Rank selection for non-negative matrix factorization

Yun Cai | Hong Gu | Toby Kenney

Department of Mathematics and Statistics, Dalhousie University, Halifax, Canada

Correspondence
Yun Cai, Department of Mathematics and Statistics, Dalhousie University, Halifax, Canada.
Email: yn298893@dal.ca

Funding information
Natural Sciences and Engineering Research Council of Canada, Grant/Award Numbers: RGPIN-2017-05108, RGPIN/4945-2014

Non-Negative Matrix Factorization (NMF) is a widely used dimension reduction method that factorizes a non-negative data matrix into two lower dimensional non-negative matrices: one is the basis or feature matrix which consists of the variables and the other is the coefficients matrix which is the projections of data points to the new basis. The features can be interpreted as sub-structures of the data. The number of sub-structures in the feature matrix is also called the rank. This parameter controls the model complexity and is the only tuning parameter for the NMF model. An appropriate rank will extract the key latent features while minimizing the noise from the original data. However due to the large amount of optimization error always present in the NMF computation, the rank selection has been a difficult problem. We develop a novel rank selection method based on hypothesis testing, using a deconvolved bootstrap distribution to assess the significance level accurately. Through simulations, we compare our method with a rank selection method based on hypothesis testing using bootstrap distribution without deconvolution and a method based on cross-validation; we demonstrate that our method is not only accurate at estimating the true ranks for NMF, especially when the features are hard to distinguish, but also efficient at computation. When applied to real microbiome data (e.g., OTU data and functional metagenomic data), our method also shows the ability to extract interpretable subcommunities in the data.

KEYWORDS
metagenomics, microbial communities, non-negative matrix factorization, rank selection, subcommunities

1 | INTRODUCTION

1.1 | Non-negative matrix factorization

Consider a non-negative data matrix \( X \) with \( p \) variables (rows) and \( n \) observations (columns). Non-negative matrix factorization NMF attempts to approximate the matrix \( X \) by the product \( TW \) where \( T \) is a \( p \times k \) non-negative matrix and \( W \) is a \( k \times n \) non-negative matrix. When \( k \ll np/(n + p) \), the dimension is significantly reduced from the original problem.

Abbreviations: NMF, non-negative matrix factorization; OTU, operational taxonomic unit.

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made. © 2023 The Authors. Statistics in Medicine published by John Wiley & Sons Ltd.
The matrices $T$ and $W$ are called the feature matrix and the weight matrix respectively. Each column of $W$ represents the corresponding observation from $X$ as a non-negative linear combination of the columns of $T$.

Because of the non-negativity constraints, the features and weights selected by NMF tend to be sparse. Furthermore, the features, or types, in the matrix $T$ represent parts of the data $X$. This is different from other dimension reduction methods such as principal component analysis or vector quantization, and makes interpretation of results easier. In many contexts, the non-negative combinations can be viewed as mixtures of the types.

NMF has been widely applied in many areas including image data,\(^1\) automatic music transcription,\(^2\) computational biology,\(^3\) molecular pathology,\(^4\) and microbiology.\(^5,6\) Microbial communities, which consists of a large number of different microorganisms, have been shown to be associated with human health and global nutrient cycling.\(^7\) The non-negative structure of NMF can be neatly interpreted as expressing the microbial community as a combination of subcommunities, which may represent interacting groups of microbes. Jiang\(^8\) has applied NMF to marine microbes data to investigate the biogeography of microbial function by extracting a small number of features associated with both microbial protein profiles and sampling location. Ko\(^9\) used NMF to estimate functional microbial interactions from microbiome data.

The NMF can be fitted to minimize a loss function, very often the squared error loss is used, which is equivalent to assuming that the data entries in $X$ follow normal distributions with means given by the corresponding entries in the matrix $TW$. More generally the negative log-likelihood of $X$ under some assumed parametric distribution can be used as a loss function. Multiplicative algorithms\(^10\) can be used for fitting NMF both for squared error loss and for negative Poisson log-likelihood loss. These methods are implemented by the R\(^11\) package NMF, which we use for fitting NMF in this paper. While these methods converge robustly, the solution to NMF is not always unique,\(^12\) and even if it is unique, the likelihood surface can be multimodal,\(^13\) meaning that the optimization can converge to a local optimum.

The rank of the feature matrix $k$ controls the model complexity and is the only tuning parameter. It can be interpreted as the number of underlying sub-structures or subcommunities extracted from the data. Thus the selection of $k$ not only affects the performance of NMF but also is related to the interpretation of NMF results in the applications. Small $k$ values may lead to missing key features and large $k$ values may lead to overfitting issues. Selecting an appropriate $k$ is a non-trivial model selection problem in NMF.

Despite the crucial importance of selecting the correct rank for NMF, it has been an understudied problem in the literature (see the literature review in Section 1.2). In this paper, we develop a new rank selection method for NMF based on a goodness-of-fit test using a deconvoluted bootstrap distribution by offsetting the optimization errors for the test statistic. The application of deconvolution to bootstraps with optimization error appears to be a novel idea, which could have significant applications in a wide range of other areas. We compare our rank selection method with bootstrap distribution with and without deconvolution and a state-of-the-art existing method, Non-Negative Linear Models (NNLM), which has shown better performance than other methods in previous literature. We find that our method produces more accurate and stable estimation of ranks for NMF compared with NNLM.

### 1.2 A brief literature review for rank selection in NMF

The usual approach to NMF rank selection is to use expert knowledge.\(^14\) However, for many analyses in practice, no one has the insight needed to select the rank. Even if there are experts with a good intuition of what rank is appropriate, being able to support the choice with statistical evidence is valuable.

When expert knowledge is unavailable, a popular approach for rank selection is to compute some model measures for a range of rank values and choose the rank value according to the measure’s criteria. Cophenetic correlation coefficient and Dispersion\(^15\) are the most commonly used measurements. The Cophenetic correlation coefficient method measures the reproducibility of the clustering of observations based on the weight matrix of NMF for many runs with different initial values. Because of the multimodality of the loss functions, runs of NMF with different initial values can produce different results. The idea is that for the appropriate number of features, there should be a clear optimum, so runs with different initial values should give the same solution. If the number of features is too large, then the optimal likelihood or mean squared error will vary between runs, leading to a smaller Cophenetic correlation. The method selects the rank value where the Cophenetic correlation\(^3\) has a steep drop off. The Dispersion method uses a similar procedure to the Cophenetic coefficient method\(^16\) but calculates Dispersion coefficients instead. These two methods both measure the robustness of the clustering analysis results, instead of the goodness-of-fit of the NMF results to the data, thus could lead to suboptimal solutions.\(^17\) Other ideas include comparing residuals, sparseness and Description Length.\(^18\) These methods
are all ad hoc in that they select the rank according to the plots based on intuition instead of a solid statistical inference results, and thus cannot be used to perform large scale simulations. The results will be inaccurate when a clear drop off point is not available.

There are also some other rank selection methods such as the Bayesian method and singular value decomposition. The Bayesian method requires an assumption that the rank should be small, which is not necessarily true for real data. Singular value decomposition chooses the rank where the singular values become small. The challenge of the method arises when there isn’t a clear drop to zero in the singular values of the matrix. Moreover, the rank of singular values is not closely related to the rank of non-negative matrix factorization.

Compared to the above approaches, the approaches based on the idea of cross-validation are more objective in measuring the goodness-of-fit of the model to data and also more popularly used. For example, the 2-fold cross-validation method splits the data randomly into two halves and applies NMF to the two parts separately. The rank minimizing the average reconstruction error between the two parts is selected for NMF. Similarly, Lin introduces a rank selection method based on imputation. They randomly delete 30% of the data entries from the original data each time and treat them as missing values in the NMF procedure. The rank is chosen by minimizing the mean squared errors of the reconstruction of the missing entries. Their NMF algorithm to handle missing values is available in the R package NNLM. A systematic comparison on the accuracy of rank selection among the Cophenetic correlation coefficient method, Dispersion method, 2-fold cross validation method and the NNLM method was made by different authors on normal synthetic data sets with different properties, and it was concluded that the NNLM method estimates the rank more accurately than other methods. Thus the NNLM is chosen as the state-of-the-art method for comparison with the new method developed in this paper.

2 METHODS

Our rank selection for NMF is based on sequentially performing the following hypothesis test:

\( H_0: \) the rank of the feature matrix is \( k \).

\( H_a: \) the rank of the feature matrix is at least \( k + 1 \).

After applying the goodness-of-fit test, if \( H_0 \) is rejected, let \( k = k + 1 \) and repeat the test until \( H_0 \) cannot be rejected at the given significance level.

2.1 Likelihood ratio test

Suppose \( L(k) \) is the Likelihood of the factorization using rank \( k \) NMF and \( L(k + 1) \) is the Likelihood of the rank \( k + 1 \) NMF results. The likelihood ratio test statistic under the null hypothesis \( H_0 \) is given by:

\[
\lambda_{LR} = -2 \log \left( \frac{\sup L(k)}{\sup L(k+1)} \right) = 2(\sup l(k+1) - \sup l(k)),
\]

where sup denotes the supremum and \( l() \) is the log-likelihood function. The log-likelihood function under a normal distribution (assuming constant variance \( \sigma^2 \)) is

\[
l_n = -\frac{1}{2\sigma^2} \sum_{i=1}^{p} \sum_{j=1}^{n} (X_{ij} - (TW)_{ij})^2,
\]

and under a Poisson distribution is

\[
l_p = \sum_{i=1}^{p} \sum_{j=1}^{n} (X_{ij} \log(TW)_{ij} - (TW)_{ij}).
\]

We focused on the normal and Poisson distributions in this paper, however our method is general for any other parametric distribution.

Although this is the common likelihood ratio statistic, the standard statistical theory of the asymptotic chi-squared distribution under the null hypothesis doesn’t apply here due to the non-negative boundary constraints. This problem can be easily solved by using a bootstrap method to compute the correct null distribution, as detailed in Section 2.2. However, due to the computational difficulty that NMF solution is usually a local optimum, the results of any given runs of the NMF algorithm are subject to computational measurement errors for the likelihood ratio statistic. With the measurement errors both in the data likelihood ratio statistic and in the bootstrap sampled statistics, the hypothesis test loses its power due to the long tail in the null distribution. We thus develop a method to improve the power of the likelihood ratio test for NMF rank selection by using deconvolution methods to remove the optimization error. This work is also an application of our previous work on a general method for measurement error deconvolution density estimation using penalized likelihood.
2.2 | Direct testing with bootstrapped null distribution (Boot-test)

For our hypothesis test, we can use a parametric bootstrap to obtain the null distribution of the test statistic: under the null hypothesis, the data is a realization of a random sample from a distribution $F$ with mean a linear combination of the $k$ features. The $\hat{F}$ can be obtained by estimating the $T$ and $W$ from the original data, then the bootstrap samples of random datasets of the same dimension as the original data are drawn from $\hat{F}$. For Poisson distribution, the means in $\hat{T}$ and $\hat{W}$ are the only parameters needed for bootstrap. For normal data, assuming each entry of the data is independently normally distributed and shares the same variance, in addition to the means in $\hat{T}\hat{W}$, the variance also needs to be estimated from the original data. The variance can be estimated from the residuals of the $k$ rank NMF model. For example, if the rank $k$ NMF gives $X \approx T_{p \times k}W_{k \times n}$, the variance can be estimated by $\frac{1}{np} \sum_{ij} (X_{ij} - (TW)_{ij})^2$. Since the bootstrapped data matrix also needs to be nonnegative, the negative entries in the bootstrapped sample are replaced by 0 for normal data. We apply rank $k$ and rank $k + 1$ NMF on each bootstrapped sample to get the likelihood ratio statistic, denoted by $\lambda^*$, from which the empirical null distribution of $\lambda$ can be obtained.

Unfortunately, this hypothesis test with bootstrapped null distribution does not perform well at selecting the rank of NMF since the likelihood surface for NMF is often multimodal, and it is common for the algorithm to converge to a suboptimal factorization. This creates noise in the bootstrapped null distribution, thus reducing the power of the test.

To improve the performance, we can rerun the algorithm with multiple starting values on the original data to choose the best log-likelihood as the estimate for the mean $\hat{F}$. Since we are trying to maximize the log-likelihood, we will assume that the largest likelihood from multiple runs of different initial values is the true maximum likelihood. If the number of different initial values is large enough, this should almost always be true. For normal data, the variance is the average of variances estimated from each run of NMF to avoid overfitting. We also run NMF with multiple initial values on each bootstrap sample and choose the best log-likelihoods to build the null distribution. The procedure is shown in Algorithm 1.

**Algorithm 1.** Pseudocode for Boot-test

```plaintext
k ← 0
while p-value is significant do
   k ← k + 1
   (1) Steps for calculating the test statistic
   for m different initial values do
      apply rank k NMF and rank k + 1 NMF to the original data
   end for
   $l_0(k) ←$ largest log-likelihood of all rank $k$ NMF results;
   $l_0(k + 1) ←$ largest log-likelihood of all rank $k + 1$ NMF results;
   test statistic $\lambda ← 2(l_0(k + 1) - l_0(k))$.
   (2) Steps for calculating the null distribution:
   compute $T_0$ and $W_0$ from the rank $k$ NMF with log-likelihood $l_0(k)$
   sample bootstrap datasets from distribution with mean $T_0W_0$ (and variance estimate if data are assumed to be normal)
   for all bootstrap datasets do
      for m different initial values do
         apply rank k NMF and rank k + 1 NMF to the bootstrap data
      end for
      Compute $\lambda^* ← 2(l_0^*(k + 1) - l_0^*(k))$, where $l_0^*(k)$ and $l_0^*(k + 1)$ are the best log-likelihood of rank $k$ and rank $k + 1$ NMF results respectively.
   end for
   get p-value;
end while
```
The simulation results for this method with different numbers of initial values and different true rank values are shown in Section 3.1.

2.3 Testing with deconvolved bootstrap null distribution (Decon-boot-test)

The major problem with the method described above is that to ensure a good bootstrap distribution, we need to fit each bootstrap sample with a large number of different initial values, which is computationally expensive. We therefore develop a new approach to more efficiently compute the bootstrap distribution. The idea is to treat the convergence error as measurement error in the log-likelihood ratio statistic. We can then use a well-developed deconvolution method to estimate the null distribution of the statistic. Deconvolution is a method to estimate a distribution from data with additive measurement error. In our case, if we run only one set of initial values for NMF for each bootstrap sample for each fixed $k$, we can directly compute the estimated maximum log-likelihood $l(k) = l_0(k) + e(k)$, where $l_0(k)$ is the globally maximized log-likelihood value (unobserved) and $e(k)$ is the measurement (convergence) error. Thus

$$\lambda^* = \lambda^*_0 + e = 2(l_0(k + 1) - l_0(k)) + 2(e(k + 1) - e(k)),$$

where $\lambda^*_0 = 2(l_0(k + 1) - l_0(k))$ is the unobserved true likelihood ratio statistic without convergence error and $e = 2(e(k + 1) - e(k))$ is the convergence error. The deconvolution method will estimate the distribution of $\lambda^*_0$ when $\lambda^*_0$ is not observable but $\lambda^*_0 + e$ is available.

Most deconvolution approaches are based on the Fourier transformation. While its mathematical solution is very neat, estimation of the characteristic function from data is somewhat unstable, and division by possibly small Fourier coefficients for the error data can greatly magnify errors. So in this paper, we use the P-MLE deconvolution method which is based on maximizing a penalized log-likelihood of the data (here $\lambda^*$) with a smoothness penalty. P-MLE also has the advantage over many other competing methods that its implementation can estimate the measurement error distribution non-parametrically from a pure measurement error sample. This is important because the convergence error is unlikely to follow a standard parametric family of distributions.

Figure 1 shows an example of the bootstrapped null distribution of the likelihood ratio statistic before and after deconvolution. The deconvolved distribution of the likelihood ratio statistic (distribution of $\lambda^*_0$) has a much smaller variance than the bootstrapped null distribution of the likelihood ratio statistic $\lambda^*$ calculated with a single initial value for each bootstrap sample.

The sampling procedure for bootstrapped datasets is the same as in Section 2.2. However, for each sampled data, rank $k$ NMF and rank $k + 1$ NMF are run only once to get $\lambda^*_0 + e$. In order to get a pure error sample, we perform rank $k$ and rank $k + 1$ NMF with multiple initial values for an additional bootstrap sample. The measurement error samples of $e(k)$ and $e(k + 1)$ can be obtained from the differences between the log-likelihoods of all NMF runs and the largest log-likelihood among different initial values. The pure error sample for $e$ can be obtained as two times the difference between $e(k + 1)$ and $e(k)$ for all different pairs of $e(k + 1)$ and $e(k)$. Our method is implemented in the R package DBNMF.ranking, which is available on CRAN https://CRAN.R-project.org/package=DBNMF.ranking.

A complete description of the procedure is given in Algorithm 2.

3 SIMULATION

In this section, we compare the performance of Boot-test and Decon-boot-test introduced in Sections 2.2 and 2.3 and NNLM which is based on missing data imputation, on Poisson NMF data, normal NMF data and non-NMF structure data. NNLM minimizes Kullback-Leibler divergence loss to factorize a matrix with Poisson error and square error loss to factorize a matrix with normal error. In the simulation, following the suggestions from Reference 21, we randomly set 30% of the data entries as NA and apply NNLM with a sequence of rank values from 1 to 40 to impute the missing data. The procedure is repeated 10 times and the rank value that minimizes the average loss (Kullback-Leibler divergence for Poisson data and Mean Squared Error for normal data) of the imputed part is selected as the rank for the data.

For the Boot-test and Decon-boot-test methods, we find that the bootstrap size doesn’t affect the rank estimates too much on the simulation data. We only show the simulation results when bootstrap size is 50 in this section. Also, NMF
FIGURE 1  Null distribution of the likelihood ratio statistics for rank 4 NMF versus rank 5 NMF. The data are generated from rank 4 Poisson NMF. The black curve is the kernel estimation of bootstrapped null distribution when applying rank 4 and rank 5 NMF with a single initial value for each bootstrap sample with bootstrap size as 50. The green curve is the deconvolved density of likelihood ratio statistic after applying the P-MLE deconvolution method to the 50 bootstrapped Likelihood ratios.

with 50 different initial values consistently converges to the optimal likelihood when running on the simulated data. Thus, for the Boot-test, we follow procedure 1 and set \( m = 1, m = 3, m = 10, m = 30 \) and \( m = 50 \) for each dataset and each bootstrap sample. For the Decon-boot-test, we follow procedure 2 and set \( m = 50 \).

We generate 50 replicates for each scenario. For the generated NMF data, the feature matrix is fixed in each scenario and the weight matrix is randomly generated in each replicate. All three methods are applied to the same data in each replicate. The significance level used for Boot-test and Decon-boot-test in the simulations is 0.1.

3.1 Poisson data simulation

We simulate a range of Poisson NMF scenarios with true rank equal to 2, 4, 6, 8, 10 and 30, and a scenario where there is no NMF structure. For true rank 2, 4, 6, 8 and 10, we simulate 100 observations, while for the true rank 30, we simulate 300 observations. For the no NMF structure scenarios, we simulate 131 observations.

For rank 2 and rank 4, we simulate five scenarios to assess the sensitivity of the methods when one of the types is close to a non-negative linear combination of the others. One feature for the true rank 2 simulation and three features for the true rank 4 simulation are also estimated from the healthy individuals in the Qin dataset using NMF with ranks 1 and 3 respectively. The Qin data is a human gut metagenomic dataset extracted from 99 healthy people and 25 Irritable Bowel Disease (IBD) patients. For rank 2 simulations, the second feature is on the line in the space spanned by the first feature and a unit vector with all elements the same and in the direction from the first feature to the unit vector. For the rank 4 simulations, the fourth feature is chosen in the space spanned by the other three features and a unit vector with all elements the same, such that the perpendicular from the fourth feature to the plane of the other three features passes through the centroid of these three features. We vary the distance of this fourth type from the plane of the other three to assess the sensitivity of the methods in cases where one type is almost a linear combination of the other three. The further the distance is, the more distinct these features are. The distances are set to be \( d = 0, 0.0005, 0.0008, 0.001, 0.0015 \).
Algorithm 2. Pseudocode for Decon-boot-test

\[ k \leftarrow 0 \]

\[ \textbf{while} \text{ p-value is significant} \textbf{do} \]
\[ \quad k \leftarrow k + 1 \]

(1) Steps for calculating the test statistic
\[ \textbf{for} \ m \text{ different initial values} \textbf{do} \]
\[ \quad \text{Applying rank } k \text{ NMF and rank } k + 1 \text{ NMF to original data} \]
\[ \textbf{end for} \]
\[ l_0(k) \leftarrow \text{the largest log-likelihood of all rank } k \text{ NMF results;} \]
\[ l_0(k + 1) \leftarrow \text{the largest log-likelihood of all rank } k + 1 \text{ NMF results;} \]
\[ \text{test statistic } \lambda \leftarrow 2(l_0(k + 1) - l_0(k)). \]

(2) Steps for calculating the null distribution:
\[ \text{compute } T_0 \text{ and } W_0 \text{ from the rank } k \text{ NMF with log-likelihood } l_0(k) \]
\[ \text{sample bootstrap datasets from distribution with mean } T_0W_0 \text{ (and variance estimate if data are assumed to be normal)} \]
\[ \textbf{for} \text{ all bootstrap datasets} \textbf{do} \]
\[ \quad \text{compute } \lambda^* \text{ by applying rank } k \text{ NMF and rank } k + 1 \text{ NMF once.} \]
\[ \textbf{end for} \]
\[ \textbf{for} \ m \text{ different initial values} \textbf{do} \]
\[ \quad \text{compute } l_i(k) \text{ by applying rank } k \text{ NMF on a bootstrap sample;} \]
\[ \quad \text{compute } l_j(k + 1) \text{ by applying rank } k + 1 \text{ NMF on the same bootstrap sample;} \]
\[ \textbf{end for} \]
\[ \textbf{for} \ i \leftarrow 1 \text{ to } m \textbf{ do} \]
\[ \quad e_i(k) \leftarrow (l_i(k) - \max(l_i(k)))); \]
\[ \textbf{end for} \]
\[ \textbf{for} \ j \leftarrow 1 \text{ to } m \textbf{ do} \]
\[ \quad e_j(k + 1) \leftarrow (l_j(k + 1) - \max(l_j(k + 1)))); \]
\[ \textbf{end for} \]
\[ e \leftarrow 2(e_i(k + 1) - e_j(k)) \text{ for all } i \text{ and } j; \]
\[ \text{compute null distribution of } \lambda^*_0 \text{ by applying P-MLE on } \lambda^* \text{ and } e; \]
\[ \text{get p-value;} \]
\[ \textbf{end while} \]

and 0.002. When the distance is 0, the feature matrix degenerates to a lower rank matrix. Thus, the rank 4 feature matrix degenerates to a rank 3 feature matrix and the rank 2 feature matrix becomes rank 1.

For the higher true ranks, we simulate one scenario for each true rank. For ranks 6, 8 and 10, all features used in the simulation are estimated from the healthy individuals in the Qin dataset, using Poisson NMF with the given rank.

For rank 30, the features used in the simulation are calculated by applying rank 30 NMF to Person 2’s gut data in the moving picture data. The moving picture data is the most detailed investigation of temporal microbiome variation to date. It consists of a long-term sampling series from two human individuals at four body sites: gut, tongue, right and left palm. After removing rows consisting of all 0’s, the total number of different Operational Taxonomic Units (OTUs) is 3131 for Person 2’s gut data. Each feature is normalized to sum to one for all scenarios.

For all Poisson NMF data simulations, columns of the weight matrix are generated independently from a uniform distribution from 0 to 1. We adjust each column of the weight matrix to have a fixed sum sampled from the column sums for the healthy group of the Qin data for true NMF rank 2, 4, 6, 8 and 10. The weight matrix is rescaled to have fixed sum sampled from the column sums for Person 2’s gut data of the moving picture data for true NMF rank 30. For microbiome data, these column sums are referred to as sequencing depth. Multiplying the feature matrix and the weight matrix, we get the means for the Poisson distribution. Each entry of the data matrix simulated for the Poisson NMF data is generated from a Poisson distribution with the Poisson mean given by the corresponding entry of the matrix product TW. The estimated results of Boot-test, Decon-boot-test and NNLM in 50 replicates of different scenarios are summarized in
### Table 1
Total number of times the true rank is selected out of 50 replicates and the 50 rank estimates' averages and standard deviations when the true rank is 2 for Poisson NMF data when bootstrap size is 50.

| d     | 0      | 0.0005 | 0.0008 | 0.001  | 0.0015 | 0.002  |
|-------|--------|--------|--------|--------|--------|--------|
| m = 1 | 47     | 1.06 (0.24) | 17 | 1.54 (0.68) | 22 | 2.04 (0.75) | 31 | 2 (0.62) | 44 | 2 (0.35) | 46 | 2.06 (0.37) |
| m = 3 | 46     | 1.08 (0.27) | 28 | 2.16 (0.65) | 39 | 2.1 (0.46) | 45 | 2.12 (0.39) | 40 | 2.2 (0.40) | 42 | 2.16 (0.37) |
| **Boot-test** | m = 10 | 47 | 1.06 (0.24) | 43 | 2.2 (0.53) | 45 | 2.08 (0.28) | 44 | 2.06 (0.25) | 43 | 2.14 (0.41) | 47 | 2.06 (0.24) |
| m = 30 | 47 | 1.06 (0.23) | 43 | 2.16 (0.42) | 45 | 2.12 (0.39) | 44 | 2.12 (0.33) | 46 | 2.08 (0.27) | 45 | 2.1 (0.30) |
| m = 50 | 47 | 1.06 (0.24) | 40 | 2.28 (0.54) | 46 | 2.08 (0.27) | 44 | 2.14 (0.40) | 45 | 2.16 (0.55) | 45 | 2.04 (0.20) |
| **Decon-boot-test** | 50 | 1 (0) | 50 | 2 (0) | 50 | 2 (0) | 50 | 2 (0) | 50 | 2 (0) |
| **NNLM** | 50 | 1 (0) | 39 | 2.12 (0.52) | 43 | 2.02 (0.38) | 44 | 2.04 (0.35) | 48 | 2.04 (0.18) | 47 | 2.06 (0.24) |

Note: First column for each value of d in grey: the number of times the true rank is selected out of 50 replicates. Second column for each value of d: the average of the 50 estimated ranks (before the bracket) and the standard deviation of the 50 estimates (in the bracket). The most accurate estimates are highlighted in yellow.

Abbreviations: d, distance from one feature to a linear combination of other features; m, number of different initial values used for each NMF.

### Table 2
Average computation time of the 50 replicates in hours when the true rank is 2 for Poisson NMF data when bootstrap size is 50.

| d     | 0      | 0.0005 | 0.0008 | 0.001  | 0.0015 | 0.002  |
|-------|--------|--------|--------|--------|--------|--------|
| m = 1 | 0.2    | 0.3    | 0.4    | 0.5    | 0.4    | 0.5    |
| m = 3 | 0.8    | 1.2    | 1      | 1.3    | 1.7    | 1.6    |
| **Boot-test** | m = 10 | 1.8    | 4.6    | 3.3    | 3.1    | 4.2    | 3.3    |
| m = 30 | 5.3    | 9.3    | 9.5    | 9      | 8.5    | 9.2    |
| m = 50 | 9      | 22.8   | 17.5   | 18.4   | 19     | 17.2   |
| **Decon-boot-test** | 1     | 2      | 2      | 2      | 2      | 2      |
| **NNLM** | 1     | 2      | 2      | 2      | 2      | 2      |

Note: The average computation time for NNLM is 22 h and approximately the same for different scenarios.

Abbreviations: d, distance from one feature to a linear combination of other features; m, number of different initial values used for each NMF.

Tables 1, 3 and 5. To compare the computational costs of the Boot-test and Decon-boot-test, we recorded the average time required to calculate the estimated rank for 50 replicates in each scenario in Tables 2, 4 and 6. The average computation time for NNLM is 22 hours and is approximately the same for all scenarios. All methods are coded in R (64-bit 4.0.3) and run on 1 node of the Graham cluster of Compute Canada with 1 Intel Xeon at 2.1Ghz with 2GB of RAM and 1 core. From these tables, we can see the Decon-boot-test method achieves a good compromise in both accuracy and speed.

To compare the ability of Decon-boot-test and NNLM to detect Poisson data without NMF structure, we design a no NMF structure Poisson data simulation. The dimension of the no NMF structure Poisson data is the same as Person 1’s gut data from the moving picture dataset, which has 1864 variables and 131 observations. In each replicate, each column of the Poisson mean of the no NMF structure Poisson data is generated from a log-multivariate normal distribution. The mean of the multivariate normal is a vector of length 1864 whose entries are independently simulated following a normal distribution with mean 4 and standard deviation 3. Each sample from the multivariate normal distribution is rescaled so that the sum of all elements in the vector of length 1864 is equal to the log of the total sequencing depth from the corresponding sample in Person 1’s gut data. The covariance matrix of the multivariate normal is generated by 1864 eigenvalues evenly spaced from $3 \times 10^{-7}$ to 1 and orthogonal eigenvectors uniformly distributed over the 1863-dimensional sphere. If the Poisson data doesn’t have NMF structure (which can be also thought of as having rank equal to the sample size), then rank selection methods should select a high rank (which doesn’t satisfy the NMF rank selection rule $k < \frac{n \times p}{n + p}$). The average estimated ranks and standard deviations for the 50 replicates of the two methods are shown in Table 5.

The Boot-test results in these tables indicate that for Poisson NMF data, the accuracy of Boot-test estimates increases with the number of initial values used for each NMF when $d = 0.0005$ and $d = 0.0008$ and is stable in most other cases. The performance of Boot-test grows slowly as the separation $d$ of the types increases. Interestingly, for small $d$, Boot-test often selects too large a rank, meaning that, even though the data can be well approximated with three types, the Boot-test
Table 3: Total number of times the true rank is selected out of 50 replicates and the 50 rank estimates' averages and standard deviations when the true rank is 4 for Poisson NMF data when bootstrap size is 50.

| d   | 0  | 0.0005 | 0.0008 | 0.001 | 0.0015 | 0.002 |
|-----|----|--------|--------|-------|--------|-------|
| m = 1 | 45 | 3.1 (0.30) | 10 | 3.28 (0.54) | 14 | 3.66 (0.85) | 18 | 4 (0.81) | 35 | 4.2 (0.57) | 42 | 4.14 (0.45) |
| m = 3 | 45 | 3.12 (0.39) | 16 | 3.48 (0.65) | 31 | 4.24 (0.62) | 38 | 4.22 (0.51) | 43 | 4.14 (0.35) | 43 | 4.14 (0.35) |
| Boot-test | m = 10 | 39 | 3.22 (0.42) | 30 | 3.92 (0.63) | 34 | 4.26 (0.44) | 34 | 4.28 (0.50) | 39 | 4.17 (0.38) | 42 | 4.16 (0.37) |
| m = 30 | 40 | 3.24 (0.52) | 35 | 4.24 (0.69) | 39 | 4.28 (0.57) | 41 | 4.2 (0.45) | 42 | 4.16 (0.37) | 41 | 4.18 (0.39) |
| m = 50 | 42 | 3.22 (0.58) | 39 | 4.24 (0.59) | 38 | 4.28 (0.54) | 37 | 4.28 (0.50) | 42 | 4.2 (0.49) | 43 | 4.16 (0.42) |
| Decon-boot-test | 50 | 3 (0) | 27 | 3.54 (0.50) | 50 | 4 (0) | 50 | 4 (0) | 50 | 4 (0) | 50 | 4 (0) |
| NNLM | 50 | 3 (0) | 18 | 3.76 (0.77) | 37 | 4.14 (0.50) | 43 | 4.06 (0.37) | 49 | 4.02 (0.14) | 48 | 4.04 (0.20) |

Note: First column for each value of d in grey: the number of times the true rank is selected out of 50 replicates. Second column for each value of d: the average of the 50 estimated ranks (before the bracket) and the standard deviation of the 50 estimates (in the bracket). The most accurate estimates are highlighted in yellow.

Abbreviations: d, distance from one feature to a linear combination of other features; m, number of different initial values used for each NMF.

Table 4: Average computation time of the 50 replicates in hours when the true rank is 4 for Poisson NMF data when bootstrap size is 50.

| d   | 0  | 0.0005 | 0.0008 | 0.001 | 0.0015 | 0.002 |
|-----|----|--------|--------|-------|--------|-------|
| m = 1 | 0.6 | 0.7 | 0.8 | 0.8 | 1 | 1.2 |
| m = 3 | 2 | 2.1 | 2.6 | 2.5 | 2.2 | 2.2 |
| Boot-test | m = 10 | 9 | 9.4 | 13.2 | 13.4 | 12 | 11.8 |
| m = 30 | 26.4 | 31.2 | 32.4 | 28.4 | 27.7 | 28 |
| m = 50 | 44 | 53 | 55.3 | 55.1 | 56.2 | 54 |
| Decon-boot-test | 3.5 | 3.8 | 4.2 | 4.2 | 4.8 | 4.5 |

Note: The average computation time for NNLM is 22 h and approximately the same for different scenarios.

Abbreviations: d, distance from one feature to a linear combination of other features; m, number of different initial values used for each NMF.

The computation time of Boot-test increases with the number of different initial values used for each NMF (m). We would expect this increase to be linear, since the computation time is almost entirely from optimizing the NMF. Indeed, the computation time appears to follow a linear trend, though the variance is fairly high. Decon-boot-test's computation time is between the time taken by Boot-test for m = 3 and m = 10. Thus, Decon-boot-test is both faster and more accurate than Boot-test with m = 10. The time taken by Decon-boot-test increases as the true rank increases, which is to be expected, because of the sequential nature of the test. However, even when the true rank is 10, Decon-boot-test is faster than NNLM.
Table 5: Total number of times the true rank is selected of 50 replicates and the 50 rank estimates’ averages and standard deviations when the true rank is 6, 8 and 10 for Poisson NMF data when bootstrap size is 50.

| Rank | 6     | 8     | 10    | 30    | nonNMF |
|------|-------|-------|-------|-------|--------|
|      | m = 1 | m = 3 | m = 10 | m = 30 | m = 50 |
| Boot-test |       |       |       |       |        |
| m = 1 | 43    | 43    | 41    | 36    | 10.24  |
|       | 6.16(0.42) | 6.14(0.35) | 6.18(0.39) | 8.24(0.43) | 10.18(0.39) |
| m = 3 | 38    | 38    | 39    | 44    | 10.12  |
|       | 8.22(0.42) | 8.26(0.44) | 8.24(0.43) | 10.24(0.43) | 10.2(0.40) |
| m = 10 | 41    | 41    | 37    | 38    | 10.24  |
|       | 6.18(0.39) | 6.32(0.47) | 8.26(0.44) | 10.24(0.43) | 10.2(0.40) |
| m = 30 | 34    | 34    | 34    | 34    | 10.22  |
|       | 8.32(0.47) | 8.32(0.47) | 8.32(0.47) | 10.22(0.47) | 10.2(0.40) |
| m = 50 | 38    | 38    | 38    | 38    | 10.22  |
|       | 3.32(2.95) | 8.32(0.47) | 8.32(0.47) | 10.22(0.47) | 10.2(0.40) |
| Decon-boot-test | 50    | 50    | 47    | 0     |        |
|       | 6 (0) | 8 (0) | 10.06(0.24) | 25.04(1.4) | 39.69(8.61) |
| NNLM | 50    | 50    | 47    | 0     |        |
|       | 6 (0) | 8 (0) | 10 (0) | 2.76 (1.76) | 1 (0) |

Note: First column under each ranking rey: the number of times the true rank is selected out of 50 replicates. (first column is not available for nonNMF data). Second column under each rank: the average of the 50 estimated ranks (before the bracket) and the standard deviation of the 50 estimates (in the bracket). The most accurate estimates are highlighted in yellow.

Abbreviation: m, number of different initial values used for each NMF.

Table 6: Average computation time of the 50 replicates in hours when the true rank is 6, 8 and 10 for Poisson NMF data when bootstrap size is 50.

| Rank | 6     | 8     | 10    |
|------|-------|-------|-------|
|      | m = 1 | m = 3 | m = 10 |
| Boot-test |       |       |       |
| m = 1 | 2.8   | 3.6   | 6     |
| m = 3 | 6     | 11.5  | 20    |
| m = 10 | 19.2  | 37    | 55.2  |
| m = 30 | 63.9  | 102   | 207.6 |
| m = 50 | 101.3 | 209.9 | 323.5 |
| Decon-boot-test | 8.1   | 12.5  | 17.2  |

Note: The average computation time for NNLM is 22 h and approximately the same for different scenarios.

Abbreviation: m, number of different initial values used for each NMF.

3.2 Normal data simulation

We generate normal NMF data with true rank equal to 2, 3, 4, 6, 8 and 10. For each scenario, we simulate 50 replicates. Each element in the feature matrix for the normal NMF data is independently sampled from a gamma distribution with shape parameter 3 and rate parameter 2, then multiplied by a Bernoulli random variable with parameter 0.7 to control the sparsity of the feature matrix. All replicates in the same scenario share the same feature matrix. The dimension of the features is 2780 initially. After removing rows with all zero elements in the feature matrix, there are 2544 rows remaining in the rank 2 feature matrix and 2721 rows in the feature matrices for other ranks.

The 100 columns of the weight matrix W are generated independently from a Dirichlet distribution with parameters all set to be 1.5 and rescaled by a factor of 10.

Each entry of the data matrix is simulated following a normal distribution with mean given by the corresponding entry in the matrix product TW and variance 1. Negative entries in the data are replaced by 0 to generate nonnegative data.

Table 7 shows the results for normal NMF data. As in the Poisson case, the boot-test’s accuracy increases when more sets of initial values are used for each NMF, but even with m = 50, it is unable to select the true rank as reliably as Decon-boot-test. Both Decon-boot-test and NNLM estimate the true ranks for normal NMF data with near-perfect accuracy for all true ranks in this range. We see that Boot-test tends to overestimate the rank slightly.

Table 8 shows the average time required by Decon-boot-test to calculate the estimated rank for the 50 replicates in each normal NMF data scenario is usually between the computational expense of m = 3 and m = 10 Boot-test. The average calculation time for NNLM is 1.6 h for each scenario. The running time of Decon-boot-test increases with the true rank of the data, as is expected because of the sequential nature of the method. We see that NNLM is much faster for the normal NMF than the Poisson NMF, whereas Decon-boot-test is only slightly faster in the normal case. This means that
TABLE 7  Total number of times the true rank is selected out of 50 replicates for normal NMF data and the 50 rank estimates' averages and standard deviations when bootstrap size is 50.

| Rank | 2 | 3 | 4 | 6 | 8 | 10 |
|------|---|---|---|---|---|----|
|      |   |   |   |   |   |    |
|      |   |   |   |   |   |    |
|      |   |   |   |   |   |    |
|      |   |   |   |   |   |    |
| m = 1 | 47 | 2.1 (0.46) | 40 | 3.02 (0.14) | 39 | 4.22 (0.42) | 40 | 6.2 (0.40) | 42 | 8.16 (0.37) | 43 | 10.14 (0.35) |
| m = 3 | 41 | 2.22 (0.55) | 38 | 3.24 (0.43) | 42 | 4.18 (0.44) | 44 | 6.12 (0.33) | 41 | 8.18 (0.39) | 46 | 10.08 (0.27) |
| Boot-test | m = 10 | 42 | 2.2 (0.49) | 40 | 3.12 (0.33) | 42 | 4.16 (0.37) | 42 | 6.16 (0.37) | 47 | 8.06 (0.24) | 41 | 10.18 (0.39) |
| m = 30 | 44 | 2.14 (0.40) | 42 | 3.22 (0.46) | 42 | 4.16 (0.37) | 43 | 6.16 (0.42) | 45 | 8.1 (0.30) | 43 | 10.14 (0.35) |
| m = 50 | 46 | 2.08 (0.27) | 42 | 3.16 (0.37) | 46 | 4.08 (0.27) | 43 | 6.14 (0.35) | 42 | 8.16 (0.37) | 42 | 10.16 (0.37) |
| Decon-boot-test | 50 | 2 (0) | 49 | 3.02 (0.14) | 50 | 4 (0) | 50 | 6 (0) | 49 | 8.02 (0.14) | 50 | 10 (0) |
| NNLM | 50 | 2 (0) | 50 | 3 (0) | 50 | 4 (0) | 50 | 6 (0) | 50 | 8 (0) | 50 | 10 (0) |

Note: First column under each ranking row: the number of times the true rank is selected out of 50 replicates. Second column under each rank: the average of the 50 estimated ranks (before the bracket) and the standard deviation of the 50 estimates (in the bracket). The most accurate estimates are highlighted in yellow. Abbreviation: m, number of different initial values used for each NMF.

TABLE 8  Average computation time of the 50 replicates in hours for normal NMF data when bootstrap size is 50.

| Rank | 2 | 3 | 4 | 6 | 8 | 10 |
|------|---|---|---|---|---|----|
|      |   |   |   |   |   |    |
|      |   |   |   |   |   |    |
|      |   |   |   |   |   |    |
|      |   |   |   |   |   |    |
| m = 1 | 0.1 | 0.5 | 0.5 | 1 | 2.1 | 3.5 |
| m = 3 | 0.4 | 1.3 | 1.3 | 3 | 7.8 | 10.2 |
| Boot-test | m = 10 | 2.7 | 2.5 | 4.2 | 10 | 25.8 | 33.5 |
| m = 30 | 3.6 | 6.3 | 11.7 | 28.7 | 69.7 | 95.7 |
| m = 50 | 5.7 | 12.9 | 23 | 53.6 | 104 | 186.5 |
| Decon-boot-test | 1.5 | 2 | 3 | 5 | 8 | 12.5 |

Abbreviation: m, number of different initial values used for each NMF.

in cases where the true rank is in the range 6–10, for normal NMF, NNLM may be slightly faster than Decon-boot-test for estimating the rank.

3.3 | Conclusions from simulations

The simulation results show that Decon-boot-test is better than Boot-test in estimation accuracy for both Poisson data and normal data. By using a large number of starting points, Boot-test can achieve similar accuracy to Decon-boot-test, but is far more computationally expensive with no benefit in accuracy. When the true rank is small, the performance of Decon-boot-test for normal data is comparable with NNML, and for Poisson data, its performance is better than NNML, especially when the features are hard to distinguish from each other. When the rank is large, both methods fail to find the true rank but Decon-boot-test’s estimated rank is much closer to the true rank. When the data is not NMF structured, Decon-boot-test selects very large rank, while NNLM selects rank 1 for all replicates. Thus, Decon-boot-test gives a much clearer indication of the lack of NMF structure, which can be used to better interpret the data. The computation time for Decon-boot-test increases with the true rank of the data, but for Poisson NMF, it is generally faster than NNLM, even when the rank is as high as 10, which is the largest rank that NNLM was able to accurately estimate in our simulation. For normal NMF, NNLM is faster, but the computation times of the methods are comparable.

4 | REAL DATA APPLICATION

We apply our NMF rank selection method to the Moving Picture data. As mentioned in Section 3.1, this dataset records the microbiome of two people’s gut, tongue, left and right palm for a long time period. Person 1 was female, and was sampled over a period of 186 days, with a total of 131–135 samples for each body site. Person 2 was male and was sampled...
over a period of 446 days, with a total of 336–373 samples for each body site. The total number of variables (different OTUs) across all samples was more than 15,000. In spite of this extensive sampling, no temporal core microbiome was detected, with only a small subset of OTUs reoccurring at the same body site across all time points. These data sets have previously been analysed using NMF, but without a good way to select the rank. It is possible that there are undetected biologically important patterns in the data that can be discovered by reanalysing the data using the number of types selected by the Decon-boot-test method. We therefore apply our method to the gut data from both individuals to choose the number of types, and examine the fitted types and weights to obtain biological insights that were not observed in previous analyses of the data using a different number of types. Our method selects NMF rank as 12 for Person 1’s gut data and NMF rank as 6 for Person 2’s gut data.

To examine the microbiome community structure of the gut from the NMF results, we plot the relative abundance of each genus in each feature in Figure 2 for Person 1’s gut and Figure 4 for Person 2’s gut. The elements of each feature sum to 1 so the coefficients can be interpreted as the proportions of each OTU in that feature. As the NMF results are usually highly sparse, we use a cut-off of 3% for each type. That is, only those genera with OTUs above 3% composition in at least one type are shown in the plot. The outstanding genera for features of both persons’ guts are from the phyla Firmicutes, Bacteroidetes, Actinobacteria, Proteobacteria, Fusobacteria and Verrucomicrobia. Most of the features have abundant Firmicutes and Bacteroidetes which is consistent with the fact that Firmicutes and Bacteroidetes constitute 90% of the gut microbiota, while Actinobacteria, Proteobacteria, Fusobacteria, and Verrucomicrobia contribute to the microbial population to a lesser extent.

Figure 2 shows the abundance of the main genera in Person 1’s gut features. We find only 30 out of more than 1800 OTUs are larger than the cut-off of 3%. These 30 OTUs are from 17 genera. From the plot, we see Types 2, 4, 5 and 6 share similar compositions at genus level. They all have extremely high abundance of Bacteroides and much lower abundance of other genera. These types differ in the compositions of other genera. In particular, the secondary genus in the community is Faecalibacterium in Type 4, Phascolarctobacterium in Type 2, an unclassified genus from order Bacteroides in Type 5 and Escherichia in Type 6. Type 10 is dominated by Akkermansia rather than Bacteroides. As lean individuals have more Bacteroides, while obese individuals have more Firmicutes in their intestinal microbiota and the abundance of Akkermansia is negatively correlated to human’s weight. Type 2, Type 4, Type 5, Type 6 and Type 10 may be healthy communities with regard to body weight. Type 1 mainly consists of Escherichia and an unclassified genus from order Bacteroidales. We know that a number of species of Escherichia are pathogenic so this type could be associated with
infection with pathogens. Type 8’s major genus is Prevotella whose increased abundance is linked to inflammatory disorders in emerging studies, suggesting that at least some strains exhibit pathobiontic properties. After an intervention with a Mediterranean diet and a low-fat diet, increased Prevotella and Bacteroides levels have also been observed, so Type 8 could be associated with a short-term change of diet. Less abundant genera in Type 8 such as Finegoldia, Staphylococcus, Lactobacillus, Corynebacterium and Fusobacterium have, on rare occasions, been found to cause human infection. Therefore Type 8 could also be associated with infection. Type 3 consists of Bacteroides, Lachnospira and Akkermansia. Type 7 has high proportions of Faecalibacterium and an unclassified genus from family Lachnospiraceae. Type 11 and Type 12 also have high proportions of Faecalibacterium while Type 11 has fairly abundant Lachnospira and Type 12 has fairly abundant Roseburia. Previous studies suggest that Roseburia and Lachnospira are strongly associated with vegetable diets, and also negatively associated with the omnivore diet. Furthermore, Faecalibacterium and Roseburia may be two major proponents of weight loss. This suggests that Type 3, Type 7, Type 11 and Type 12 may all be related to low-fat diets. Type 9 has fairly abundant Phascolarctobacterium which has been found to increase under high fat diet and to be positively correlated to a positive mood and body weight.
To investigate the temporal dynamics of Person 1’s gut microbiome, the weight matrix is plotted in Figure 3. The weights for each time point are normalized to sum to 1. We see that there is substantial short-term fluctuation for all types. Types 6 and 8 have consistently low weights, with the exception of single-day spikes. This would be consistent with Types 6 and 8 being associated with short-term infections. It could also be consistent with Type 8 being associated with diet, if the association is short-term, then the spikes might correspond to the individual eating particular foods. Type 9 is dominant with a relatively stable weight for around one week at end of December which would be consistent with the diet change and happy mood. Other types show elevated levels for time periods of weeks or months, indicating changes to the microbial community over time. This seems plausible dynamics for healthy types.

Figure 4 shows the abundance of the main genera in Person 2’s gut features. 33 out of 3131 OTUs have larger than 3% proportions in at least one feature. These 33 OTUs are from 20 genera. Person 2’s Type 3, Type 5 and Type 6 are similar. They are all dominated by Bacteroides with the next most abundant genera, Faecalibacterium and Parabacteroides, appearing in much smaller proportions. Type 4 has high proportions of Bacteroides and Akkermansia. Type 1 is more diverse, with no genus exceeding 20% of the total abundance of this type. The most abundant genera are Prevotella, Escherichia, Finegoldia and Porphyromonas. Many species from these genera have previously been associated with periodontitis. Although the species in this dataset are not the ones previously associated with periodontitis, it seems plausible that this type might be associated with periodontic infections. Type 2 consists of Streptococcus (gordonii species and oralis species), Prevotella (nanceiensis species, melaninogenica species and an unclassified species), Neisseria (subflava species) and a lower abundance of Staphylococcus (epidermidis species), Veillonella (parvula species and an unclassified species), an unclassified genus from family Pasteurellaceae, Haemophilus (parainfluenzae species) and Fusobacterium (unclassified species). The Prevotella species in Type 2 are different from the species in Type 1. S. gordonii, V. parvula and Fusobacterium are known to be opportunistic pathogens that can cause periodontitis. Also, S. gordonii, V. parvula, H. parainfluenzae and N. subflava are recognized in opportunistic infections such as endocarditis and meningitis. P. melaninogenica is also an important human pathogen in various anaerobic infections, often mixed with other aerobic and anaerobic bacteria. Thus, Type 2 might also be associated with infection.

Figure 5 shows the weights of each type in Person 2’s gut over time. We see that Types 1 and 2 are usually very low abundance, but have occasional spikes. This is consistent with these types corresponding to infections. The other features show much more stability over time, having similar abundance levels over long periods of time (when normalized with
FIGURE 5  Gut weight matrix time series plot for Person 2. The top plot shows Person 2’s gut weight matrix normalized on Type 3, Type 4, Type 5 and Type 6 from rank 6 NMF. The bottom plot shows the weight matrix for other types. Each weight is colored the same as its corresponding type.

respect to the total of these four types, with Types 1 and 2 excluded). There are some very abrupt changes to the abundance levels of these types at various times during the study. This is consistent with these types representing small variations on a healthy community.

5  CONCLUSION

We have developed an NMF rank selection method based on hypothesis testing using a deconvolved bootstrap to estimate the null distribution. The simulations show that our rank selection method can estimate the rank accurately for both Poisson and normal NMF data when the true rank is small. When the NMF features are close or the true NMF rank is large, our method has better performance than NNLM. We applied our method to a microbiome data set. With the number of ranks selected, we were able to gain new insights beyond previous analyses of the same data. The new insights are biologically extremely plausible, and will hopefully lead to new research in the field.
There are a number of directions for future research. Firstly, the sequential nature of the test means that we need to recompute the null distribution for each rank, which is computationally expensive. The test could be made computationally more efficient for large ranks by using a more efficient search. Instead of increasing the tested rank by 1 each time, we could increase it by more. If the test does not reject the null hypothesis, it would then be necessary to perform the tests for lower ranks. This would increase computation when the rank is low, but could decrease it significantly for higher rank. More heuristics could be added to decide which hypothesis tests to perform for maximum efficiency.

Another issue is that a single failure to reject the null hypothesis can cause the method to select the current rank. It might be possible to develop a more robust method that combines the output of additional hypothesis tests to estimate the rank more reliably.

Another direction for future research is estimation of variance for the parametric bootstrap in the normal case. The variance is estimated from the residuals of the model. However, because the model is fitted to the data, these residuals will be smaller when the rank is larger. For linear regression, there is a correction to get an unbiased estimate for the variance. However, for NMF, the nonnegativity constraint means that this correction is not applicable, so another method is needed to obtain a more stable estimate for the variance. This will make the critical value of the test based on the parametric bootstrap more accurate.

The application of deconvolution to deal with optimization errors in bootstrap samples has potential applications in any field where full optimization is computationally expensive. This is common for discrete optimization problems such as variable selection, clustering, phylogenetics and many other areas. Applying the deconvolved bootstrap in these areas could prove a very fruitful topic for future research.

AUTHOR CONTRIBUTIONS
All three authors contributed to the development of methodology, design of the simulations, interpretation of the real data analysis, and writing of the paper. All three authors contributed to the implementation of the method. All authors read and approved the final manuscript.

FUNDING INFORMATION
The second author is supported by NSERC grant RGPIN-2017-05108. The third author is supported by NSERC grant RGPIN/04945-2014.

CONFLICT OF INTEREST STATEMENT
The authors declare no potential conflict of interests.

DATA AVAILABILITY STATEMENT
The data analyzed in this manuscript have all been previously published in other papers. The Moving picture data set is in Qiita study 550. The Qin data set is available in the format used for our analysis with the BiomeNet package at http://sourceforge.net/projects/biomenet/.

ORCID
Yun Cai  https://orcid.org/0000-0001-8163-4694
Toby Kenney  https://orcid.org/0000-0002-5421-4325

REFERENCES
1. Lee DD, Seung HS. Learning the parts of objects by non-negative matrix factorization. Nature. 1999;401(6755):788-791.
2. Wu H, Marmoret A, Cohen JE. Semi-supervised Convolutive NMF for automatic music transcription. arXiv preprint arXiv:2202.04989. 2022.
3. Brunet JP, Tamayo P, Golub TR, Mesirov JP. Metagenes and molecular pattern discovery using matrix factorization. Proc Natl Acad Sci. 2004;101(12):4164-4169.
4. Boccarelli A, Del Buono N, Esposito F. Analysis of fibroblast genes selected by NMF to reveal the potential crosstalk between ulcerative colitis and colorectal cancer. Exp Mol Pathol. 2021;123(104713):104713-104723.
5. Cai Y, Gu H, Kenney T. Learning microbial community structures with supervised and unsupervised non-negative matrix factorization. Microbiome. 2017;5(1):1-27.
6. Zhao J, Ma Y, Liu L. Microbiome data analysis by symmetric non-negative matrix factorization with local and global regularization. Front Mol Biosci. 2021;8(643014):643014-643021.
Gilbert JA, Steele JA, Caporaso JG, et al. Defining seasonal marine microbial community dynamics. *Isme J.* 2012;6(2):298-308.

Jiang X, Langille MG, Neches RY, et al. *Functional Biogeography of Ocean Microbes Revealed through Non-negative Matrix Factorization.* USA: Public Library of Science San Francisco; 2012.

Ko YJ, Kim S, Pan CH, Park K. Identification of functional microbial modules through network-based analysis of meta-microbial features using matrix factorization. *IEEE/ACM Trans Comput Biol Bioinform.* 2021;19(5):2851-2862.

Seung HS, Lee DD. Algorithms for non-negative matrix factorization. *Adv Neural Inf Process Syst.* 2001;13:556-562.

R Core Team. R: a language and environment for statistical computing. R foundation for statistical Computing Secure. https://www.R-project.org 2018.

Laurberg H, Christensen MG, Plumbley MD, Hansen LK, Jensen SH. Theorems on positive data: on the uniqueness of NMF. *Comput Intell Neurosci.* 2008;2008:764206.

Chu MT, Diele F, Plemmons RJ, Ragni S. Optimality, computation, and interpretation of nonnegative matrix factorizations. https://mtchu.math.ncsu.edu/Research/Papers/nnmf.pdf 2004.

Gillis N. The why and how of nonnegative matrix factorization. *Regul Optim Kernels Support Vector Mach.* 2014;12(257):257-291.

Muzzarelli L, Weis S, Eickhoff SB, Patil KR. Rank selection in non-negative matrix factorization: systematic comparison and a new MAD metric. Paper presented at: IEEE 2019 International Joint Conference on Neural Networks (IJCNN). 2019 1-8.

Kim H, Park H. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics.* 2007;23(12):1495-1502.

Frigyesi A, Höglund M. Non-negative matrix factorization for the analysis of complex gene expression data: identification of clinically relevant tumor subtypes. *Cancer Inform.* 2008;6:CIN–S606.

Squires S, Prügel-Bennett A, Niranjan M. Rank selection in nonnegative matrix factorization using minimum description length. *Neural Comput.* 2017;29(9):2164-2176.

Hoffman MD, Blei DM, Cook PR. Bayesian nonparametric matrix factorization for recorded music. ICML’10. Proceedings of the 27th International Conference on International Conference on Machine Learning. Omnipress, Madison, WI, USA. 2010 439–446.

Sotiras A, Toledo JB, Gur RE, Gur RC, Satterthwaite TD, Davatzikos C. Patterns of coordinated cortical remodeling during adolescence and their associations with functional specialization and evolutionary expansion. *Proc Natl Acad Sci.* 2017;114(13):3527-3532.

Lin X, Boutros PC. Optimization and expansion of non-negative matrix factorization. *Bioinform.* 2020;21(1):1-10.

Cai Y, Gu H, Kenney T. Deconvolution density estimation with penalised MLE. arXiv preprint. https://arxiv.org/abs/2105.11205 2021.

Qin J, Li R, Raes J, et al. A human gut microbial gene catalogue established by metagenomic sequencing. *Nature.* 2010;464(7285):59-65.

Caporaso JG, Lauber CL, Costello JE, et al. Moving pictures of the human microbiome. *Genome Biol.* 2011;12(5):1-8.

Scarpellini E, Ianiro G, Attili F, Bassanelli C, De Santis A, Gasbarrini A. The human gut microbiota and virome: potential therapeutic implications. *Dig Liver Dis.* 2015;47(12):1007-1012.

Turnbaugh PJ, Ley RE, Mahowald MA, Magrini V, Mardis ER, Gordon JI. An obesity-associated gut microbiome with increased capacity for energy harvest. *Nature.* 2006;444(7122):1027-1031.

Zhou Q, Zhang Y, Wang X, et al. Gut bacteria Akkermansia is associated with reduced risk of obesity: evidence from the American gut project. *Nutrit Metabol.* 2020;17(1):1-9.

Hogan CM. Bacteria. Encyclopedia of Earth. Washington DC: National Council for Science and the Environment; 2010.

Larsen JM. *The immune response to Prevotella bacteria in chronic inflammatory disease.* *Immunology.* 2017;151(4):363-374.

Preocup G, Vodnar DC. Gut Prevotella as a possible biomarker of diet and its eubiotic versus dysbiotic roles: a comprehensive literature review. *Br J Nutr.* 2019;122(2):131-140.

Cobo F. Infections caused by anaerobic microorganisms. In: Rezaei N, ed. Encyclopedia of Infection and Immunity. Amsterdam, Netherlands: Elsevier; 2022:641-627.

Lee E, Anjum F. *Staphylococcus epidermidis.* In: Aboubaker S, Abu-Ghosh A, Acharya AB, et al., eds. *StatPearls [Internet].* Treasure Island, FL: StatPearls Publishing; 2021 https://www.ncbi.nlm.nih.gov/books/NBK563240/

Rossi F, Amadoro C, Colavita G. Members of the lactobacillus genus complex (LGC) as opportunistic pathogens: a review. *Microorganisms.* 2019;7(5):126.

Bernard K. The genus Corynebacterium and other medically relevant coryneform-like bacteria. *J Clin Microbiol.* 2012;50(10):3152-3158.

Afka K, Laupland K, Leal J, Lloyd T, Gregson D. Incidence, risk factors, and outcomes of fusobacterium species bacteremia. *BMJ Infect Dis.* 2013;13(1):1-6.

Vacca M, Celano G, Calabrese FM, Portincasa P, Gobbetti M, De Angelis M. The controversial role of human gut lachnospiraceae. *Microorganisms.* 2020;8(4):573.

Chakraborti CK. New-found link between microbiota and obesity. *World J Gastrointest Pathophysiol.* 2015;6(4):110.

Li L, Su Q, Xie B, et al. Gut microbes in correlation with mood: case study in a closed experimental human life support system. *Neurogastroenterol Motility.* 2016;28(8):1233-1240.

Lecomte V, Kaakoush NO, Maloney CA, et al. Changes in gut microbiota in rats fed a high fat diet correlate with obesity-associated metabolic parameters. *PloS One.* 2015;10(5):e0126931.

Mei F, Xie M, Huang X, et al. Porphyromonas gingivalis and its systemic impact: current status. *Pathogens.* 2020;9(11):944.

Park JO, Kwon Y, Park C, et al. Streptococcus gordonii: pathogenesis and host response to its cell wall components. *Microorganisms.* 2020;8(12):1852.

Bongaerts G, Schreurs B, Lunel FV, Lemmens J, Pruszczyński M, Merkx M. Was isolation of Veillonella from spinal osteomyelitis possible due to poor tissue perfusion? *Med Hypotheses.* 2004;63(4):659-661.
43. Broadley M, Schweon SJ. Get the facts about fusobacterium. *Nursing*. 2017;47(5):64-65.
44. Latyshev Y, Mathew A, Jacobson JM, Sturm E. Purulent pericarditis caused by haemophilus parainfluenzae. *Tex Heart Inst J*. 2013;40(5):608.
45. Deza G, Martin-Ezquerra G, Gómez J, Villar-Garcia J, Supervia A, Pujol RM. Isolation of haemophilus influenzae and haemophilus parainfluenzae in urethral exudates from men with acute urethritis: a descriptive study of 52 cases. *Sex Transm Infect*. 2016;92(1):29-31.
46. Kaplan JB, Fine DH. Biofilm dispersal of Neisseria subflava and other phylogenetically diverse oral bacteria. *Appl Environ Microbiol*. 2002;68(10):4943-4950.
47. Brook I. *Anaerobic Infections: Diagnosis and Management*. Washington, DC: CRC Press; 2007.

**How to cite this article:** Cai Y, Gu H, Kenney T. Rank selection for non-negative matrix factorization. *Statistics in Medicine*. 2023;42(30):5676-5693. doi: 10.1002/sim.9934