bayesNMF Tutorial

While language and simulation examples are in the context of mutational signatures analysis, this package can be used for any application of NMF.

Quick Start

Installation

```
library(devtools)
devtools::install_github("jennalandy/bayesNMF")
```

```
library(bayesNMF)
```

Use

For example purposes, we use a dataset of 64 samples simulated from 4 signatures.

```
data <- readRDS(system.file("extdata", "example_data.rds", package = "bayesNMF"))
dim(data$M)
#> [1] 96 64
```

The `bayesNMF` R software package allows users to fit all models defined in the paper with the `bayesNMF` function. Model specifications can be adjusted by the `likelihood` (default `"poisson"`) and `prior` (default `"truncnormal"`) parameters.

Our paper introduces computationally efficient MH-within-Gibbs steps (called Poisson+MH models in the paper) using proposal distributions based on the full conditionals from a Normal-likelihood model—this approach is used by default when possible, but can be controlled by setting `MH = FALSE`.

The `rank` parameter can be a fixed value if the latent dimension is known. For example:

```
sampler_fixed_rank <- bayesNMF(
    data$M, rank = 4,
    output_dir = "../examples/fixed_rank",
    overwrite = TRUE
)
```

```
sampler_fixed_rank <- readRDS("../examples/fixed_rank/sampler.rds")
```

If a range vector is provided, rank is learned using the `"SBFI"` appraoch introduced in our paper (controlled by `rank_method`, can also be set to `"BFI"` or `"BIC"` (heuristic)). For example:

```
sampler_learn_rank <- bayesNMF(
    data$M, rank = 1:10,
    output_dir = "../examples/learn_rank",
    overwrite = TRUE
)
```

```
sampler_learn_rank <- readRDS("../examples/learn_rank/sampler.rds")
```

Details

Models

The following models have been implemented and benchmarked in our paper. In general, the Poisson+MH approach can only be used when the prior leads to easily sampled conditionals in the Normal Bayesian NMF, regardless of how the full conditional factors in the Poisson model. For this reason, Gamma priors cannot be used with this setup. Similarly, we cannot implement a standard Gibbs sampler for the Poisson-Truncated Normal model because the full conditionals are not from distributions that can be directly sampled, even with Poisson augmentation. The Poisson-Gamma model is the closest comparison.

Poisson - Truncated Normal (default)

$M_{kg} \sim \text{Poisson}((PE)_{kg})$, $P, E \sim \text{TruncatedNormal}$

Example: `bayesNMF(data$M, rank = 1:10)`

Only implemented with `MH = TRUE`.

Poisson - Exponential

$M_{kg} \sim \text{Poisson}((PE)_{kg})$, $P, E \sim \text{Exponential}$

Example: `bayesNMF(data$M, rank = 1:10, prior = "exponential", MH = FALSE)`

Implemented with `MH = TRUE` by default, but `MH = FALSE` can be specified.

Poisson - Gamma

$M_{kg} \sim \text{Poisson}((PE)_{kg})$, $P, E \sim \text{Gamma}$

Example: `bayesNMF(data$M, rank = 1:10, prior = "gamma")`.

Only implemented with `MH = FALSE` (automatically set).

Normal - Truncated Normal

$M_{kg} \sim \text{Normal}((PE)_{kg}, \sigma^2_{kg})$, $P, E \sim \text{TruncatedNormal}$, $\sigma^2_{kg} \sim \text{InvGamma}$

Example: `bayesNMF(data$M, rank = 1:10, likelihood = "normal")`

Normal - Exponential

$M_{kg} \sim \text{Normal}((PE)_{kg}, \sigma^2_{kg})$, $P, E \sim \text{Exponential}$, $\sigma^2_{kg} \sim \text{InvGamma}$

Example: `bayesNMF(data$M, rank = 1:10, likelihood = "normal", prior = "exponential")`

`bayesNMF` output:

The returned object (the same object saved in `sampler.rds`) is of the R6 Class `bayesNMF_sampler`.

Four files will be created and updated as the sampler progresses in `output_dir`. For improved speed, set `periodic_save = FALSE` to reduce I/O operations to once at the end of the run. For improved memory, users can choose to set `save_all_samples = FALSE` to only save the last 1000 samples for MAP inference.

log.txt

Logs the model specifications and progress of the Gibbs sampler.

Each progress update looks like this:

```
[2025-10-19 15:45:35]   iter = 1200
[2025-10-19 15:45:35]       Computing MAP
[2025-10-19 15:45:35]       1001001010    1001101010    1001001011    1001001110    10010110
                              873            9             8             8             8
[2025-10-19 15:45:35]       Checking convergence
[2025-10-19 15:45:35]       logposterior = 19717.62 | 1.13% change | 0 no change | 0 no best
```

The third and fourth lines here report the mode(s) of the A matrix (signature inclusion matrix). The example above shows that the mode of A is `1001001010`, which corresponds to the inclusion of signatures 1, 4, 7, and 9, and that this matrix appeared 873 times in the last 1000 samples. The second most frequent A matrix is `1001101010`, which corresponds to the inclusion of signatures 1, 4, 5, 7, and 9, and that this matrix appeared 9 times in the last 1000 samples.

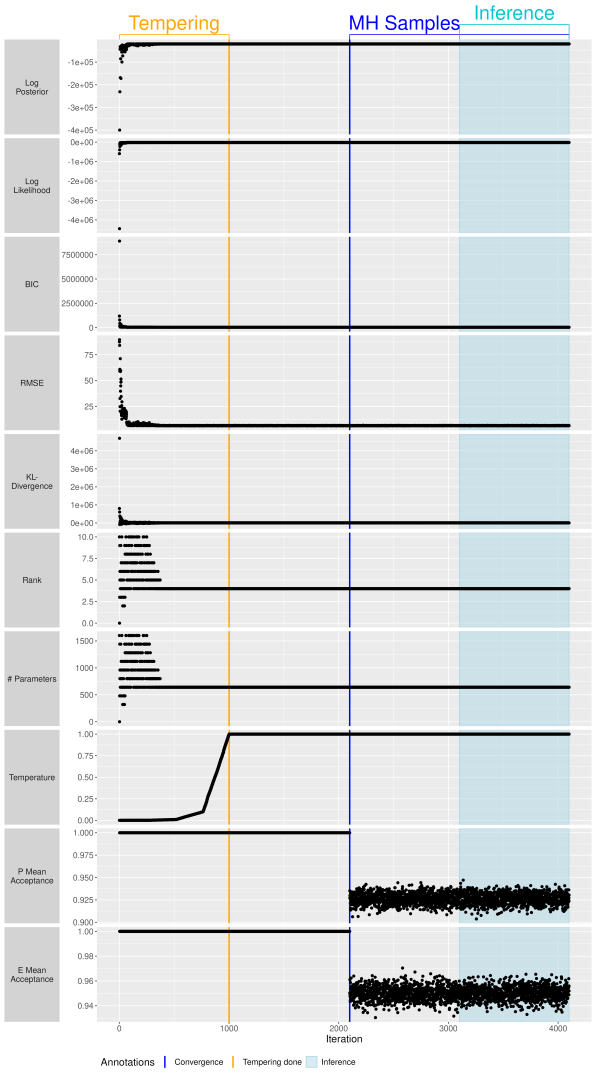
The last line reports the current metric and the percentage change from the previous metric. It also reports how many consecutive progress updates have had no change, no new "best" metric, or NA metric. Here, the metric is log posterior, which has increased by 1.13% since the previous update. The default tolerance for what is a meaningful "change" in MAP metric is 0.1% (controlled by `tol` in `new_convergence_control`), so there have been zero consecutive updates with a meaningful change. This is also an improvement over all previous updates, so there have been zero consecutive updates with no new "best" metric. Finally, this metric is not NA, so there have been zero consecutive updates with an NA metric.

sampler.rds

Periodically records Gibbs sampler object, which is be useful if your run is cut short (the dreaded OOM error). Once the run is complete, this records complete results for future access.

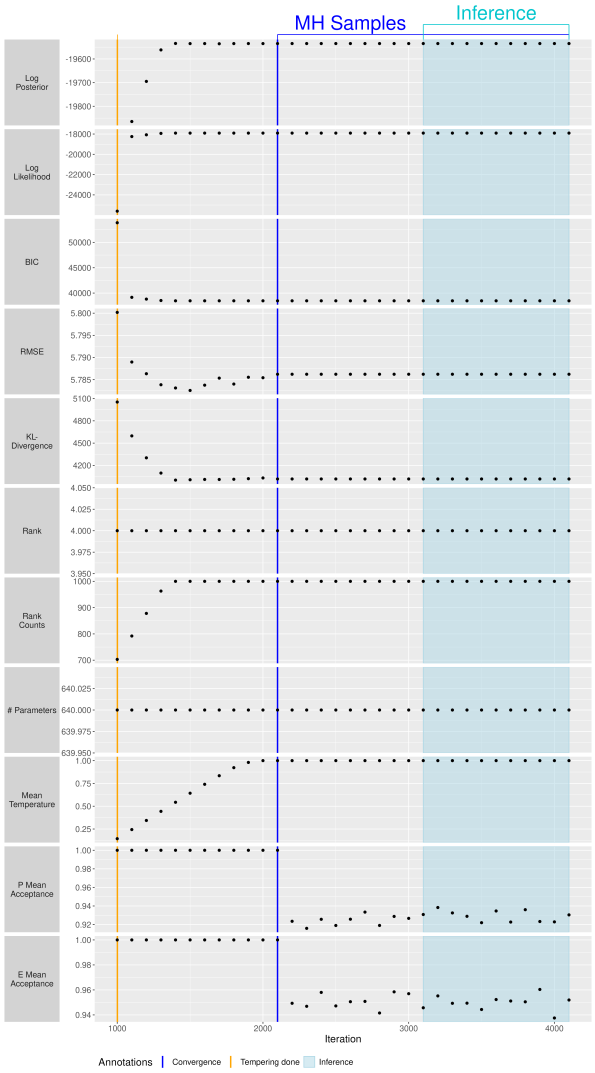
trace_plot.png

Holds trace plots of metrics: log posterior, log likelihood, BIC, RMSE, KL-Divergence. If rank is being learned, this also reports rank, number of functional parameters (used in BIC), and average acceptance rate for elements of P and E (initally all 1 to find high posterior region, then true MH samples for inference). Note that for log likelihood and log posterior, values from the Poisson models are not comparable to those from Normal models.



`trace_plot_MAP.png`

Holds trace plots of periodically computed MAP estimates (see the section on convergence below for details). If rank is being learned, also reports the number of considered samples that contribute to the mode of A (`MAP_A_counts`). Note that for log likelihood and log posterior, values from the Poisson models are not comparable to those from Normal models.



`bayesNMF_sampler` class attributes:

`data`

Data matrix input by user

```
dim(sampler_learn_rank$data)
#> [1] 96 64
```

specs

Model specifications.

```
sampler_learn_rank$specs
#> $rank
#>  [1]  0  1  2  3  4  5  6  7  8  9 10
#>
#> $likelihood
#> [1] "poisson"
#>
#> $prior
#> [1] "truncnormal"
#>
#> $MH
#> [1] TRUE
#>
#> $learning_rank
#> [1] TRUE
#>
#> $convergence_control
#> $convergence_control$MAP_over
#> [1] 1000
#>
#> $convergence_control$MAP_every
#> [1] 100
#>
#> $convergence_control$tol
#> [1] 0.001
#>
#> $convergence_control$Ninarow_nochange
#> [1] 5
#>
#> $convergence_control$Ninarow_nobest
#> [1] 10
#>
#> $convergence_control$miniters
#> [1] 1000
#>
#> $convergence_control$maxiters
#> [1] 5000
#>
#> $convergence_control$minA
```

```
#> [1] 0
#>
#> $convergence_control$metric
#> [1] "logposterior"
#>
#>
#> $output_dir
#> [1] "../examples/learn_rank"
#>
#> $overwrite
#> [1] TRUE
#>
#> $verbosity
#> [1] 1
#>
#> $periodic_save
#> [1] TRUE
#>
#> $save_all_samples
#> [1] TRUE
#>
#> $prop_temp
#> [1] 0.2
#>
#> $rank_method
#> [1] "SBFI"
#>
#> $post_warmup
#> [1] 2000
```

MAP

Maximum a-posteriori (MAP) estimates for P and E, and A if learning rank. It also holds `idx` to indicate the indices of the samples that contribute to the MAP estimates. If learning rank, `A_counts` indicates the number of samples match the mode of A (which is the subset of samples used for inference), and `keep_sigs` indicates the indices of the signatures that are kept in the MAP estimates. MAP estimates are already reduces to kept signatures, but these indices are useful when inspecting individual samples.

```
dim(sampler_learn_rank$MAP$P)
#> [1] 96  4
dim(sampler_learn_rank$MAP$E)
```

```
#> [1]  4 64
sampler_learn_rank$MAP$keep_sigs
#> [1] 2 3 8 9
```

credible_intervals

95% credible intervals for P and E.

```
names(sampler_learn_rank$credible_intervals$P)
#> [1] "lower" "upper"
dim(sampler_learn_rank$credible_intervals$P$lower)
#> [1] 96  4
```

samples

All samples of all parameters and prior parameters.

```
names(sampler_learn_rank$samples)
#>  [1] "P"                 "E"                 "A"
#>  [4] "R"                 "Mu_p"              "Sigmasq_p"
#>  [7] "Mu_e"              "Sigmasq_e"         "P_acceptance_rate"
#> [10] "E_acceptance_rate"
```

Samples are not reduced to kept signatures, so they are full size of the data matrix. This is where `sampler_learn_rank$MAP$keep_sigs` is useful.

```
dim(sampler_learn_rank$samples$P[[1]])
#> [1] 96 10
```

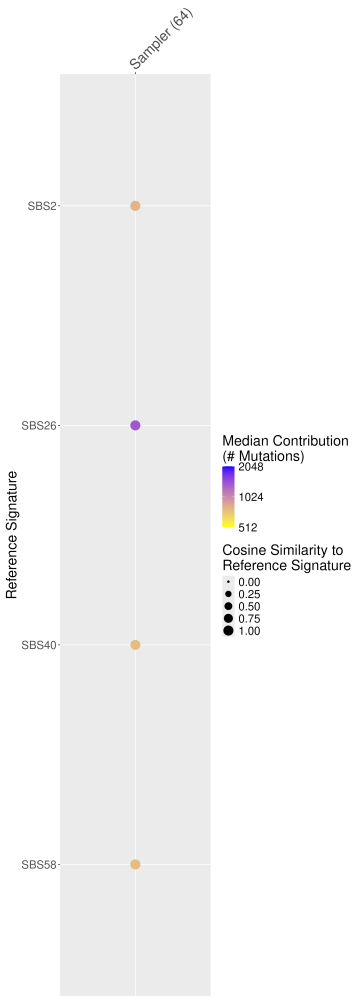
Analysis

The `plot` function creates several plots comparing sampler output to a reference signatures matrix (controlled by `reference_P`, default `"cosmic"` uses COSMIC v3.3.1 SBS GRCh37 signatures, but a matrix can be provided) and saves them to the `output_dir`. Each plot also has a separate function that can be called to create the plot without saving.

```
plot(sampler_learn_rank, sigs = TRUE)
```

summary.png

Plots median contribution to each signature and the cosine similarity between the estimated and reference signatures.



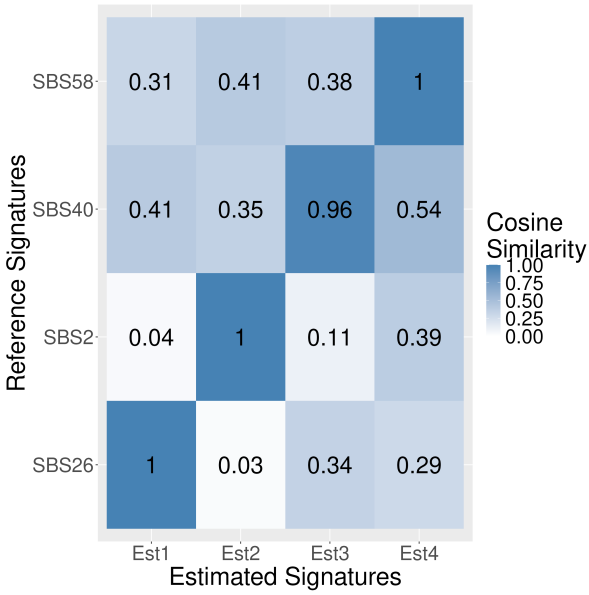
The exact values from this plot can be accessed with the `summary` function.

```
summary(sampler_learn_rank)
#>    G  N  K Signature Med_Contribution Prop_atleast_1 Reference_Signature
#> 1 64 10 96         1        1396.2510              1               SBS26
#> 2 64 10 96         2         799.2390              1                SBS2
#> 3 64 10 96         3         761.1403              1               SBS40
#> 4 64 10 96         4         761.5873              1               SBS58
#>   Cosine_Similarity
#> 1         0.9992944
```

```
#> 2          0.9996010
#> 3          0.9642135
#> 4          0.9992822
```

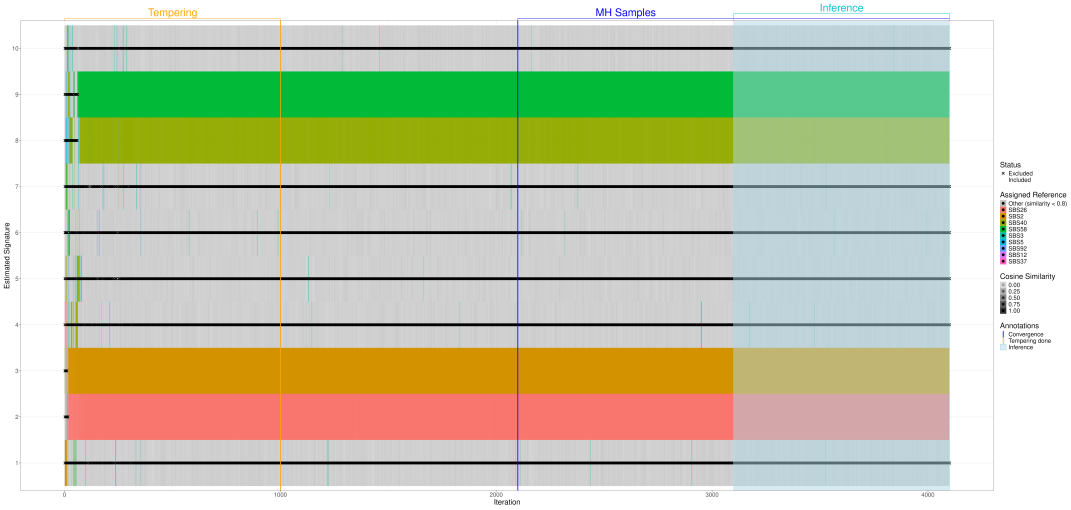
`similarity_heatmap.png`

Heatmap of the cosine similarity between the estimated and assigned reference signatures.



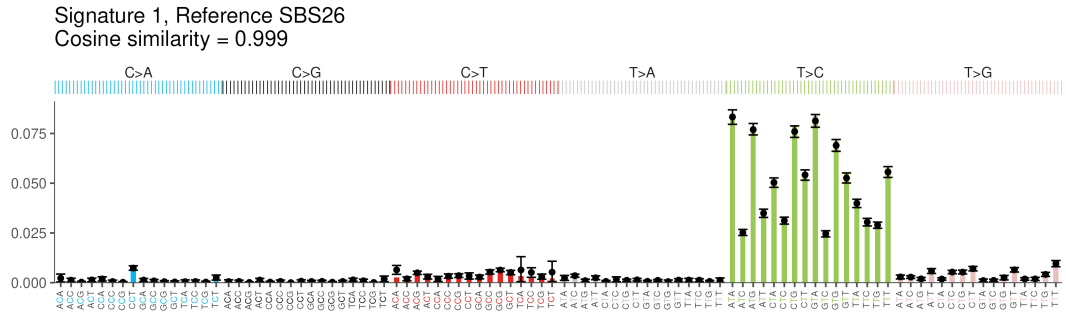
`label_switching.png`

Diagnostic plot of label switching, showing the closest reference match for each signature for each posterior sample. Only created if learning rank and `save_all_samples = TRUE` (default). If label switching occurs, the color of a latent factor will switch part way through the chain.



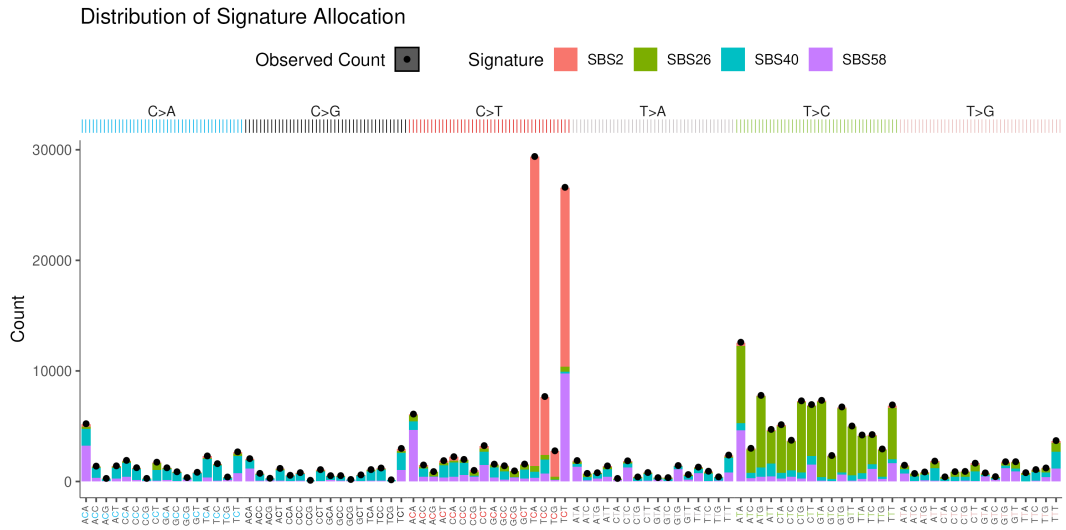
`sig_1.png` to `sig_K.png`

Plots of the sampler's estimated signatures with their assigned signature. The reference signature is a barplot, and the MAP estimate is a point with 95% credible intervals.



`signature_dist.png`

Plots the inferred MAP number of mutations attributed to each signature summed over all samples.



If `reference_P = NULL`, the `sig_k.png` plots will still be created but without the reference barplot.

Signature Assignments

After running `plot` (or using the `assign_signatures_ensemble` method), the `MAP$assignment_res` attribute of the sampler object will be populated.

```
sampler_learn_rank$reference_comparison$assignments
#> # A tibble: 4 x 5
#>   sig_est sig_ref MAP_cosine lower_cosine upper_cosine
#>     <dbl> <chr>        <dbl>        <dbl>        <dbl>
#> 1       1 SBS26        0.999        0.997        0.999
#> 2       2 SBS2         1.00         0.999        1.00
#> 3       3 SBS40        0.964        0.947        0.975
#> 4       4 SBS58        0.999        0.997        0.999
sampler_learn_rank$reference_comparison$votes
#>   sig_est sig_ref prop_votes
#> 1       1   SBS26          1
#> 2       2    SBS2          1
#> 3       3   SBS40          1
#> 4       4   SBS58          1
```

The assignment procedure assigns signatures for each posterior sample, then ensembles with majority vote to get the final assignment (see paper for details).

The `reference_comparison$assignments` dataframe holds final assignments for each signature, cosine similarities between MAP estimates and reference signatures, and 95% credible intervals for the cosine similarities.

The `reference_comparison$votes` dataframe holds the proportion of votes for each signature assignment for each posterior sample. This includes signatures that did not receive majority vote, allowing users to understand posterior uncertainty in assignment.