p NMF model becomes degenerate unless the constant vector $\mathbf{1}$ is contained in the cone spanned by the columns of $\mathbf{F}$, which can be achieved by adding such an intercept.

While this initialization approach may improve interpretability for quasi-multiplicative models, empirically we find that the log-likelihoods from this initialization strategy are no better, and perhaps somewhat worse, than those from random initializations (see Figure K.2), so further study of initialization strategies may be warranted.

4. Visualization and evaluation of log1p NMF fits

Before evaluating the log1p NMF model on real data, we discuss a few matters related to the evaluation and visualization of these models.

4.1 Re-scaling and re-ordering inferred loadings and factors

As with most matrix factorization models, there is some inherent non-identifiability in the scale of the estimates of $\mathbf{L}$ and $\mathbf{F}$. In particular, one can multiply each element of $\mathbf{l}_{\cdot k}$ by any constant a_k, and divide each element of $\mathbf{f}_{\cdot k}$ by the same constant, and the resulting likelihood will be unchanged (because $\mathbf{LF}^\top$ is unchanged). Thus, before plotting or comparing results from log1p models with different settings of c (or different runs of the algorithm with the same c) it is important to scale the factors and loadings in some standardized way.

Here, after obtaining estimates of $\mathbf{L}$ and $\mathbf{F}$, we scale each column of these matrices such that $\max_i l_{ik} = 1$ (i.e., we rescale using $a_k = 1/\max_i l_{ik}$). If one thinks of l_{ik} as representing the "membership" of sample i in factor k then this corresponds to assuming that the maximum membership in each factor is 1. This also means that the interpretation of f_{jk} from (8) is relative to "maximal membership" in factor k. Thus, for example, as $c \to \infty$, f_{jk} represents the additive change in gene expression associated with full membership in factor k, and as $c \to 0$, f_{jk} represents the log-fold change in gene expression associated with full membership in factor k. We note that the scaling, while not affecting the fit, does affect visualization of the results. With our scaling, every column of $\mathbf{L}$ will show up somewhat equally in plots of the loadings (since each column has a maximum value of 1), no matter how strong the impact of that factor on the data (i.e., how large the corresponding column of $\mathbf{F}$ is). This can be useful for highlighting subtler structure, since factors that have a small effect on the data will still be visible in the plot. However, in some settings it might be preferable to down-weight factors with small effects in the visualization (which could be achieved, for example, by using $a_k = \max_j f_{jk}$ or $a_k = \sum_j f_{jk}$).

We discuss additional choices we made in the Results regarding visualization of the log1p NMF models in Appendix D.

4.2 Evaluating Sparsity

One advantage of NMF methods, relative to other methods that do not have non-negative constraints (e.g., PCA), is that they often produce sparse estimates of $\mathbf{L}$ and $\mathbf{F}$. Below, we assess how different settings of c impact the sparsity of the estimated $\mathbf{L}$ and $\mathbf{F}$. To measure sparsity, we use Hoyer's metric (Hoyer, 2004), defined for a non-negative n-vector $\mathbf{x}$ (with at least 1 non-zero entry) as

$$H(\mathbf{x}) = \frac{\sqrt{n} - ||\mathbf{x}||_1/||\mathbf{x}||_2}{\sqrt{n} - 1}. \tag{17}$$

While H is arguably measuring skewness, rather than sparsity, it has the advantage that it is invariant to re-scaling of $\mathbf{x}$, which is particularly important when comparing log1p NMF fits with different values of c (see Hurley and Rickard (2009) for a broad discussion of sparsity metrics). For simplicity, when examining the sparsity of the matrices $\mathbf{L}$ and $\mathbf{F}$, we examine the mean sparsity of the columns of these matrices.

5. Results

5.1 Simulations

5.1.1 A simple toy example

We first show that even when the true data generating process is such that the log1p NMF model is correctly specified for *any* c, the value of c used to fit the model affects downstream interpretation.

To illustrate this, we generated a 1000×400 data matrix $\mathbf{Y}$ from a rank-3 Poisson NMF model. We first defined a "baseline" factor $\mathbf{f}_0$ and two sparse change factors $\mathbf{f}_A$ and $\mathbf{f}_B$. Ordering the $p = 400$ features in four blocks of 100, we defined these factors as

$$\begin{aligned}
\mathbf{f}_0 &= \big(\underbrace{2,\ldots,2}_{100},\ \underbrace{1000,\ldots,1000}_{100},\ \underbrace{2,\ldots,2}_{100},\ \underbrace{1000,\ldots,1000}_{100}\big),\\
\mathbf{f}_A &= \big(\underbrace{18,\ldots,18}_{100},\ \underbrace{100,\ldots,100}_{100},\ \underbrace{0,\ldots,0}_{100},\ \underbrace{0,\ldots,0}_{100}\big),\\
\mathbf{f}_B &= \big(\underbrace{0,\ldots,0}_{100},\ \underbrace{0,\ldots,0}_{100},\ \underbrace{18,\ldots,18}_{100},\ \underbrace{100,\ldots,100}_{100}\big).
\end{aligned}$$

We designed the change factors $\mathbf{f}_A$ and $\mathbf{f}_B$ so that each one acts on two kinds of features, distinguished by their baseline rate. The two kinds respond very differently when their corresponding change factor has loading of 1:

Feature Type	Baseline Rate ($\mathbf{f}_0$)	Rate if Loading = 1 (on $\mathbf{f}_A, \mathbf{f}_B$)	Fold Δ	Absolute Δ
High-Baseline	1000	1100	1.1×	+100
Low-Baseline	2	20	10×	+18

That is, high-baseline features undergo a large *absolute* change (i.e., $\mathbf{f}_X - \mathbf{f}_0$) but a small *relative* (fold) change (i.e., $\mathbf{f}_X/\mathbf{f}_0$), whereas low-baseline features undergo a large *relative* change but a small *absolute* change.

We then defined four groups of samples based on their loadings on these factors:

$$\mathbf{l}_{i\cdot} = \begin{cases} (1,0,0) & \text{if } 1 \le i \le 250 \\ (1,1,0) & \text{if } 250 < i \le 500 \\ (1,0,1) & \text{if } 500 < i \le 750 \\ (1,1,1) & \text{if } 750 < i \le 1000, \end{cases}$$

where each loading is ordered (l_{i0}, l_{iA}, l_{iB}). For each sample i we defined

$$\boldsymbol{\lambda}_i = l_{i0}\mathbf{f}_0 + l_{iA}\mathbf{f}_A + l_{iB}\mathbf{f}_B,$$

and drew $y_{ij} \sim \text{Poisson}(\lambda_{ij})$ independently across i and j. The group definitions are shown in Figure 4A.

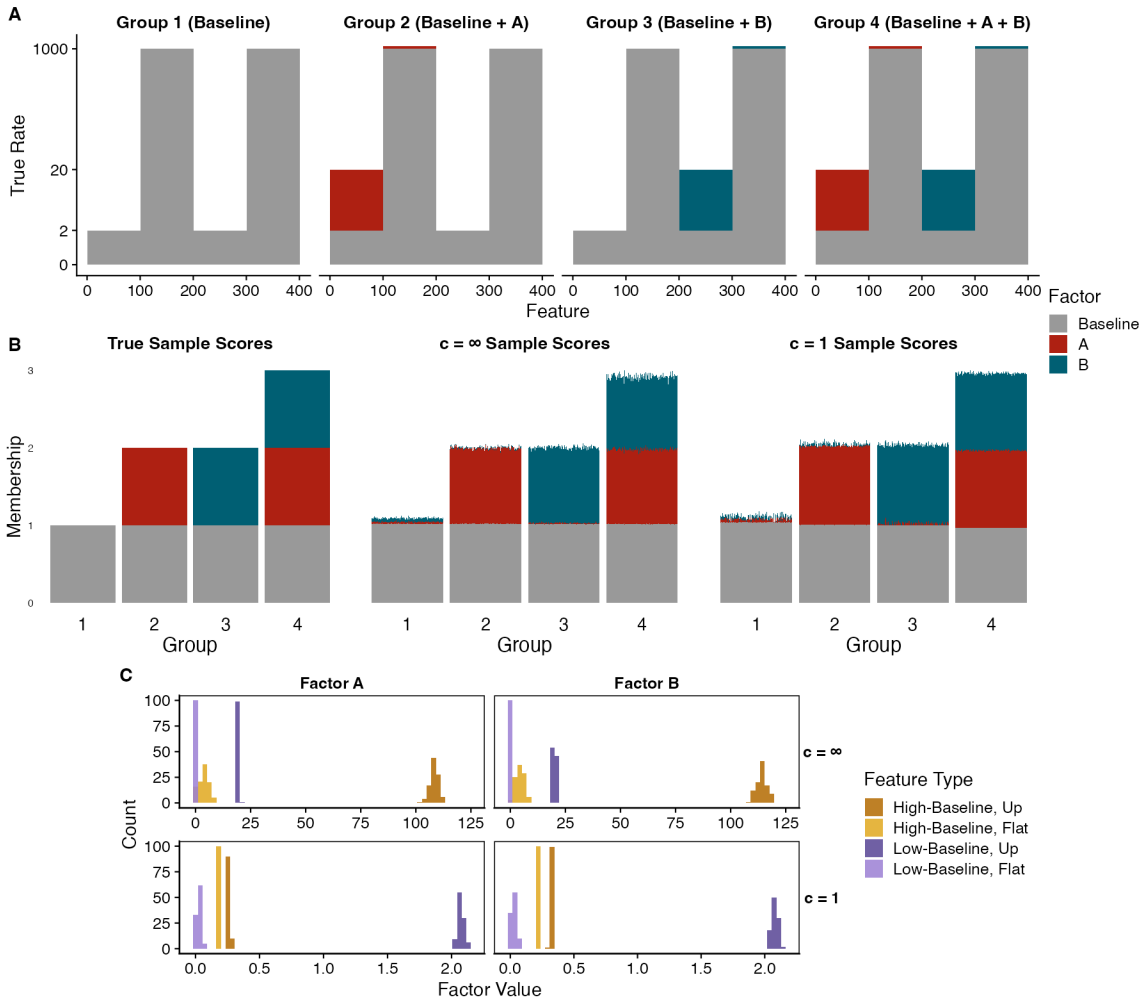


Figure 4: Toy example results. (A) True feature rates in each of the four sample groups. (B) True sample scores, and estimated sample scores (**L**) for log1p NMF with $c = \infty$ and $c = 1$. (C) Distributions of estimated factor values (columns of **F**) for the two change factors, with features classified by their baseline level and by whether the change factor increases them.

While we have written this model using an additive link, it can equivalently be written as a log1p NMF model with any value of c. Specifically, given c, suppose we keep the loadings $\mathbf{l}_{i\cdot}$ unchanged and replace the factors as follows:

$$\begin{aligned}
\mathbf{f}_0^{(c)} &= \alpha_c \log\bigl(1 + \tfrac{1}{c}\,\mathbf{f}_0\bigr)\,, \\
\mathbf{f}_A^{(c)} &= \alpha_c \left[\log\bigl(1 + \tfrac{1}{c}(\mathbf{f}_0 + \mathbf{f}_A)\bigr) - \log\bigl(1 + \tfrac{1}{c}\,\mathbf{f}_0\bigr)\right], \\
\mathbf{f}_B^{(c)} &= \alpha_c \left[\log\bigl(1 + \tfrac{1}{c}(\mathbf{f}_0 + \mathbf{f}_B)\bigr) - \log\bigl(1 + \tfrac{1}{c}\,\mathbf{f}_0\bigr)\right],
\end{aligned}$$

where $\log(\cdot)$ is applied elementwise. With these factors,

$$\alpha_c \log(1 + \lambda_{ij}/c) = l_{i0} f_{0j}^{(c)} + l_{iA} f_{Aj}^{(c)} + l_{iB} f_{Bj}^{(c)}$$

for every sample i and feature j. Such a construction exists for any c only because factors A and B have disjoint support, in the sense that no feature is non-zero in both change factors, so the nonlinear link never has to represent an interaction between them. (And also because the loadings l take only two values, 0 and 1.)

We show the results of fitting the log1p model with $c = 1$ and $c = \infty$ in Figure 4B,C. Unsurprisingly, since models with both values of c are correctly specified, the estimated loadings returned by the two fits are nearly indistinguishable, and match the true values closely (Figure 4B). However, the estimated factors differ greatly between the $c = 1$ and $c = \infty$ fits (Figure 4C). Indeed, because the link function is additive when $c = \infty$, the top features in each factor for the $c = \infty$ fit are those that have a high baseline (1000) and increase by 100 when the factor has a loading of 1. In absolute terms, these features increase by much more than the features with a low baseline (2) that increase by only 18. Conversely, because the link function with $c = 1$ behaves closely to a log, which represents changes on a relative scale, we see that the highest features in the $c = 1$ estimated factors are the low-baseline features. These features experience a fold change of 10 when the change factors are loaded at 1, whereas the high-baseline features experience a fold change of only 1.1.

This simple example highlights that even when changing the value of c does not change the goodness of fit of the model to the data, it can substantially change the interpretation of the results. In particular, when c is small, the log1p NMF model will tend to produce factors that have high scores on features with large fold changes in some samples relative to others, whereas with large c the estimated factor values tend to emphasize large absolute changes. The example also illustrates that, at least in principle, c can be non-identifiable, a point we return to below.

5.1.2 Impact of c on sparsity and log-likelihood of fit

In preliminary investigations of real datasets, fitting the log1p model using a range of values of c, we observed the following two general trends. First, fits using larger values of c tend to produce sparser estimates of both $\mathbf{L}$ and $\mathbf{F}$ (by Hoyer's measure of sparsity (17); Figure 11). Second, fits using smaller values of c often produced higher log-likelihoods (e.g., Figure K.2). In this section we illustrate these trends on simulated data and discuss potential explanations.

For these simulations we generated counts from the log1p NMF model (2)–(4) with $c_{\text{true}} \in \{10^{-3}, 1, \infty\}$, representing the quasi-multiplicative, intermediate, and additive regimes. For each of 30 random seeds we simulated an $n \times p$ count matrix with $n = p = 500$ and $K = 5$ factors, as follows:

1. The rows of the loadings matrix $\mathbf{L} \in \mathbb{R}_{\geq 0}^{n \times K}$ were drawn independently as

$$\mathbf{l}_{i\cdot} \sim \text{Dirichlet}(a, \ldots, a)$$

with $a = 0.3$.

2. The entries of the factor matrix $\mathbf{F} \in \mathbb{R}_{\geq 0}^{p \times K}$ were generated as a mixture of a point mass at 0 and a truncated log-t distribution. That is, for each feature j and factor k we drew

$$u_{jk} \sim \text{Uniform}(0, 1), \qquad \phi_{jk} \sim t_4$$

independently, and set

$$\begin{aligned} \widetilde{m}_{jk} &= \exp\left(\mu^{(c_{\text{true}})} + \sigma^{(c_{\text{true}})}\phi_{jk}\right), \\ m_{jk} &= \min\left(\max\left(\widetilde{m}_{jk},\, T_{\text{low}}^{(c_{\text{true}})}\right),\, T_{\text{high}}^{(c_{\text{true}})}\right), \\ f_{jk} &= m_{jk}\, \mathbf{1}\left\{u_{jk} > \pi_0^{(c_{\text{true}})}\right\}. \end{aligned}$$

In the case where this procedure resulted in a feature j with $u_{jk} \leq \pi_0^{(c_{\text{true}})}$ for every k, we calculated $k^* = \arg\max_k u_{jk}$, and set $f_{jk^*} = m_{jk^*}$, leaving the remaining entries at zero.

3. With $\mathbf{B} = \mathbf{L}\mathbf{F}^\top$, we drew $y_{ij} \sim \text{Poisson}(\lambda_{ij})$ with $\lambda_{ij} = g^{-1}(b_{ij}; c_{\text{true}})$. In the rare case that an entire row or column of $\mathbf{Y}$ was 0, the Poisson draw was repeated for that row or column.

The constants $\left(\pi_0^{(c_{\text{true}})}, \mu^{(c_{\text{true}})}, \sigma^{(c_{\text{true}})}, T_{\text{low}}^{(c_{\text{true}})}, T_{\text{high}}^{(c_{\text{true}})}\right)$ were calibrated at each c_{true} to match the characteristics of the pancreas dataset of Section 5.2.2, as described in Appendix I.

To each simulated dataset we fit the log1p NMF model with $K = 5$ and optimized the exact log-likelihood at every value of c on the grid $\{10^{-4}, 10^{-3}, \ldots, 10^4\}$, together with $c = \infty$. All fits used the rank-1 initialization of Section 3.5 and were fit until the change in the log-likelihood between successive iterations fell below 10^{-6}.

The mean Hoyer sparsity for the estimated $\mathbf{L}, \mathbf{F}$ are shown in Figure 5A,B. In both cases H is a nearly monotonically increasing function of c. That is, large c fits are sparser than small c fits *regardless of the value of* c_{true}.

The consistency of this trend (which also occurs in real data; see Figure 11) suggests that there may be a deeper theoretical explanation. Although directly analyzing the maximum likelihood estimate of $\mathbf{L}$ and $\mathbf{F}$ as a function of c seems challenging, we have developed a less direct argument that we believe helps to explain the trend.

To describe this argument, we first relate the sparsity of $\mathbf{L}$ and $\mathbf{F}$ to a simpler quantity, the peak-to-mean ratio of the matrix $\mathbf{B} = \mathbf{L}\mathbf{F}^\top$, defined as

$$\text{pm}(\mathbf{B}) = \frac{\max(\mathbf{B})}{\text{mean}(\mathbf{B})}. \tag{18}$$

In Theorem 7, we bound the sparsity of each column of $\mathbf{L}$ and $\mathbf{F}$ in terms of an increasing function of $\mathrm{pm}(\mathbf{B})$. Specifically, for large n and p, the Theorem implies (to a good approximation)

$$H(\mathbf{l}_{\cdot k}) + H(\mathbf{f}_{\cdot k}) \leq 2\left(1 - \left(\frac{w_k}{\mathrm{pm}(\mathbf{B})}\right)^{1/4}\right), \tag{19}$$

where $w_k = \frac{\mathrm{mean}(\mathbf{l}_{\cdot k})\,\mathrm{mean}(\mathbf{f}_{\cdot k})}{\mathrm{mean}(\mathbf{B})}$ and $\sum_{k=1}^{K} w_k = 1$.

The next key observation is that the pm value of the "true" $\mathbf{B}$, in the log1p model with parameter c, is monotonic in c. Specifically, given Poisson rates $\mathbf{\Lambda}$, define the corresponding true value of $\mathbf{B}_c$ entrywise as $b_{ij}(c) = g(\lambda_{ij}; c)$. Proposition 8 shows that for any $0 < c_1 < c_2 \leq \infty$,

$$\mathrm{pm}(\mathbf{B}_{c_1}) \leq \mathrm{pm}(\mathbf{B}_{c_2}). \tag{20}$$

While the fitted $\mathbf{B}$ will inevitably differ from the true $\mathbf{B}$, in our simulation the pm of the fitted $\mathbf{B}$ closely matches the pm of the true $\mathbf{B}$ (Figure 5C).

Putting these observations together, Equation (20) shows that increasing c increases the peak-to-mean ratio of the true $\mathbf{B}$, which will tend to increase the peak-to-mean ratio of the fitted $\mathbf{B}$, which by (19) may allow the sparsity of the estimated $\mathbf{L}$ and $\mathbf{F}$ to increase.

Next we turn to how the log-likelihood of the model depends on the assumed value of c. Interestingly, even when the value of c used to fit the model does not match the simulation truth, the log1p NMF model appears to provide reasonably accurate fitted values of $\mathbf{B}$, both in terms of the log-likelihood of the data (Figure 5D) and in terms of the reconstruction error of the true $\mathbf{B}$ (Figure K.3A). The largest effects of model mis-specification occur when c_{true} is small and the value of c used for fitting is large.

Perhaps the most surprising observation is that the log-likelihood is not always maximized at the true value of c, and indeed tends to increase slightly as c gets very small ($c = 10^{-4}$ achieves the highest mean and median log-likelihoods for $c_{\mathrm{true}} = 10^{-3}$ and $c_{\mathrm{true}} = \infty$). We do not fully understand this observation, but our best explanation is that, for fixed K, the non-linearity of the log1p NMF model with small c provides some additional flexibility compared with the more linear model (large c). In Appendix J we provide some preliminary theoretical support for this. Focusing on the simplest case where $K = 2$, we characterize, as a function of $\mathbf{F}$, the set of directions the rate vector $\boldsymbol{\lambda}_i$ can achieve in the limit $c \to 0^+$. We show that there are choices of $\mathbf{F}$ (in the log1p NMF model with $c \to 0^+$) for which this achievable set is larger than the achievable set of the standard (linear) Poisson NMF model under *any* choice of $\mathbf{F}$. This suggests that, for a fixed value of K, the log1p NMF model with small c has more "effective" parameters than the standard Poisson NMF model. We thus suspect that there may be issues with identifying c in practice, and instead leave selection of this value as an exploratory choice.

5.2 Real Data Applications

5.2.1 MCF-7 bulk RNA-seq

As a simple initial real-data example, we examine bulk RNA-sequencing data from the human breast carcinoma cell line MCF-7 under four conditions (Sanford et al., 2020). This dataset is particularly well suited for evaluating log1p NMF across values of c, as it was explicitly generated to probe whether the effects of multiple treatments on gene expression

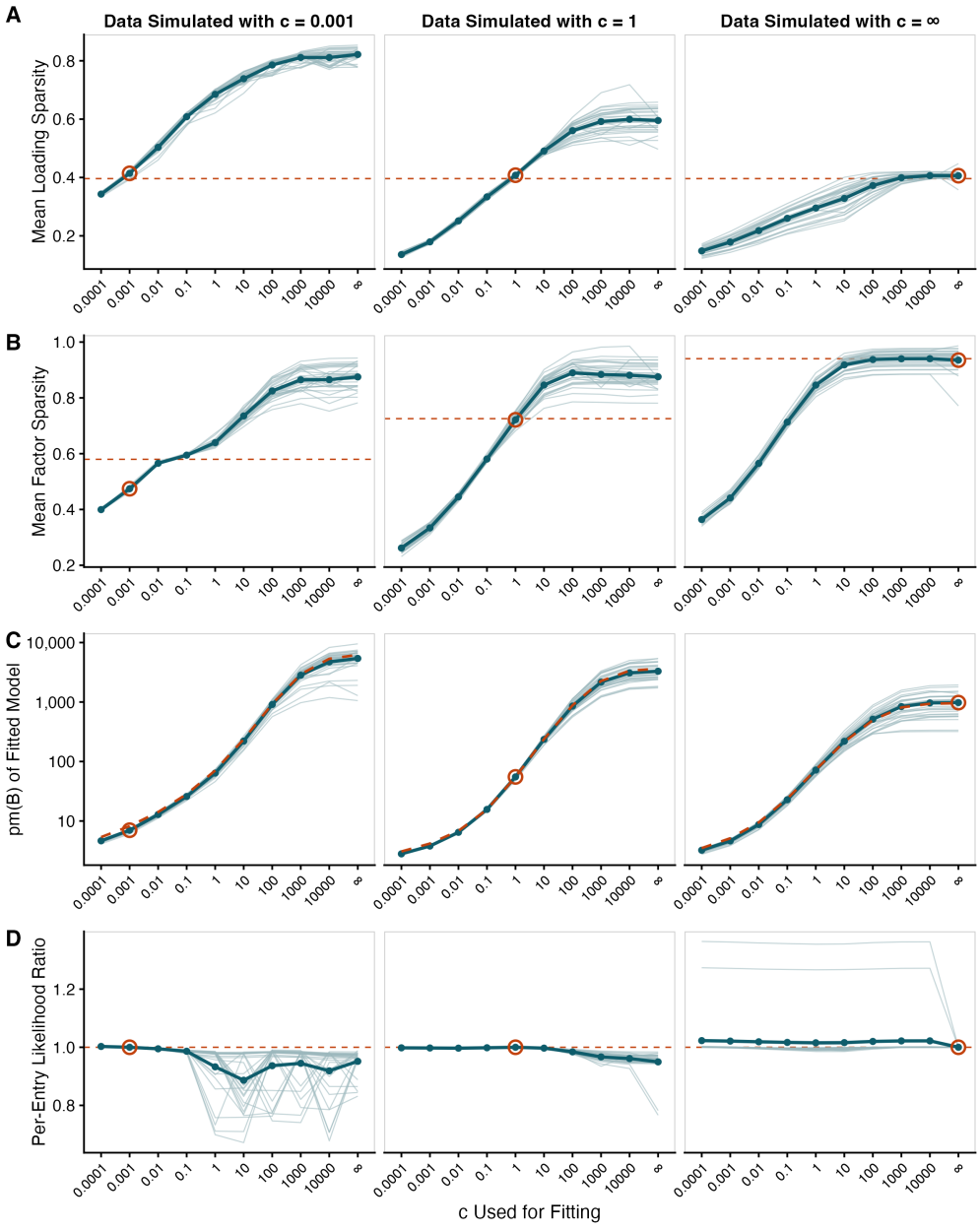


Figure 5: Results of mis-specification simulation: Thin lines show individual simulation replicates, thick lines their mean, and the open circle marks $c_{\text{fit}} = c_{\text{true}}$. (A, B) Mean Hoyer sparsity of the columns of $\mathbf{L}$ and of $\mathbf{F}$; the dashed line shows the same quantity for the true loadings and factors. (C) Peak-to-mean ratio pm($\mathbf{B}$) of the fitted model; the dashed line shows the value predicted from the true rates alone — pm($\log(1 + \mathbf{\Lambda}/c)$) at finite fitting c, and pm($\mathbf{\Lambda}$) at $c = \infty$ — averaged over replicates. (D) Per-entry likelihood ratio relative to the same replicate's fit at c_{true}, i.e. $\exp\{[\ell_{\text{log1p}}(c) - \ell_{\text{log1p}}(c_{\text{true}})]/np\}$; the dashed line marks 1.

combine additively or multiplicatively. Briefly, MCF-7 cells were either treated with all-trans retinoic acid (RA), transforming growth factor beta (TGF-β), or their combination at various concentrations. Some cells were also kept as a control (CTRL). In total, mRNA expression across 41 samples (10 CTRL, 11 RA, 9 TGF-β, 11 RA + TGF-β) was measured. After filtering down to mRNA corresponding to protein-coding genes, and excluding any genes that were not detected in at least 4 samples, we obtained a $41 \times 16{,}733$ count matrix with approximately 9% 0 entries.

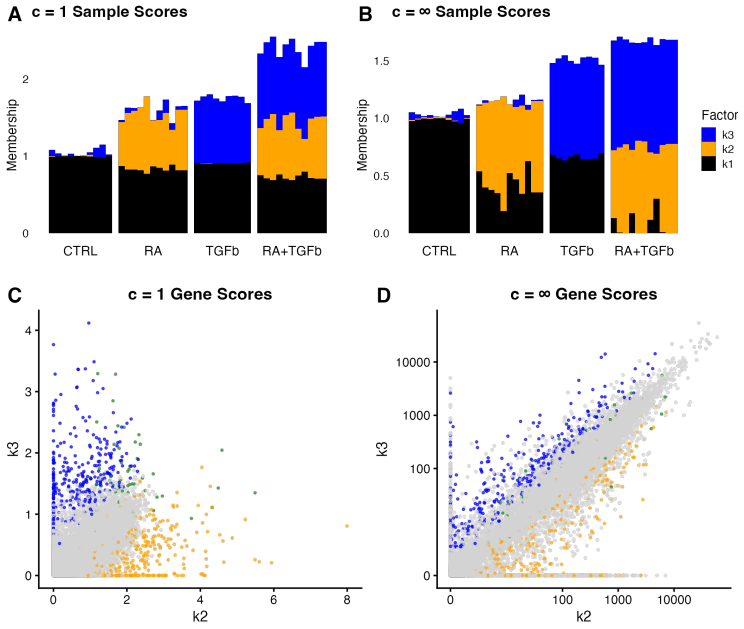
Figure 6 shows the results of fitting the log1p model with $c = 1$ and $c = \infty$ (i.e., standard Poisson NMF) to the MCF-7 data with rank 3. The inferred "sample scores" ($\mathbf{L}$) are shown in Figure 6 A,B. Although different, the two model fits share some key similarities: both use one factor to capture treatment with RA (k2, orange), and another to capture treatment with TGF-β (k3, blue). Both also use a factor to capture the control condition (k1, black). However, for $c = 1$ all samples have a high score on this factor, so it effectively acts as a "baseline" factor, whereas for $c = \infty$ the sample scores decrease in the single-treatment groups, and are close to 0 in the double-treatment (RA+TGF-β) group.

The gene scores ($\mathbf{F}$) for the treatment-related factors (k2, k3) in each model are shown in Figure 6C,D. It is immediately visually apparent that the gene scores for the two factors are more correlated under $c = \infty$ than for $c = 1$ (Spearman $\rho \approx 0.91$ vs. 0.67). Coloring the genes according to which ones are differentially expressed in each treatment vs. control (using *DESeq2*; Love et al. (2014)) shows that the genes with highest scores for $c = 1$ in each factor are generally identified as differentially expressed by *DESeq2*, consistent with the interpretation of the sample scores for this model as capturing treatment effects.

One way to view these results is that the $c = 1$ results separate out a baseline factor from the treatment factors, whereas the $c = \infty$ results absorb some of the baseline into all three factors, which causes the gene scores for different factors to be highly correlated, and more focused on the highest-expressed genes. Note that this happens for $c = \infty$ even though we used an initialization strategy that might encourage it to separate out a baseline factor. The results for $c = \infty$ are not "wrong" and the sample scores do capture the treatment structure in the data. However, by separating out a baseline factor, the $c = 1$ results make it easier to identify the "key genes" that are responding to each treatment, simply by looking at which genes have the highest gene scores in each factor (see Table in Figure 6). For example, the top two genes in factor k2, *CYP26B1* and *CYP26A1*, are the main enzymes responsible for metabolizing RA in the human body (Topletz et al., 2012), and the next two genes with the highest scores, *SLC5A5* and *STRA6*, have been previously implicated in cellular response to RA: *STRA6* is believed to be the main protein responsible for transport of RA across cell membranes (Kelly and von Lintig, 2015), and *SLC5A5* is known to be up-regulated in response to RA in MCF-7 (Kogai et al., 2000). For $c = \infty$ the genes with the highest scores are very similar across factors, and tend to be genes that are highly expressed across all samples. It is possible that a more sophisticated approach to identifying "key genes" could help here; e.g., see Carbonetto et al. (2023).

5.2.2 Murine pancreas single cell RNA-seq data

Our second example is a more complex single cell RNA-seq dataset, derived from murine pancreas cells stimulated with cytokines (Stancill et al., 2021). Briefly, cells isolated from the



Factor	Top Genes – $c = 1$	Top Genes – $c = \infty$
k1	*KRT8, COX1, ND4, ND5, ATP6, CYTB, COX2, COX3, ACTB, EEF1A1* GO: aerobic electron transport chain	*KRT8, COX1, ND4, ND5, ATP6, CYTB, COX2, COX3, ACTB, EEF1A1* GO: aerobic electron transport chain
k2	*CYP26B1, CYP26A1, SLC5A5, STRA6, SHROOM1, HOXA1, LCN2, KDR, PTPRH, SERPINA3* GO: negative regulation of retinoic acid receptor signaling pathway	*ND4, CYTB, ATP6, ND5, COX1, COX2, EEF1A1, COX3, ACTB, PABPC1* GO: aerobic electron transport chain
k3	*TP63, GABRP, TCIM, SPOCK1, DOCK4, MGP, KIAA2012, COL4A3, NCF2, PTPRB* GO: osteoblast differentiation	*KRT8, COX1, ND4, ACTB, EEF1A1, ACTG1, COX2, EEF2, ND5, ATP6* GO: oxidative phosphorylation

Figure 6: Combined figure and table for the MCF-7 analysis. (A-B) Visual representation of estimated $\mathbf{L}$ matrices for log1p NMF with $c = 1$ and $c = \infty$. Each column represents a row of $\mathbf{L}$, where each color corresponds to a column of $\mathbf{L}$. (C-D) Scatterplots of factors 2 and 3. Each point corresponds to a single gene (row of $\mathbf{F}$). Points are colored based on results of differential expression using DESeq2 (Love et al., 2014). Points in green have Benjamini-Hochberg (Benjamini and Hochberg, 1995) adjusted p-values < 0.01 and log2FC > 1 in both the RA and TGF-β groups. Points in orange meet these conditions in only the RA group, points in blue meet these conditions only in the TGF-β group, and points in grey meet these conditions in neither group. (Table): Top 10 genes in each column of $\mathbf{F}$ from the fitted models. Below each gene list, in upright type, is the most significantly enriched Gene Ontology (Ashburner et al., 2000; The Gene Ontology Consortium, 2023) biological process term among the factor's top 30 genes, computed by a hypergeometric test against all genes in the dataset (Boyle et al., 2004), with Benjamini–Hochberg adjusted $p < 0.05$.

pancreas of eight mice were first pooled and then separated into four samples. One sample was treated with interleukin-1 beta (IL-1β), another was treated with interferon gamma (IFNγ), a third was treated with both IL-1β and IFNγ, and a final sample was left untreated. After filtering for cells with between 2000 and 60000 unique molecular identifiers (UMIs), removing cells with greater than 10% of UMIs coming from mitochondrial genes, removing genes expressed in fewer than 3 cells, and removing mitochondrial genes, genes coding for ribosomal proteins, and the gene *Malat1*, we obtained a 7,606 $\times$ 18,195 (cells $\times$ genes) count matrix with approximately 82% 0 counts. These data contain eight different "cell types" ("acinar", "ductal", "endothelial/mesenchymal", "macrophage", "alpha", "beta", "delta", and "gamma" cells, with labels assigned based on the marker genes used in Stancill et al. (2021)), and we are interested in how the different NMF models capture this cell type structure in addition to the effects of cytokine treatment.

Setting $K = 13$, we fit the log1p NMF model with $c = 1$ and $c = \infty$. Both models produced some cell type-related factors and some treatment-related factors, with little overlap between these sets (the exception being factor k10 for $c = \infty$). Figure 7 shows the cell scores ($\mathbf{L}$) for the cell type-related factors and Figure 9 shows them for the treatment-related factors, and we now discuss these results in turn.

Examining the cell type-related factors, both models show clear differences among cell types, but the two representations are nonetheless quite different. In broad terms, the $c = \infty$ results are more "clustered", with each cell type being associated with one or two factors, whereas the $c = 1$ results are more "modular", with several cell types being represented as a combination of three or more factors. If one is primarily interested in clustering the cells into cell types then $c = \infty$ results may appear cleaner. On the other hand, if one is interested in understanding the underlying processes that define cell types, and which processes are shared among cell types, the $c = 1$ results may be more useful. For example, in $c = 1$ the factor k8 (red) is strongly present in most alpha, delta and gamma cells, and also present in beta cells. Similarly, the factor k13 is strongly present in gamma cells, but also present in delta and alpha cells. These factors therefore represent processes that are shared across these cell types (all of which are islet cells).

To further illustrate these differences, we consider in more detail the delta and gamma cells, which are transcriptionally very similar, with just a few genes – most notably *Ppy*, *Sst*, and *Rbp4* – showing strong differences in expression (Figure 8C). The *Ppy* gene is the canonical marker gene for gamma cells (which, indeed, are also called PP cells; Inzani et al. (2020)), while *Sst* and *Rbp4* are canonical markers for delta cells in the murine pancreas (Thielert et al., 2025). The $c = \infty$ model essentially assigns a factor to each cell type (k3, k13) with these factors having correlated gene scores (Figure 8B) and the key genes lying away from the strong main diagonal. The $c = 1$ model also captures the difference between the cells with two factors (k3,k13) but these factors show much less correlation (no strong diagonal) and focus primarily on the key genes. And k13, while strongest in gamma cells, is also present in other cells, again perhaps highlighting some shared processes. At a high level, the $c = \infty$ model captures similarity/differences between the gamma and delta cells by using two similar (but different) factors, whereas the $c = 1$ model captures it by their shared membership in rather different factors. This type of behavior explains why, in Figure 7, the cell scores for $c = \infty$ more clearly delineate cell types, whereas those for $c = 1$ better convey which cell types are similar to one another.

Besides this high-level difference between the $c = 1$ and $c = \infty$ fits, we note two other differences between the results. First, the $c = \infty$ results suggest a gradient of variation across alpha cells that is not evident in $c = 1$. This is due to a factor (k10) that is also related to treatment, and is discussed further below. Second, the $c = \infty$ results highlight a gradient of variation in the endothelial and mesenchymal cells (factor k9 vs. k12) that is absent or less evident in the $c = 1$ results. The top genes in these two factors include genes specifically related to endothelial cells (e.g., *Igfbp7*; Van Breevoort et al. (2012); He et al. (2024)) and to mesenchymal cells (e.g., *Vim*; Usman et al. (2021)) and this gradient may be related to the documented endothelial to mesenchymal transition (Piera-Velazquez and Jimenez, 2019).

Turning now to the treatment-associated factors (Figure 9) the $c = 1$ results (panel A) are somewhat analogous to our first data example above: one factor (k10) captures treatment with IL-1β and another factor (k4) captures treatment with IFNγ; samples treated with both IL-1β and IFNγ show membership in both these factors. Many of the key genes for these treatment-associated factors are biologically connected with the treatments. For example, the top gene in factor k4 (associated with IFNγ treatment) is *Cxcl10*, which is also known as interferon gamma induced protein 10 (Liu et al., 2011), and the top genes in factor k10 include several genes known to be regulated by IL-1β (e.g., *Lcn2*; Hu et al. (2015), *Cebpd*; Moore et al. (2012), *Cxcl1*; Diana and Lehuen (2014)).

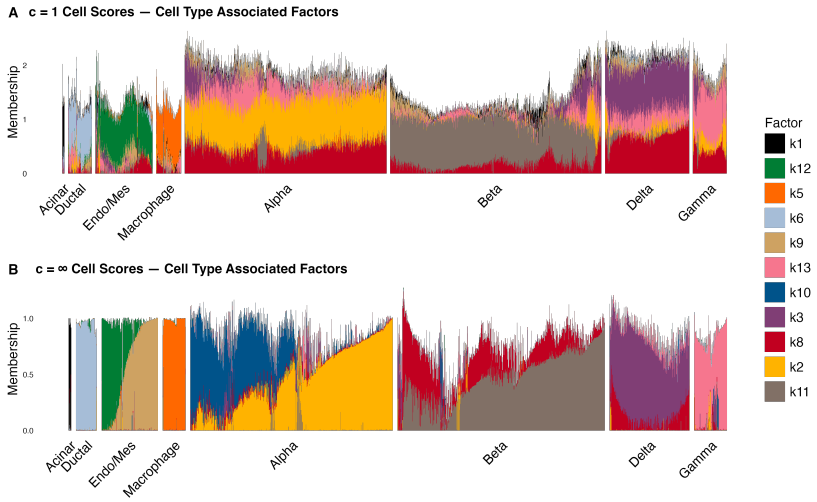
In comparison, for $c = \infty$, the treatment associated factors are harder to interpret (Figure 9B). One factor (k10) is specific to IL-1β treatment, but it is also exclusive to alpha cells (Figure 9D). As a result, the top gene in this factor (by far) is *Gcg*, which is a canonical marker of alpha cells (Stancill et al., 2021), and unlikely to be specifically related to treatment. The factor most associated with IFNγ treatment (k7) is also present in untreated cells and in cells treated with IL-1β, but largely absent from the cells treated with both cytokines, making it hard to interpret in terms of the treatments. One possible reason for these results is that the effects of cell type and treatment in these data may combine more multiplicatively than additively, and thus align better with $c = 1$ than $c = \infty$.

5.2.3 BBC News data

Finally, we analyze a text dataset of news articles collected from the BBC between 2004 and 2005 (Greene and Cunningham, 2005). Each article is labelled based on its editorial category in the BBC publication as one of the following: "business", "entertainment", "politics", "sports", or "tech". After removing SMART stop words (Salton, 1971), stripping punctuation, converting words to lower case, removing numbers, stemming the words using Porter's algorithm (Porter and Boulton, 2001), and removing any word that was not used in at least 5 articles, we created a document-term matrix of size $2{,}127 \times 5{,}861$. Approximately 98% of entries in this matrix were 0. We fit all models to these data with $K = 10$.

For these data, results for $c = 1$ looked somewhat similar to $c = \infty$, so to better highlight differences between small and large c we focus on comparing $c = 0.001$ vs. $c = \infty$ (Figure 10). Results for a range of values of c are shown in Appendix K (Figure K.6).

Examining the document scores ($\mathbf{L}$), $c = \infty$ again produces a relatively "clustered" representation, with each document having strong membership in just one or two components, whereas $c = 0.001$ produces a much more modular representation, with many documents



Factor	Top Genes – $c = 1$	Top Genes – $c = \infty$
k1	*Ctrb1, Prss2, Try5, Reg1, Cela1, Cela3b, Cpa1, Try4, Reg3b, Ins2* GO: peptide hormone processing	*Ctrb1, Prss2, Try5, Reg1, Cela1, Clu, Cela3b, Cpa1, Reg3b, Try4* GO: peptide hormone processing
k2	*Gcg, Ttr, Spp1, Pyy, Gpx3, Rbp4, Tmem27, Gnas, Gc, Higd1a* GO: hormone secretion	*Gcg, Pyy, Ttr, Tpt1, Gnas, Chga, Tmem27, Eef1a1, Resp18, Spp1* GO: hormone secretion
k3	*Sst, Pyy, Ppy, Rbp4, Arg1, Clu, Chgb, Ins1, Fam159b, Cd24a* GO: pancreas development	*Sst, Pyy, Iapp, Chgb, Rbp4, Tpt1, Gnas, Resp18, Scg2, Meg3* GO: peptide hormone processing
k5	*Fth1, Apoe, Tmsb4x, Ftl1, Lgals3, Ctsb, Actb, Prdx1, C1qb, Fcer1g* GO: antigen processing and presentation of exogenous peptide antigen via MHC class II	*Fth1, Ftl1, Tmsb4x, Apoe, Lgals3, Actb, Ctsb, Mt1, Prdx1, Psap* GO: lymphocyte mediated immunity
k6	*Clu, Spp1, Lcn2, Tmsb4x, Krt8, Krt18, Eef1a1, Cxcl5, Tpt1, Epcam* GO: response to tumor necrosis factor	*Spp1, Clu, Lcn2, Eef1a1, Tpt1, Tmsb4x, Krt18, Krt8, Actg1, Actb* GO: cell killing
k8	*Iapp, Tpt1, Scg2, Ubb, Cd63, Resp18, Ssr4, Eef1a1, Scg5, Chgb* GO: platelet aggregation	*Iapp, Chga, Tpt1, Chgb, Eef1a1, Scg2, Resp18, Cd63, Cpe, Pcsk2* GO: peptide hormone processing
k9	*Ghrl, Ppy, Hspa1a, Mt1, Gcg, Mgp, Hspa1b, Mt2, Mif, Bnip3* GO: canonical glycolysis	*Mgp, Tpt1, Igfbp7, Eef1a1, Igfbp5, Ifitm3, Cxcl10, Actb, Tmsb4x, Ptma* GO: response to interferon-beta
k11	*Ins2, Ins1, Iapp, Chga, Ftl1, Tpt1, Ubb, Eef1a1, Scg2, Resp18* GO: digestion	*Ins2, Ins1, Iapp, Chga, Ftl1, Ubb, Resp18, Scg2, Pcsk1n, Tpt1* GO: digestion
k12	*Mgp, Igfbp7, Igfbp5, Tmsb4x, Sparc, Vim, Ifitm3, Eef1a1, Tpt1, Ccl2* GO: cell chemotaxis	*Tpt1, Tmsb4x, Eef1a1, Cxcl10, Fth1, Ccl2, Vim, Actb, Cxcl1, Spp1* GO: neutrophil chemotaxis
k13	*Ppy, Pyy, Ins2, Spp1, Chga, Tspan8, Resp18, Fth1, Pcsk1n, Clu* GO: peptide hormone processing	*Ppy, Pyy, Tpt1, Resp18, Eef1a1, Ubb, Cst3, Chgb, Scg2, Spp1* GO: peptide hormone processing

Figure 7: Combined figure and table for the cell type associated factors of the Pancreas analysis. (A-B) Visual representation of estimated $\mathbf{L}$ matrices for the log1p NMF with $c = 1$ and $c = \infty$, grouped by cell type. Each column represents a row of $\mathbf{L}$, where each color corresponds to a column of $\mathbf{L}$. (Table) Top genes for the factors of each model, excluding treatment associated factors. GO terms are computed as described in the caption of Figure 6.

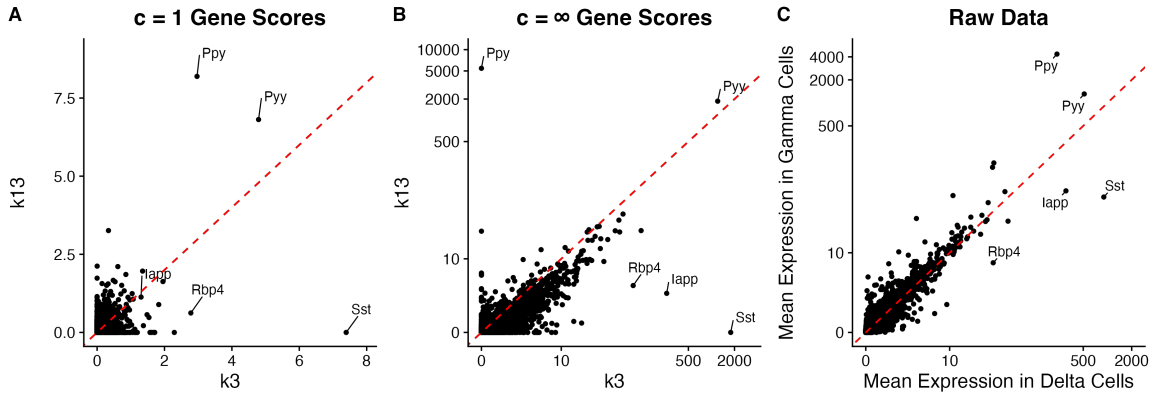
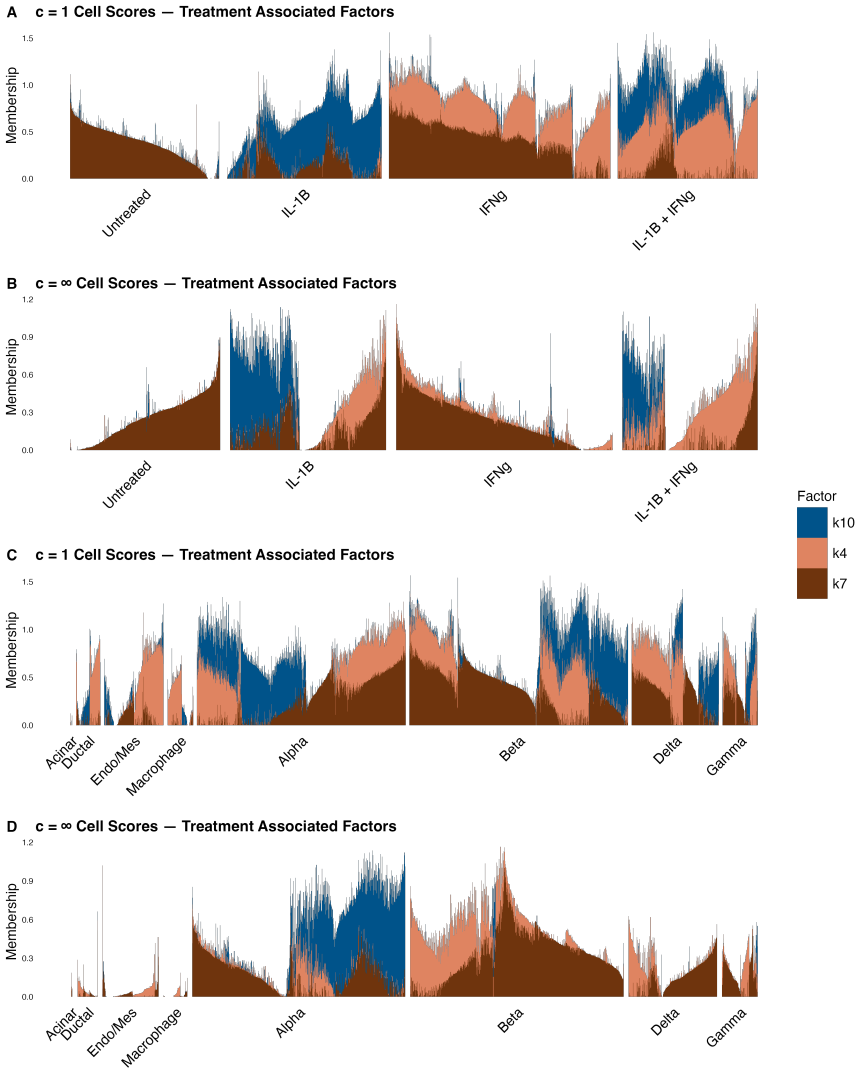


Figure 8: Comparison between delta and gamma cells in single cell pancreas data. (A-B) Gene scores ($\mathbf{F}$) from $c = 1$ and $c = \infty$ models corresponding to the factor most associated with delta cells (k3) and the factor most associated with gamma cells (k13). Before plotting, each matrix $\mathbf{L}$ was normalized so that the maximum value of each column was 1, and $\mathbf{F}$ was scaled accordingly. (C) Mean expression of genes in delta and gamma cells.

having membership in multiple components. Comparing the top words ("keywords") in each component, the two fits capture some similar topics with highly overlapping keywords (Figure 10, Table). For example, both fits identify a politics topic (k1, whose shared keywords include "labour", "tori(es)", "blair", "vote", and "tax"); a business topic (k6, with shared keywords "market", "price", "growth", etc); and an entertainment topic (k2, with shared keywords "film", "actor", "oscar", "nomin(ate)", "actress"). However, there are some differences in keywords that reflect the additive vs. multiplicative behavior of the two models: for example, in the entertainment topic (k2) the keywords for $c = 0.001$ results place greater emphasis on rarer proper nouns (e.g., "dicaprio", "foxx"). These proper nouns are not close to the most used words in entertainment articles ("dicaprio" and "foxx" are the 317^{th} and 162^{nd} most used words among all entertainment articles, respectively), but they are *relatively* much more frequent in entertainment documents than in other documents (indeed, neither "dicaprio" nor "foxx" occur in documents in other categories).

The difference between a more clustered as opposed to a more modular/layered representation is nicely illustrated by the different ways the two models represent heterogeneity among sports articles. The $c = \infty$ result essentially divides these articles into three distinct clusters (k5, k7, k10), which from the keywords seem to correspond roughly to rugby, football (soccer), and other sports. In contrast the $c = 0.001$ results yield one factor (k5) that almost every sports article is loaded on, and heterogeneity among sports articles is captured by loadings on a variety of additional components (e.g., k7 captures the soccer-related articles). Some of these additional components seem to be doing "double-duty": for example, the keywords of k8 include "athlet(e)" and "olymp(ic)", but also "music", "hip", "hop" and "soul". The existence of such double-duty components may be a side-effect of the quasi-



Factor	Top Genes – $c = 1$	Top Genes – $c = \infty$
k4	*Cxcl10, Gbp2, Iigp1, Gbp4, Igtp, Gbp7, Gbp3, Irgm1, Stat1, Cd274* GO: cellular response to type II interferon	*Ins2, Mt1, Tpt1, Eef1a1, Mt2, Cxcl10, Gbp2, Actg1, Hspa8, Fth1* GO: cell killing
k7	*Kcnq1ot1, Acly, Atp2a2, Peg3, Eef1a1, Zbtb20, Ccnd2, Gm42418, Gatsl2, Zdhhc2* GO: insulin secretion	*Tpt1, Eef1a1, Ftl1, Ins2, Chga, Calr, Gm42418, Fau, Cd63, Eif1* GO: striated muscle contraction
k10	*Defb1, Lcn2, Mt1, Mt2, Cebpd, Sod2, Cxcl1, Steap4, Ccl2, Pabpc1* GO: inorganic ion homeostasis	*Gcg, Spp1, Tpt1, Ttr, Pyy, Eef1a1, Gnas, Resp18, Hamp, Pcsk2* GO: peptide hormone processing

Figure 9: Visualization of the treatment associated factors of the Pancreas analysis. (A-B) Cell scores grouped by treatment. (C-D) Cell scores grouped by cell type. (Table) Top genes for the factors of each model, excluding cell type associated factors. GO terms are computed as described in the caption of Figure 6.

multiplicative nature of the model, where the effect of a component can depend very much on what it is being added to: intuitively, adding component k8 to other components that already contain music-related keywords will most increase the fitted values of those music-related words, whereas adding it to other components that contain sport-related words will most increase the fitted values of those sport-related words (to state the obvious, multiplying big numbers by something creates a bigger absolute change than multiplying small numbers by the same thing). These double-duty components are also somewhat harder to interpret (k10 for c=0.001 is particularly challenging); it is possible that increasing the number of factors, and/or introducing methods that encourage sparsity of the word scores could help here.

While the more modular/layered representation of $c = 0.001$ may sometimes be harder to interpret, it can also sometimes highlight subtler structure that is missing in the clustered representation of $c = \infty$. For example, the soccer factor (k7) appears only in sports articles for $c = \infty$, but appears across all 5 document classes in the $c = 0.001$ fit. Manually examining the documents with high scores on k7 (with $c = 0.001$) reveals that they typically mention words related to sports, and sometimes specifically British soccer, once or twice in the article. For example, the politics article with the highest score on k7 is about a controversial comment made by then London Mayor Ken Livingstone, and mentions soccer in reference to a similarly-controversial comment made by Boris Johnson regarding Liverpool fans (BBC, 2005).

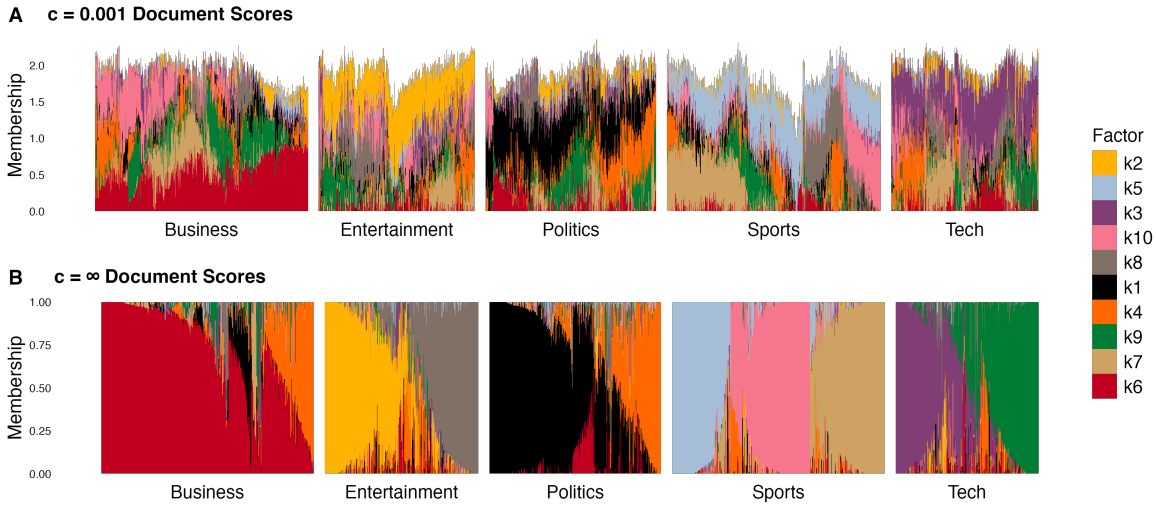
5.3 Quantitative assessment of log1p NMF fits across datasets

In discussing the results above, particularly for the last two datasets, we noted that standard Poisson NMF ($c = \infty$) produces more "clustered" representations, where most samples (cells or documents) have appreciable scores on only one or two factors, and that the inferred factors tend to be highly correlated; in contrast, smaller values of c produced more modular/layered representations, with each sample being represented as a combination of more factors (although still a modest number) that are less correlated. In brief: larger values of c produced sparser loadings and more correlated factors. Interestingly, the first of these two observations is well supported by our simulations of Section 5.1.2 (Figure 5A,B), whereas the second is less so (Figure K.3B).

To quantify this trend more systematically, we fit log1p NMF to each dataset for a grid of values between $c = 10^{-3}$ and $c = 10^3$ (as well as $c = \infty$). For each fit, we measured the mean (absolute) Spearman correlation between columns of $\mathbf{F}$, and the mean column-wise sparsity (as defined in equation (17)) of both $\mathbf{L}$ and $\mathbf{F}$. The results (Figure 11) confirm that larger c produced sparser loadings (and factors) and more correlated factors. Visualizations of the sample scores of these fitted models are provided in Appendix K.

6. Discussion

We have introduced log1p Poisson NMF, which is, to our knowledge, the first version of Poisson NMF with a non-identity link function. Our new approach has a parameter, c, which controls the behavior of the link function: large values of c make it more linear, similar to standard Poisson NMF, and small values of c make it more logarithmic (factors combine quasi-multiplicatively), and different from standard Poisson NMF. In three examples we



Factor	Top Words - $c = 0.001$	Top Words - $c = \infty$
k1	tori, labour, elect, parti, blair, kennedi, lib, conserv, dem, howard, liber, democrat, voter, prime, chancellor, tax, brown, vote, toni, immigr	labour, elect, parti, govern, peopl, blair, minist, tori, plan, brown, tax, howard, public, leader, prime, campaign, work, polit, vote, year
k2	film, actor, oscar, nomin, actress, award, comedi, star, foxx, nomine, aviat, drama, sideway, scorses, staunton, dicaprio, bafta, hollywood, ceremoni, drake	film, award, star, year, show, actor, director, oscar, nomin, includ, won, movi, actress, role, win, prize, comedi, festiv, bbc, british
k3	user, technolog, broadband, digit, mobil, devic, phone, web, content, network, net, microsoft, comput, download, program, search, internet, video, pcs, gadget	game, mobil, technolog, peopl, phone, digit, video, year, player, make, time, play, music, devic, develop, market, gadget, high, work, comput
k4	hunt, sentenc, crimin, guilti, prosecut, murder, law, suspect, arrest, virus, detain, lawyer, evid, trial, investig, fraud, polic, offenc, ban, prosecutor	law, court, govern, lord, case, rule, polic, legal, told, claim, right, charg, yuko, offic, compani, bill, trial, ban, hunt, year
k5	wale, win, ireland, england, victori, injuri, match, play, franc, squad, coach, itali, side, half, william, minut, cup, player, game, score	england, wale, ireland, game, nation, rugbi, franc, play, half, side, win, player, back, year, coach, scotland, team, robinson, injuri, time
k6	growth, economi, price, rate, economist, rise, profit, bank, inflat, quarter, market, forecast, econom, rose, analyst, manufactur, consum, export, dollar, retail	year, market, compani, bank, sale, price, firm, share, growth, economi, month, rate, expect, econom, countri, rise, busi, report, profit, china
k7	chelsea, arsenal, liverpool, wenger, gerrard, parri, rover, club, mourinho, ferguson, everton, striker, fiat, consol, manchest, villa, footbal, anfield, aston, morient	club, game, play, player, unit, time, chelsea, manag, footbal, leagu, goal, team, liverpool, win, arsenal, back, year, manchest, side, cup
k8	athlet, olymp, women, urban, marathon, album, radcliff, artist, indoor, athen, hip, medal, hop, gadget, song, band, music, drug, holm, soul	music, year, song, band, record, album, show, number, includ, top, singl, chart, award, artist, perform, rock, singer, releas, peopl, radio
k9	iran, straw, palestinian, blog, isra, aid, embargo, tsunami, turkey, peac, egypt, china, foreign, iraqi, israel, wto, cyprus, india, blogger, rugbi	peopl, user, net, servic, network, site, softwar, firm, internet, system, mail, comput, search, secur, call, inform, onlin, websit, broadband, virus
k10	deutsch, yuko, boers, lse, sharehold, takeov, euronext, russian, mci, file, seed, gazprom, bid, ukip, auction, houston, kilroy, chart, silk, khodorkovski	year, world, win, final, set, open, olymp, play, time, champion, athlet, match, race, titl, game, won, drug, test, roddick, beat

Figure 10: Combined figure and table for the BBC analysis. (A-B) Visual representation of estimated $\mathbf{L}$ matrices for log1p NMF with $c = 0.001$ and $c = \infty$, grouped by document type. Each column represents a row of $\mathbf{L}$, where each color corresponds to a column of $\mathbf{L}$. (Table) Top words for each factor.

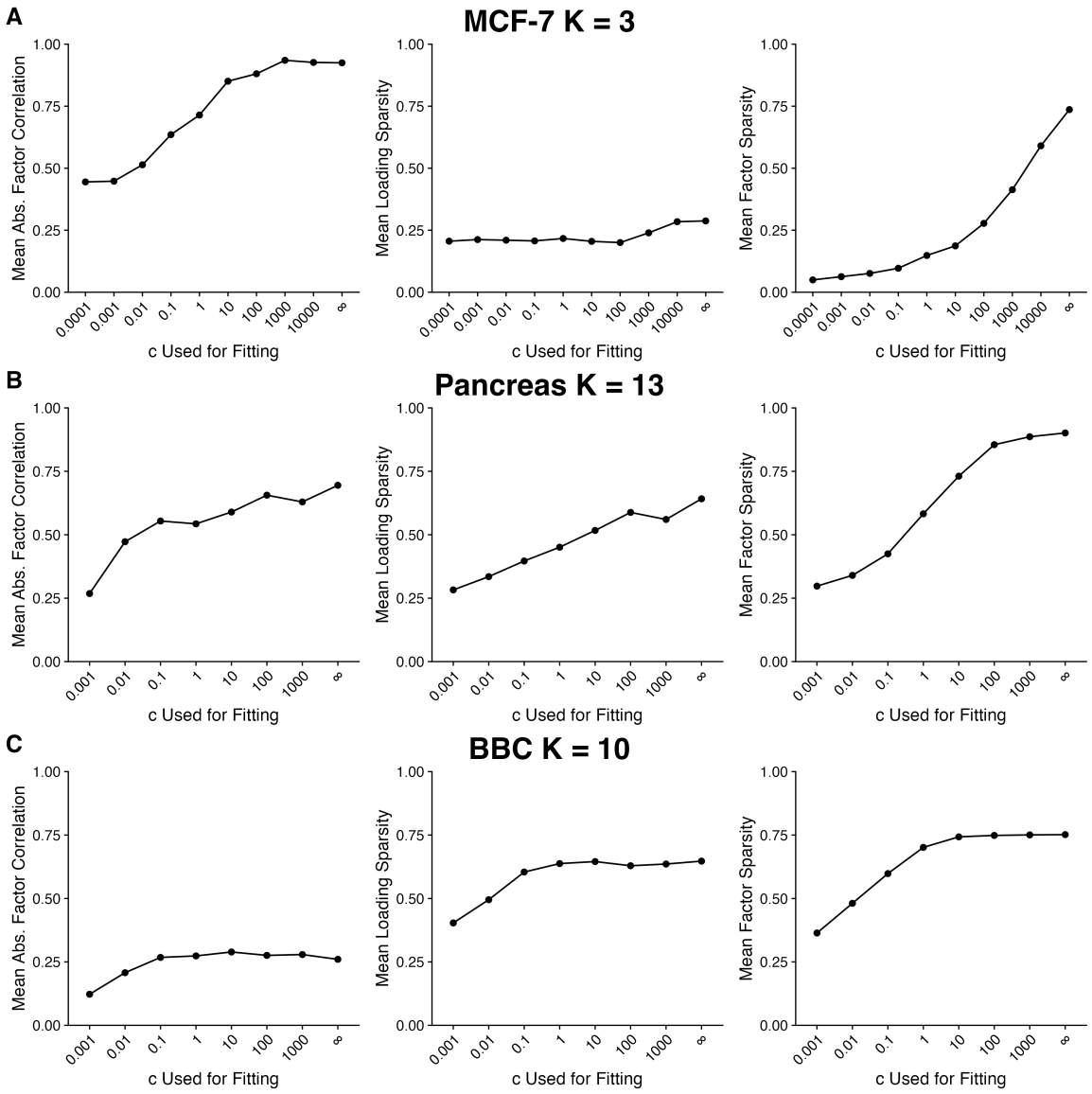


Figure 11: Summary of model fits across values of c for real data applications. Spearman correlations are calculated pairwise between all factors (columns of $\mathbf{F}$) and averaged. Sparsity values are calculated for each loading / factor (columns of $\mathbf{L}$ or columns of $\mathbf{F}$) and then averaged. (A) Correlation and sparsity values for the MCF-7 dataset with $K = 3$. (B) Correlation and sparsity values for the pancreas dataset with $K = 13$. (C) Correlation and sparsity values for the BBC dataset with $K = 10$.

showed that different link functions (different values of c) can produce quite different results, potentially yielding different insights. Since Poisson NMF is widely used in practice, these new methods have the potential to provide new insights in a wide range of applications.

Given the new ability to fit Poisson NMF with a range of link functions, it may seem natural to ask what the "correct" link function is for any given application. However, Poisson NMF may perhaps be best viewed as an exploratory data analysis tool, and this view encourages a different perspective: different link functions that provide different representations and different insights might complement one another rather than compete. For example, in our applications we saw that $c = \infty$ may produce a more clustered view of the data, whereas smaller c produced a more modular/layered view. Neither view is "correct", but both may be useful in their own way (and so, in this sense, neither is "incorrect"). In practice, we therefore suggest beginning NMF analyses of count data by running log1p NMF with both $c = \infty$ and a smaller value of c (e.g., $c = 1$) to compare the results. Even in settings where one might argue for an additive link function on scientific grounds, comparing the results with $c = 1$ could be an interesting exercise.

One trend we saw in our applications is that log1p NMF with small c tends to produce less sparse solutions. Usually sparsity is considered helpful for interpretability of results, so producing less sparse solutions may seem a step backwards. However, in our applications this does not appear clear-cut: indeed, in these applications small c arguably produces representations that are more "parts-based", in the sense of Lee and Seung (1999), who highlighted this as a major benefit of NMF. Nonetheless, all things being equal, sparsity may indeed aid interpretation, and investigating sparse versions of log1p NMF could be an interesting area for future work. In particular, adapting empirical Bayes methods for Gaussian matrix factorization (e.g., Wang and Stephens (2021)) to the log1p NMF model could improve the practical utility of the model, and may also aid in determining appropriate values of K, another challenging practical problem when using NMF. It may also be interesting to extend other approaches to selecting K, such as the one in Zito and Miller (2024), to log1p NMF.

Finally, we also found empirically that small c produced results with less-correlated factors. It is possible that this result is particular to the data sets we examined here, but it is also possible that this is a more general phenomenon. We have performed some very preliminary investigations in this direction (see Appendix J), and can show that, for small c ($c \to 0^+$) the expressivity of a regression model with two factors depends on the upper right boundary of the convex hull when these two factors are plotted against one another. If the two factors are highly correlated then there will be few points on this hull, resulting in low expressivity. (Indeed, if the factors have the same maximal element then the upper right convex hull is a single point, and the 2-factor regression model collapses to a single factor). Extending this intuition to K factors suggests that small c models may prefer less correlated factors simply because they are more expressive, resulting in a better fit to data. A rigorous mathematical study of this, and more generally of identifiability issues with log1p NMF, could also be an interesting area for further work.

Acknowledgments

This research was supported by NIH grant HG002585 to MS and NIH grants R35GM131802 and R01HG005220 to RAI. We thank members of the Stephens and Irizarry labs for helpful comments and feedback. Large language models were used to assist with proofreading, copy editing, and some derivations in Section H and Section J. The authors carefully reviewed and verified all AI-assisted content and take full responsibility for the contents of the manuscript.

Appendix A. Derivations and proofs

A.1 Proof of Theorem 1

Proof First, with $\alpha_c := \max(1, c)$, define the functions

$$\lambda_{\text{log1p}}(\mathbf{l}, \mathbf{f}, c) = c\left(\exp\left\{\frac{1}{\alpha_c}\sum_{k=1}^{K} l_k f_k\right\} - 1\right)$$
$$\lambda_{\text{id}}(\mathbf{l}, \mathbf{f}) = \sum_{k=1}^{K} l_k f_k,$$

where $\mathbf{l}, \mathbf{f} \in \mathbb{R}^K_{\geq 0}$, which are the Poisson rate parameters of a single arbitrary element of $\mathbf{Y}$ for the log1p model and Poisson NMF model with identity link, respectively. Since the log-likelihood of data generated from a Poisson distribution depends on its parameters only through its rate, it suffices to show that for any $\mathbf{l}, \mathbf{f} \in \mathbb{R}^K_{\geq 0}$,

$$\lim_{c\to\infty} \lambda_{\text{log1p}}(\mathbf{l}, \mathbf{f}, c) = \lambda_{\text{id}}(\mathbf{l}, \mathbf{f}).$$

Now, let $b = \sum_{k=1}^{K} l_k f_k$ and let $x = \frac{b}{c}$. If $b = 0$, then $\lambda_{\text{log1p}}(\mathbf{l}, \mathbf{f}, c) = \lambda_{\text{id}}(\mathbf{l}, \mathbf{f}) = 0$, in which case the statement holds for any c. If $b > 0$, we have

$$\begin{aligned}
\lim_{c\to\infty} \lambda_{\text{log1p}}(\mathbf{l}, \mathbf{f}, c) &= \lim_{c\to\infty} c\left(\exp\left\{\frac{1}{\alpha_c}\sum_{k=1}^{K} l_k f_k\right\} - 1\right) \\
&= b \lim_{x\to 0^+} \frac{e^x - 1}{x} \quad (\text{since } \alpha_c = c \text{ for } c \geq 1) \\
&= b \lim_{x\to 0^+} e^x \quad (\text{L'Hospital's rule}) \\
&= b
\end{aligned}$$

Substituting this back into the definition of $\lambda_{\text{log1p}}(\cdot)$, we have

$$\lim_{c\to\infty} \lambda_{\text{log1p}}(\mathbf{l}, \mathbf{f}, c) = \sum_{k=1}^{K} l_k f_k = \lambda_{\text{id}}(\mathbf{l}, \mathbf{f}).$$

■

A.2 Proof of Theorem 2

Proof Substituting the definition of ε_{ij} in equation (9) into equation (8), we can write

$$\begin{aligned}
\frac{1}{\alpha_c} l_{ik} f_{jk} + \varepsilon_{ij} &= \frac{1}{\alpha_c} l_{ik} f_{jk} + \log\left(1 + c/\lambda_{ij}^{(-k)}\right) - \log(1 + c/\lambda_{ij}) \\
&= \frac{1}{\alpha_c} l_{ik} f_{jk} + \log\left(\frac{1 + c/\lambda_{ij}^{(-k)}}{1 + c/\lambda_{ij}}\right) \\
&= \frac{1}{\alpha_c} l_{ik} f_{jk} + \log\left(\frac{\left(\frac{\lambda_{ij}^{(-k)} + c}{\lambda_{ij}^{(-k)}}\right)}{\left(\frac{\lambda_{ij} + c}{\lambda_{ij}}\right)}\right) \\
&= \frac{1}{\alpha_c} l_{ik} f_{jk} + \log(\lambda_{ij}) - \log\left(\lambda_{ij}^{(-k)}\right) + \log\left(\lambda_{ij}^{(-k)} + c\right) - \log\left(\lambda_{ij} + c\right) \\
&= \frac{1}{\alpha_c} l_{ik} f_{jk} + \log\left(\frac{\lambda_{ij}}{\lambda_{ij}^{(-k)}}\right) - \frac{1}{\alpha_c} l_{ik} f_{jk} \\
&= \log\left(\frac{\lambda_{ij}}{\lambda_{ij}^{(-k)}}\right),
\end{aligned}$$

where the second to last line follows by the definition of the link function (5). ■

A.3 Proof of bi-concavity of log1p NMF log-likelihood

Theorem 3 *With c fixed, $\ell_{log1p}(\mathbf{L}, \mathbf{F}, c; \mathbf{Y})$ is a bi-concave function of $\mathbf{L}$ and $\mathbf{F}$ on the domain $\Omega = \left\{\mathbf{L} \in \mathbb{R}_{\geq 0}^{n \times K}, \mathbf{F} \in \mathbb{R}_{\geq 0}^{p \times K} \mid \sum_{k=1}^{K} l_{ik} f_{jk} > 0 \; \forall \, (i,j) \in \{1, \ldots, n\} \times \{1, \ldots, p\}\right\}$. That is, with c and $\mathbf{L}$ fixed, $\ell_{log1p}(\mathbf{L}, \mathbf{F}, c; \mathbf{Y})$ is a concave function of $\mathbf{F}$, and with c and $\mathbf{F}$ fixed, $\ell_{log1p}(\mathbf{L}, \mathbf{F}, c; \mathbf{Y})$ is a concave function of $\mathbf{L}$.*

Proof First, for $b > 0$ and $y \geq 0$, define

$$h_y(b) = y \log(e^{b/\alpha_c} - 1) - c e^{b/\alpha_c},$$

where $\alpha_c := \max(1, c)$ for $c > 0$. Then, we have that

$$\frac{\partial^2 h_y}{\partial b^2} = -\frac{e^{b/\alpha_c}}{\alpha_c^2} \left(\frac{y}{(e^{b/\alpha_c} - 1)^2} + c\right).$$

Since $y, c, \alpha_c \geq 0$, for all $b > 0$ we have that $\frac{\partial^2 h_y}{\partial b^2}(b) \leq 0$. Thus, $h_y(b)$ is a concave function of b on the domain $(0, \infty)$ for any $y \geq 0$. Now, note that (up to a constant with respect to $\mathbf{L}$ and $\mathbf{F}$), we can write

$$\ell_{\text{log1p}}(\mathbf{L}, \mathbf{F}, c; \mathbf{Y}) = \sum_{i=1}^{n} \sum_{j=1}^{p} h_{y_{ij}}(b_{ij}) = \sum_{i=1}^{n} \sum_{j=1}^{p} h_{y_{ij}} \left(\sum_{k=1}^{K} l_{ik} f_{jk}\right). \tag{21}$$

Now, with $\mathbf{F}$ fixed and $\mathbf{L}, \mathbf{F} \in \Omega$, $h_{y_{ij}}\left(\sum_{k=1}^{K} l_{ik} f_{jk}\right)$ is the composition of an affine function (in $\mathbf{L}$) and a concave function, and is thus itself concave in $\mathbf{L}$ (Boyd and Vandenberghe, 2004, Section 3.2.2). Then, since equation (21) is a sum of concave functions of $\mathbf{L}$, the sum itself must be concave in $\mathbf{L}$ (Boyd and Vandenberghe, 2004, Section 3.2.1). This shows that with $\mathbf{F}$ fixed and $\mathbf{L}, \mathbf{F} \in \Omega$, $\ell_{\text{log1p}}(\mathbf{L}, \mathbf{F}, c; \mathbf{Y})$ is a concave function of $\mathbf{L}$. An analogous argument shows that $\ell_{\text{log1p}}(\mathbf{L}, \mathbf{F}, c; \mathbf{Y})$ is a concave function of $\mathbf{F}$ with $\mathbf{L}$ fixed. ■

Appendix B. Computational complexity of approximate log-likelihood

In the main text, we suggest an approximation to the log-likelihood of the log1p model:

$$\ell_{\text{log1p}}(\mathbf{L}, \mathbf{F}, c; \mathbf{Y}) \approx \sum_{(i,j) \notin \mathcal{I}_0} y_{ij} \log\left(\exp\left\{\frac{1}{\alpha_c}\sum_{k=1}^{K} l_{ik} f_{jk}\right\} - 1\right) - c \sum_{(i,j) \notin \mathcal{I}_0} \exp\left(\frac{1}{\alpha_c}\sum_{k=1}^{K} l_{ik} f_{jk}\right) - \frac{\eta_1 c}{\alpha_c} \sum_{(i,j) \in \mathcal{I}_0} \sum_{k=1}^{K} l_{ik} f_{jk} - \frac{\eta_2 c}{\alpha_c^2} \sum_{(i,j) \in \mathcal{I}_0} \left(\sum_{k=1}^{K} l_{ik} f_{jk}\right)^2.$$

The first two terms in the above equation clearly require $\mathcal{O}(\omega K)$ operations, where ω is the number of non-zero entries of $\mathbf{Y}$. Naively, it appears that the final two terms would require $\mathcal{O}(|\mathcal{I}_0|K)$ operations, as they require computing $|\mathcal{I}_0|$ terms, each of which requires K operations. However, observe the identities

$$\sum_{i=1}^{n}\sum_{j=1}^{p}\sum_{k=1}^{K} l_{ik} f_{jk} = \sum_{k=1}^{K}\left(\sum_{i=1}^{n} l_{ik}\right)\left(\sum_{j=1}^{p} f_{jk}\right), \tag{22}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{p}\left(\sum_{k=1}^{K} l_{ik} f_{jk}\right)^2 = \operatorname{tr}\left(\mathbf{F}\mathbf{L}^\top\mathbf{L}\mathbf{F}^\top\right). \tag{23}$$

Thus, equation (22) can be computed in $\mathcal{O}((n+p)K)$ operations and (23) can be computed in $\mathcal{O}((n+p)K^2)$ operations.

Using the above identities, we can re-write the approximate log-likelihood as

$$\ell_{\text{log1p}}(\mathbf{L}, \mathbf{F}, c) \approx \sum_{(i,j) \notin \mathcal{I}_0} \left[y_{ij} \log\left(\exp\left\{\frac{1}{\alpha_c}\sum_{k=1}^{K} l_{ik} f_{jk}\right\} - 1\right) \right. \tag{24}$$

$$\left. - c\left(\exp\left\{\frac{1}{\alpha_c}\sum_{k=1}^{K} l_{ik} f_{jk}\right\} - \frac{\eta_1}{\alpha_c}\sum_{k=1}^{K} l_{ik} f_{jk} - \frac{\eta_2}{\alpha_c^2}\left\{\sum_{k=1}^{K} l_{ik} f_{jk}\right\}^2\right)\right] \tag{25}$$

$$- c\left[\frac{\eta_1}{\alpha_c}\sum_{k=1}^{K}\left(\sum_{i=1}^{n} l_{ik}\right)\left(\sum_{j=1}^{p} f_{jk}\right) + \frac{\eta_2}{\alpha_c^2} \cdot \operatorname{tr}\left(\mathbf{F}\mathbf{L}^\top\mathbf{L}\mathbf{F}^\top\right)\right].$$

The first two lines (24)-(25) above can still be computed in $\mathcal{O}(\omega K)$ operations because each term involves the sum of the same K values $\left(\sum_{k=1}^{K} l_{ik} f_{jk}\right)$, and the final line has complexity

as described above. This brings the total computational complexity of the approximate log-likelihood to

$$\mathcal{O}\left((\omega + n + p)K + (n + p)K^2\right),$$

as described in the main text.

Appendix C. Fitting non-negative Poisson GLMs with a log1p link using cyclic coordinate ascent

Algorithm 1 involves repeatedly fitting the model

$$y_i \overset{\text{indep.}}{\sim} \text{Poisson}(\lambda_i)$$
$$\alpha_c \log\left(1 + \frac{\lambda_i}{c}\right) = \mathbf{x}_i^\top \boldsymbol{\beta},$$

where $\mathbf{y} \in \mathbb{N}_0^N$ is a vector of counts, $\mathbf{X} \in \mathbb{R}_{\geq 0}^{N \times q}$ is a fixed matrix of non-negative "covariates", $\boldsymbol{\beta} \in \mathbb{R}_{\geq 0}^q$ is an unknown vector of non-negative regression coefficients, $c \in \mathbb{R}_{>0}$ is a fixed constant, and $\alpha_c := \max(1, c)$. This model's log-likelihood, $\ell_{\text{log1pReg}}(\boldsymbol{\beta}, c; \boldsymbol{y}, \mathbf{X})$, is written in equation (13). Fitting the non-negative Poisson GLM via maximum likelihood thus reduces to solving the problem

$$\hat{\boldsymbol{\beta}} = \text{argmax}_{\boldsymbol{\beta}} \;\; \ell_{\text{log1pReg}}(\boldsymbol{\beta}, c; \boldsymbol{y}, \mathbf{X}) \tag{26}$$
$$\text{subject to} \quad \beta_j \geq 0, \;\; j = 1, \ldots, q. \tag{27}$$

We solve (26)-(27) using cyclic coordinate ascent (CCA) (Bertsekas, 1999; Wright, 2015) due to its simplicity and good performance in similar Poisson matrix factorization problems (Carbonetto et al., 2021; Weine et al., 2024). The CCA algorithm performs the following 1-d optimization for each $j = 1, \ldots, q$,

$$\beta_j^{\text{new}} \leftarrow \text{argmax}_{\beta_j \geq 0} \, \ell_{\text{log1pReg}}(\boldsymbol{\beta}, c; \boldsymbol{y}, \mathbf{X}), \tag{28}$$

and repeats these 1-d optimizations for some fixed number of cycles or until the iterates reach a stationary point (by default our implementation uses a maximum of 3 cycles).

To solve each 1-d optimization problem of the form (28), we use a simple projected Newton's method with a line search (Bertsekas, 1982). This algorithm is very efficient, and the projection step can be performed in constant time because we are solving a 1-d problem.

We note that when maximizing our approximate log-likelihood of equation (15), we also solve an approximate version of problem (26)-(27). Specifically, using the approximate log-likelihood of equation (15) and performing a decomposition analogous to equation (14) leads to a GLM with approximate log-likelihood

$$\begin{aligned}\ell_{\text{log1pReg}}(\boldsymbol{\beta}, c; \boldsymbol{y}, \mathbf{X}) \approx \sum_{i \notin \mathcal{I}_0} \Bigg[& y_i \log\left\{\exp\left\{\mathbf{x}_i^\top \boldsymbol{\beta}/\alpha_c\right\} - 1\right\} \\ & - c\left(\exp\left\{\mathbf{x}_i^\top \boldsymbol{\beta}/\alpha_c\right\} - \frac{\eta_1}{\alpha_c}\mathbf{x}_i^\top \boldsymbol{\beta} - \frac{\eta_2}{\alpha_c^2}\left\{\mathbf{x}_i^\top \boldsymbol{\beta}\right\}^2\right)\Bigg] \\ & - c\left(\frac{1}{\alpha_c}\boldsymbol{\beta}^\top \mathbf{X}^\top \boldsymbol{\eta_1} + \frac{\eta_2}{\alpha_c^2}\boldsymbol{\beta}^\top \mathbf{X}^\top \mathbf{X}\boldsymbol{\beta}\right),\end{aligned}$$

where $\mathcal{I}_0$ is the index set corresponding to the 0 counts of $\mathbf{y}$, η_1 and η_2 are the coefficients used in the approximation of $\exp(z)$, and $\boldsymbol{\eta_1}$ is an N-vector with η_1 in each entry. We note that over multiple iterations of CCA (and indeed, over the updates over the rows of $\mathbf{L}$ and $\mathbf{F}$), $\mathbf{X}^\top\boldsymbol{\eta_1}$ and $\mathbf{X}^\top\mathbf{X}$ only need to be computed *once*. That is, for each outer loop of Algorithm 1, before updating $\mathbf{L}$ we can pre-compute $\mathbf{F}^\top\boldsymbol{\eta_1}$ and $\mathbf{F}^\top\mathbf{F}$, and before updating $\mathbf{F}$ we can pre-compute $\mathbf{L}^\top\boldsymbol{\eta_1}$ and $\mathbf{L}^\top\mathbf{L}$.

Appendix D. Additional details on visualization of log1pNMF fits

In addition to the invariance to re-scaling discussed in Section 4.1, the log1p NMF model is also invariant to re-ordering of the columns of $\mathbf{L}$ and $\mathbf{F}$. That is, if we swap columns k and m of both $\mathbf{L}$ and $\mathbf{F}$, the fitted Poisson rates λ_{ij} are unchanged, and therefore so is the log-likelihood of the model. This invariance can hinder comparison of log1p NMF fits with different values of c, as in the Results we assign each factor a distinct color when visualizing its scores (where we use the same color to column mapping across all fits). Whenever we display multiple fits of the same dataset together, we therefore match their factors before plotting. One fit is designated as the reference (in our analyses, the $c = 1$ fits for the MCF-7 and pancreas data, and the $c = 0.001$ fit for the BBC data), and the factors of every other fit are placed in one-to-one correspondence with the reference factors by maximizing the sum of Spearman correlations between the paired columns of $\mathbf{F}$. The optimal matching between the factors can be computed efficiently with the Hungarian algorithm (Kuhn, 1955).

Finally, log1p NMF fits are also invariant to re-ordering of the rows of $\mathbf{L}$ (so long as the same re-ordering is done to the rows of $\mathbf{Y}$), but this ordering can substantially impact visualizations. For each dataset, we therefore first re-order the rows of $\mathbf{L}$ according to a categorization given to the corresponding row of $\mathbf{Y}$ by the original authors (for genomics applications, this is typically cell type or treatment, for text applications this is document class). Then, within each category, we re-order the rows of $\mathbf{L}$ using the function `structure_plot` in the `R` package `fastTopics` (Carbonetto et al., 2021), which computes a one-dimensional t-SNE embedding (van der Maaten and Hinton, 2008) of the rows of $\mathbf{L}$ and then sorts the rows within each category by their position along this embedding. Generally, this results in samples with similar values of $\mathbf{l}_{i\cdot}$ being placed next to one another. Note that because this ordering is recomputed separately for each fit, when comparing fits across values of c the precise ordering of the rows of $\mathbf{L}$ may not be the same.

Appendix E. Reproducibility of results

An `R` package with associated code to fit log1p NMF on our real data examples and reproduce our figures can be found at `https://github.com/eweine/log1pNMF`.

Appendix F. Benchmarking the approximate log-likelihood approach on real data

In Section 3.3 we assessed the accuracy of the approximate log-likelihood in simulations. Here, we compare the exact and approximate approaches on real data in terms of the quality of fit achieved per unit of wall-clock time. Specifically, we fit the log1p NMF model with

$c = 1$ to three datasets: the pancreas dataset of Section 5.2.2 ($K = 13$), the BBC dataset of Section 5.2.3 ($K = 10$), and — to provide an example at the scale of modern single-cell atlases — the mouse organogenesis cell atlas (MOCA) of Cao et al. (2019), a dataset of approximately 1.33 million cells with 26,182 genes measured ($K = 25$). The MOCA dataset is also extremely sparse, with approximately 2.6% of entries greater than 0.

For each dataset we fit the model twice: once optimizing the exact log-likelihood (12) and once optimizing the approximate log-likelihood (15), with the polynomial coefficients set by the Chebyshev approximation. Each fit was initialized with the rank-1 procedure of Section 3.5, carried out under the fit's own objective. Each fit was then allowed up to 10 hours of optimization time on 64 threads, stopping earlier if the absolute change in its objective fell below 10^{-8}. Throughout each fit we recorded the wall-clock time and the exact log-likelihood of the current iterate. Figure F.1 shows, for each dataset, the gap between this exact log-likelihood and the best exact log-likelihood attained by either approach on that dataset.

The benefits of the approximate approach are most apparent for the MOCA dataset, which is by far the largest dataset under consideration (and is quite sparse). Interestingly, on the BBC dataset, the approximate approach converges to a better exact log-likelihood than directly optimizing the exact log-likelihood. This presumably results from the exact approach getting stuck in a local optimum on this dataset, which is unsurprising given the non-convex nature of the optimization problem. Finally, on the pancreas dataset, the exact approach clearly beats the approximate approach. While the approximate approach appears to make faster initial progress (as it is able to complete far more iterations per unit time), eventually the exact approach exceeded the log-likelihood of the approximate approach and continued to improve. Given enough time (and, save for convergence to local optima), the exact approach will *eventually* achieve a higher exact log-likelihood than the approximate approach. However, in practice, especially with extremely large datasets, fitting NMF models to convergence (and in particular log1p NMF models) will take an extremely long time, and thus running the approximate approach some fixed number of iterations will often lead to reasonable results in a much shorter period of time.

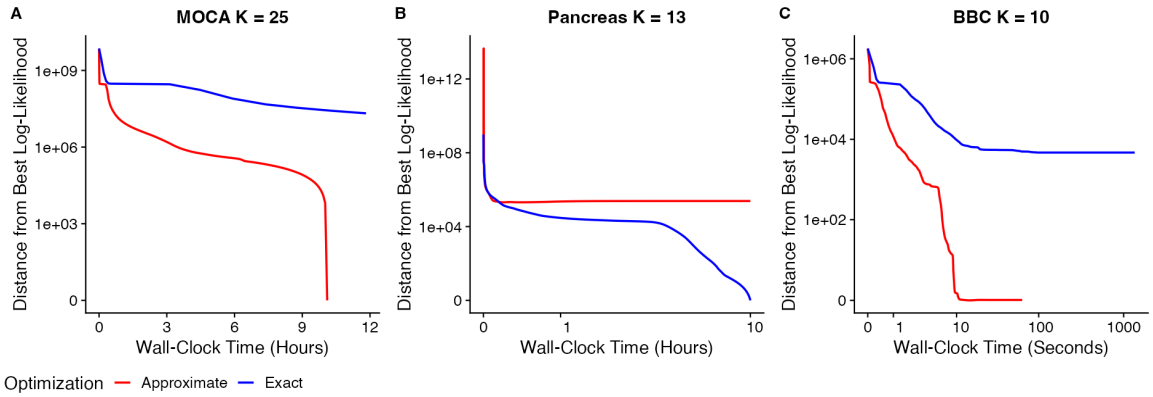


Figure F.1: Progress of the exact and approximate (Chebyshev) optimization approaches on three real datasets, fitting log1p NMF with $c = 1$ from the rank-1 initialization. Each panel shows, as a function of wall-clock time (including initialization time), the distance between the exact log-likelihood of the current iterate and the best exact log-likelihood achieved by either approach on that dataset; a curve reaching 0 attained the best solution found. Distances are plotted on a $\log(1+x)$ scale, and the time axes of panels (B) and (C) are on the same scale, so that 0 (the initialization) remains representable. (A) MOCA. (B) Pancreas. (C) BBC.

Appendix G. Comparing log1p NMF to Frobenius NMF on the log1p transformed counts

In the main text, we introduced an approximate log-likelihood that is much faster to compute than the exact likelihood of the log1p NMF model and was shown to be reasonably accurate in simulations. Here, we compare the results of using this approximation as opposed to the exact log-likelihood for fitting the log1p model to the pancreas dataset with $c = 1$. In addition, we compare these approaches to fitting Frobenius NMF to the log1p transformed count data (i.e., equations (10)-(11) with $c = 1$).

All models were fit from a common initialization, constructed by first fitting the log1p model with the exact log-likelihood and $K = 1$, and then running a single iteration of the $K = 13$ model (again with the exact log-likelihood) initialized from that rank-1 fit. This was done in order to encourage the factors of the different models to be comparable.

The cell scores for each of these three approaches are shown in Figure G.1 and Figure G.2. Generally speaking, all three models discover relatively similar "cell type associated" structure. However, interestingly, only the two approximate methods appear to fit factors that model a spectrum between endothelial and mesenchymal cells. However, the cost of this factor appears to be a poor representation of the acinar cells, which do not appear to be represented very cleanly in either approximation approach.

The "treatment associated" cell scores are extremely similar between the model fit with the exact log-likelihood (Figure G.2A) and the model fit with the quadratic approximation to the log-likelihood (Figure G.2B). Specifically, both models have 1 factor each that represents the Untreated, IL-1β, and IFNγ treatments, respectively, and use a combination of these factors to represent the combination of treatments. However, the results of fitting Frobenius NMF to the log1p transformed counts are not as parsimonious. While this model does appear to identify one factor each associated with IL-1β and IFNγ treatment (k10 and k4, respectively), the combination group is represented as a combination of k4 and an entirely separate factor k13. We additionally fit Frobenius NMF to the log1p transformed counts with $K = 12$ and $K = 11$ using the same initialization strategy as above (data not shown), and these models similarly identified a unique factor to the combination treatment group.

Overall, both approximation approaches appear qualitatively reasonably similar for this setting of c on this particular dataset. However, the quadratic approximation approach is clearly more similar to the exact log-likelihood approach, and in particular is more parsimonious in its representation of the combined treatment group than fitting Frobenius NMF to the log1p transformed counts.

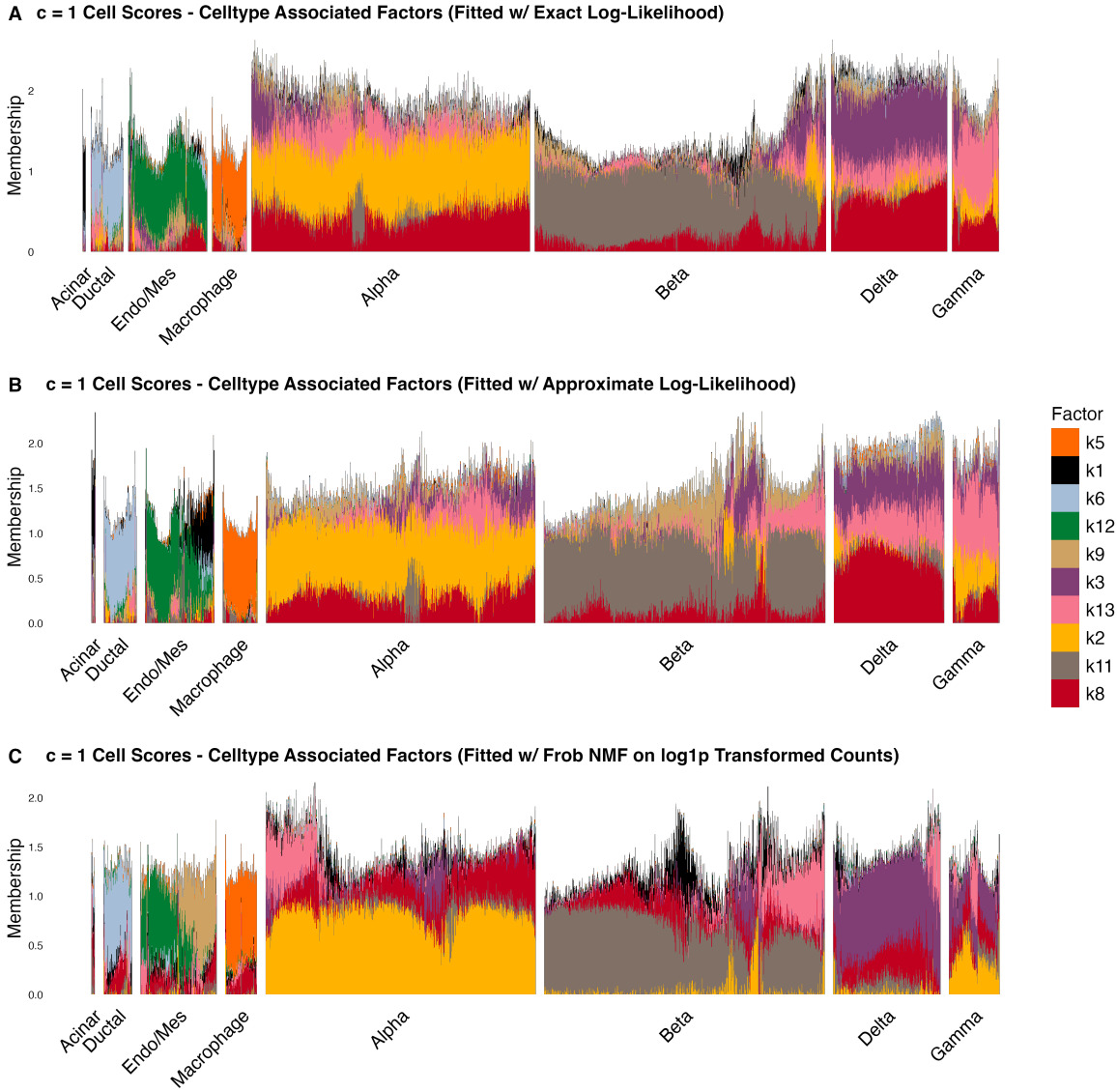


Figure G.1: Celltype associated factors for the log1p NMF model fit by maximizing the exact log-likelihood (A), as well as with the Chebyshev approximation approach (B), and by fitting Frobenius NMF to the log1p transformed counts (C).

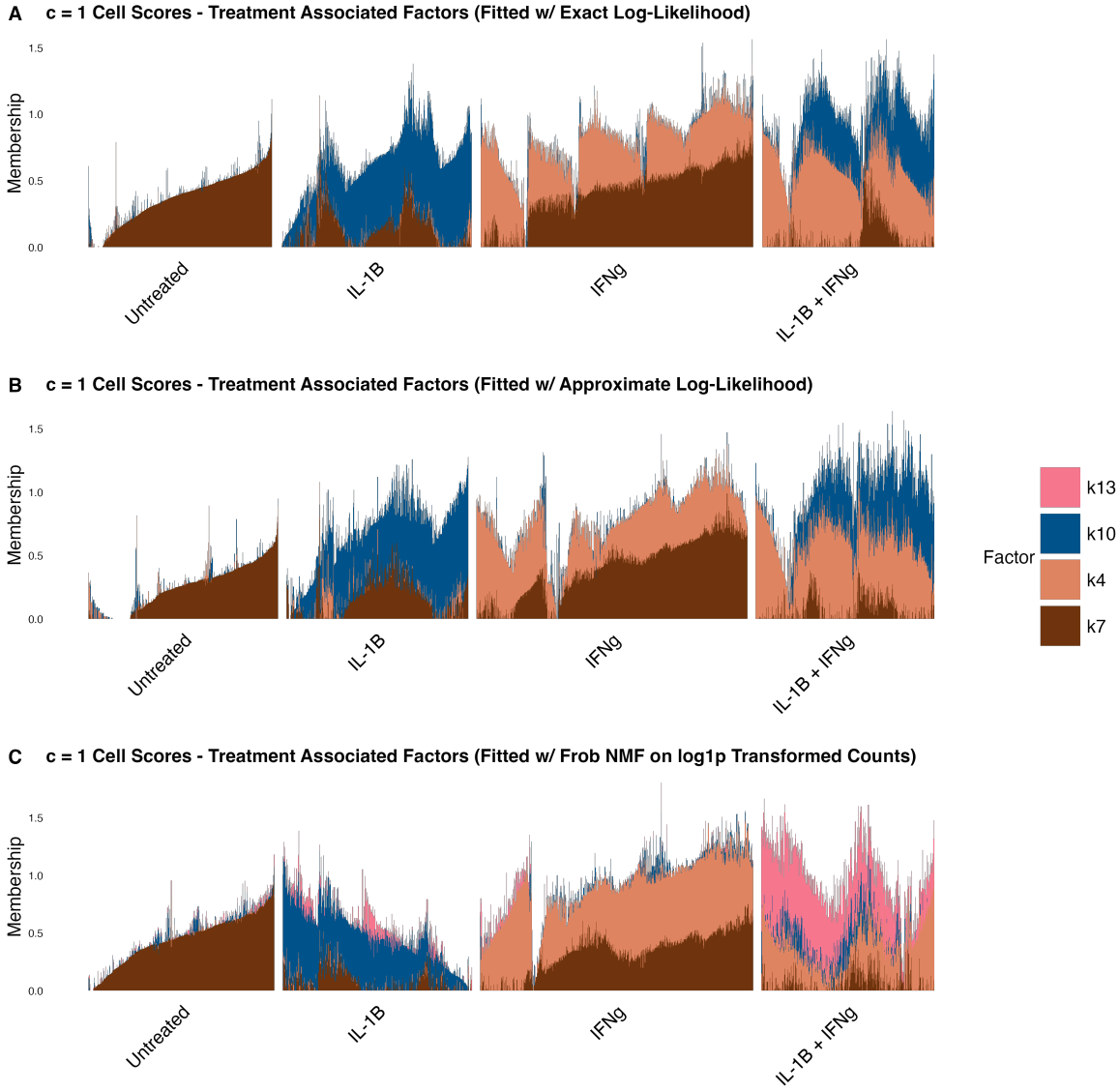


Figure G.2: Treatment associated factors for the log1p NMF model fit by maximizing the exact log-likelihood (A), as well as with the Chebyshev approximation approach (B), and by fitting Frobenius NMF to the log1p transformed counts (C).

Appendix H. Hoyer Sparsity Results

Lemma 4 *Let* $\mathbf{x}$ *be a non-negative* n*-vector (for* $n \geq 2$*) with at least one non-zero entry, and define* $\mathrm{pm}(\mathbf{x})$ *as in* (18)*. Then,*

$$\frac{\|\mathbf{x}\|_1}{\|\mathbf{x}\|_2} \geq \sqrt{\frac{n}{\mathrm{pm}(\mathbf{x})}}.$$

Proof Because $\|\mathbf{x}\|_2^2 \leq \|\mathbf{x}\|_\infty \|\mathbf{x}\|_1$ and $\frac{\|\mathbf{x}\|_1}{\|\mathbf{x}\|_\infty} = n\frac{\mathrm{mean}(\mathbf{x})}{\max(\mathbf{x})} = \frac{n}{\mathrm{pm}(\mathbf{x})}$, we can write

$$\frac{\|\mathbf{x}\|_1}{\|\mathbf{x}\|_2} \;\geq\; \frac{\|\mathbf{x}\|_1}{\sqrt{\|\mathbf{x}\|_\infty \|\mathbf{x}\|_1}} = \sqrt{\frac{\|\mathbf{x}\|_1}{\|\mathbf{x}\|_\infty}} = \sqrt{\frac{n}{\mathrm{pm}(\mathbf{x})}}.$$

■

Lemma 5 *Let* $\mathbf{L} \in \mathbb{R}_{\geq 0}^{n \times K}$ *and* $\mathbf{F} \in \mathbb{R}_{\geq 0}^{p \times K}$*, where* $\mathbf{B} = \mathbf{L}\mathbf{F}^\top$*. Let* $\mathbf{l}_{\cdot k}$ *and* $\mathbf{f}_{\cdot k}$ *denote columns* k *of* $\mathbf{L}$ *and* $\mathbf{F}$*, respectively, and suppose that every column of* $\mathbf{L}$ *and of* $\mathbf{F}$ *has at least one non-zero entry. Further, denote*

$$w_k = \frac{\mathrm{mean}(\mathbf{l}_{\cdot k})\,\mathrm{mean}(\mathbf{f}_{\cdot k})}{\mathrm{mean}(\mathbf{B})}.$$

Then,

$$\mathrm{pm}(\mathbf{l}_{\cdot k})\,\mathrm{pm}(\mathbf{f}_{\cdot k}) w_k \leq \mathrm{pm}(\mathbf{B}).$$

Proof First, let i^* be the smallest $i \in \{1, \ldots, n\}$ such that $l_{i^*k} = \max(\mathbf{l}_{\cdot k})$, and similarly let j^* be the smallest $j \in \{1, \ldots, p\}$ such that $f_{j^*k} = \max(\mathbf{f}_{\cdot k})$. Then, we have

$$\begin{aligned}
\max(\mathbf{B}) &\geq b_{i^*j^*} \\
&= \sum_{m=1}^{K} l_{i^*m} f_{j^*m} \\
&\geq l_{i^*k} f_{j^*k} \\
&= \max(\mathbf{l}_{\cdot k})\max(\mathbf{f}_{\cdot k}).
\end{aligned}$$

Then, we have

$$\begin{aligned}
\mathrm{pm}(\mathbf{B}) &= \frac{\max(\mathbf{B})}{\mathrm{mean}(\mathbf{B})} \\
&\geq \frac{\max(\mathbf{l}_{\cdot k})\max(\mathbf{f}_{\cdot k})}{\mathrm{mean}(\mathbf{B})} \\
&= \frac{\max(\mathbf{l}_{\cdot k})\max(\mathbf{f}_{\cdot k})}{\mathrm{mean}(\mathbf{B})} \cdot \frac{\mathrm{mean}(\mathbf{l}_{\cdot k})\,\mathrm{mean}(\mathbf{f}_{\cdot k})}{\mathrm{mean}(\mathbf{l}_{\cdot k})\,\mathrm{mean}(\mathbf{f}_{\cdot k})} \\
&= \mathrm{pm}(\mathbf{l}_{\cdot k})\,\mathrm{pm}(\mathbf{f}_{\cdot k}) w_k.
\end{aligned}$$

■

Note that $\sum_{k=1}^{K} w_k = 1$ because

$$\begin{aligned}\text{mean}(\mathbf{B}) &= \frac{1}{n}\frac{1}{p}\sum_{i=1}^{n}\sum_{j=1}^{p}\sum_{k=1}^{K} l_{ik}f_{jk} \\ &= \sum_{k=1}^{K}\left(\frac{1}{n}\sum_{i=1}^{n} l_{ik}\right)\left(\frac{1}{p}\sum_{j=1}^{p} f_{jk}\right) \\ &= \sum_{k=1}^{K}\text{mean}(\mathbf{l}_{\cdot k})\,\text{mean}(\mathbf{f}_{\cdot k}).\end{aligned}$$

While Lemma 5 bounds the pm of $\mathbf{l}_{\cdot k}$ and $\mathbf{f}_{\cdot k}$, our ultimate goal is to bound the sparsity of these objects, which we do with the following Lemma and Theorem.

Lemma 6 *For a non-negative n-vector $\mathbf{x}$ with at least one non-zero entry, define*

$$d(\mathbf{x}) = \frac{\|\mathbf{x}\|_1}{\sqrt{n}\|\mathbf{x}\|_2}.$$

Then, under the conditions of Lemma 5, for each $k \in \{1, \ldots, K\}$,

$$d(\mathbf{l}_{\cdot k})\, d(\mathbf{f}_{\cdot k}) \geq \sqrt{\frac{w_k}{\text{pm}(\mathbf{B})}}.$$

Proof Starting from Lemma 4, we have

$$\begin{aligned} \frac{\|\mathbf{x}\|_1}{\|\mathbf{x}\|_2} &\geq \sqrt{\frac{n}{\text{pm}(\mathbf{x})}} \\ \Rightarrow \quad \frac{\|\mathbf{x}\|_1}{\sqrt{n}\|\mathbf{x}\|_2} &\geq \frac{1}{\sqrt{\text{pm}(\mathbf{x})}} \\ \Rightarrow \quad d(\mathbf{x}) &\geq \frac{1}{\sqrt{\text{pm}(\mathbf{x})}} \end{aligned}$$

Re-arranging the final line to $\frac{1}{d(\mathbf{x})^2} \leq \text{pm}(\mathbf{x})$ and applying this to both $\mathbf{l}_{\cdot k}$ and $\mathbf{f}_{\cdot k}$, and then using Lemma 5, we have

$$\frac{1}{d(\mathbf{l}_{\cdot k})^2 d(\mathbf{f}_{\cdot k})^2} \leq \text{pm}(\mathbf{l}_{\cdot k})\,\text{pm}(\mathbf{f}_{\cdot k}) \leq \frac{\text{pm}(\mathbf{B})}{w_k}.$$

Taking reciprocals and then square roots gives the result. ■

Theorem 7 *Under the conditions of Lemma 5, and writing $\kappa_m = \frac{\sqrt{m}}{\sqrt{m}-1}$,*

$$H(\mathbf{l}_{\cdot k}) + H(\mathbf{f}_{\cdot k}) \leq \kappa_n + \kappa_p - 2\sqrt{\kappa_n \kappa_p}\left(\frac{w_k}{\text{pm}(\mathbf{B})}\right)^{1/4}.$$

Proof First, note that for a non-negative m-vector (for $m \geq 2$) $\mathbf{x}$ with at least one non-zero entry, we have

$$\begin{aligned}\kappa_m(1 - d(\mathbf{x})) &= \frac{\sqrt{m}}{\sqrt{m}-1} - \frac{\sqrt{m}}{\sqrt{m}-1}\frac{\|\mathbf{x}\|_1}{\sqrt{m}\|\mathbf{x}\|_2} \\ &= \frac{\sqrt{m} - \|\mathbf{x}\|_1/\|\mathbf{x}\|_2}{\sqrt{m}-1} \\ &= H(\mathbf{x})\end{aligned}$$

Applying this fact to $\mathbf{l}_{\cdot k}$ and $\mathbf{f}_{\cdot k}$, we have

$$\begin{aligned}H(\mathbf{l}_{\cdot k}) + H(\mathbf{f}_{\cdot k}) &= \kappa_n(1 - d(\mathbf{l}_{\cdot k})) + \kappa_p(1 - d(\mathbf{f}_{\cdot k})) \\ &= \kappa_n + \kappa_p - (\kappa_n d(\mathbf{l}_{\cdot k}) + \kappa_p d(\mathbf{f}_{\cdot k})) \\ &\leq \kappa_n + \kappa_p - 2\sqrt{\kappa_n\kappa_p\, d(\mathbf{l}_{\cdot k})d(\mathbf{f}_{\cdot k})} \qquad \text{(AM-GM inequality)} \\ &\leq \kappa_n + \kappa_p - 2\sqrt{\kappa_n\kappa_p}\left(\frac{w_k}{\mathrm{pm}(\mathbf{B})}\right)^{1/4} \qquad \text{(Lemma 6).}\end{aligned}$$

■

For sufficiently large n and p we have $\kappa_n, \kappa_p \approx 1$, in which case the bound is approximately:

$$H(\mathbf{l}_{\cdot k}) + H(\mathbf{f}_{\cdot k}) \leq 2\left(1 - \left(\frac{w_k}{\mathrm{pm}(\mathbf{B})}\right)^{1/4}\right).$$

Finally, we relate these bounds back to c with the following Proposition.

Proposition 8 *Let $\mathbf{\Lambda} \in \mathbb{R}^{n\times p}_{\geq 0}$ be a fixed matrix of rates with at least one positive entry, and for $c \in (0, \infty]$ define $\mathbf{B}_c$ entrywise by $b_{ij}(c) = g(\lambda_{ij}; c)$, with $\mathbf{B}_\infty = \mathbf{\Lambda}$. Then, for any $0 < c_1 < c_2 \leq \infty$,*

$$\mathrm{pm}(\mathbf{B}_{c_1}) \leq \mathrm{pm}(\mathbf{B}_{c_2}).$$

Proof Since pm is invariant to multiplying its argument by a positive constant, we may drop the factor α_c from (5) and work with $\theta_{ij}(c) = \log(1 + \lambda_{ij}/c)$.

The proof has two steps. First, we show that $\boldsymbol{\theta}(c_1)$ is obtained from $\boldsymbol{\theta}(c_2)$ by applying, entrywise, an increasing concave function φ with $\varphi(0) = 0$. Second, we show that applying such a function can never increase pm.

For the first step, suppose first that $c_2 < \infty$, and let $a = c_2/c_1 > 1$. Then, defining

$$\varphi(t) = \log\left(1 + a(e^t - 1)\right)$$

for $t \geq 0$, we have

$$\varphi(\theta_{ij}(c_2)) = \log\left(1 + \frac{c_2}{c_1}(1 + \lambda_{ij}/c_2 - 1)\right) = \log(1 + \lambda_{ij}/c_1) = \theta_{ij}(c_1).$$

Clearly $\varphi(0) = 0$, and differentiating, we have

$$\varphi'(t) = \frac{ae^t}{1 + a(e^t - 1)} > 0, \qquad \varphi''(t) = \frac{ae^t(1-a)}{(1 + a(e^t - 1))^2} \leq 0,$$

where the sign of φ'' follows from $a > 1$. If instead $c_2 = \infty$, then $\theta_{ij}(c_2) = \lambda_{ij}$ and the same conclusions hold for $\varphi(t) = \log(1 + t/c_1)$.

For the second step, we use the fact that φ is concave and $\varphi(0) = 0$, and thus the graph of φ lies on or above the line connecting the origin to any point on the graph. Precisely, let $M = \max_{ij} \theta_{ij}(c_2)$, which is positive because at least one rate is positive. Any $t \in [0, M]$ can be written as the convex combination $t = \frac{t}{M} \cdot M + \left(1 - \frac{t}{M}\right) \cdot 0$, so concavity gives

$$\varphi(t) \;\geq\; \frac{t}{M}\,\varphi(M) + \left(1 - \frac{t}{M}\right)\varphi(0) \;=\; \frac{t}{M}\,\varphi(M).$$

Since every entry of $\boldsymbol{\theta}(c_2)$ lies in $[0, M]$, we can apply this bound with $t = \theta_{ij}(c_2)$ for every (i, j), which gives

$$\theta_{ij}(c_1) = \varphi\left(\theta_{ij}(c_2)\right) \;\geq\; \frac{\varphi(M)}{M}\,\theta_{ij}(c_2),$$

and averaging over all np entries,

$$\operatorname{mean}\left(\boldsymbol{\theta}(c_1)\right) \;\geq\; \frac{\varphi(M)}{M}\,\operatorname{mean}\left(\boldsymbol{\theta}(c_2)\right).$$

Finally, because φ is increasing, the largest entry of $\boldsymbol{\theta}(c_1)$ is the image of the largest entry of $\boldsymbol{\theta}(c_2)$ under φ, so $\max\left(\boldsymbol{\theta}(c_1)\right) = \varphi(M)$. That is, passing from $\boldsymbol{\theta}(c_2)$ to $\boldsymbol{\theta}(c_1)$ shrinks the maximum by exactly the factor $\varphi(M)/M$, while the mean shrinks by at most that factor. Combining, we have

$$\operatorname{pm}\left(\boldsymbol{\theta}(c_1)\right) = \frac{\varphi(M)}{\operatorname{mean}\left(\boldsymbol{\theta}(c_1)\right)} \;\leq\; \frac{\varphi(M)}{\frac{\varphi(M)}{M}\operatorname{mean}\left(\boldsymbol{\theta}(c_2)\right)} = \frac{M}{\operatorname{mean}\left(\boldsymbol{\theta}(c_2)\right)} = \operatorname{pm}\left(\boldsymbol{\theta}(c_2)\right).$$

■

Appendix I. Calibration of simulation parameters to the pancreas data

Here we describe how the constants $\left(\pi_0^{(c_{\text{true}})}, \mu^{(c_{\text{true}})}, \sigma^{(c_{\text{true}})}, T_{\text{low}}^{(c_{\text{true}})}, T_{\text{high}}^{(c_{\text{true}})}\right)$ of the simulation in Section 5.1.2 were chosen.

We set $T_{\text{low}}^{(c_{\text{true}})} = \log(1 + 0.05/c_{\text{true}})$ (and $T_{\text{low}}^{(\infty)} = 0.05$). The remaining parameters were set by matching (as closely as possible) the simulated dataset to two properties of the pancreas dataset analyzed in Section 5.2.2. Specifically, we set the parameters so that simulated data had the same mean count as the pancreas dataset (≈ 0.674) and approximately the same peak to mean curve

$$c \;\mapsto\; \operatorname{pm}\left(\log(1 + \mathbf{Y}/c)\right)$$

as the pancreas dataset.

To do this, we minimize the sum of the squared distances between the real curve $c \mapsto \operatorname{pm}\left(\log(1 + \mathbf{Y}/c)\right)$ for the pancreas dataset and the expected pm curve, evaluated at the

points $c = (10^{-4}, \ldots, 10^4)$. Precisely, each c_{true} is calibrated separately (we suppress the superscripts below), and we minimize

$$\ell(\pi_0, \sigma, T_{\text{high}}) = \sum_{c \in \{10^{-4}, \ldots, 10^4\}} \left[\log \text{pm}\left(\log(1 + \mathbf{Y}^*/c)\right) - \log \widehat{\text{pm}}_c\left(\pi_0, \sigma, T_{\text{high}}\right)\right]^2,$$

where $\widehat{\text{pm}}_c$ is the value of $\text{pm}\left(\log(1 + \mathbf{Y}_{\text{sim}}/c)\right)$ averaged over three simulated datasets $\mathbf{Y}_{\text{sim}}$. Because a 500×500 simulated matrix cannot be expected to attain the maximum count of the much larger pancreas matrix, $\mathbf{Y}^*$ denotes the pancreas matrix with its maximum count replaced by $t^* = 8{,}820$, the median of the maximum of $n \cdot p = 250{,}000$ draws from the empirical distribution of the pancreas counts.

We minimize ℓ over $(\pi_0, \sigma, T_{\text{high}})$ using Nelder–Mead. The parameter μ is not included in this search; instead, it is *profiled*: at every evaluation of ℓ, given the current $(\pi_0, \sigma, T_{\text{high}})$, we set μ to the value that makes the expected mean count of the simulated data equal to the mean count of the pancreas dataset, found by one-dimensional root-finding. The calibrated values are shown below.

c_{true}	π_0	μ	σ	T_{high}	T_{low}
10^{-3}	0.959	2.196	0.099	15.57	3.93
1	0.838	−0.963	0.665	7.82	0.049
∞	0.901	−2.418	1.467	911.22	0.05

Appendix J. The geometry of log1p NMF for $c \to 0^+$

Let $\boldsymbol{\lambda}_i = (\lambda_{i1}, \ldots, \lambda_{ip})$ denote the vector of mean values for the ith row of $\mathbf{Y}$. In standard Poisson NMF ($c \to \infty$) the range of possible values of $\boldsymbol{\lambda}_i$ has a simple geometric relationship with the factors (columns of $\mathbf{F}$): it is simply the conical hull of the columns of $\mathbf{F}$ (i.e., the set of non-negative linear combinations of the columns). The implications of this simple geometry for identifiability of NMF have been well studied (e.g., Donoho and Stodden (2003)). Here we provide preliminary results on the corresponding geometry for the log1p NMF model in the case $c \to 0^+$.

Since the range of possible values of the vector $\boldsymbol{\lambda}_i$ is the same for each i we drop the subscript i, and study the achievable values of $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p)$ where

$$\alpha_c \log(1 + \lambda_j/c) = \mathbf{l} \cdot \mathbf{f}_{j\cdot} = \eta_j \text{ say.} \tag{29}$$

The question we aim to answer is: what are the achievable values of $\boldsymbol{\lambda}$ when $\mathbf{l} \in \mathbb{R}^K_{\geq 0}$ as $c \to 0^+$? And, how does this achievable set compare to the achievable set under standard Poisson NMF?

To simplify matters, in this section we impose the constraint $\sum_j \lambda_j = 1$ on model (29). This normalization allows us to study the geometry of the achievable *directions* of $\boldsymbol{\lambda}$. Note that these directions are invariant to the scaling factor; the set of achievable directions would be identical if we imposed $\sum_j \lambda_j = A$ for any $A > 0$. Consequently, the set of achievable values of $\boldsymbol{\lambda}$ (in the limit $c \to 0^+$) forms a cone generated by the achievable directions (the set $\mathcal{L}_{0^+}(\mathbf{F})$ defined below).

Before characterizing the achievable directions of $\boldsymbol{\lambda}$ in the log1p NMF model as $c \to 0^+$, it is useful to first understand the asymptotic behavior of the link function under the sum to 1 constraint. Let $\Delta^{p-1} = \{\boldsymbol{\lambda} \geq 0 : \sum_j \lambda_j = 1\}$ denote the probability simplex in $\mathbb{R}^p$, and let $\Delta^{p-1}_{>0} = \{\boldsymbol{\lambda} \in \Delta^{p-1} : \lambda_j > 0 \text{ for all } j\}$ denote its strictly positive elements. Recalling the softmax function $\sigma : \mathbb{R}^p \to \Delta^{p-1}$ defined component-wise as

$$\sigma(\mathbf{x})_i = \frac{\exp(x_i)}{\sum_{j=1}^p \exp(x_j)},$$

we have the following lemma:

Lemma 9 *Consider the vector $\boldsymbol{\lambda}$ defined in (29) with the constraint $\sum_{j=1}^p \lambda_j = 1$. For $c \in (0,1)$ (so that $\alpha_c = 1$), let $\eta_{\max} := \max_j \eta_j$. As $c \to 0^+$, the following properties hold:*

1. *$\eta_{\max} \to \infty$.*
2. *The vector $\boldsymbol{\lambda}$ converges to $\sigma(\boldsymbol{\eta})$ with the following bound:*

$$\lambda_j = \sigma(\boldsymbol{\eta})_j + \mathcal{O}(pe^{-\eta_{\max}}).$$

This lemma allows us to characterize the achievable directions of $\boldsymbol{\lambda}$ as $c \to 0^+$ in terms of the softmax function σ. By analogy to standard Poisson NMF, one might suspect that the achievable directions of $\boldsymbol{\lambda}$ are simply the image of the conical hull of the columns of $\mathbf{F}$ under σ. This turns out to be true for the *strictly positive* achievable directions, *provided that $\mathbf{F}$ contains the constant vector $\mathbf{1}$ in the conical hull of its columns.*

Before describing this more precisely, we introduce additional definitions and notation. First, to represent the cone generated by the columns of $\mathbf{F}$, we write $\text{cone}(\mathbf{F}) := \{\mathbf{F}\mathbf{v} : \mathbf{v} \geq 0\}$. Second, we introduce notation for the range of achievable *directions* of $\boldsymbol{\lambda}$, viewed as a function of $\mathbf{F}$. For fixed c, let

$$\Lambda_c(\mathbf{F}) := \Big\{\boldsymbol{\lambda} \in \Delta^{p-1} \ : \ \alpha_c \log(1 + \boldsymbol{\lambda}/c) \in \text{cone}(\mathbf{F})\Big\},$$

where the logarithm is applied entrywise. Then, for $c' \in [0, \infty]$, define

$$\mathcal{L}_{c'}(\mathbf{F}) := \Big\{\boldsymbol{\lambda}^* \in \Delta^{p-1} \ : \ \boldsymbol{\lambda}(t) \to \boldsymbol{\lambda}^* \text{ for some sequences } c_t \to c' \text{ and } \boldsymbol{\lambda}(t) \in \Lambda_{c_t}(\mathbf{F})\Big\},$$

the set of achievable limiting directions of $\boldsymbol{\lambda}$ as $c \to c'$. We focus here on the two extreme cases $\mathcal{L}_{0^+}(\mathbf{F})$ and $\mathcal{L}_\infty(\mathbf{F})$.

We now state our main result:

Theorem 10 *Let $\mathbf{F} \in \mathbb{R}^{p \times K}_{\geq 0}$ where each column has at least 1 non-zero entry. There exists a $\boldsymbol{\lambda}^* \in \mathcal{L}_{0^+}(\mathbf{F})$ with all non-zero entries if and only if $\mathbf{1} \in \text{cone}(\mathbf{F})$. When $\mathbf{F}$ meets this condition, the set of achievable directions of $\boldsymbol{\lambda}$ where all entries are non-zero, $\mathcal{L}_{0^+}(\mathbf{F}) \cap \Delta^{p-1}_{>0}$, is*

$$\sigma(\bar{\mathcal{C}}) := \{\sigma(\mathbf{x}) : \mathbf{x} \in \bar{\mathcal{C}}\}, \tag{30}$$

where $\bar{\mathcal{C}} = \text{cone}(\mathbf{g}_1, \ldots, \mathbf{g}_K)$ is the cone of the centered factors $\mathbf{g}_k = \mathbf{f}_{\cdot k} - \Big(\frac{1}{p}\sum_j f_{jk}\Big)\mathbf{1}$.

It is natural to ask, when $\mathbf{1} \in \text{cone}(\mathbf{F})$, what equation (30) looks like, and how its size compares to the achievable sets for choices of $\mathbf{F}$ in standard Poisson NMF. Unfortunately, the high-dimensional, curved sets of equation (30) are difficult to characterize simply. However, for the simple case where $K = 2$, equation (30) can be described completely as a one-dimensional curve. To see this, note that we can write

$$\mathbf{1} = a\mathbf{f}_{\cdot 1} + b\mathbf{f}_{\cdot 2}$$

for some $a, b \geq 0$. Averaging the coordinates of both sides gives $1 = a\bar{f}_1 + b\bar{f}_2$, where $\bar{f}_k = \frac{1}{p}\sum_j f_{jk}$; subtracting this identity from each coordinate of the equation above then gives $a\mathbf{g}_1 + b\mathbf{g}_2 = \mathbf{0}$, so the two centered factors point in opposite directions. The cone $\bar{\mathcal{C}}$ is therefore a ray from the origin (when exactly one factor is constant, i.e., an intercept), or a full line through the origin (when neither factor is constant, which forces $a, b > 0$). In either case its image $\sigma(\bar{\mathcal{C}})$ is a smooth curve in the simplex passing through the uniform vector $\sigma(\mathbf{0}) = \frac{1}{p}\mathbf{1}$. (In the degenerate case where both factors are constant, $\bar{\mathcal{C}} = \{\mathbf{0}\}$ and the curve collapses to the single uniform point.)

Similarly, for standard Poisson NMF when we impose the constraint $\boldsymbol{\lambda} \in \Delta^{p-1}$, the achievable set is a line segment, since for any loadings $\mathbf{v} \geq 0$ with $\mathbf{v} \neq \mathbf{0}$ we can write

$$\boldsymbol{\lambda} \;=\; \frac{\mathbf{F}\mathbf{v}}{\mathbf{1}^\top \mathbf{F}\mathbf{v}} \;=\; w\tilde{\mathbf{f}}_1 + (1-w)\tilde{\mathbf{f}}_2,$$

where $\tilde{\mathbf{f}}_k = \mathbf{f}_{\cdot k}/s_k$ is the kth column normalized by its sum $s_k = \sum_j f_{jk}$, and $w = v_1 s_1/(v_1 s_1 + v_2 s_2) \in [0, 1]$. As $\mathbf{v} \geq 0$ varies, w spans all of $[0, 1]$, so the achievable set is exactly the segment joining the two normalized columns $\tilde{\mathbf{f}}_1$ and $\tilde{\mathbf{f}}_2$. In the notation above, this segment is exactly $\mathcal{L}_\infty(\mathbf{F})$, since the link function converges to the identity as $c \to \infty$.

Thus, both achievable sets are curves when $K = 2$, and we can measure their size by computing the arclength. Note that the arclength computed from a parameterization of a curve agrees with the intrinsic size of the curve (its one-dimensional Hausdorff measure) whenever the parameterization is injective. Both parameterizations here are injective: the segment $w \mapsto w\tilde{\mathbf{f}}_1 + (1-w)\tilde{\mathbf{f}}_2$ is affine in w, and σ is injective on $\bar{\mathcal{C}}$, since $\sigma(\mathbf{x}) = \sigma(\mathbf{y})$ requires $\mathbf{x} - \mathbf{y}$ to be a constant vector, and every element of $\bar{\mathcal{C}}$ has mean 0. Thus, comparing arclengths is a genuine comparison of the sizes of the two achievable sets. This yields the following theorem:

Theorem 11 *For $K = 2$ and $p \geq 5$, there exists an $\mathbf{F} \in \mathbb{R}^{p\times 2}_{\geq 0}$ with $\mathbf{1} \in \text{cone}(\mathbf{F})$ such that*

$$\text{arclength}\left(\mathcal{L}_{0^+}(\mathbf{F})\right) > \text{arclength}\left(\mathcal{L}_\infty(\widetilde{\mathbf{F}})\right)$$

for any $\widetilde{\mathbf{F}} \in \mathbb{R}^{p\times 2}_{\geq 0}$.

We prove Theorems 10 and 11, as well as supporting Lemmas, in the section below.

J.1 Proofs

We first prove Lemma 9:

Proof Since $c \to 0^+$, we have $\alpha_c = 1$. Inverting the link function $\log(1 + \lambda_j/c) = \eta_j$ yields $\lambda_j = c(\exp(\eta_j) - 1)$. Summing over j and applying the constraint $\sum \lambda_j = 1$, we have

$$1 = \sum_{j=1}^{p} c(\exp(\eta_j) - 1) = c\left(\sum_{j=1}^{p} \exp(\eta_j) - p\right).$$

Rearranging for c gives:

$$c = \frac{1}{\sum_{j=1}^{p} \exp(\eta_j) - p}. \tag{31}$$

As $c \to 0^+$, the denominator in (31) must approach ∞. Since p is finite, this requires $\sum \exp(\eta_j) \to \infty$, which implies $\eta_{\max} \to \infty$.

To establish the softmax equivalence, we can substitute (31) back into the expression for λ_j and write

$$\lambda_j = \frac{\exp(\eta_j) - 1}{\sum_{j'=1}^{p}(\exp(\eta_{j'}) - 1)}.$$

Let $\sigma(\boldsymbol{\eta})_j = \frac{\exp(\eta_j)}{\sum_{j'=1}^{p} \exp(\eta_{j'})}$ denote the standard softmax function. We analyze the ratio

$$\begin{aligned}\frac{\lambda_j}{\sigma(\boldsymbol{\eta})_j} &= \frac{\exp(\eta_j) - 1}{\exp(\eta_j)} \cdot \frac{\sum_{j'=1}^{p} \exp(\eta_{j'})}{\sum_{j'=1}^{p} \exp(\eta_{j'}) - p} \\ &= (1 - e^{-\eta_j}) \cdot \left(1 - \frac{p}{\sum_{j'=1}^{p} \exp(\eta_{j'})}\right)^{-1}.\end{aligned}$$

Using the Taylor expansion $(1 - x)^{-1} = 1 + x + \mathcal{O}(x^2)$ for the second term, we have

$$\begin{aligned}\lambda_j &= \sigma(\boldsymbol{\eta})_j(1 - e^{-\eta_j})\left(1 + \frac{p}{\sum_{j'=1}^{p} \exp(\eta_{j'})} + \mathcal{O}\left(\frac{p^2}{(\sum_{j'=1}^{p} e^{\eta_{j'}})^2}\right)\right) \\ &= \sigma(\boldsymbol{\eta})_j + \sigma(\boldsymbol{\eta})_j\left(\frac{p}{\sum_{j'=1}^{p} \exp(\eta_{j'})} - e^{-\eta_j}\right) + \text{h.o.t.}\end{aligned}$$

The residual term is dominated by $\max(\frac{p\sigma(\boldsymbol{\eta})_j}{\sum_{j'} e^{\eta_{j'}}}, \sigma(\boldsymbol{\eta})_j e^{-\eta_j})$, where we note that $\sigma(\boldsymbol{\eta})_j e^{-\eta_j} = (\sum_{j'=1}^{p} e^{\eta_{j'}})^{-1}$. Since $\sum_{j'=1}^{p} \exp(\eta_{j'}) \geq \exp(\eta_{\max})$, the error is $\mathcal{O}(pe^{-\eta_{\max}})$. As $\eta_{\max} \to \infty$, $\boldsymbol{\lambda} \to \sigma(\boldsymbol{\eta})$. ■

Next, we state and prove an intermediate result that is used in the proof of Theorem 10.

Lemma 12 *Let $\mathbf{F} \in \mathbb{R}_{\geq 0}^{p \times K}$ where each column has at least 1 non-zero entry. For a unit vector $\mathbf{w} \in \mathbb{R}_{\geq 0}^{K}$, let*

$$m(\mathbf{w}) = \max_j \mathbf{w} \cdot \mathbf{f}_{j\cdot}, \qquad S(\mathbf{w}) = \{j : \mathbf{w} \cdot \mathbf{f}_{j\cdot} = m(\mathbf{w})\},$$

and define

$$p^*(\mathbf{F}) \;=\; \max_{\mathbf{w} \geq 0,\, \|\mathbf{w}\| = 1} |S(\mathbf{w})|.$$

For every $\boldsymbol{\lambda}^* \in \mathcal{L}_{0^+}(\mathbf{F})$, there exists some unit vector $\mathbf{w} \geq 0$ such that $S(\mathbf{w})$ contains the support of $\boldsymbol{\lambda}^*$. That is, $\boldsymbol{\lambda}^*_j = 0$ for all $j \notin S(\mathbf{w})$. Consequently $\boldsymbol{\lambda}^*$ has at most $p^*(\mathbf{F})$ non-zero entries.

Proof Fix some $\mathbf{F}$ and $\boldsymbol{\lambda}^* \in \mathcal{L}_{0^+}(\mathbf{F})$. By the definition of $\mathcal{L}_{0^+}(\mathbf{F})$, there exist sequences $c_t \to 0^+$ and $\boldsymbol{l}(t) \geq 0$ such that $\boldsymbol{\lambda}(t) \to \boldsymbol{\lambda}^*$ elementwise. For this sequence, let $\eta_j(t) = \boldsymbol{l}(t) \cdot \boldsymbol{f}_{j\cdot}$ and $\eta_{\max}(t) = \max_j \eta_j(t)$. By Lemma 9, $\eta_{\max}(t) \to \infty$, and since $\mathbf{F}$ is fixed, we must have $||\boldsymbol{l}(t)|| \to \infty$. Let $\boldsymbol{u}(t) = \frac{\boldsymbol{l}(t)}{||\boldsymbol{l}(t)||}$ be the sequence of directions of $\boldsymbol{l}(t)$. Since $||\boldsymbol{u}(t)|| = 1$ for all t, by Bolzano-Weierstrass there must exist some subsequence such that $\boldsymbol{u}(t) \to \mathbf{w}$ elementwise, where $\mathbf{w} \geq 0$ is a unit vector. Note that $\boldsymbol{\lambda}(t) \to \boldsymbol{\lambda}^*$ along this subsequence.

Recalling that $S(\mathbf{w})$ is the index set of rows of $\mathbf{F}$ with maximal inner product with $\mathbf{w}$, fix $j \in S(\mathbf{w})$ (which must exist because $\mathbf{F}$ has a finite number of rows) and $i \notin S(\mathbf{w})$ (which we may assume exists, since otherwise the statement is vacuously true). Then, we can write

$$\frac{\eta_j(t) - \eta_i(t)}{\|\boldsymbol{l}(t)\|} \;=\; \frac{\boldsymbol{l}(t)}{\|\boldsymbol{l}(t)\|} \cdot (\mathbf{f}_{j\cdot} - \mathbf{f}_{i\cdot}) \;\longrightarrow\; \mathbf{w} \cdot (\mathbf{f}_{j\cdot} - \mathbf{f}_{i\cdot}) \;>\; 0.$$

Then, since $\|\boldsymbol{l}(t)\| \to \infty$, we must have that $\eta_j(t) - \eta_i(t) \to \infty$. As a result, we can write

$$\sigma(\boldsymbol{\eta}(t))_i \;=\; \frac{e^{\eta_i(t)}}{\sum_{i'} e^{\eta_{i'}(t)}} \;\leq\; e^{\eta_i(t) - \eta_{\max}(t)} \;\leq\; e^{-(\eta_j(t) - \eta_i(t))} \;\longrightarrow\; 0,$$

and the additive error in Lemma 9 is $\mathcal{O}(p\, e^{-\eta_{\max}(t)}) \to 0$, so $\lambda_i(t) \to 0$.

Thus, we have shown the existence of a unit vector $\mathbf{w}$ such that the support of $\boldsymbol{\lambda}^*$ must be contained in $S(\mathbf{w})$. Thus, letting $\mathbf{w}^*$ be a maximizer of $|S(\mathbf{w})|$ over all unit vectors $\mathbf{w}$, the size of the support of $\boldsymbol{\lambda}^*$ must be at most $|S(\mathbf{w}^*)| = p^*(\mathbf{F})$. ■

We are now able to show Theorem 10:

Proof Before proving the if and only if statement, we first show the following: (a) Every $\boldsymbol{\lambda}^* \in \mathcal{L}_{0^+}(\mathbf{F})$ with $\boldsymbol{\lambda}^* > 0$ elementwise also lies in $\sigma(\bar{\mathcal{C}})$. (b) If $\mathbf{1} \in \text{cone}(\mathbf{F})$, then every element of $\sigma(\bar{\mathcal{C}})$ is an achievable limit.

Proof of (a): For convenience, define the centering function $\Pi(\boldsymbol{\eta}) = \boldsymbol{\eta} - (\frac{1}{p}\sum_j \eta_j)\mathbf{1}$. Note that $\sigma(\Pi(\boldsymbol{\eta})) = \sigma(\boldsymbol{\eta})$ since for each i,

$$\sigma(\Pi(\boldsymbol{\eta}))_i = \frac{\exp(\eta_i - 1/p \sum_j \eta_j)}{\sum_{i'} \exp(\eta_{i'} - 1/p \sum_j \eta_j)} = \frac{\exp(-1/p \sum_j \eta_j)\exp(\eta_i)}{\exp(-1/p \sum_j \eta_j)\sum_{i'} \exp(\eta_{i'})} = \sigma(\boldsymbol{\eta})_i. \tag{32}$$

Now fix $\boldsymbol{\lambda}^* \in \mathcal{L}_{0^+}(\mathbf{F})$ with $\boldsymbol{\lambda}^* > 0$ elementwise, achieved along sequences $c_t \to 0^+$ and $\boldsymbol{l}(t) \geq 0$, and write $\boldsymbol{\eta}(t) = \mathbf{F}\boldsymbol{l}(t) \in \text{cone}(\mathbf{F})$.

We first show that every pairwise difference $\eta_i(t) - \eta_j(t)$ converges to a finite limit. By Lemma 9, $\boldsymbol{\lambda}(t) = \sigma(\boldsymbol{\eta}(t)) + o(1)$, and since $\boldsymbol{\lambda}(t) \to \boldsymbol{\lambda}^*$ it follows that $\sigma(\boldsymbol{\eta}(t)) \to \boldsymbol{\lambda}^*$ as well. Because $\boldsymbol{\lambda}^* > 0$, for every pair i, j we have

$$e^{\eta_i(t) - \eta_j(t)} \;=\; \frac{\sigma(\boldsymbol{\eta}(t))_i}{\sigma(\boldsymbol{\eta}(t))_j} \;\longrightarrow\; \frac{\lambda^*_i}{\lambda^*_j} \in (0, \infty),$$

and taking logarithms gives $\eta_i(t) - \eta_j(t) \to \log(\lambda^*_i / \lambda^*_j)$.

Next, convergence of the pairwise differences implies convergence of the centered vector, because each coordinate of $\Pi(\boldsymbol{\eta}(t))$ is an average of pairwise differences:

$$\Pi(\boldsymbol{\eta}(t))_i \;=\; \frac{1}{p}\sum_{k=1}^{p}\big(\eta_i(t)-\eta_k(t)\big) \;\longrightarrow\; \frac{1}{p}\sum_{k=1}^{p}\log(\lambda_i^*/\lambda_k^*) \;=\; \Pi(\log\boldsymbol{\lambda}^*)_i,$$

where the logarithm is applied entrywise. Write $\mathbf{x}^* = \Pi(\log\boldsymbol{\lambda}^*)$ for this limit.

We now argue that $\mathbf{x}^* \in \bar{\mathcal{C}}$. Since Π is linear, the image of $\mathrm{cone}(\mathbf{F})$ under Π is the cone generated by the centered columns:

$$\Pi(\mathrm{cone}(\mathbf{F})) \;=\; \mathrm{cone}\big(\Pi(\mathbf{f}_{\cdot 1}),\ldots,\Pi(\mathbf{f}_{\cdot K})\big) \;=\; \mathrm{cone}(\mathbf{g}_1,\ldots,\mathbf{g}_K) \;=\; \bar{\mathcal{C}}.$$

In particular, $\Pi(\boldsymbol{\eta}(t)) \in \bar{\mathcal{C}}$ for every t. A finitely generated cone is a closed set, and a closed set contains the limit of any convergent sequence of its elements; therefore $\mathbf{x}^* \in \bar{\mathcal{C}}$.

Finally, combining the shift invariance (32) with the continuity of σ,

$$\boldsymbol{\lambda}^* \;=\; \lim_t \sigma(\boldsymbol{\eta}(t)) \;=\; \lim_t \sigma\big(\Pi(\boldsymbol{\eta}(t))\big) \;=\; \sigma(\mathbf{x}^*) \;\in\; \sigma(\bar{\mathcal{C}}),$$

which completes the proof of (a).

Proof of (b): Suppose $\mathbf{1} \in \mathrm{cone}(\mathbf{F})$, i.e., $\mathbf{1} = \mathbf{F}\mathbf{v}$ for some $\mathbf{v} \geq 0$. Fix some $\sigma(\mathbf{x}) \in \sigma(\bar{\mathcal{C}})$; since $\bar{\mathcal{C}} = \Pi(\mathrm{cone}(\mathbf{F}))$, we can write $\mathbf{x} = \Pi(\mathbf{F}\mathbf{u})$ for some $\mathbf{u} \geq 0$. Under the sequence of loadings

$$\boldsymbol{l}(t) \;=\; \mathbf{u} + t\mathbf{v},$$

we have,

$$\boldsymbol{\eta}(t) \;=\; \mathbf{F}\mathbf{u} + t\mathbf{1},$$

so $\Pi(\boldsymbol{\eta}(t)) = \Pi(\mathbf{F}\mathbf{u}) = \mathbf{x}$ for every t since centering removes the $t\mathbf{1}$ term. Note that $\eta_{\max}(t) = \max_j(\mathbf{F}\mathbf{u})_j + t \to \infty$, so by the identity (31) the value of c enforcing the constraint $\sum_j \lambda_j = 1$, namely

$$c_t \;=\; \frac{1}{\sum_{j=1}^{p}\exp(\eta_j(t)) - p},$$

is positive for all sufficiently large t and satisfies $c_t \to 0^+$. Hence $\boldsymbol{\lambda}(t) \in \Lambda_{c_t}(\mathbf{F})$ with $c_t \to 0^+$. Finally, Lemma 9 together with the shift invariance (32) gives

$$\boldsymbol{\lambda}(t) \;=\; \sigma(\boldsymbol{\eta}(t)) + \mathcal{O}\big(pe^{-\eta_{\max}(t)}\big) \;=\; \sigma\big(\Pi(\boldsymbol{\eta}(t))\big) + o(1) \;=\; \sigma(\mathbf{x}) + o(1) \;\longrightarrow\; \sigma(\mathbf{x}).$$

Therefore $\sigma(\mathbf{x}) \in \mathcal{L}_{0^+}(\mathbf{F})$, and since $\mathbf{x} \in \bar{\mathcal{C}}$ was arbitrary, every element of $\sigma(\bar{\mathcal{C}})$ is an achievable limit. This completes the proof of (b).

Thus, (a) shows that every strictly positive element of $\mathcal{L}_{0^+}(\mathbf{F})$ lies in $\sigma(\bar{\mathcal{C}})$, and (b) shows that, when $\mathbf{1} \in \mathrm{cone}(\mathbf{F})$, every element of $\sigma(\bar{\mathcal{C}})$ is achievable. Since every softmax value is strictly positive, $\sigma(\bar{\mathcal{C}}) \subseteq \Delta_{>0}^{p-1}$, and combining the two inclusions gives $\mathcal{L}_{0^+}(\mathbf{F}) \cap \Delta_{>0}^{p-1} = \sigma(\bar{\mathcal{C}})$ when $\mathbf{1} \in \mathrm{cone}(\mathbf{F})$.

Now, we can show the if and only if statement. First, suppose $\mathbf{1} \in \mathrm{cone}(\mathbf{F})$. Trivially $\mathbf{x} = \mathbf{0}$ is in $\bar{\mathcal{C}}$, and $\sigma(\mathbf{0}) = \frac{1}{p}\mathbf{1} > 0$, which establishes that a strictly positive achievable value of $\boldsymbol{\lambda}$ exists. Now, suppose there exists $\boldsymbol{\lambda}^* \in \mathcal{L}_{0^+}(\mathbf{F})$ with all non-zero entries. By Lemma

12, there exists some unit vector $\mathbf{w} \geq 0$ such that $S(\mathbf{w}) = \{1, \ldots, p\}$. For this to hold, it must be that $\mathbf{w} \cdot \mathbf{f}_{j\cdot} = m(\mathbf{w})$ for all $j \in \{1, \ldots, p\}$. Thus,

$$\mathbf{F}\mathbf{w} = m(\mathbf{w})\mathbf{1}. \tag{33}$$

Because no column of $\mathbf{F}$ is all 0, we must have $m(\mathbf{w}) > 0$. We can then divide both sides of equation (33) by $m(\mathbf{w})$ and write $\mathbf{F}(\mathbf{w}/m(\mathbf{w})) = \mathbf{1}$, which means that $\mathbf{1} \in \text{cone}(\mathbf{F})$. ■

Finally, we are able to show Theorem 11. First, observe that when $K = 2$,

$$\text{arclength}\left(\mathcal{L}_\infty(\mathbf{F})\right) \leq \sqrt{2} \quad \text{for every } \mathbf{F} \in \mathbb{R}^{p\times 2}_{\geq 0}.$$

To see this, recall that $\mathcal{L}_\infty(\mathbf{F})$ is the segment joining the normalized columns $\tilde{\mathbf{f}}_1, \tilde{\mathbf{f}}_2 \in \Delta^{p-1}$, so its arclength is $\|\tilde{\mathbf{f}}_1 - \tilde{\mathbf{f}}_2\|$. For any $\mathbf{x}, \mathbf{y} \in \Delta^{p-1}$, we have $\|\mathbf{x}\|^2 = \sum_j x_j^2 \leq \sum_j x_j = 1$ (since each $x_j \in [0, 1]$) and $\mathbf{x} \cdot \mathbf{y} \geq 0$, so

$$\|\mathbf{x} - \mathbf{y}\|^2 = \|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - 2\,\mathbf{x} \cdot \mathbf{y} \leq 2.$$

To show that, for all $p \geq 5$, there exists an $\mathbf{F} \in \mathbb{R}^{p\times 2}_{\geq 0}$ such that $\text{arclength}(\mathcal{L}_{0^+}(\mathbf{F})) > \sqrt{2}$, we first show that for $p \geq 2$, we can construct an $\mathbf{F}$ such that $\text{arclength}(\mathcal{L}_{0^+}(\mathbf{F}))$ gets arbitrarily close to

$$\psi(p) = \sum_{j=2}^{p} \frac{1}{\sqrt{j(j-1)}}.$$

Since $\psi(p)$ is monotonically increasing in p, and $\psi(5) \approx 1.6276 > \sqrt{2}$, Theorem 11 immediately follows. We formalize this in the following Lemma:

Lemma 13 *Fix $p \geq 2$. For every $\varepsilon > 0$, there exists an $\mathbf{F} \in \mathbb{R}^{p\times 2}_{\geq 0}$ with $\mathbf{1} \in \text{cone}(\mathbf{F})$ such that*

$$\text{arclength}\left(\mathcal{L}_{0^+}(\mathbf{F})\right) \;>\; \psi(p) - \varepsilon.$$

Proof Take the first column of $\mathbf{F}$ to be the intercept $\mathbf{1}$, so that $\mathbf{1} \in \text{cone}(\mathbf{F})$ automatically, and take the second column to be the vector $\mathbf{f}$ with entries

$$f_j \;=\; \sum_{k=j}^{p-1} M^k, \qquad j = 1, \ldots, p-1, \qquad f_p = 0$$

where $M > 1$ is a parameter. The entries of $\mathbf{f}$ are decreasing, with gaps $f_j - f_{j+1} = M^j$ that grow by a factor of M each time j increases.

Since the first column of $\mathbf{F}$ is constant, its centered version is $\mathbf{g}_1 = \mathbf{0}$, so $\bar{\mathcal{C}} = \{t\mathbf{g}_2 : t \geq 0\}$. By Theorem 10 together with the shift invariance of the softmax (equation (32)),

$$\{\sigma(t\mathbf{f}) : t \geq 0\} \;=\; \sigma(\bar{\mathcal{C}}) \;\subseteq\; \mathcal{L}_{0^+}(\mathbf{F}).$$

Since the curve $\boldsymbol{\lambda}(t) = \sigma(t\mathbf{f})$ is injectively parameterized, it therefore suffices to lower bound its arclength.

The result of our construction is that as t grows, the curve proceeds down a "staircase" of barycenters. Let $\mathbf{b}_j = \frac{1}{j}\sum_{i=1}^{j}\mathbf{e}_i$ denote the barycenter of the face of the simplex spanned by the first j coordinates, so that $\mathbf{b}_p$ is the uniform vector and $\mathbf{b}_1 = \mathbf{e}_1$. The curve starts at $\boldsymbol{\lambda}(0) = \mathbf{b}_p$ and passes near $\mathbf{b}_{p-1}, \mathbf{b}_{p-2}, \ldots, \mathbf{b}_1$ in order. To make this precise, for $j = 1, \ldots, p-1$, let $t_j = M^{1/2-j}$, and note that $0 < t_{p-1} < t_{p-2} < \cdots < t_1$. At t_j, for any $i \le j$,

$$0 \;\le\; t_j(f_i - f_j) \;=\; t_j \sum_{k=i}^{j-1} M^k \;\le\; t_j\, pM^{j-1} \;=\; pM^{-1/2},$$

while for any $i > j$,

$$t_j(f_i - f_j) \;\le\; -t_j M^j \;=\; -M^{1/2}.$$

Writing

$$\sigma(t_j\mathbf{f})_i \;=\; \frac{e^{t_j(f_i - f_j)}}{\sum_{l=1}^{p} e^{t_j(f_l - f_j)}},$$

the bounds above show that, as $M \to \infty$, the numerator converges to 1 for $i \le j$ and to 0 for $i > j$, so $\sigma(t_j\mathbf{f}) \to \mathbf{b}_j$. Since there are only finitely many j, for any $\varepsilon' > 0$ we can choose M large enough that $\|\sigma(t_j\mathbf{f}) - \mathbf{b}_j\| < \varepsilon'$ for all j simultaneously.

It remains to compute the distances between consecutive barycenters. The vector $\mathbf{b}_{j-1} - \mathbf{b}_j$ has its first $j-1$ coordinates equal to $\frac{1}{j-1} - \frac{1}{j} = \frac{1}{j(j-1)}$ and its jth coordinate equal to $-\frac{1}{j}$, so

$$\|\mathbf{b}_{j-1} - \mathbf{b}_j\|^2 \;=\; \frac{j-1}{j^2(j-1)^2} + \frac{1}{j^2} \;=\; \frac{1}{j(j-1)}.$$

Now, taking $t_p = 0$ (so that $\sigma(t_p\mathbf{f}) = \mathbf{b}_p$ exactly), the curve visits the points $\boldsymbol{\lambda}(t_p), \boldsymbol{\lambda}(t_{p-1}), \ldots, \boldsymbol{\lambda}(t_1)$ at increasing values of t, so by the triangle inequality its arclength is at least the length of the polygonal chain through these points:

$$\text{arclength}\left(\mathcal{L}_{0^+}(\mathbf{F})\right) \;\ge\; \sum_{j=2}^{p} \|\boldsymbol{\lambda}(t_{j-1}) - \boldsymbol{\lambda}(t_j)\| \;\ge\; \sum_{j=2}^{p} \Big(\|\mathbf{b}_{j-1} - \mathbf{b}_j\| - 2\varepsilon'\Big) \;=\; \psi(p) - 2(p-1)\varepsilon'.$$

Taking $\varepsilon' < \varepsilon/(2(p-1))$ completes the proof. ∎

Theorem 11 now follows by applying Lemma 13 with $\varepsilon = \psi(p) - \sqrt{2}$, which is positive for all $p \ge 5$.

Appendix K. Additional figures

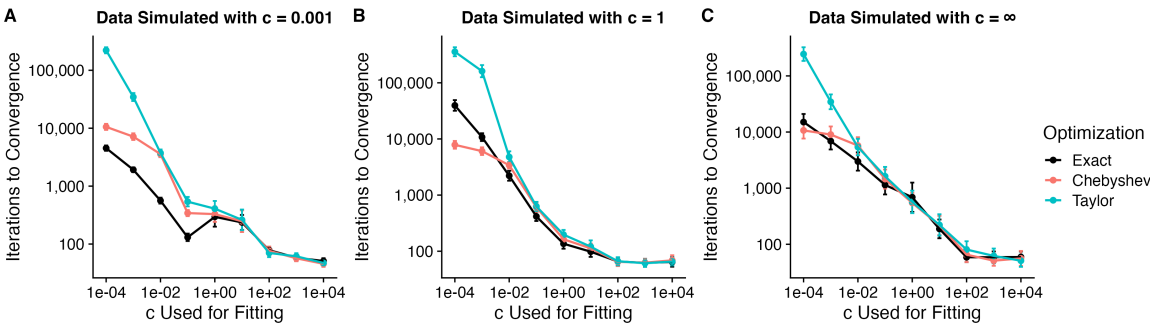
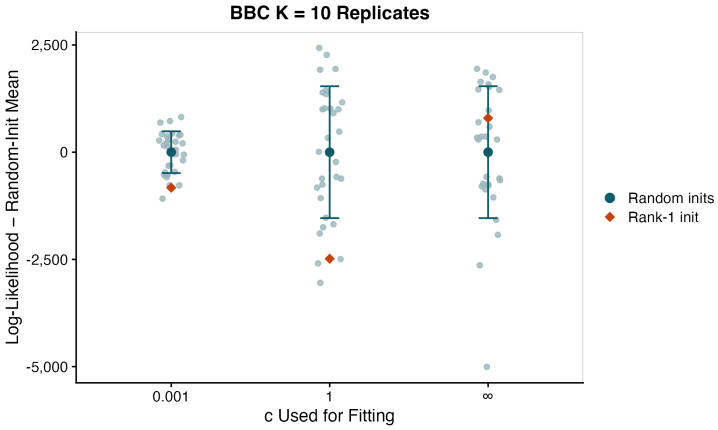


Figure K.1: Number of iterations until convergence for the fits of the approximation accuracy experiment of Section 3.3. For each generative regime and each finite value of c used for fitting, points show the geometric mean of the number of iterations until convergence over simulation replicates (30 per regime; 100 for data simulated with $c = 10^{-3}$), with error bars showing 95% confidence intervals computed on the log scale.



c	Mean	SD
0.001	−1,065,397	487
1	−1,124,985	1,538
∞	−1,125,533	1,538

Figure K.2: Stability of the $K = 10$ BBC fits to initialization. For each $c \in \{0.001, 1, \infty\}$, the model was fit to convergence (optimizing the exact log-likelihood) from 30 random initializations and from the rank-1 initialization used for the results in the main text. The random initializations are matched across values of c: for each replicate we drew matrices $\mathbf{L}_{\text{raw}}$ and $\mathbf{F}_{\text{raw}}$ with i.i.d. Uniform$(0, 1)$ entries, depending only on the replicate so that all values of c share the same draws, and initialized each model at $\mathbf{L}_0 = \sqrt{\tau_c}\,\mathbf{L}_{\text{raw}}$, $\mathbf{F}_0 = \sqrt{\tau_c}\,\mathbf{F}_{\text{raw}}$, where $\tau_c > 0$ was found numerically such that the Poisson rates implied by $\mathbf{B}_0 = \tau_c \mathbf{L}_{\text{raw}} \mathbf{F}_{\text{raw}}^\top$ under that model's link (i.e., $s_i g^{-1}(b_{ij}; c)$, and $\tau_c \mathbf{L}_{\text{raw}} \mathbf{F}_{\text{raw}}^\top$ directly for $c = \infty$) have mean equal to the mean of the observed counts. Light points show the exact log-likelihood of each converged random-initialization fit, centered at the mean over the 30 random initializations for that value of c; error bars show ± 1 standard deviation around this mean. (Table): mean and standard deviation of the exact log-likelihoods of the random-initialization fits for each value of c.

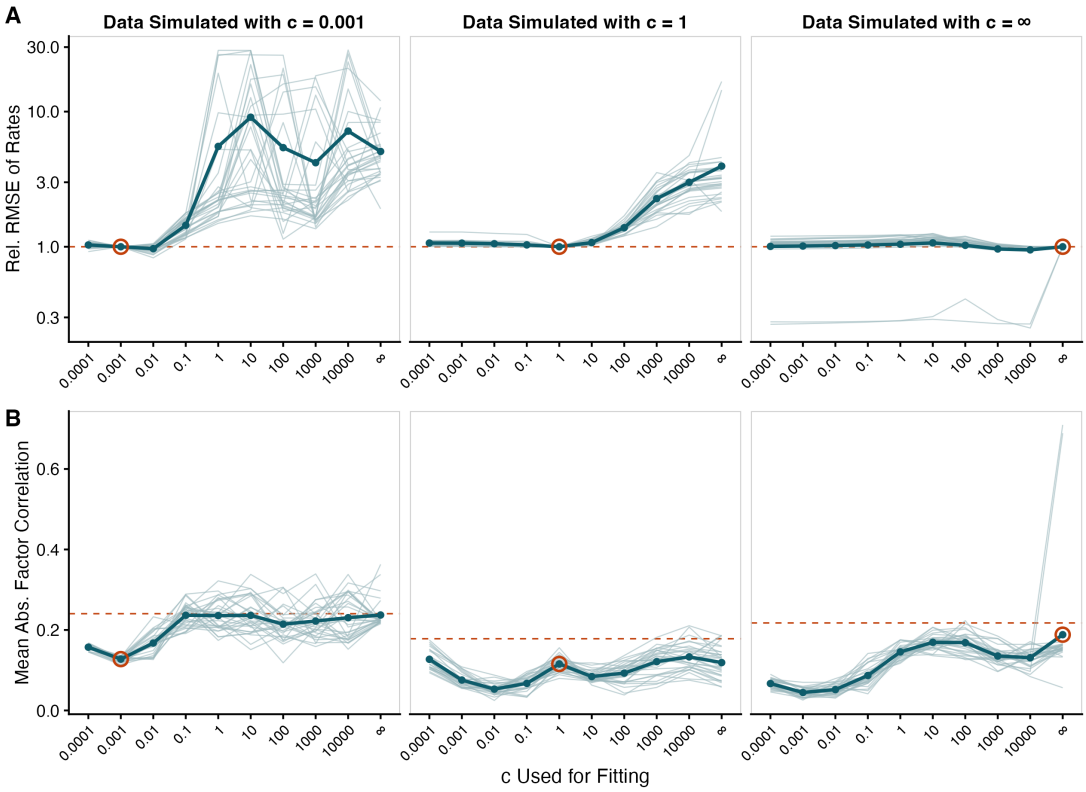


Figure K.3: Additional results for Simulation 2. Thin lines show individual simulation replicates, thick lines their mean, and the open circle marks $c_{\text{fit}} = c_{\text{true}}$. (A) RMSE of the fitted rates against the true rates, relative to the same replicate's fit at $c_{\text{fit}} = c_{\text{true}}$ (log scale); the dashed line marks 1. (B) Mean absolute pairwise Spearman correlation between the columns of the estimated $\mathbf{F}$ (as in Figure 11); the dashed line shows the same quantity for the true factors.

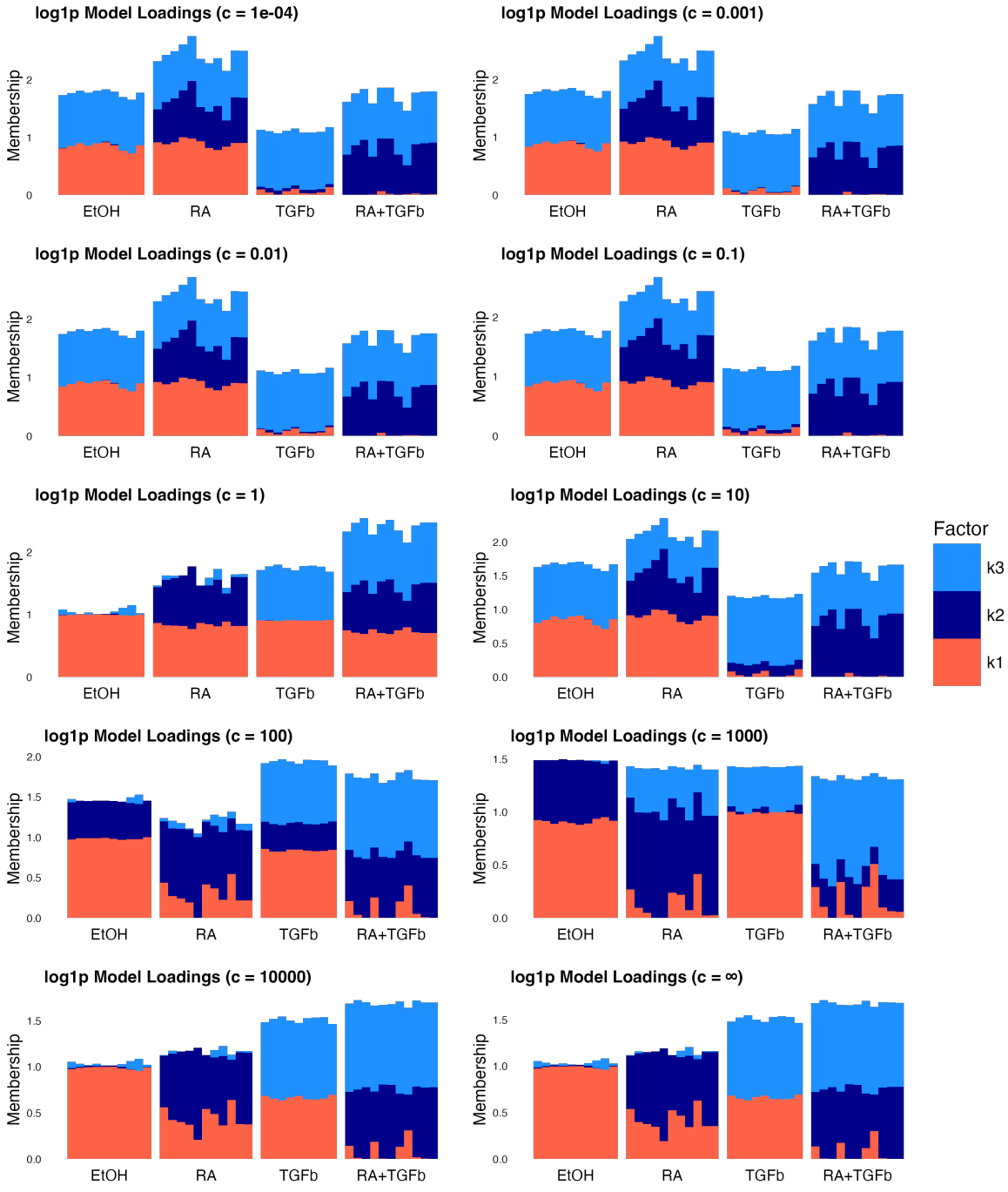


Figure K.4: Structure plots for the MCF-7 dataset across various values of c.

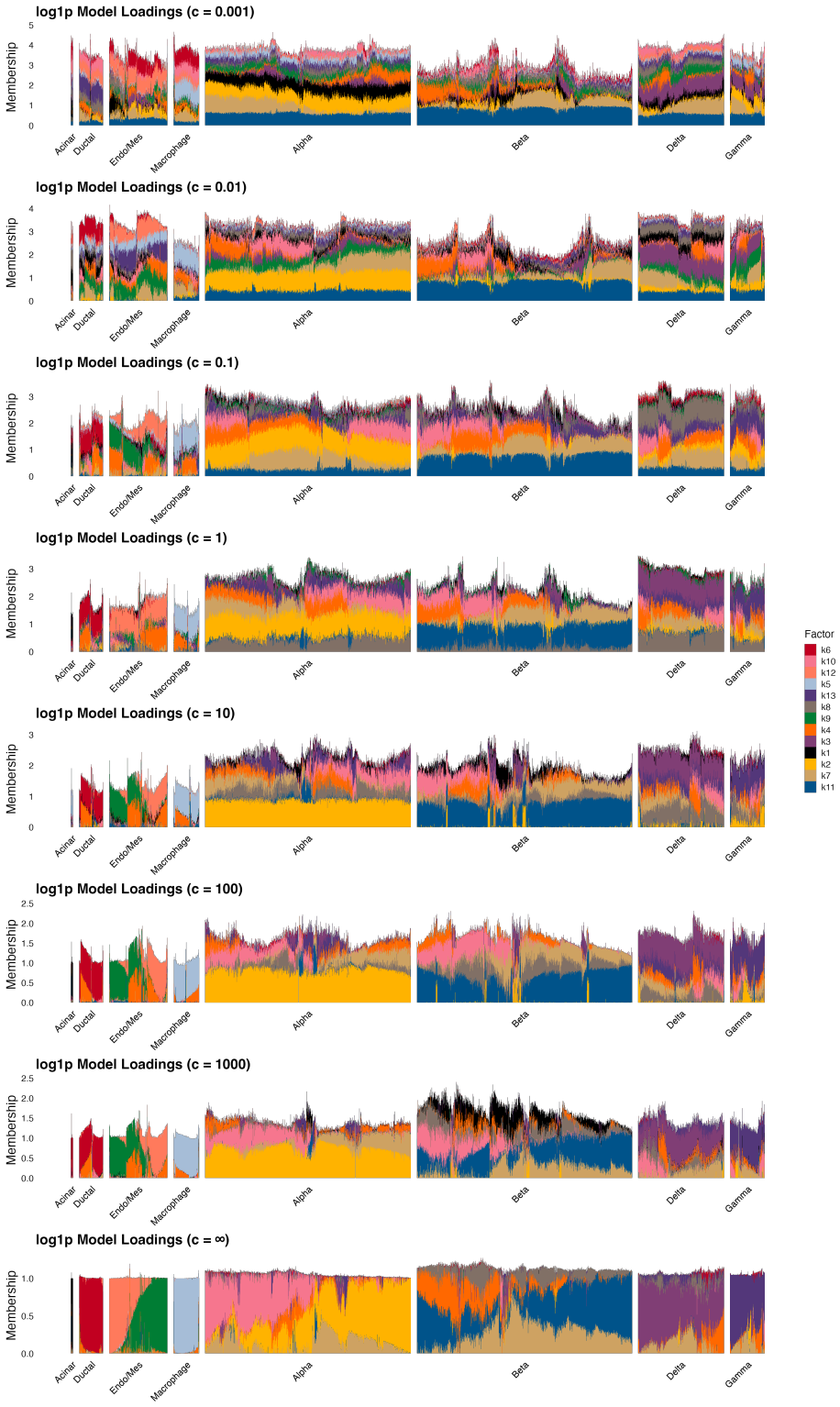


Figure K.5: Structure plots for the pancreas dataset across various values of c.

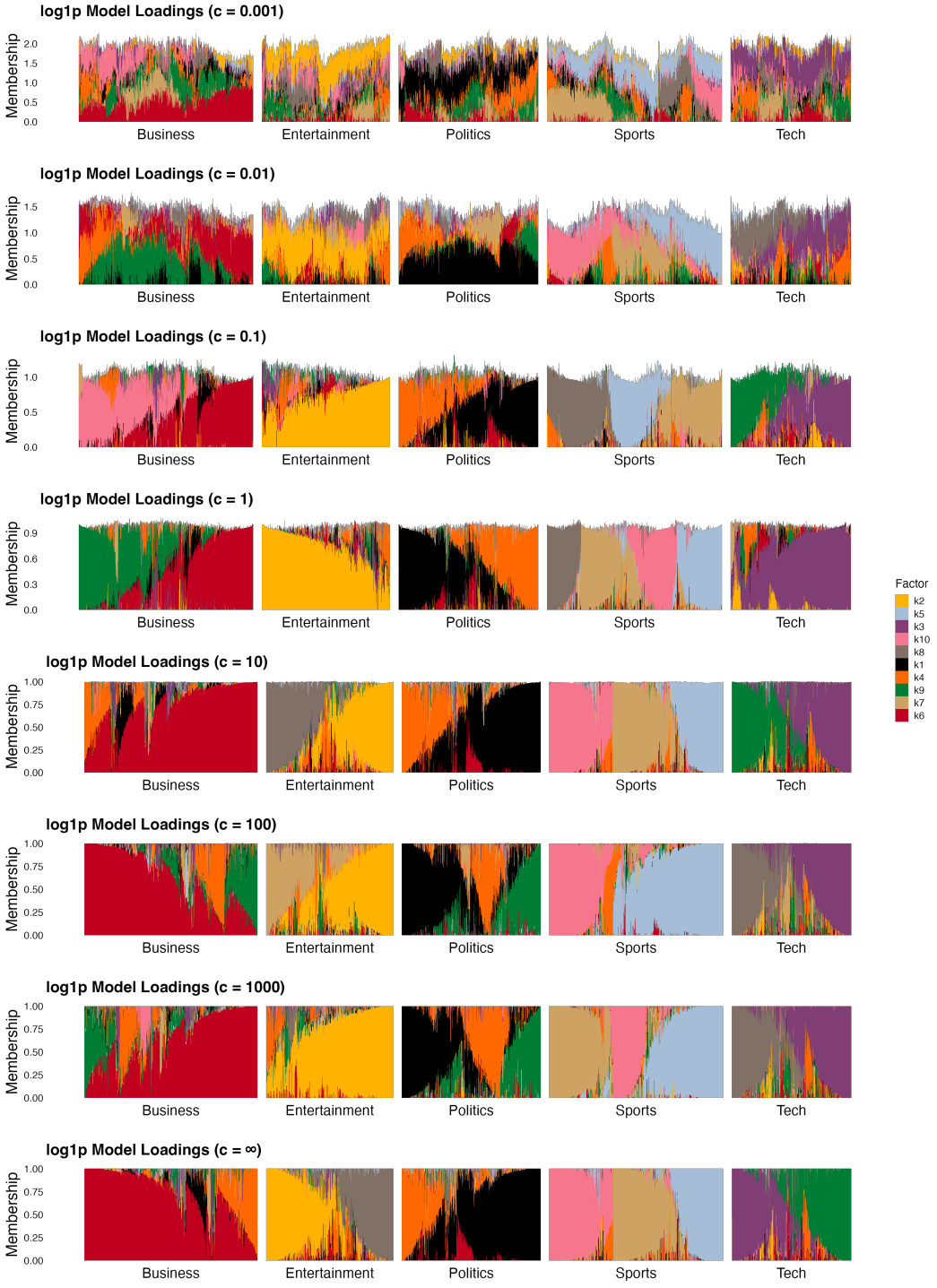


Figure K.6: Structure plots for the BBC dataset across various values of c.

References

Constantin Ahlmann-Eltze and Wolfgang Huber. Comparison of transformations for single-cell RNA-seq data. *Nature Methods*, 20(5):665–672, 2023.

Ludmil B Alexandrov, Jaegil Kim, Nicholas J Haradhvala, Mi Ni Huang, Alvin Wei Tian Ng, Yang Wu, Arnoud Boot, Kyle R Covington, Dmitry A Gordenin, Erik N Bergstrom, et al. The repertoire of mutational signatures in human cancer. *Nature*, 578(7793):94–101, 2020.

Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.

Stuart G Baker. The multinomial-Poisson transformation. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 43(4):495–504, 1994.

BBC. Boris opposes livingstone apology. `http://news.bbc.co.uk/2/hi/uk_news/politics/4272895.stm`, 2005. [Accessed 28-07-2025].

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995.

Dimitri P Bertsekas. Projected Newton methods for optimization problems with simple constraints. *SIAM Journal on control and Optimization*, 20(2):221–246, 1982.

Dimitri P. Bertsekas. *Nonlinear programming*. Athena Scientific, Belmont, MA, 2nd edition, 1999.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Elizabeth I Boyle, Shuai Weng, Jeremy Gollub, Heng Jin, David Botstein, J Michael Cherry, and Gavin Sherlock. GO::TermFinder—open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. *Bioinformatics*, 20(18):3710–3715, 2004.

Junyue Cao, Malte Spielmann, Xiaojie Qiu, Xingfan Huang, Daniel M Ibrahim, Andrew J Hill, Fan Zhang, Stefan Mundlos, Lena Christiansen, Frank J Steemers, Cole Trapnell, and Jay Shendure. The single-cell transcriptional landscape of mammalian organogenesis. *Nature*, 566(7745):496–502, 2019.

Peter Carbonetto, Abhishek Sarkar, Zihao Wang, and Matthew Stephens. Non-negative matrix factorization algorithms generally improve topic model fits. *arXiv*, 2105.13440, 2021.

Peter Carbonetto, Kaixuan Luo, Abhishek Sarkar, Anthony Hung, Karl Tayeb, Sebastian Pott, and Matthew Stephens. GoM DE: interpreting structure in sequence count data with differential expression analysis allowing for grades of membership. *Genome Biology*, 24(1):236, 2023.

Vartan Choulakian. Generalized bilinear models. *Psychometrika*, 61(2):271–283, 1996.

Michael Collins, Sanjoy Dasgupta, and Robert E Schapire. A generalization of principal components analysis to the exponential family. *Advances in Neural Information Processing Systems*, 14, 2001.

Kushal K Dey, Chiaowen Joyce Hsiao, and Matthew Stephens. Visualizing the structure of RNA-seq expression data using grade of membership models. *PLoS Genetics*, 13(3): e1006599, 2017.

Julien Diana and Agnès Lehuen. Macrophages and β-cells are responsible for CXCR2-mediated neutrophil infiltration of the pancreas during autoimmune diabetes. *EMBO molecular medicine*, 6(8):1090–1104, 2014.

David Donoho and Victoria Stodden. When does non-negative matrix factorization give a correct decomposition into parts? *Advances in neural information processing systems*, 16, 2003.

Prem Gopalan, Jake M Hofman, and David M Blei. Scalable recommendation with hierarchical poisson factorization. In *UAI*, pages 326–335, 2015.

Derek Greene and Pádraig Cunningham. Producing accurate interpretable clusters from high-dimensional data. In *European conference on principles of data mining and knowledge discovery*, pages 486–494. Springer, 2005.

Rui He, Bo Feng, Yuezhou Zhang, Yuqing Li, Daoxing Wang, and Linchao Yu. IGFBP7 promotes endothelial cell repair in the recovery phase of acute lung injury. *Clinical Science*, 138(13):797–815, 2024.

Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

Ying Hu, Jihua Xue, Ying Yang, Xiaotang Zhou, Chaochao Qin, Min Zheng, Haihong Zhu, Yanning Liu, Weixia Liu, Guohua Lou, et al. Lipocalin 2 upregulation protects hepatocytes from IL1-β-induced stress. *Cellular Physiology and Biochemistry*, 36(2):753–762, 2015.

Jonathan Huggins, Ryan P Adams, and Tamara Broderick. PASS-GLM: polynomial approximate sufficient statistics for scalable Bayesian GLM inference. *Advances in Neural Information Processing Systems*, 30, 2017.

Niall Hurley and Scott Rickard. Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(10):4723–4741, 2009.

Frediano Inzani, Guido Rindi, and Stefano Serra. The endocrine pancreas. In Ozgur Mete and Sylvia L. Asa, editors, *Endocrine Pathology*, pages 718–742. Cambridge University Press, Cambridge, 2020.

Jeanette AI Johnson, Ashley P Tsang, Jacob T Mitchell, David L Zhou, Julia Bowden, Emily Davis-Marcisak, Thomas Sherman, Ted Liefeld, Melanie Loth, Loyal A Goff, et al. Inferring cellular and molecular processes in single-cell data with non-negative matrix factorization using python, r and genepattern notebook implementations of cogaps. *Nature protocols*, 18(12):3690–3731, 2023.

Stephen Keeley, David Zoltowski, Yiyi Yu, Spencer Smith, and Jonathan Pillow. Efficient non-conjugate Gaussian process factor models for spike count data using polynomial approximations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 5177–5186, 2020.

Mary Kelly and Johannes von Lintig. STRA6: role in cellular retinol uptake and efflux. *Hepatobiliary surgery and nutrition*, 4(4):229, 2015.

Takahiko Kogai, James J Schultz, Laura S Johnson, Min Huang, and Gregory A Brent. Retinoic acid induces sodium/iodide symporter gene expression and radioiodide uptake in the MCF-7 breast cancer cell line. *Proceedings of the National Academy of Sciences*, 97(15):8519–8524, 2000.

Harold W Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, 1955.

Jenna M Landy, Nishanth Basava, and Giovanni Parmigiani. bayesNMF: Fast Bayesian Poisson NMF with automatically learned rank applied to mutational signatures. *arXiv preprint arXiv:2502.18674*, 2025.

Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

Mingli Liu, Shanchun Guo, Jacqueline M Hibbert, Vidhan Jain, Neeru Singh, Nana O Wilson, and Jonathan K Stiles. CXCL10/IP-10 in infectious diseases pathogenesis and potential therapeutic implications. *Cytokine & growth factor reviews*, 22(3):121–130, 2011.

Michael I. Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12):550, 2014.

Robert Luce, Peter Hildebrandt, Uwe Kuhlmann, and Jörg Liesen. Using separable nonnegative matrix factorization techniques for the analysis of time-resolved raman spectra. *Applied spectroscopy*, 70(9):1464–1475, 2016.

Emily L Mackevicius, Andrew H Bahle, Alex H Williams, Shijie Gu, Natalia I Denisenko, Mark S Goldman, and Michale S Fee. Unsupervised discovery of temporal sequences in high-dimensional datasets, with applications to neuroscience. *Elife*, 8:e38471, 2019.

Peter McCullagh and John A. Nelder. *Generalized linear models*. Chapman & Hall, New York, NY, second edition, 1989.

Fabrice Moore, Izortze Santin, Tatiane C Nogueira, Esteban N Gurzov, Lorella Marselli, Piero Marchetti, and Decio L Eizirik. The transcription factor C/EBP delta has anti-apoptotic and anti-inflammatory roles in pancreatic beta cells. *PloS one*, 7(2):e31062, 2012.

Phillip B Nicol and Jeffrey W Miller. Model-based dimensionality reduction for single-cell RNA-seq using generalized bilinear models. *Biostatistics*, 26(1):kxaf024, 2025. doi: 10.1093/biostatistics/kxaf024.

Sonsoles Piera-Velazquez and Sergio A Jimenez. Endothelial to mesenchymal transition: role in physiology and in the pathogenesis of human diseases. *Physiological reviews*, 99 (2):1281–1324, 2019.

Martin Porter and Richard Boulton. *The English (Porter2) stemming algorithm*, 2001. URL `http://snowball.tartarus.org/algorithms/english/stemmer.html`.

Jonathan K Pritchard, Matthew Stephens, and Peter Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959, 2000.

G. Salton. *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc., USA, 1971.

Eric M Sanford, Benjamin L Emert, Allison Coté, and Arjun Raj. Gene regulation gravitates toward either addition or multiplication when combining the effects of two signals. *Elife*, 9:e59388, 2020.

Abhishek Sarkar and Matthew Stephens. Separating measurement and expression models clarifies confusion in single-cell RNA sequencing analysis. *Nature genetics*, 53(6):770–777, 2021.

Matthias Seeger and Guillaume Bouchard. Fast variational Bayesian inference for non-conjugate matrix factorization models. In *Artificial Intelligence and Statistics*, pages 1012–1018. PMLR, 2012.

Jennifer S Stancill, Moujtaba Y Kasmani, Achia Khatun, Weiguo Cui, and John A Corbett. Single-cell RNA sequencing of mouse islets exposed to proinflammatory cytokines. *Life science alliance*, 4(6), 2021.

The Gene Ontology Consortium. The Gene Ontology knowledgebase in 2023. *Genetics*, 224 (1):iyad031, 2023.

Marvin Thielert, Adrian Villalba, Vincenth Brennsteiner, Maria Wahle, Constantin Ammar, Andreas-David Brunner, Alexis Fouque, Chloé Lourenço, Masaya Oshima, Latif Rachdi, et al. Decoding adult murine pancreatic islet cell diversity through cell type-resolved proteomics and phosphoproteomics. *Communications Biology*, 8(1):1483, 2025.

Ariel R Topletz, Jayne E Thatcher, Alex Zelter, Justin D Lutz, Suzanne Tay, Wendel L Nelson, and Nina Isoherranen. Comparison of the function and expression of CYP26A1 and CYP26B1, the two retinoic acid hydroxylases. *Biochemical pharmacology*, 83(1): 149–163, 2012.

F William Townes, Stephanie C Hicks, Martin J Aryee, and Rafael A Irizarry. Feature selection and dimension reduction for single-cell RNA-Seq based on a multinomial model. *Genome Biology*, 20:295, 2019.

Saima Usman, Naushin H Waseem, Thuan Khanh Ngoc Nguyen, Sahar Mohsin, Ahmad Jamal, Muy-Teck Teh, and Ahmad Waseem. Vimentin is at the heart of epithelial mesenchymal transition (EMT) mediated metastasis. *Cancers*, 13(19):4985, 2021.

Dorothee Van Breevoort, Ellen L Van Agtmaal, Bieuwke S Dragt, Jacqueline Klein Gebbinck, Ilze Dienava-Verdoold, Astrid Kragt, Ruben Bierings, Anton JG Horrevoets, Karine M Valentijn, Jeroen C Eikenboom, et al. Proteomic screen identifies IGFBP7 as a novel component of endothelial cell-specific Weibel-Palade bodies. *Journal of proteome research*, 11(5):2925–2936, 2012.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

Wei Wang and Matthew Stephens. Empirical bayes matrix factorization. *Journal of Machine Learning Research*, 22(120):1–40, 2021.

Eric Weine, Peter Carbonetto, and Matthew Stephens. Accelerated dimensionality reduction of single-cell RNA sequencing data with fastglmpca. *Bioinformatics*, 40(8):btae494, 2024.

Jason Willwerscheid. *Empirical Bayes matrix factorization: methods and applications*. PhD thesis, The University of Chicago, 2021.

Stephen J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151:3–34, 2015.

Jessica L Zhou, Karthik Guruvayurappan, Shushan Toneyan, Hsiuyi V Chen, Aaron R Chen, Peter Koo, and Graham McVicker. Analysis of single-cell CRISPR perturbations indicates that enhancers predominantly act multiplicatively. *Cell Genomics*, 4(11), 2024.

Alessandro Zito and Jeffrey W Miller. Compressive Bayesian non-negative matrix factorization for mutational signatures analysis. *arXiv preprint arXiv:2404.10974*, 2024.

David Zoltowski and Jonathan W Pillow. Scaling the Poisson GLM to massive neural datasets through polynomial approximations. In *Advances in Neural Information Processing Systems*, volume 31, 2018.