

countprop

Introduction

The `countprop` package allows estimation of several types of proportionality metrics for count-based compositional data such as 16S, metagenomic, and single-cell sequencing data. The package includes functions that allow standard empirical estimates of these proportionality metrics, as well as estimates based on the multinomial logit-normal model.

First, we'll define the model. Assume n samples and $J + 1$ features. Suppose the counts for sample i for feature j are denoted by y_{ij} for $i = 1, \dots, n$ and $j = 1, \dots, J + 1$. They are modelled using the multinomial distribution:

$$y_i \sim \text{Multinomial}(M_i; p_{i1}, \dots, p_{i(J+1)}),$$

where $M_i = \sum_{j=1}^{J+1} y_{ij}$ and proportion vector $\mathbf{p}_i = (p_{i1}, \dots, p_{i(J+1)})$. The proportions themselves are modelled using a logit-normal model, which can be formulated through a set of latent vectors (w_{i1}, \dots, w_{iJ}) which are related to the proportions by:

$$\begin{aligned} p_{ij} &= \text{alr}^{-1}(\mathbf{w}_i) \\ &= \begin{cases} \frac{\exp\{w_{ij}\}}{1 + \sum_{j=1}^J \exp\{w_{ij}\}} & \text{if } j = 1, \dots, J \\ \frac{1}{1 + \sum_{j=1}^J \exp\{w_{ij}\}} & \text{if } j = J + 1. \end{cases} \end{aligned}$$

The latent vectors are distributed as multivariate normal:

$$(w_{i1}, \dots, w_{iJ}) \sim \text{MV-Normal}_J(\boldsymbol{\mu}, \boldsymbol{\Sigma}).$$

The read-depths are assumed to be distributed as log-normal:

$$M_i \sim \text{Log-Normal}(\mu_\ell, \sigma_\ell^2).$$

Finally, to guard against spurious correlations, we apply the L_1 -penalty to the inverse covariance matrix $\boldsymbol{\Sigma}$ (i.e. the "graphical lasso" penalty).

$$\ell(w_{i1}, \dots, w_{iJ}) = \log \det \boldsymbol{\Sigma}^{-1} - \text{tr}(S \boldsymbol{\Sigma}^{-1}) - \lambda \|\boldsymbol{\Sigma}^{-1}\|_1$$

Fitting the model

The `countprop` package has a built-in function to estimate the model parameters. First, let's load the `countprop` library and look at the first few lines of the murine single cell sequencing dataset included with the package:

```
library(countprop)
#>
#> Attaching package: 'countprop'
#> The following object is masked from 'package:stats':
#>
#> logLik
data(singlecell)

head(singlecell, 2)
#> ENSMUSG00000064351 ENSMUSG00000064339 ENSMUSG00000064370
#> G1_cell1_count      40852                45108                31004
#> G1_cell2_count      67986                52596                57246
#> ENSMUSG00000023944 ENSMUSG00000029580 ENSMUSG00000057113
#> G1_cell1_count      16235                19137                15962
#> G1_cell2_count      19273                20124                18578
#> ENSMUSG00000037742 ENSMUSG00000020368 ENSMUSG00000064341
#> G1_cell1_count      11512                8614                 17692
#> G1_cell2_count      9652                 13785                24139
#> ENSMUSG00000054766
#> G1_cell1_count      8902
#> G1_cell2_count      18429
```

To fit the multinomial logit-normal model, we can use the `mleLR()` function:

```
mle <- mleLR(singlecell)

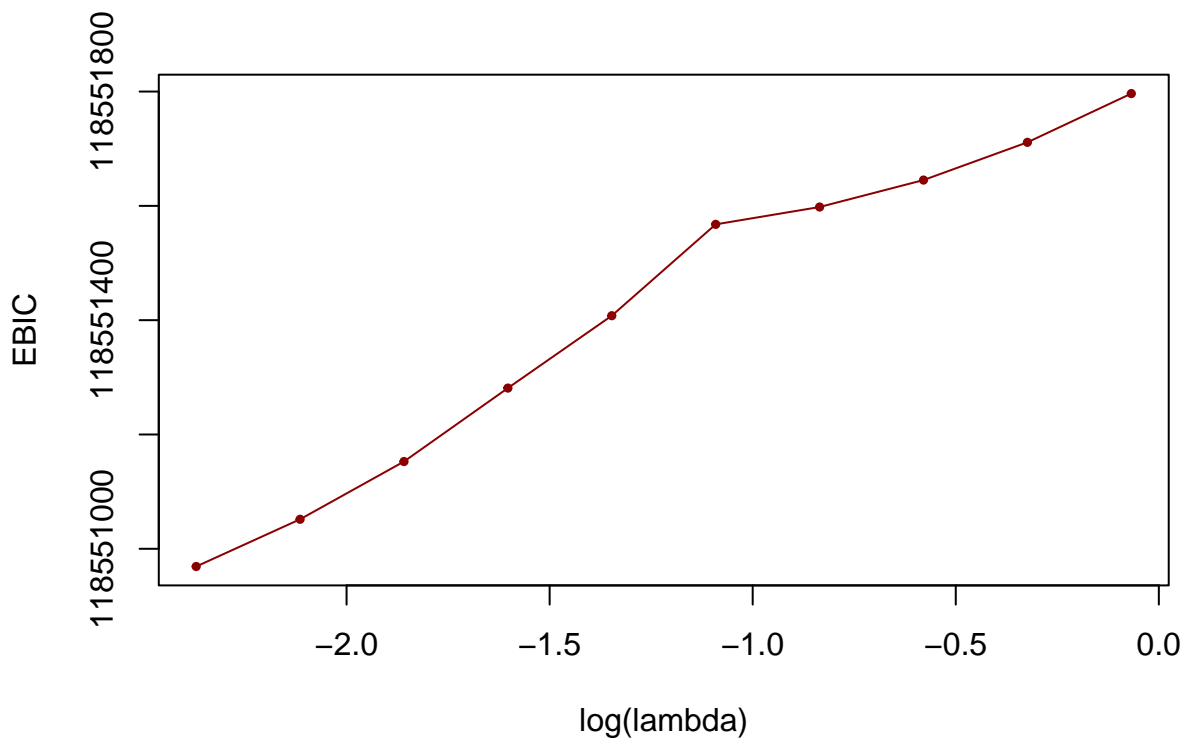
# Maximum likelihood estimates of model parameters
mle$mu
#> [1] 1.08972166 0.69105543 0.56031725 0.34323847 0.25345590 0.19768001
#> [7] 0.12912964 -0.02145714 -0.10901549
mle$Sigma.inv
#>      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
#> [1,] 21.44710128 -9.1006784 -10.894588 -5.1174332 2.447743 -0.3932173
#> [2,] -9.10013470 19.1978605 -2.045736 1.7209571 -2.808991 -0.1172743
#> [3,] -10.88951159 -2.0454178 27.490281 6.9191915 3.731340 -8.5116687
#> [4,] -5.11634377 1.7202705 6.919483 24.2221804 -2.833603 -3.9216127
#> [5,] 2.44864368 -2.8074752 3.731312 -2.8334097 22.554437 -1.3602528
#> [6,] -0.39286959 -0.1168179 -8.513339 -3.9223575 -1.360460 20.6824940
#> [7,] 2.90225574 0.2463541 -1.843747 -13.2077787 -3.662659 -4.6353533
#> [8,] -2.09514692 -0.6046498 -4.078285 0.2203073 -1.094891 2.4296618
#> [9,] -0.02713065 -5.8662579 -10.516366 -5.5981544 -5.101741 3.2801105
#>      [,7]      [,8]      [,9]
#> [1,] 2.9031726 -2.0957806 -0.02558981
#> [2,] 0.2477855 -0.6045823 -5.86419959
#> [3,] -1.8438336 -4.0771074 -10.51541009
#> [4,] -13.2077869 0.2210259 -5.59699319
#> [5,] -3.6627851 -1.0945800 -5.10133297
#> [6,] -4.6355185 2.4295730 3.27986643
#> [7,] 18.0886569 -1.0038532 4.16440953
```

```
#> [8,] -1.0034046 7.6764749 1.90561397
#> [9,] 4.1645004 1.9059161 16.60315147
```

For the `mleLR()` function, it is necessary to specify a value for λ , which is the graphical lasso penalty parameter. The default is 0. However, we can also run multiple values of λ to find which one leads to the best fit based on the Extended Bayesian Information Criterion (EBIC). To do this, we use the `mlePath()` function. This allows us to choose the number of λ values we want to run the model on (`n.lambda` parameter). This can also be parallelized by setting `n.cores`>1. Once we've obtained the model fit, we can visualize the EBIC values for each λ value using `ebicPlot()`.

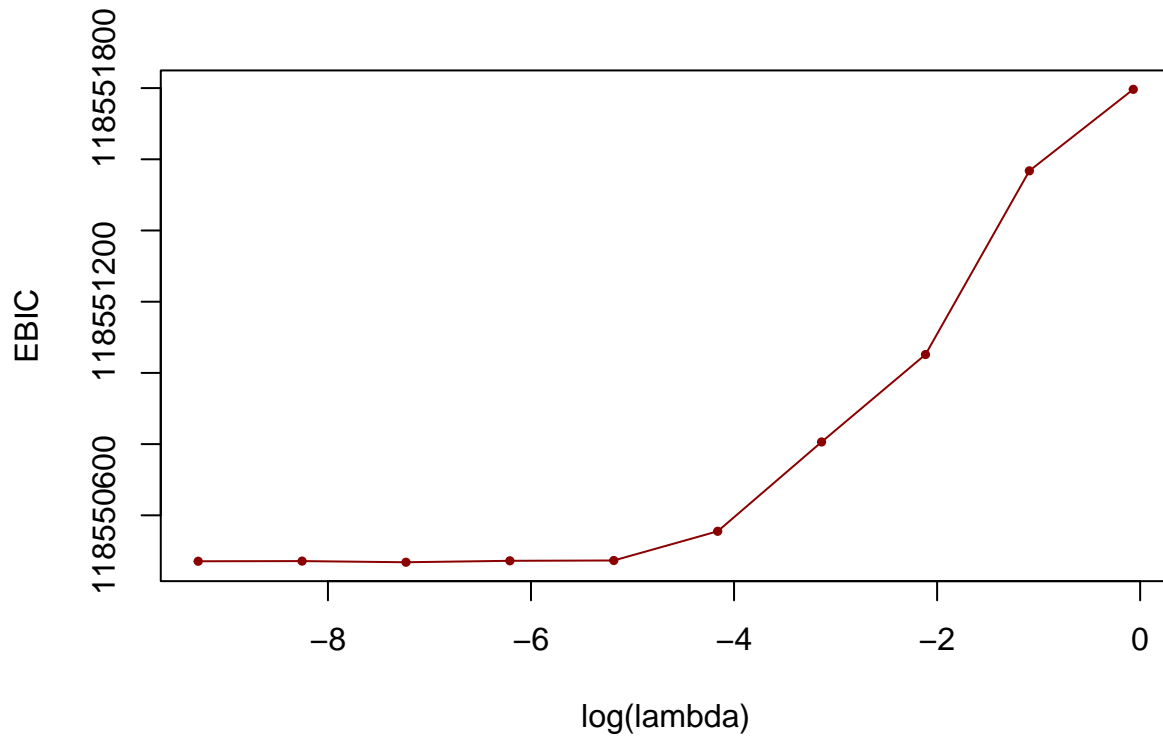
```
mle2 <- mlePath(singlecell, n.lambda=10, n.cores=1)
mle2$min.idx # Index of smallest lambda value
#> [1] 1
```

```
# Plot EBIC for different lambda values
ebicPlot(mle2)
```



In this case, the optimal value of λ is the one in the first position of the `lambda` vector. When the optimal λ value is the smallest one considered, then it's possible that an even smaller λ value would be optimal and was not considered. In this case, the argument `lambda.min.ratio` can be reduced from its default of 0.1:

```
mle3 <- mlePath(singlecell, n.lambda=10, lambda.min.ratio = 0.0001, n.cores=1)
mle3$min.idx
#> [1] 3
ebicPlot(mle3)
```



The minimum EBIC now corresponds to the 3rd smallest value of λ .

Estimating the proportionality metrics

Once the model parameters have been estimated, the model-based proportionality metrics can be estimated:

```
# Variation matrix
logitNormalVariation(mle3$est.min$mu, mle3$est.min$Sigma)
#>      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
#> [1,] 0.00000000 0.08223963 0.07155602 0.5100901 0.44032828 0.37281434
#> [2,] 0.08223963 0.00000000 0.09422343 0.4724987 0.37644140 0.34607945
#> [3,] 0.07155602 0.09422343 0.00000000 0.5050141 0.41572056 0.33307552
#> [4,] 0.51009009 0.47249872 0.50501409 0.0000000 0.10530745 0.11702161
#> [5,] 0.44032828 0.37644140 0.41572056 0.1053074 0.00000000 0.11564416
#> [6,] 0.37281434 0.34607945 0.33307552 0.1170216 0.11564416 0.00000000
#> [7,] 0.72512140 0.67375771 0.69687232 0.0886797 0.16033935 0.16848805
#> [8,] 0.23966764 0.24571246 0.22236016 0.3764686 0.30275075 0.29647659
#> [9,] 0.12782306 0.10277001 0.10009460 0.4856819 0.39130452 0.38111717
#> [10,] 0.41911591 0.37189939 0.39041370 0.1068904 0.06366782 0.08565271
#>      [,7]      [,8]      [,9]      [,10]
#> [1,] 0.7251214 0.2396676 0.1278231 0.41911591
#> [2,] 0.6737577 0.2457125 0.1027700 0.37189939
#> [3,] 0.6968723 0.2223602 0.1000946 0.39041370
#> [4,] 0.0886797 0.3764686 0.4856819 0.10689043
#> [5,] 0.1603394 0.3027507 0.3913045 0.06366782
#> [6,] 0.1684881 0.2964766 0.3811172 0.08565271
#> [7,] 0.0000000 0.4978141 0.7138370 0.15792003
#> [8,] 0.4978141 0.0000000 0.3012230 0.26633669
#> [9,] 0.7138370 0.3012230 0.0000000 0.40606973
#> [10,] 0.1579200 0.2663367 0.4060697 0.00000000
```

```

# Phi matrix
logitNormalVariation(mle3$est.min$mu, mle3$est.min$Sigma, type="phi")
#>      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
#> [1,] 0.0000000 0.1962217 0.1707309 1.2170621 1.0506122 0.8895256 1.7301214
#> [2,] 0.2211341 0.0000000 0.2533573 1.2705015 1.0122130 0.9305728 1.8116666
#> [3,] 0.1832826 0.2413425 0.0000000 1.2935358 1.0648206 0.8531348 1.7849587
#> [4,] 4.7720837 4.4204024 4.7245959 0.0000000 0.9851906 1.0947809 0.8296318
#> [5,] 6.9160252 5.9125846 6.5295235 1.6540136 0.0000000 1.8163674 2.5183733
#> [6,] 4.3526276 4.0404964 3.8886747 1.3662336 1.3501518 0.0000000 1.9671072
#> [7,] 4.5916999 4.2664486 4.4128177 0.5615481 1.0153199 1.0669201 0.0000000
#> [8,] 0.8998671 0.9225633 0.8348837 1.4135063 1.1367219 1.1131646 1.8691159
#> [9,] 0.3147811 0.2530846 0.2464961 1.1960553 0.9636387 0.9385510 1.7579174
#> [10,]      Inf      Inf      Inf      Inf      Inf      Inf      Inf
#>      [,8]      [,9] [,10]
#> [1,] 0.5718409 0.3049826      1
#> [2,] 0.6606961 0.2763382      1
#> [3,] 0.5695501 0.2563809      1
#> [4,] 3.5220046 4.5437357      1
#> [5,] 4.7551609 6.1460325      1
#> [6,] 3.4613803 4.4495636      1
#> [7,] 3.1523179 4.5202436      1
#> [8,] 0.0000000 1.1309858      1
#> [9,] 0.7418012 0.0000000      1
#> [10,]      Inf      Inf      NaN

# Rho matrix
logitNormalVariation(mle3$est.min$mu, mle3$est.min$Sigma, type="rho")
#>      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
#> [1,] 1.00000000 0.89603283 0.91160790 0.030258667 0.08793887 0.2614154
#> [2,] 0.89603283 1.00000000 0.87639798 0.013139583 0.13574441 0.2436283
#> [3,] 0.91160790 0.87639798 1.00000000 -0.015503518 0.08448036 0.3003591
#> [4,] 0.03025867 0.01313958 -0.01550352 1.000000000 0.38257196 0.3922318
#> [5,] 0.08793887 0.13574441 0.08448036 0.382571965 1.00000000 0.2255308
#> [6,] 0.26141537 0.24362830 0.30035912 0.392231751 0.22553076 1.0000000
#> [7,] -0.25663126 -0.27167424 -0.27089085 0.665120119 0.27640732 0.3082639
#> [8,] 0.65035126 0.61501321 0.66142364 -0.008684981 0.08258605 0.1577116
#> [9,] 0.84509781 0.86789962 0.87432934 0.053178192 0.16697203 0.2249344
#> [10,] 0.00000000 0.00000000 0.00000000 0.000000000 0.00000000 0.0000000
#>      [,7]      [,8]      [,9] [,10]
#> [1,] -0.2566313 0.650351261 0.84509781      0
#> [2,] -0.2716742 0.615013212 0.86789962      0
#> [3,] -0.2708908 0.661423639 0.87432934      0
#> [4,] 0.6651201 -0.008684981 0.05317819      0
#> [5,] 0.2764073 0.082586046 0.16697203      0
#> [6,] 0.3082639 0.157711591 0.22493436      0
#> [7,] 1.0000000 -0.173379489 -0.26569146      0
#> [8,] -0.1733795 1.000000000 0.55202240      0
#> [9,] -0.2656915 0.552022397 1.00000000      0
#> [10,] 0.0000000 0.000000000 0.00000000      1

```

The package also provides the standard naive (empirical) estimates of the proportionality metrics.

```

# Naive (empirical) variation matrix
naiveVariation(singleCell)

```

```

#>           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
#> [1,] 0.00000000 0.08152267 0.06974364 0.5087255 0.4447607 0.3750271 0.7259202
#> [2,] 0.08152267 0.00000000 0.09152832 0.4728165 0.3741705 0.3578873 0.6751740
#> [3,] 0.06974364 0.09152832 0.00000000 0.5051328 0.4168222 0.3400802 0.6945633
#> [4,] 0.50872551 0.47281649 0.50513276 0.0000000 0.1034765 0.1255767 0.0865833
#> [5,] 0.44476070 0.37417054 0.41682223 0.1034765 0.0000000 0.1316500 0.1603955
#> [6,] 0.37502706 0.35788732 0.34008017 0.1255767 0.1316500 0.0000000 0.1731814
#> [7,] 0.72592022 0.67517400 0.69456335 0.0865833 0.1603955 0.1731814 0.0000000
#> [8,] 0.23904178 0.25093706 0.22512118 0.3830622 0.3119511 0.2972780 0.4988237
#> [9,] 0.12486115 0.10187624 0.09768942 0.4840748 0.3926279 0.3869858 0.7145046
#>           [,8]      [,9]
#> [1,] 0.2390418 0.12486115
#> [2,] 0.2509371 0.10187624
#> [3,] 0.2251212 0.09768942
#> [4,] 0.3830622 0.48407480
#> [5,] 0.3119511 0.39262793
#> [6,] 0.2972780 0.38698581
#> [7,] 0.4988237 0.71450463
#> [8,] 0.0000000 0.30353326
#> [9,] 0.3035333 0.00000000

```

Naive (empirical) Phi matrix

```
naiveVariation(singlecell, type="phi")
```

```

#>           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
#> [1,] 0.00000000 0.1945533 0.1664427 1.2140701 1.0614185 0.8949996 1.7324038
#> [2,] 0.2189841 0.0000000 0.2458610 1.2700675 1.0050873 0.9613477 1.8136351
#> [3,] 0.1788519 0.2347169 0.0000000 1.2953717 1.0689066 0.8721078 1.7811510
#> [4,] 4.7624863 4.4263203 4.7288524 0.0000000 0.9687057 1.1755996 0.8105585
#> [5,] 6.7866329 5.7094929 6.3603179 1.5789544 0.0000000 2.0088565 2.4474853
#> [6,] 4.0360080 3.8515516 3.6599127 1.3514457 1.4168060 0.0000000 1.8637633
#> [7,] 4.6059091 4.2839281 4.4069521 0.5493645 1.0176973 1.0988232 0.0000000
#> [8,] 0.8866078 0.9307275 0.8349762 1.4207805 1.1570289 1.1026065 1.8501409
#> [9,] 0.3076443 0.2510119 0.2406961 1.1927075 0.9673924 0.9534908 1.7604615
#>           [,8]      [,9]
#> [1,] 0.5704716 0.2979803
#> [2,] 0.6740607 0.2736573
#> [3,] 0.5773049 0.2505165
#> [4,] 3.5860760 4.5317161
#> [5,] 4.7600821 5.9911354
#> [6,] 3.1992799 4.1647070
#> [7,] 3.1649987 4.5334780
#> [8,] 0.0000000 1.1258072
#> [9,] 0.7478729 0.0000000

```

Naive (empirical) Rho matrix

```
naiveVariation(singlecell, type="rho")
```

```

#>           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
#> [1,] 1.00000000 0.89697647 0.91378783 0.03255458 0.08213424 0.2674468
#> [2,] 0.89697647 1.00000000 0.87992012 0.01310692 0.14536149 0.2306757
#> [3,] 0.91378783 0.87992012 1.00000000 -0.01683163 0.08488623 0.2957140
#> [4,] 0.03255458 0.01310692 -0.01683163 1.00000000 0.39962866 0.3712978
#> [5,] 0.08213424 0.14536149 0.08488623 0.39962866 1.00000000 0.1691651
#> [6,] 0.26744677 0.23067570 0.29571403 0.37129778 0.16916510 1.0000000

```

```

#> [7,] -0.25889881 -0.27419464 -0.26847388  0.67256083 0.28119253 0.3087303
#> [8,]  0.65287781  0.60906566  0.65868280 -0.01760995 0.06921592 0.1800000
#> [9,]  0.84863241  0.86907703  0.87724593  0.05579807 0.16709698 0.2241390
#>      [,7]      [,8]      [,9]
#> [1,] -0.2588988  0.65287781  0.84863241
#> [2,] -0.2741946  0.60906566  0.86907703
#> [3,] -0.2684739  0.65868280  0.87724593
#> [4,]  0.6725608 -0.01760995  0.05579807
#> [5,]  0.2811925  0.06921592  0.16709698
#> [6,]  0.3087303  0.18000000  0.22413904
#> [7,]  1.0000000 -0.16760330 -0.26804738
#> [8,] -0.1676033  1.00000000  0.55063798
#> [9,] -0.2680474  0.55063798  1.00000000

```