Microbiome Data Simulation

Dahun Seo

2025-1-14

Contents

1 Data Generation

1.1 The logistic normal (LN) distribution

- Generate random binary tree with p (leaves) variables.
- Calculate cophenetic distanace between variables d_{ij} for $i, j = 1, \dots, p$.
- Define variance-covariance matrix Σ:
  The (i, j)-th element of Σ is defined as follows:

  $$\Sigma_{ij} = \exp(-d_{ij})/2,$$

  where $i, j = 1, \dots, p$.
- Generate p-dimensional data vector from multivariate normal distribution, for $i = 1, \dots, n$,

  $$M_i \sim \mathcal{N}_p(\alpha_0, \Sigma),$$

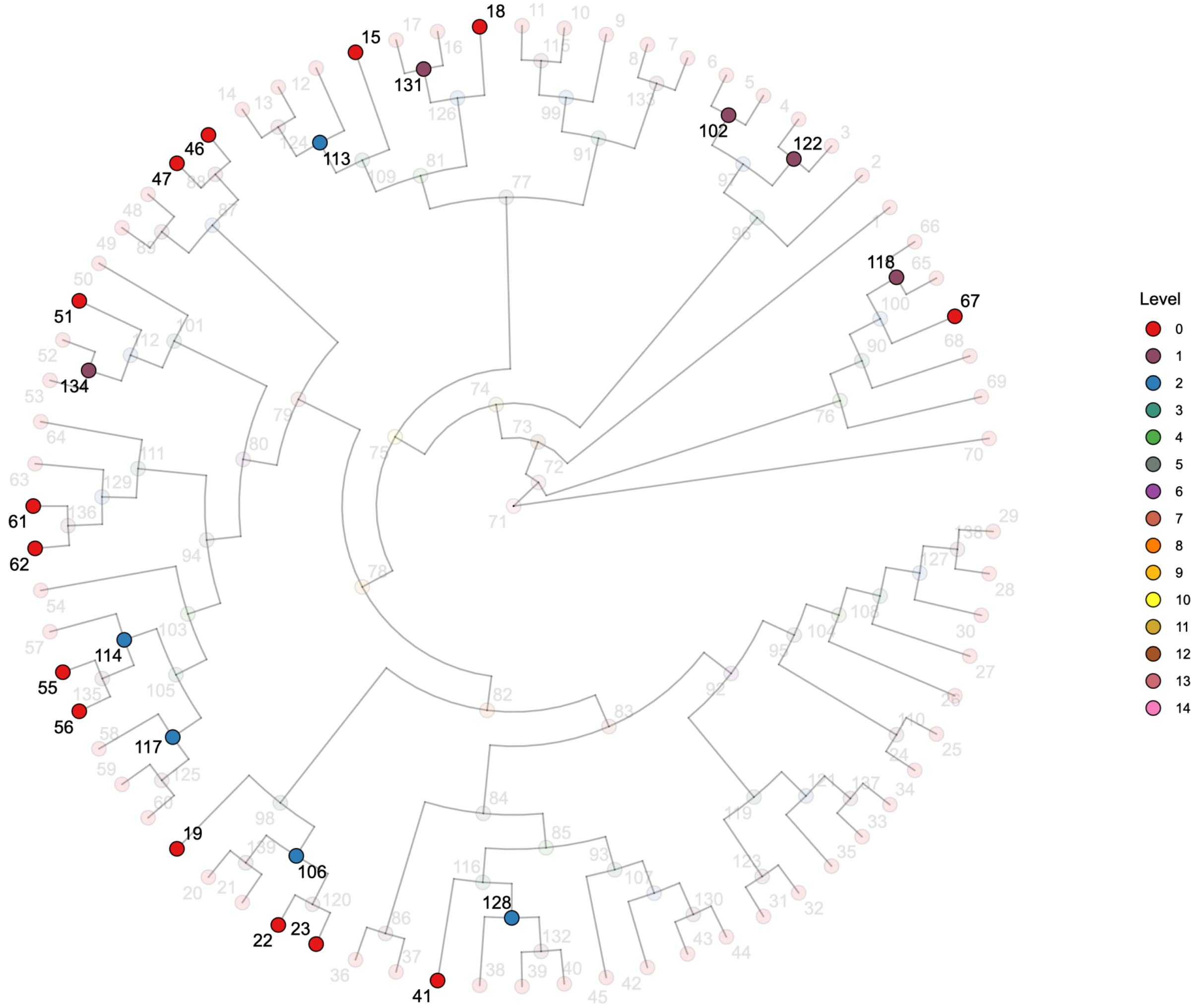
  where α_0 is pre-defined vector (e.g., $\alpha_0 = 0$).
- Transformation for compositional characteristic, for $i = 1, \dots, n$ and $j = 1, \dots, p$,

  $$Z_{ij} = \log\left(\frac{\exp(M_{ij})}{\sum_{k=1}^{p} \exp(M_{ik})}\right)$$

- Calculate expanded feature $\tilde{Z}_i$ using Algorithm 1 in the main paper.
- Generate outcome variable for $i = 1, \dots, n$,

  $$Y_i = \beta_0 + \tilde{Z}_i^\top \beta + \varepsilon_i$$

  where $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$, positive noise level σ, and β is pre-defined true beta vector.
  The variables with non-zero coefficients were randomly selected, and the effect sizes ranged from -2 to 2.


18
15
131
46
47
113
102
122
118
67
51
134
61
62
114
55
56
117
19
106
22
23
128
41
Level
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14

2 Simulation Study Results

How well Y is modeled?

Original:

- MSE (cross validation): 0.1437
- Correlation (y vs predicted): 0.9712

Expanded:

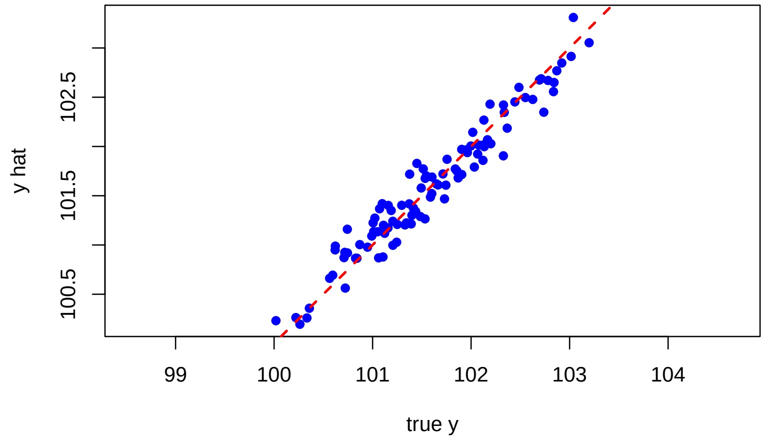
- MSE (cross validation): 0.1106
- Correlation (y vs predicted): 0.9397

Even though the true data was generated using the expanded features created by Algorithm 1, the modeling performance for Y worse than that of existing methods.

This phenomenon is suspected to be due to the excessive number of transformations introduced by the subcomposition process. As a result, I am currently considering improvements to Algorithm 1.

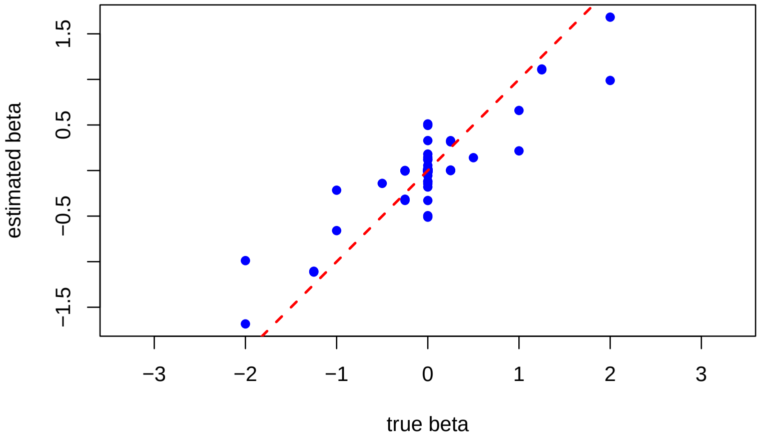
2.1 Original

```
C <- matrix(1, p, ncol = 1)

res2 <- ConstrLassoCrossVal(y = y, x = log_z, C = C,
    nfolds = 10)

res2$cvm[res2$sel]
#> [1] 0.143687
res2$Rsq.sel
#> [1] 0.9416038

cor(y, log_z %*% res2$bet.sel + res2$int.sel)
#>           [,1]
#> [1,] 0.971235

{
    plot(y, log_z %*% res2$bet.sel + res2$int.sel,
        xlab = "true y", ylab = "y hat", pch = 16,
        col = "blue", asp = 1)
    abline(a = 0, b = 1, col = "red", lty = 2, lwd = 2)
}
```

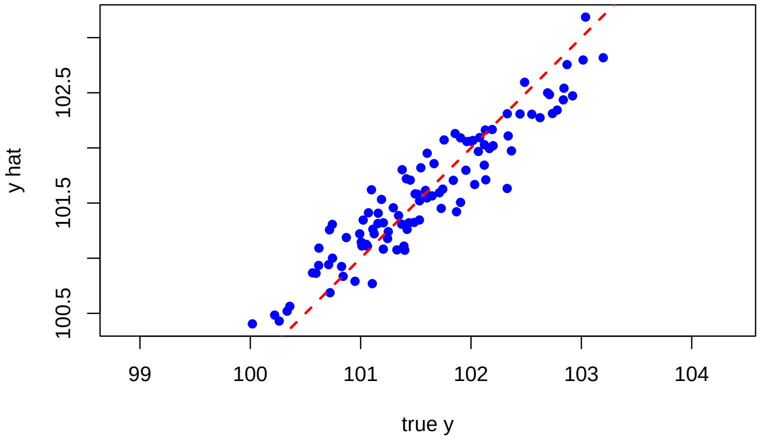


2.2 Expanded

```
res3 <- ConstrLassoCrossVal(y = y, x = expanded_z$data,
    C = expanded_z$C, nfolds = 10)

res3$cvm[res3$sel]
#> [1] 0.1105758
res3$Rsq.sel
#> [1] 0.8674755

cor(y, expanded_z$data %*% res3$bet.sel + res3$int.sel)
#>           [,1]
#> [1,] 0.9396877

{
    plot(beta, res3$bet.sel, xlab = "true beta", ylab = "estimated beta",
        pch = 16, col = "blue", asp = 1)
    abline(a = 0, b = 1, col = "red", lty = 2, lwd = 2)
}
```



```
{
    plot(y, expanded_z$data %*% res3$bet.sel + res3$int.sel,
        xlab = "true y", ylab = "y hat", pch = 16,
        col = "blue", asp = 1)
    abline(a = 0, b = 1, col = "red", lty = 2, lwd = 2)
}
```


y hat
true y