Plasma amyloid-beta biomarker for Alzheimer's disease

Shuai Zhu

Introduction

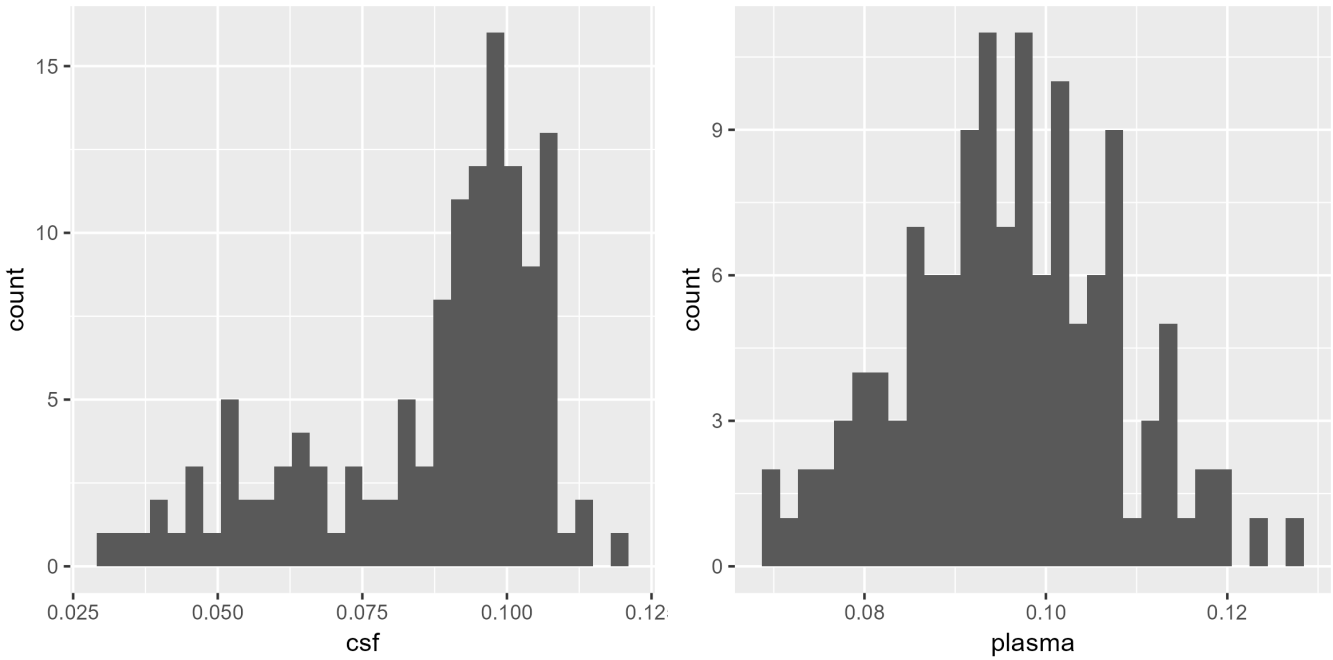
- The only valid method for detecting Alzheimer's disease(AD) is the measurement of amyloid-β(Aβ) in cerebrospinal fluid(CSF).
- Lower $A\beta_{42}/A\beta_{40}$ ratio, higher risk of developing AD dementia
- Extracting CSF from human's brain is so expensive, a cost-effective blood test amyloid-β biomarker is desirable.

Data

- The sample of CSF and plasma are from Longitudinal Innate Immunity and Aging(LIIA) cohort.
- Older adults who are 60 years or older, In good general health, and have not been diagnosed with a memory disorder was recruited to LIIA study.
- We only use baseline data and sample size is 130.

Descriptive analysis

	Overall
	(N=130)
CSF.AB42/40.Ratio	
Mean (SD)	0.0865 (0.0205)
Median [Min, Max]	0.0943 [0.0300, 0.119]
Plasma.AB42/40.Ratio	
Mean (SD)	0.0960 (0.0118)
Median [Min, Max]	0.0957 [0.0689, 0.127]



- A two component Gaussian mixture model was fitted to CSF Aβ ratio to classify subjects as Amyloid negative and Amyloid positive. But we are not able to fit same model to plasma Aβ ratio.
- A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters

$$f(y) = \sum_{k=1}^{2} \pi_k \cdot N\left(\mu_k \ , \sigma^2 \right)$$

Joint distribution of CSF and plasma

- A bivariate Gaussian mixture model was used to fit the joint distribution of CSF and plasma, incorporating information from both sources while ensuring that CSF plays a dominant role in the classification.

$$f\begin{pmatrix} y^{\text{csf}} \\ y^{\text{plasma}} \end{pmatrix} = \sum_{k=1}^{2} \pi_k \cdot N\begin{pmatrix} \mu_k \end{pmatrix}, \Sigma\Big)$$

- Gibbs sampler was used to estimate the parameter of bivariate Gaussian mixture model
  - The Gibbs sampler is a Markov Chain Monte Carlo (MCMC) algorithm commonly used for sampling from complex, high-dimensional probability distributions
  - Parameters: $\mu_1^{csf}, \mu_1^{plasma}, \mu_2^{csf}, \mu_2^{plasma}, \Sigma, \pi_1, z_{1-130}$

Simulation

We simulate 1,000 data for each of sample size 100, 200, 500 using parameter from real data and sensitivity, specificity, accuracy were calculated across 1,000 datasets. The cutpoint for z is 0.5.

$$\mu_1 = (0.05, 0.08), \mu_2 = (0.1, 0.1), \Sigma = \begin{pmatrix} 0.0001082 & 0.0000375 \\ 0.0000375 & 0.0001030 \end{pmatrix}$$

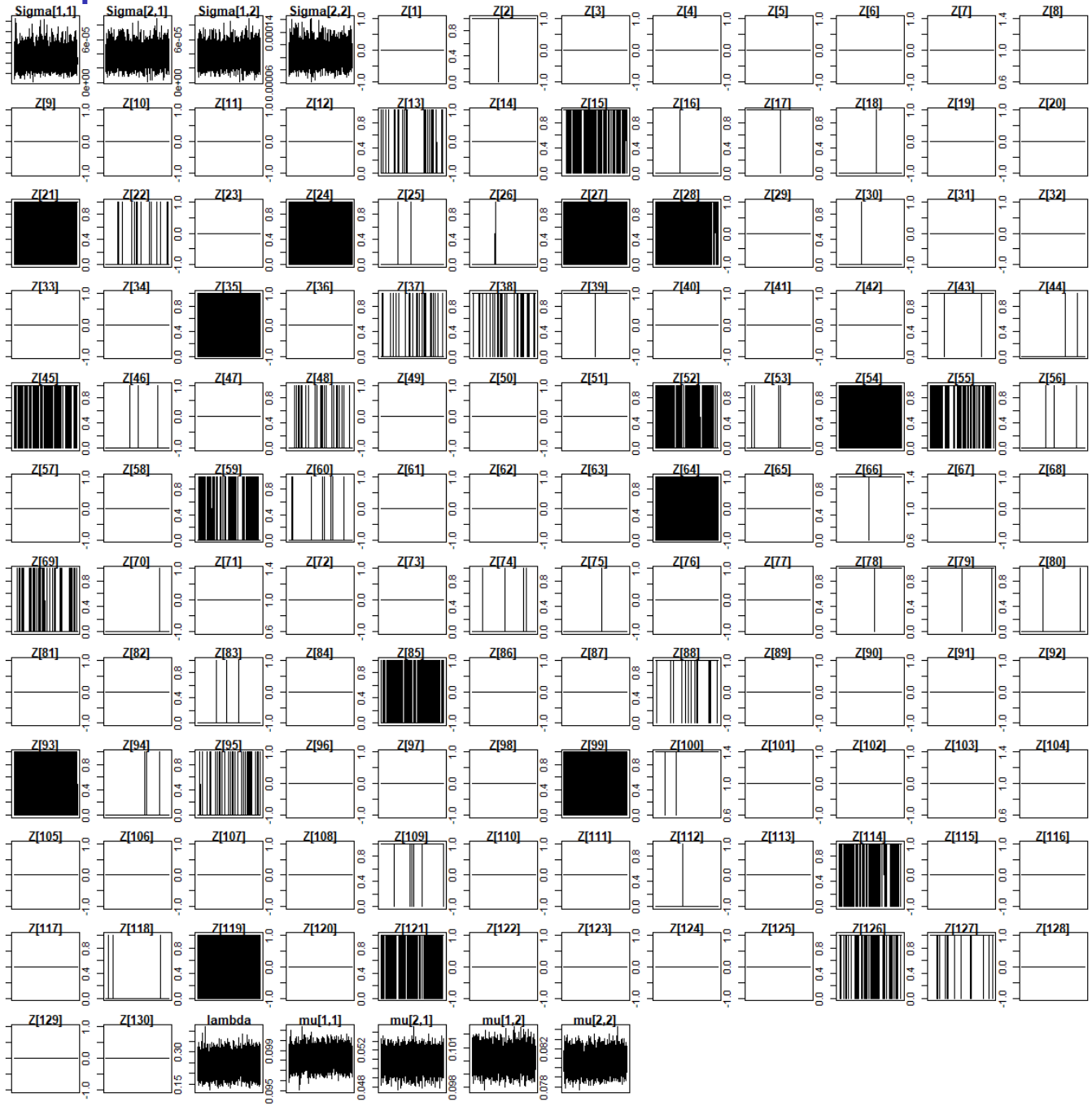
$$\pi_1 = 0.3, \pi_2 = 0.7$$

Findings

The estimated labels are very close to true labels. The standard deviation tend to decrease as sample size increased.

	Accuracy	Sensitivity	Specificity
simulated data with sample size 100	0.9940 (0.0074)	0.9966 (0.0076)	0.9880 (0.0198)
simulated data with sample size 200	0.9924 (0.0063)	0.9953 (0.0057)	0.9858 (0.0158)
simulated data with sample size 500	0.9925 (0.0039)	0.9950 (0.0039)	0.9867 (0.0094)

Trace plots


Sigma[1,1]
Sigma[2,1]
Sigma[1,2]
Sigma[2,2]
Z[1]
Z[2]
Z[3]
Z[4]
Z[5]
Z[6]
Z[7]
Z[8]
Z[9]
Z[10]
Z[11]
Z[12]
Z[13]
Z[14]
Z[15]
Z[16]
Z[17]
Z[18]
Z[19]
Z[20]
Z[21]
Z[22]
Z[23]
Z[24]
Z[25]
Z[26]
Z[27]
Z[28]
Z[29]
Z[30]
Z[31]
Z[32]
Z[33]
Z[34]
Z[35]
Z[36]
Z[37]
Z[38]
Z[39]
Z[40]
Z[41]
Z[42]
Z[43]
Z[44]
Z[45]
Z[46]
Z[47]
Z[48]
Z[49]
Z[50]
Z[51]
Z[52]
Z[53]
Z[54]
Z[55]
Z[56]
Z[57]
Z[58]
Z[59]
Z[60]
Z[61]
Z[62]
Z[63]
Z[64]
Z[65]
Z[66]
Z[67]
Z[68]
Z[69]
Z[70]
Z[71]
Z[72]
Z[73]
Z[74]
Z[75]
Z[76]
Z[77]
Z[78]
Z[79]
Z[80]
Z[81]
Z[82]
Z[83]
Z[84]
Z[85]
Z[86]
Z[87]
Z[88]
Z[89]
Z[90]
Z[91]
Z[92]
Z[93]
Z[94]
Z[95]
Z[96]
Z[97]
Z[98]
Z[99]
Z[100]
Z[101]
Z[102]
Z[103]
Z[104]
Z[105]
Z[106]
Z[107]
Z[108]
Z[109]
Z[110]
Z[111]
Z[112]
Z[113]
Z[114]
Z[115]
Z[116]
Z[117]
Z[118]
Z[119]
Z[120]
Z[121]
Z[122]
Z[123]
Z[124]
Z[125]
Z[126]
Z[127]
Z[128]
Z[129]
Z[130]
lambda
mu[1,1]
mu[2,1]
mu[1,2]
mu[2,2]

Train test partition

The sample size is 200, and the data was split into training and testing sets with varying proportions: 30% training data, 50% training data, and 80% training data. The remaining data in each case was used for testing performance. As expected, the performance of our method improves as the size of the training data increases.

train_size	optimal_cutpoint	Accuracy	Sensitivity	Specificity
0.3	0.0898 (0.0032)	0.8278 (0.0476)	0.8302 (0.0882)	0.8224 (0.1001)
0.5	0.0900 (0.0028)	0.8288 (0.0471)	0.8296 (0.0812)	0.8271 (0.1006)
0.8	0.0899 (0.0024)	0.8319 (0.0629)	0.8341 (0.0901)	0.8268 (0.1247)