Classification Using Functional Data Analysis for Temporal Gene Expression Data

Functional PCA Simulation Study

July 31, 2019

Contents

1. Introduction

- A simulation study was performed based on the first five estimated FPCs($\hat{\rho}_m$) from the yeast cell-cycle data
- Five random coefficients $\epsilon_m (m = 1, ..., 5)$ were generated for each subject from normal distributions with means 0.6, 0.5, 0.4, 0.3, and 0.2 for group 1 and -0.6, -0.5, -0.4, -0.3, and -0.2 for group 2
- The variances of ϵ_m correspond to the estimated eigenvalues($\hat{\lambda}_m$)

$$\hat{X}_i(t) = \hat{\mu}(t) + \sum_{m=1}^{M} \epsilon_{im} \hat{\rho}_m(t) \quad 0 \leq t \leq T.$$

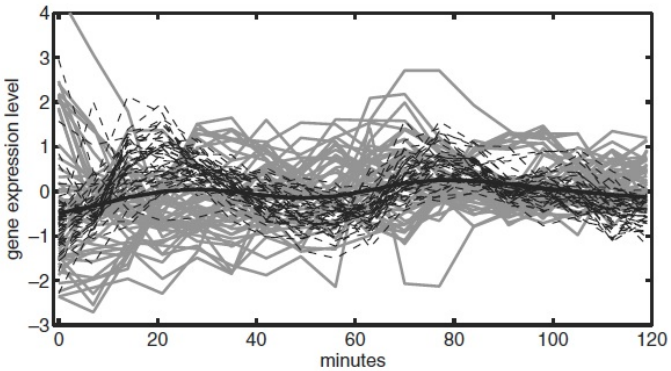


Fig. 1. Temporal gene expression profiles of yeast cell cycle. Dashed lines: G_1 phase; Gray solid lines: non-G_1 phases; Black solid line: overall mean curve.

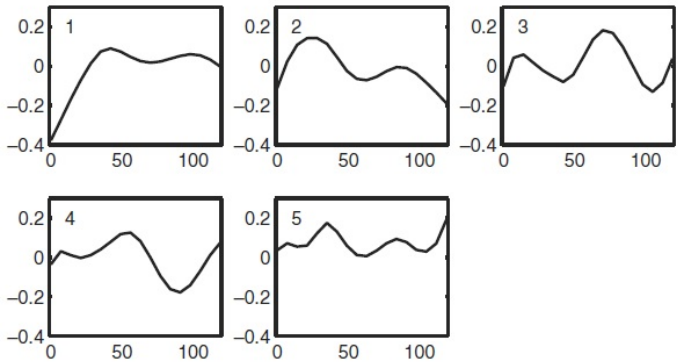


Fig. 5. The first five FPCs for the known 90 genes for yeast cell-cycle data. These five FPCs account for 98.9% of the total variation, with the first FPC accounting for 66.5%, the second for 21.9%, the third for 4.7%, the fourth for 3.1% and the fifth for 2.7%.

1. Introduction

- Using training set from generated $\hat{X}_i$'s, calculate $\hat{\epsilon}'_{im}$ for FPCA, and $\hat{\gamma}_{il}$ for B-spline method($i = 1, 2, ..., n,\ m = 1, 2, ..., M,\ l = 1, 2, ..., L$)

$$\hat{X}_i(t) = \hat{\mu}'(t) + \sum_{m=1}^{M} \hat{\epsilon}'_{im}\hat{\rho}'_m(t) \quad 0 \le t \le T,$$

$$\hat{X}_i(t) = \sum_{k=1}^{p} \hat{\beta}_k \hat{\bar{B}}_k(t_{ij}) + \sum_{l=1}^{q} \hat{\gamma}_{il}\hat{B}_l(t_{ij}) => \hat{\mu}'(t) + \sum_{l=1}^{q} \hat{\gamma}_{il}\hat{B}_l(t_{ij})$$

- With these $\hat{\epsilon}'_{im}$, and $\hat{\gamma}_{il}$, classify test set's group via logistic regression analysis, where the inverse link function is

$$g^{-1}\left(\hat{\alpha} + \sum_{m=1}^{M} \hat{\beta}_m \hat{\epsilon}'_m\right) = \hat{\pi}_i, \quad i = 1, 2, ..., n.$$

- Then compare the classification error rates of both methods with the fitted logistic regression models

2. Generating Samples

- http://genome-www.stanford.edu/cellcycle/data/rawdata/
- Rawdata contains unnecessary data(other than α factor synchronized data) and NA values

	gene	cln3experiment1	cln3experiment2	clb2experiment2	clb2experiment1	alpha0min	alpha7min	alpha14min	alpha21min	alpha28min	alpha35min	alpha42min
1	YAL001C	0.15	NA	-0.22	0.07	-0.15	-0.15	-0.21	0.17	-0.42	-0.44	-0.15
2	YAL002W	-0.07	-0.76	-0.12	-0.25	-0.11	0.10	0.01	0.06	0.04	-0.26	0.04
3	YAL003W	-1.22	-0.27	-0.10	0.23	-0.14	-0.71	0.10	-0.32	-0.40	-0.58	0.11
4	YAL004W	-0.09	1.20	0.16	-0.14	-0.02	-0.48	-0.11	0.12	-0.03	0.19	0.13
5	YAL005C	-0.60	1.01	0.24	0.65	-0.05	-0.53	-0.47	-0.06	0.11	-0.07	0.25
6	YAL007C	0.65	1.39	-0.29	-0.54	-0.60	-0.45	-0.13	0.35	-0.01	0.49	0.18
7	YAL008W	-0.36	-0.22	-0.20	0.10	-0.28	-0.22	-0.06	0.22	0.25	0.13	0.34
8	YAL009W	0.25	-0.79	-0.22	-0.54	-0.03	-0.27	0.17	-0.12	-0.27	0.06	0.23
9	YAL010C	-0.30	-0.60	-0.18	0.01	-0.05	0.13	0.13	-0.21	-0.45	-0.21	0.06
10	YAL011W	-0.15	-0.71	-0.15	-0.25	-0.31	-0.43	-0.30	-0.23	-0.13	-0.07	0.08
11	YAL012W	-1.22	0.66	-0.64	-0.17	0.02	-0.33	-0.49	-0.30	-0.15	-0.24	0.40
12	YAL013W	-0.34	-1.06	-0.45	-0.29	-0.36	-0.19	0.00	-0.32	-0.27	-0.12	0.04
13	YAL014C	-0.84	-1.29	-0.12	0.18	-0.10	-0.15	-0.01	-0.25	-0.16	-0.13	0.06
14	YAL015C	-0.12	-0.54	-0.12	-0.18	0.00	-0.01	0.12	-0.23	-0.13	0.25	0.30
15	YAL016W	-0.42	0.23	0.14	0.32	0.06	0.01	0.17	-0.14	0.01	-0.24	0.15
16	YAL017W	0.29	-0.40	-0.09	-0.32	-0.40	-0.22	0.19	-0.20	-0.09	0.41	0.13
17	YAL018C	-0.29	NA	-0.42	-0.01	0.46	0.28	0.16	-1.72	0.33	0.05	0.22
18	YAL019W	0.26	-0.17	-0.23	-0.12	-0.24	-0.95	-0.23	0.12	-0.02	0.23	-0.11
19	YAL020C	0.44	-0.51	-0.22	0.15	-0.02	-0.29	-0.07	-0.22	-0.06	-0.07	0.20

2. Generating Samples

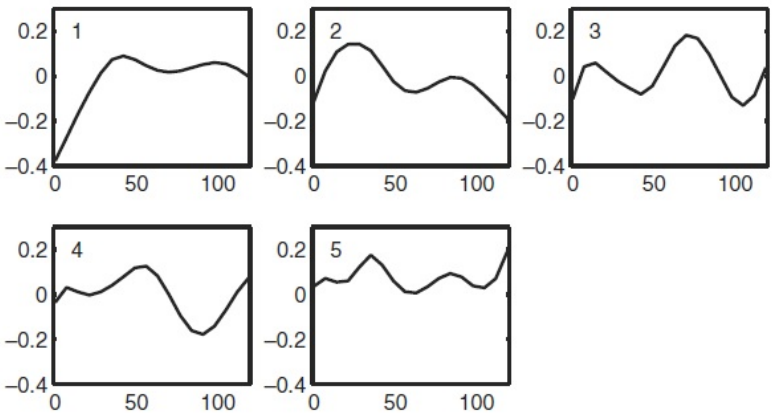
- Select α factor synchronized data and omit NA values
- Transform the rawdata into fd(functional data) object
- Obtain PCA basis from original data

```
library(fda.usc)
cell <- read.delim("mbc_9_12_3273__CDCDATA.txt")
gene <- cell[, c(1, 6:23)]
gene <- na.omit(gene)
train <- gene[, -1] # remove the column of gene's names
train.fdata <- fdata(train)
train.fd <- fdata2fd(train.fdata, nbasis=5)
train.pca <- create.pc.basis(train.fdata, l=1:5, lambda=1)
```

2. Generating Samples

- The plots of the five FPC curves are similar to paper's plots

```
train.pca$basis$data <- -train.pca$basis$data
plot(train.pca$basis, ylim=c(-0.6, 0.6))
par(mfrow=c(2,3))
for (i in 1:5) {
  plot(train.pca$basis[i,], main=paste("FPC", i), ylim=c
    (-0.6, 0.6))
}
```



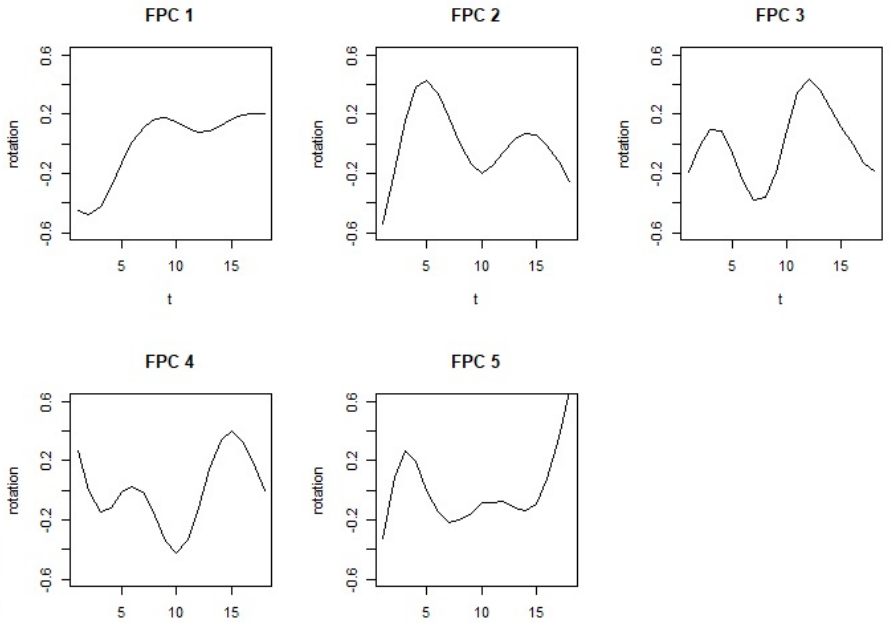


Fig. 5. The first five FPCs for the known 90 genes for yeast cell-cycle data. These five FPCs account for 98.9% of the total variation, with the first FPC accounting for 66.5%, the second for 21.9%, the third for 4.7%, the fourth for

2. Generating Samples

- From $\hat{X}_i(t) = \hat{\mu}(t) + \sum_{m=1}^{M} \epsilon_{im}\hat{\rho}_m(t) \quad 0 \leq t \leq T$, generate 100 datasets(100 train and test data for each dataset) $\hat{X}_i(t_j),\ j = 1, .., 18$ with the ϵ_{im}'s given by rnorm function

- $\hat{\mu}(t_j)$

```
> train.pca$mean$data
             [,1]        [,2]       [,3]        [,4]       [,5]       [,6]       [,7]          [,8]       [,9]
[1,] -0.001483627 -0.06765872 0.02064157 -0.00565382 0.03527957 0.00833816 0.07849855 -0.0004566719 0.05561818
            [,10]      [,11]       [,12]      [,13]      [,14]       [,15]       [,16]       [,17]       [,18]
[1,] -0.008652261 0.03202941 -0.01520383 0.04179996 -0.1583738 0.001347739 0.007527289 0.004328358 -0.02778347
```

- $\hat{\rho}_m(t_j),\ j = 1, .., 18$

```
> train.pca$basis$data
            1           2          3           4           5           6          7           8          9
PC1 -0.4561404 -0.479346676 -0.4247908 -0.29182332 -0.126654438  0.01429060  0.1138374  0.167619140  0.1778224
PC2 -0.5411512 -0.186631395  0.1605178  0.38051140  0.425509450  0.34040726  0.1803649  0.004681526 -0.1353060
PC3 -0.1898493 -0.020566941  0.0975911  0.08482354 -0.064600250 -0.25186520 -0.3810106 -0.359326018 -0.1877626
PC4  0.2695601 -0.005837465 -0.1442123 -0.11876424 -0.009465060  0.02516988 -0.0204302 -0.171052862 -0.3399338
PC5 -0.3229185  0.084633595  0.2655946  0.19445872 -0.007339227 -0.14848975 -0.2188865 -0.202355596 -0.1529007
           10          11          12          13          14          15          16          17          18
PC1  0.14841734  0.10541349  0.07989280  0.08783613  0.12572348  0.16424892  0.193680854  0.2009294  0.199314663
PC2 -0.19609004 -0.15040227 -0.05606380  0.02964161  0.07065622  0.06271406 -0.013454951 -0.1197625 -0.255791659
PC3  0.08564740  0.34353622  0.43444108  0.36855158  0.24246307  0.11089664 -0.006290314 -0.1250567 -0.181603360
PC4 -0.42292993 -0.33849766 -0.11923254  0.14104835  0.33351809  0.40326009  0.329094820  0.1916163 -0.003462398
PC5 -0.08170919 -0.08148318 -0.07663211 -0.12252707 -0.14050875 -0.09324863  0.080089810  0.3547865  0.671519340
```

3. Simulation Results : FPCA Method

- Estimate $\hat{\mu}'(t)$ and $\hat{\rho}'_m(t)$ with the generated train set via create.pc.basis function

- Then calculate the $\hat{\epsilon}'_{im}$ of train and test set for each $m = 1, ..., 5$, where

$$\hat{\epsilon}'_{im} = \sum_{k=1}^{S}((\hat{X}_i(k) - \hat{\mu}'(k))\hat{\rho}'_m(k), \ S = 18$$

```
> set.e$train
                e1          e2          e3           e4           e5
1   -0.869215856 -0.23562203  0.433777067  5.537948e-01  1.063533e-02
2    0.556755587  0.98388789  0.411870967 -2.808091e-01 -2.318547e-01
3   -2.820701032  0.66934749  0.122762361 -5.452579e-01  2.245622e-01
4   -1.025282104  0.19911721  0.827533058  8.093052e-02 -2.422870e-01
5   -0.593736680 -1.42365411  0.649620527  1.639880e-01 -5.136332e-01
6   -1.092823476  0.44827997 -0.026507738 -5.537899e-01 -1.683859e-01
7   -1.186541380  0.66016536  0.543837469 -3.542383e-02  5.154803e-01
8   -3.024055210 -0.24716041  0.343763698  9.113634e-02 -6.634528e-01
9   -3.789155918  0.99908339 -0.144872988 -8.804926e-02  3.332286e-01
10  -1.969398241 -1.13112182 -1.070065773 -1.809187e-01  1.896484e-01
```

3. Simulation Results : FPCA Method

- Fit the logistic regression model with the train set, and inverse link function is,

$$g^{-1}\left(\hat{\alpha} + \sum_{m=1}^{M} \hat{\beta}_m \hat{\epsilon}'_m\right) = \hat{\pi}_i, \quad i = 1, 2, ..., n.$$

- Compare the predicted group with real group

```
model <- glm(group ~ ., data=set.e$train, family = binomial)
pi.hat <- predict(model, set.e$test, type="response")
pred <- ifelse(pi.hat > 0.5, 1, 0)
c.tab <- table(set.e$test$group, pred)
```

```
> c.tab
    pred
      0  1
g1 45  5
g2  2 48
```

3. Simulation Results : B-Spline Method

- Estimate $\hat{\gamma}_l(t)$ with $\hat{\mu}'(t)$ and the generated train set via fdata2fd function

$$\hat{X}_i(t) = \hat{\mu}'(t) + \sum_{l=1}^{q} \hat{\gamma}_{il}\hat{B}_l(t_{ij}) \rightarrow \sum_{l=1}^{q} \hat{\gamma}_{il}\hat{B}_l(t_{ij}) = \hat{X}_i(t) - \hat{\mu}'(t)$$

- Then, compare the predicted group with real group again

```
> set.g$train
        bspl4.1     bspl4.2     bspl4.3     bspl4.4       bspl4.5
1   -0.62616469  0.24564096 -0.34526021  0.79890592 -0.2799767746
2    0.80388548 -1.23495369  0.90478820 -0.55219458  0.3871989234
3   -1.52332520 -0.58778991  0.79114818  0.27916429  0.9497160121
4   -0.57219810 -0.24336485 -0.28915390  0.66955174  0.1559336510
5   -0.97656065  1.82230530 -2.27876606  1.37854016 -0.2618415784
6   -0.58135191 -0.41349177  0.49447364 -0.28318202  0.6959187765
7   -0.34948635 -0.78836300  0.44450431  0.56145459  0.2002993960
8   -2.23394579  0.45480243 -0.22007230  0.68499553  0.5731021470
9   -2.07672726 -1.03434167  1.74478320  0.13489758  0.9155496583
10  -1.95039882  1.52990490 -0.67697929  0.17207756  0.1265386771
```

3. Simulation Results : Classification Error Rates

- For each of the 100 simulated datasets, classification error rates were calculated for the test data based on FPCA and B-spline methods
- Except for the case with No=1, the overall classification error rates based on FPCA are always lower that those observed for B-splines

Table: Classification error rates based on FPCA ans B-Splines(B-S)

No. of FPCs or base functions	Group 1 FPCA	B-S	Group 2 FPCA	B-S	overall FPCA	B-S
1	32.72 (8.41)	27.32 (7.86)	32.70 (8.31)	26.90 (7.86)	32.71 (5.26)	27.11 (4.58)
2	22.16 (6.65)	24.08 (6.37)	22.06 (6.15)	24.80 (6.55)	22.11 (4.33)	24.44 (3.97)
3	7.58 (4.58)	7.92 (3.96)	8.26 (5.34)	8.76 (4.71)	7.92 (3.35)	8.34 (2.70)
4	7.14 (4.14)	8.18 (4.18)	7.62 (5.10)	8.98 (5.00)	7.38 (3.11)	8.58 (2.96)
5	7.40 (4.07)	7.68 (4.29)	7.86 (5.26)	8.58 (5.01)	7.63 (3.06)	8.13 (3.15)

References I

Hans-Georg Müller Xiaoyan Leng.
Classification using functional data analysis for temporal gene expression data.
BIOINFORMATICS, 22(1):68–76, 2006.

Thank You!