12-18-2012

Classification of Genotype and Age of Eyes Using RPE Cell Size and Shape

Jie Yu

CLASSIFICATION OF GENOTYPE AND AGE OF EYES USING RPE CELL SIZE AND SHAPE

by

JIE YU

Under the Direction of Yi Jiang

ABSTRACT

Retinal pigment epithelium (RPE) is a principal site of pathogenesis in age-related macular degeneration (AMD). AMD is a main source of vision loss even blindness in the elderly and there is no effective treatment right now. Our aim is to describe the relationship between the morphology of RPE cells and the age and genotype of the eyes. We use principal component analysis (PCA) or functional principal component method (FPCA), support vector machine (SVM), and random forest (RF) methods to analyze the morphological data of RPE cells in mouse eyes to classify their age and genotype. Our analyses show that amongst all morphometric measures of RPE cells, cell shape measurements (eccentricity and solidity) are good for classification. But combination of cell shape and size (perimeter) provide best classification.

INDEX WORDS: Principal component analysis, Functional principal component analysis, Support vector machine, Random forest, Retinal pigment epithelium

CLASSIFICATION OF GENOTYPE AND AGE OF EYES USING RPE CELL SIZE AND SHAPE

by

JIE YU

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science

in the College of Arts and Sciences

Georgia State University

2012

CLASSIFICATION OF GENOTYPE AND AGE OF EYES USING RPE CELL SIZE AND SHAPE

by

JIE YU

Committee Chair: Yi Jiang

Committee: Xin Qi

Yichuan Zhao

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

December 2012

ACKNOWLEDGEMENTS

First and foremost, I offer my sincere gratitude to my advisor, Dr Yi Jiang, who has supported and helped me throughout my thesis research. Without her help, my thesis would not have been written. It is an honor for me to have such an encouraging and patient professor.

I would like to thank Dr Xin Qi, he has helped me with many statistical methods and helped me go through some problems that I encountered in the process.

The Department of Mathematics and Statistics has provided the support that I needed to complete my thesis. I am indebted to many professors and classmates who have helped me.

Finally, I would like to thank my parents who have supported me throughout all my studies in University.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

Age-related macular de-generation (AMD) is the leading cause of severe irreversible central vision loss and legal blindness in individuals 65 years of age or older in the United States and other developed countries [1-3]. Since the number of elderly persons will double by 2020, AMD is expected to become a major public health problem. Two forms of AMD are recognized [4, 5]. The non-neovascular form (also known as ‘‘dry’’ or ‘‘nonexudative’’) represents an early form of AMD usually associated with little visual acuity loss. It is characterized by atrophic abnormalities of the retinal pigment epithelium (RPE) and drusen, small lesions at the level of the RPE that contain granular and vesicular lipid-rich material. Over time, however, this form of AMD often progresses to the neovascular (also known as ‘‘wet’’ or ‘‘exudative’’) form of AMD that results in significant vision loss due to the appearance of choroidal neovascularization (CNV). Although the precise events that contribute to the development of AMD remain uncertain, recent studies have implicated various immunological and inflammatory mechanisms [6] and adhesion failure [7].

RPE is a principal site of pathogenesis of AMD. Situated just outside the neurosensory retina, firmly attached to the underlying choroid and overlying retinal visual cells, RPE not only shields the retina from excess light, but also nourishes retinal visual cells. Aging and disease progression, including lipofuscin deposition, drusen formation, and inflammation, all pose many different stresses on the RPE. Our hypothesis is that different stresses cause different deformations on the RPE cells, such that the RPE morphology reflects the various underlying causes and thus is descriptive of AMD status and age. To test this hypothesis, we will analyze the relationship between RPE cell morphology and the age and disease progression of the eye. This thesis focuses on the classification analysis of age and genotype of the eye using RPE morphometric data in mouse eyes from different ages and two genotypes.

Available to us through collaboration with Emory Eye Center are a large data set of mouse RPE flatmount images and the morphometric measurements. The morphometric analysis for RPE has only very recently commenced [8]. Professor Xin Qi has performed classification of age and genotype of the mouse eyes using RPE cells according to Genotype and age based on the parameters area (a measure of size) and aspect ratio (a measure of cell shape) [9]. We will extend this work to testing other cell morphometric parameters, including number of neighbors, eccentricity, solidity, and perimeter, to classify the genotype and age of the mouse eyes. I will use principal component analysis, in junction with support vector machine, and random forest methods.

Principal component analysis (PCA) is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components [10]. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to (i.e., uncorrelated with) the preceding components. By using only the first two principal components, the dimensionality of the data is significantly reduced. For discrete variables, PCA is applied in order to reduce the dimensionality. For the RPE data set that we used, the original sample size is large. Because each variable is measured for all individual cells identified from the RPE, we have approximately 10000 data points for every variable for each of the 123 eyes. By applying PCA, the size of the data is reduced to 2 principal components for every variable for each of the 123 eyes. For continuous variables, functional principal component analysis (FPCA) is applied in order to reduce the dimensionality and keep its property of functional data. In FPCA, an eigenfunction is associated with each eigenvalue, rather than an eigenvector in PCA. These eigenfunctions describe major variational components. Applying a rotation to them often results in a

more interpretable picture of the dominant modes of variation in the functional data, without changing the total amount of variation [11]. Similar to principal component analysis, FPCA also reduces the data from around 10000 to 2 principal components for every variable for each of the 123 eyes.

Supervised classification algorithms have been developed to classify data according to some given learning samples. Linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), support vector machine (SVM) and random forest (RF) are some of the representatives of supervised classification methods. LDA and QDA are parametric methods, while SVM is distribution-free, and RF is a voting method. LDA is used in statistics, pattern recognition and machine learning to find a linear combination of features that characterizes or separates two or more classes. The resulting combination of features may be used as a linear classifier, or more commonly, for dimensionality reduction before further classification [12]. QDA is closely related to LDA, but with the assumption that the measurements from each class are normally distributed and no assumption that the covariance of each of the classes is identical [13]. SVM is a set of related supervised learning methods that analyze data and recognize patterns, used for classification and regression analysis [14]. RF is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the classes output by individual trees [15]. Different methods have their pros and cons. Mr Folarinde has recently performed a series of LDA and QDA analysis on the RPE data [16].

1.1 Purpose of the Study

The main purpose of the study is to classify the genotype and age of mouse eyes using RPE cell morphology data. In particular we will test cell size and shape measures, including number of neighbors, eccentricity, solidity and perimeter to identify which ones best classify the RPE cells according to Genotype and age.

1.2 Expected Results

We expect that some of the morphometric measures for RPE cells will act as better classifiers than others; and that some combination of the morphometric measures will serve as much better classifiers for genotype and age of the eyes. We expect the results will confirm the central hypothesis by demonstrating the connection between RPE morphology and age and AMD status of the eye.

2 METHODOLOGY AND RESULTS

2.1 Data description

The flatmount RPE images were obtained at John Nickerson's Lab at the Emory Eye Center. The protocols for obtaining flatmount RPE images are briefly as follows.

The mouse eye was fixed with formalin for 10 minutes. Then on a microscope slide, any extra scleral tissue from eye including optic nerve was cut away. From puncture 4 cuts were extended using 3 mm scissors from cornea back towards optic nerve; each section was unfurled to reveal and remove the lens. 4.5 μl of Zymed rabbit anti-ZO-1 antibody and 0.45 μl of Oregon green conjugated anti-rabbit IgG (Invitrogen) was added to 450 μl of antibody buffer. The images were taken using a Nikon C1 confocal microscopy with 3 optical sections 5 μm apart as the Z-stacks; each image was 1024x1024 pixels in size.

Confocal images were stitched together using Adobe Photoshop CS2. Cut-boxes of equal size (181 x 266 pixels, or 225 x 331 μm) were cropped from the merged flatmount image from areas devoid of dissection artifacts. As many cut-boxes as possible were taken from each image (45-60 cut-boxes per image). Figure 1 shows a typical merged flatmount image of a C57BL/6J eye. Over all 123 eyes of three genotypes were collected. C57BL/6J is a wildtype, RD10 and RPE65 are mutants with deletions in the RPE related genes.

Twenty-one (21) morphometric measurements, including cell location, cell area, solidity, eccentricity, form factor, and number of neighbors were calculated using CellProfiler [17].

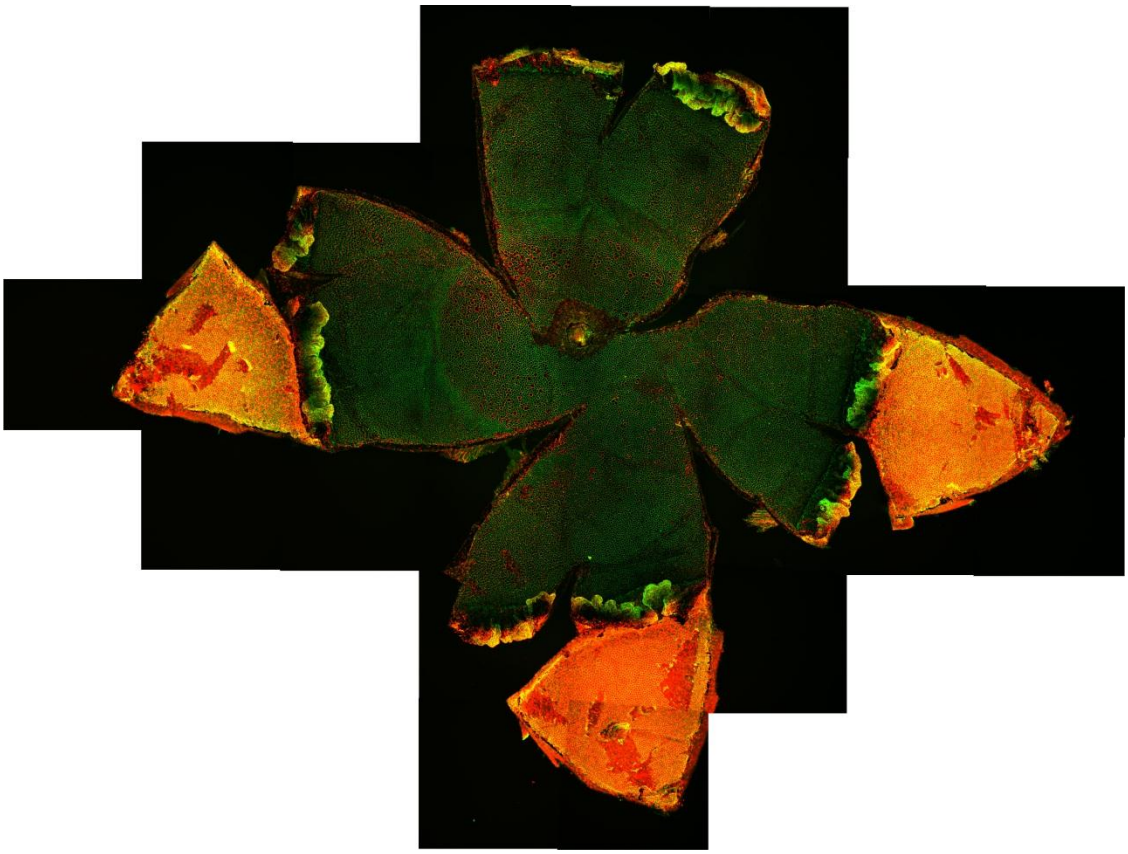

Figure 1 Flatmount RPE image of the whole mouse eye, photomerged from individual high resolution microscopy images. Green represents the cell boundary while red represents the cell nucleus.

Each eye contains 28 variables and approximately 8500 observations, each observation representing one cell in the eye. There are a total of 123 eyes belonging to three different genotypes and two age groups. We used genotype and one cutoff value of age for each genotype to classify the eyes into the following six groups. 400 days (post natal) was the best to differentiate for the genotype RPE65-/-. 70 days was the best cutoff value in age for c57BL/6J and rd10 [9]. We then segregated the data into two age groups: below the cutoff age (Young) and above the cutoff age (Old).

Table 1 Definition and sample size of the six groups. Sample size refers to number of eyes.

Group	Sample Size	Genotype	Age (days)
1	16	RPE65-/-	<400
2	3	RPE65-/-	>=400
3	23	c57BL/6J	<70
4	20	c57BL/6J	>=70
5	27	rd10	<70
6	34	rd10	>=70

2.2 Cell number of neighbors alone is not a good classifier

We first chose the number of neighbors as a potential good classifier for the eyes based on close observations of the RPE images (Figure 1). We see that RPE cells in a C57BL/6J eye would have more homogeneous size and rather hexagonal packing, while RPE cells in an rd10 eye would have more varied sizes and distorted shapes, and the cell packing is far from hexagonal.

For each eye, there are approximately 8500 cells, which mean that there are around 8500 observations. Since the number of neighbors is a discrete variable, the range of the value for the number of neighbors is from 3 to 15. Each eye has one such frequency. As a result, there are a total of 123 of such frequencies.

The following three steps were followed to classify the six groups.

(a) Frequencies were generated for the number of neighbors for each eye.

(b) Principal component analysis (PCA) method was applied to the frequencies for all eyes to reduce the dimension. The first two principal components which have the largest variance were chosen.

(c) Four classification methods were applied to the two components obtained from step (b): Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Support Vector Machine

(SVM) and Random Forest (RF).One observation is selected as the testing set and the rest of the observations are selected as the training set. This is iterated until all the observations have been selected as the testing set.

The plot of the first and second principal component scores were shown below.

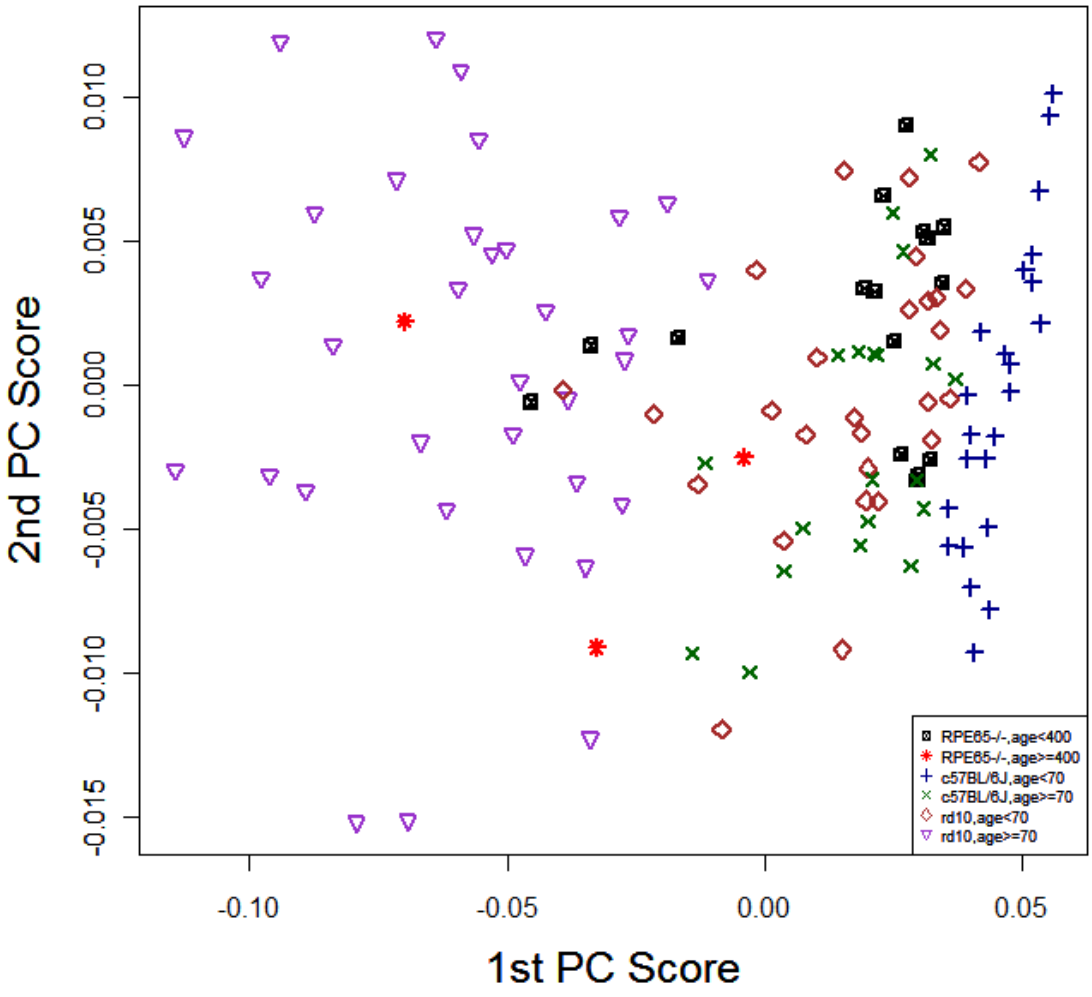

Figure 2 Scatterplot of the 1st and 2nd PC scores of the number of neighbors for all six classes

The scatterplot (figure 2) of the first and second PC scores shows that all the six groups are mixed together and cannot be distinguished cleanly.

The accuracies of classification using LDA, QDA, SVM and RF are listed in Table 2. Note that they are better than random guesses of 1/6, but even the highest prediction rate (83.2% by RF) is not very impressive.

Table 2 Average prediction rate for six groups using number of neighbors

Methods	LDA	QDA	SVM	RF
Prediction Rate	58.5%	61.8%	62%	83.2%

One of the problems that I noticed while classifying the six groups is that the two groups with genotype: RPE65-/- has limited numbers of observations, which makes them difficult to be distinguished from other groups. As a result, excluding these two groups from the classification might be the best way to improve the prediction rate.

2.3 Cell perimeters alone is not a good classifier

Cells of different genotypes and different ages have different sizes, so the perimeters are different. We test if perimeter could be a good classification parameter. The tools used to analyze the perimeter are a little different from the tools used to analyze the number of neighbors, because the number of neighbors is a discrete variable and perimeter is continuous. It is better to treat perimeter as functional data and use the density of perimeter. We use functional principal component analysis for perimeters. We follow the following four steps to classify the six groups.

(a) Density functions of perimeters were generated for the perimeters for all cells in each eye. Density curves of perimeters were shown below.

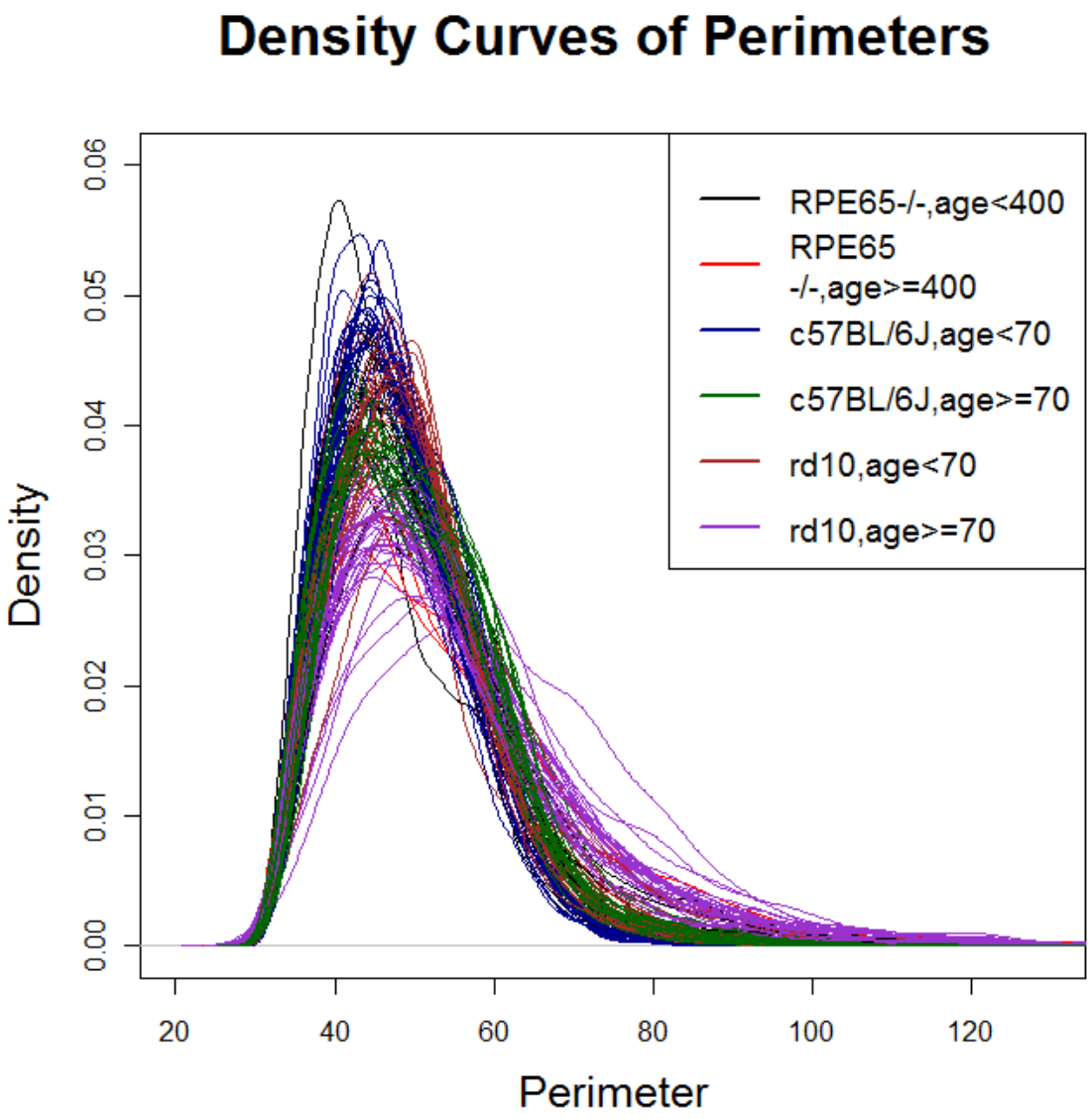

Figure 3 Density Curves of Perimeters

(b) Functional Principal component analysis (FPCA) method was applied to the density functions for all eyes to reduce the dimension. The first two principal components which have the largest variance were chosen.

(c) SVM method was applied to the two components obtained from step (b).

(d) One observation is selected as the testing set and the rest of the observations are selected as the training set. This is iterated until all the observations have been selected as the testing set.

From the scatter plot of the first and second principal component scores, we see that the six groups are without clear distinction between the groups.

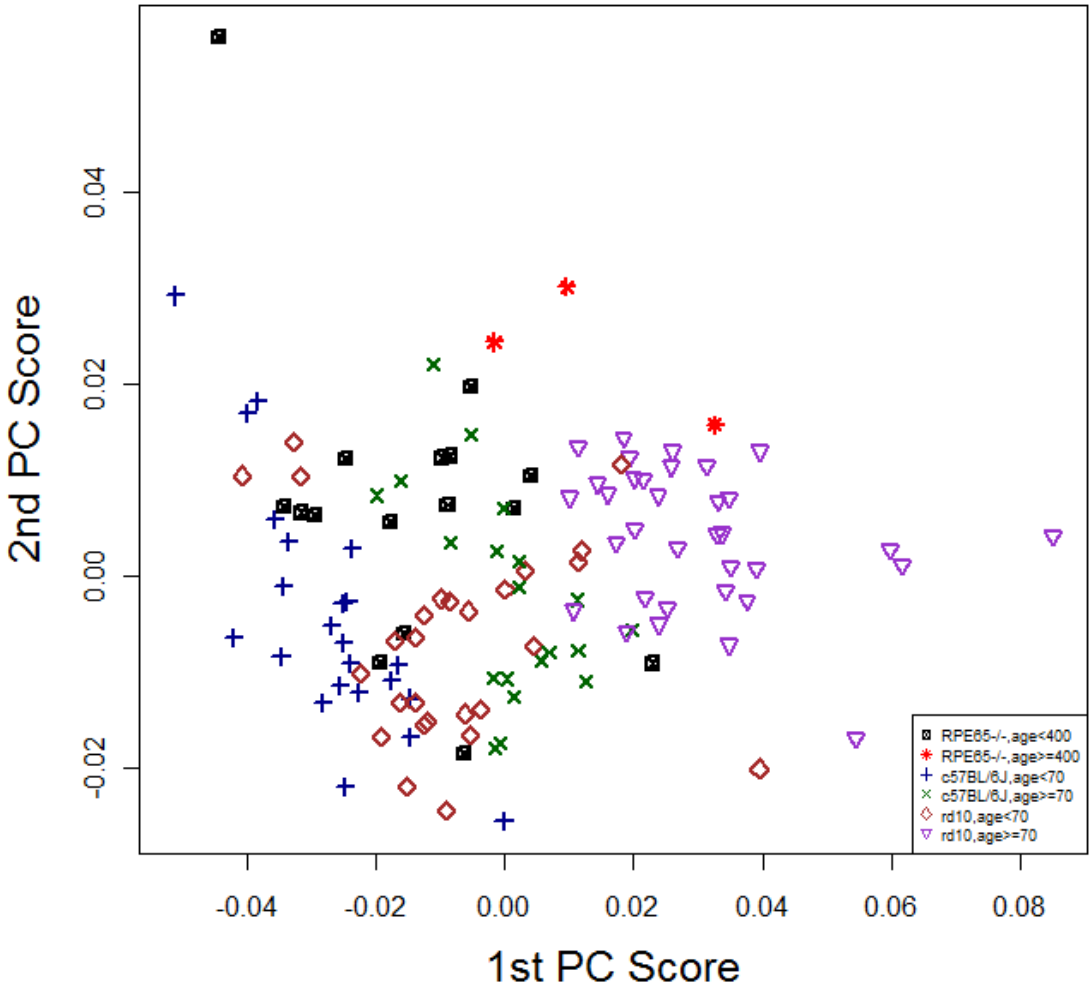

Figure 4 Scatterplot of 1st and 2nd PC scores of the perimeter for all six groups

The prediction rate using the support vector machine method is 58%. This is similar to the number of neighbors.

Again, in order to improve the prediction rate, only the last four groups were included.

Steps (a) and (b) are the same as before.

(c) LDA, QDA, SVM and RF methods were applied to the two components obtained from step (b)

The new classification results are listed in Table 3. We see similar accuracy in predicting these groups. Based on these analyses, we suggest that perimeter is not a good variable for classifying genotype and age of the mouse eyes.

Table 3 Prediction rate using Perimeter on four groups

Group	Method	c57BL/6J& Age<70	c57BL/6J& Age>=70	rd10& Age<70	rd10& Age>=70
Prediction Rate	LDA	75%	66.7%	50%	89.1%
	QDA	64.3%	72.2%	45%	86.8%
	SVM	73.9%	77.7%	53.8%	86.5%
	RF	42.6%	23.8%	25.4%	70.6%

2.4 Cell orientation alone is not a good classifier based on four groups

The orientation of an object is defined as the imaginary rotation that is needed to move the object from a reference placement to its current placement. It is a good description of how a cell is placed in space.

In order to improve the prediction rate, only the last four groups were included.

We follow the four steps as in 2.3 to classify the four groups. We see the scatter plot of the first two PC scores (Figure 4) the four groups are mixed together, making it impossible to distinguish the four groups.

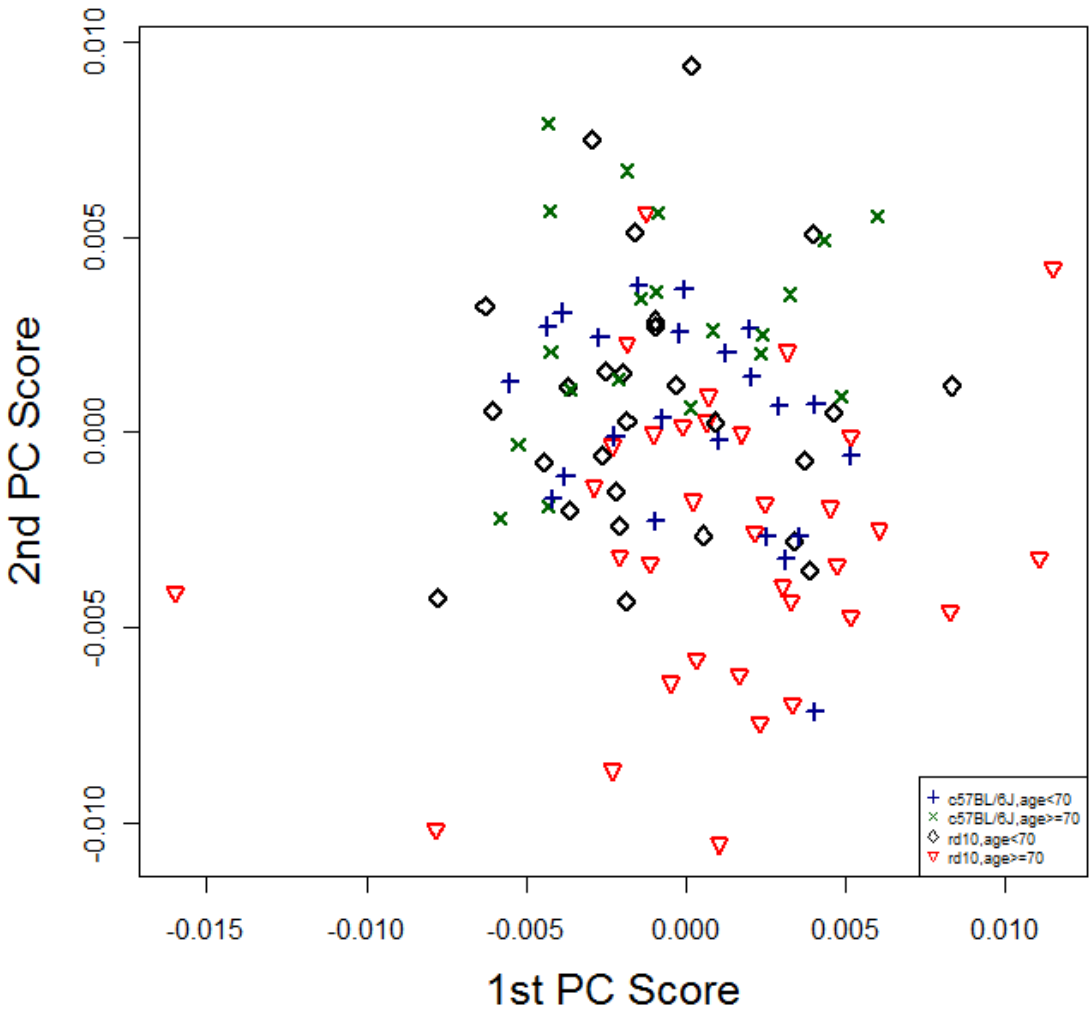

Figure 5 Scatterplot of the 1st and 2nd PC scores of orientation for four groups

(c) LDA, QDA, SVM and RF methods were applied to the two components obtained from step (b)

Results:

Table 4 Prediction rate using Orientation

Group	Method	c57BL/6J& Age<70	c57BL/6J& Age>=70	rd10& Age<70	rd10& Age>=70
Prediction Rate	LDA	0%	28.6%	24.4%	48.9%
	QDA	0%	20%	27.3%	50%
	SVM	0%	25%	33%	44.9%
	RF	7.3%	16%	17.8%	50%

From the table, we can see that the prediction rate is very low, confirms that orientation is not a good variable for prediction.

2.5 Cell eccentricity is a good classifier based on four groups

Then we tried with the variable eccentricity. Eccentricity is the amount by which its orbit deviates from a perfect circle. Eccentricity is a shape parameter with the potential to differentiate cells according to genotype and age.

In order to improve the prediction rate, only the last four groups were included.

We followed the four steps described in section 2.4 to classify the four steps.

The plot of the first and second principal component scores were shown below.

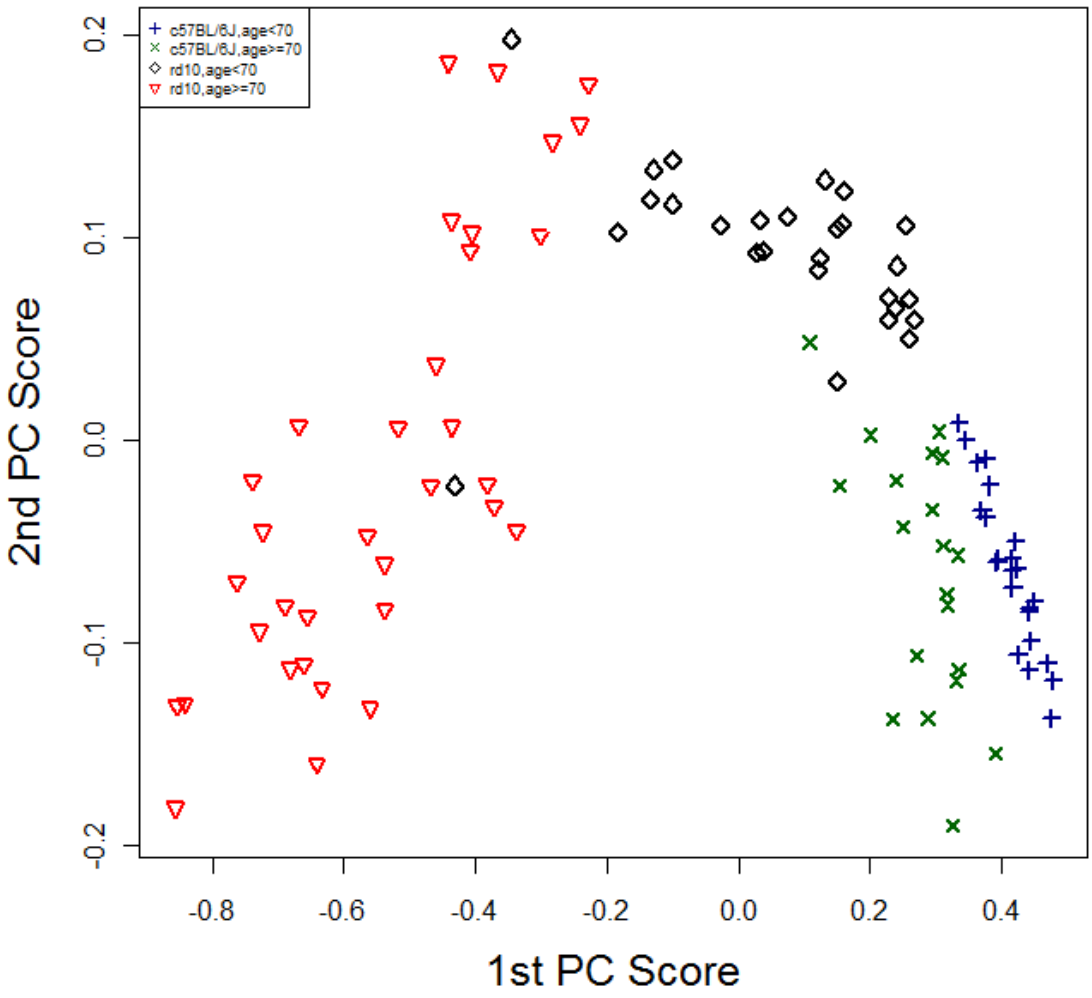

Figure 6 Scatterplot of the 1st and 2nd PC scores of eccentricity for four groups

It can be seen that the four groups can be easily classified according to this plot. This proved that eccentricity might be a good variable for classification.

(c) LDA, QDA, SVM and RF methods were applied to the two components obtained from step (b)

Results:

Table 5 Prediction rate using eccentricity

Group	Method	c57BL/6J& Age<70	c57BL/6J& Age>=70	rd10& Age<70	rd10& Age>=70
Prediction Rate	LDA	74%	80%	89.3%	94.1%
	QDA	100%	100%	96.1%	94.4%
	SVM	73.3%	91.7%	96.2%	94.4%
	RF	95.45%	85.71%	95.83%	91.89%

From the table, we can see that eccentricity is a good variable for prediction.

2.6 Cell solidity is a good classifier based on four groups

Solidity is a variable with similar characteristic of Eccentricity. Therefore, it has the potential to be a good classifier.

In order to improve the prediction rate, only the last four groups were included.

We followed the four steps described in section 2.4 to classify the four steps.

The plot of the first and second principal component scores were shown below.

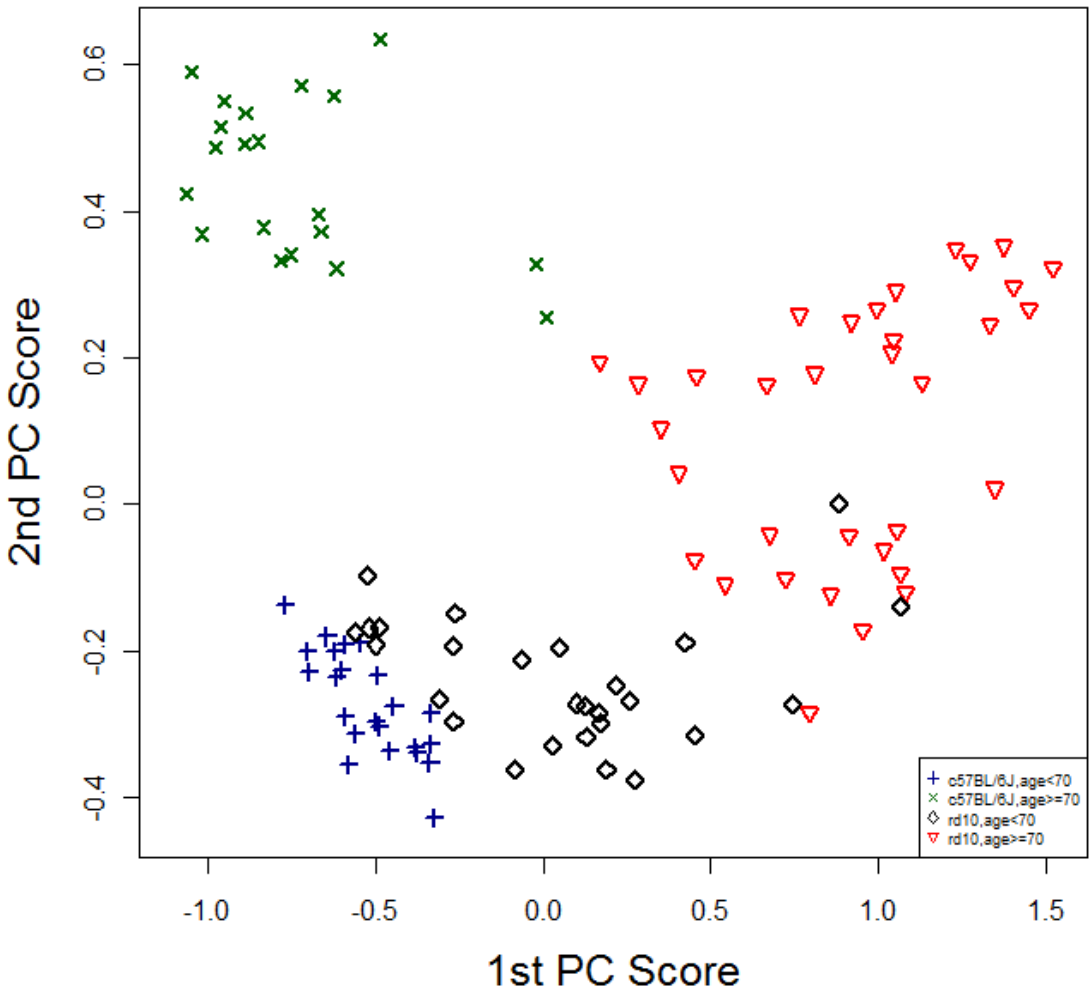

Figure 7 Scatterplot of the 1st and 2nd PC scores of solidity for four groups

It can be seen that the four groups can be easily classified according to this plot. This proved that solidity might be a good variable for classification.

(c) LDA, QDA, SVM and RF methods were applied to the two components obtained from step (b)

Results:

Table 6 Prediction rate using Solidity

Group	Method	c57BL/6J& Age<70	c57BL/6J& Age>=70	rd10& Age<70	rd10& Age>=70
Prediction Rate	LDA	76.7%	100%	85.7%	91.7%
	QDA	87.5%	100%	88%	91.7%
	SVM	76.7%	100%	94.4%	89.2%
	RF	87.5%	94.4%	86.3%	86.6%

From the table, we can see that solidity is a good variable for prediction.

2.7 Combination of cell solidity, cell eccentricity and cell perimeter provide to be the best classifier in our study

Combination of solidity, eccentricity and perimeter together might produce a better result since they contain more information than one variable or two variables. As can be seen from previous results, solidity and eccentricity have proven to be very good classifier alone. However, both of them are shape parameters while perimeter describes the cells in a different aspect. It might be better to include perimeter in the combination.

The following three steps were followed to classify the four groups.

(a) Density functions were generated for solidity, eccentricity and perimeter separately for all cells in each eye.

(b) Functional Principal component analysis (FPCA) method was applied to the density functions of solidity, eccentricity and perimeter separately for all eyes to reduce the dimension. The first two principal components which have the largest variance were chosen for each variable.

(c) LDA, QDA, SVM and RF methods were applied to the six components (6 by 123 matrix) obtained from step (b)

Results:

Table 7 Prediction rate using Eccentricity, Solidity and Perimeter

Group	Method	c57BL/6J& Age<70	c57BL/6J& Age>=70	rd10& Age<70	rd10& Age>=70
Prediction Rate	LDA	95.8%	100%	92.3%	94.1%
	QDA	100%	95%	85.7%	94.1%
	SVM	100%	100%	88.9%	91.9%
	RF	100 %	100%	100%	94.4%

From the table, we can see that the combination of solidity, eccentricity and perimeter maximizes the predictive power.

3 CONCLUSIONS

We used PCA and FPCA to reduce the dimension of the raw data. LDA, QDA, SVM and RF methods were then applied to classify the cells according to genotype and age. Eccentricity and solidity proved to be relatively good classifier. Numbers of neighbors, perimeter and orientation cannot classify the cells very accurately. This suggests that shape parameters might be good classifiers in our case. Parameters like number of neighbors or perimeter are not very good for classification. On the other hand, this also implies that cells of different genotype and different age different greatly in shape but are similar in terms of area.

In terms of which method is the best for prediction, there is no clear winner in our cases. LDA, QDA, SVM, and RF methods showed similar prediction rates. In building predictive models, parameter is a more important factor than method.

Combining more variables could improve the prediction rate since more information is incorporated. By combining eccentricity, solidity and perimeter, the prediction rate is almost 100% for three groups, indicating that there is little room for improvement. However, there is a trade-off of the accuracy and the computing time. When three variables are selected, the computing time would be at least tripled while the improvement in prediction rate is not significant. It all depends on the need. If accuracy is the most important factor in consideration, then perhaps more variables, not just three variables should be used.

REFERENCES

[1] Klein R, Klein BE, Linton KL (1992) Prevalence of age-related maculopathy. The Beaver Dam Eye Study. Ophthalmology 99: 933–943.

[2] Klein R, Klein BE, Jensen SC, Meuer SM (1997) The five-year incidence and progression of age-related maculopathy: the Beaver Dam Eye Study. Ophthalmology 104: 7–21.

[3] Smith W, Assink J, Klein R, Mitchell P, Klaver CC, et al. (2001) Risk factors for age-related macular degeneration: Pooled findings from three continents. Ophthalmology 108: 697–704.

[4] Hagedorn CL, Adelman RA (2006) Age-related macular degeneration. In: Tombran-Tink J, Barnstable CJ, eds. Ocular Angiogenesis: Diseases, Mecha- nisms, and Therapeutics. Totowa, NJ: Humana Press Inc. pp 3–22.

[5] O'Connell SR, Bressler N (1999) Age-related macular degeneration. In: Regillo G, Flynn Jr. H, eds. Vitreoretinal diseases: The essentials. New York: Thieme Medical Publishers. pp 213–240.

[6] Cousins SW, Espinosa-Heidmann DG, Miller DM, Pereira-Simon S, Hernandez EP, et al. (2012) Macrophage Activation Associated with Chronic Murine Cytomegalovirus Infection Results in More Severe Experimental Choroidal Neovascularization. PLoS Pathog 8(4): e1002671. doi:10.1371/journal.ppat.1002671

[7] Shirinifard A, Glazier JA, Swat M, Gens JS, Family F, et al. (2012) Adhesion Failures Determine the Pattern of Choroidal Neovascularization in the Eye: A Computer Simulation Study. PLoS Comput Biol 8(5): e1002440. doi:10.1371/journal.pcbi.1002440

[8] Micah A. Chrenek, Nupur Dalal, Christopher Gardner, Hans Grossniklaus and Yi Jiang, et al. (2012) Analysis of the RPE Sheet in the rd10 Retinal Degeneration Model. Advances in Experimental Medicine and Biology, 1, Volume 723, Retinal Degenerative Diseases, Part 8, Pages 641-647

[9] Jiang, Qi, et al., (2012) Functional Principal Component Analysis Reveals Discriminating Categories of Retinal Pigment Epithelial Morphology in Mice, in preparation.

[10] Principal components method: In Wikipedia, http://en.wikipedia.org/wiki/Principle_components

[11] J.O. Ramsay & B.W. Silverman (2009) Functional data analysis with R and Matlab.

New York: Springer.

[12] Linear discriminant analysis: In Wikipedia,

http://en.wikipedia.org/wiki/Linear_discriminant_analysis

[13] Quadratic classifier: In Wikipedia, http://en.wikipedia.org/wiki/Quadratic_classifier

[14] Support vector machine: In Wikipedia, http://en.wikipedia.org/wiki/Support_vector_machine

[15] Random forest: In Wikipedia, http://en.wikipedia.org/wiki/Random_forest

[16] MICHEAL S FOLARINDE, (2012) A COMPARATIVE STUDY BETWEEN GENOTYPES AND AGES OF EYES USING MORPHOMETRIC MEASURES OF RETINAL PIGMENT EPITHELIAL CELLS, MS. Thesis, Georgia State University.

[17] Lamprecht, M.R., D.M. Sabatini, and A.E. Carpenter, (2007) CellProfiler: free, versatile software for automated biological image analysis. Biotechniques, 2007. 42(1): p. 71-5.

APPENDIX

R code:

```
#Using Linear Discriminant Analysis and Quadratic Discriminant Analysis to classify cells based on Number of neighbors
library(foreign)
setwd("C:/Users/matyyx/Dropbox/Jie Yu/RPE/original")
a<-list.files()
length<-length(a)
min=5;
max=5;
for (x in a)
{u<-read.csv(x)
if (min(u$Neighbors_NumberOfNeighbors_0)<min)
{min=min(u$Neighbors_NumberOfNeighbors_0)}
if (max(u$Neighbors_NumberOfNeighbors_0)>max)
{max=max(u$Neighbors_NumberOfNeighbors_0)}
}
width<-max-min+2
b<-matrix(1:length*width,length,width)
i<-1
for (x in a)
{u<-read.csv(x)
for (y in (min:max))
{b[i,y-min+1]<-sum(u$Neighbors_NumberOfNeighbors_0==y)}
```

```
        b[i,width]<-1*sum(u$Genotype[1]=="RPE65-/-
")+2*sum(u$Genotype[1]=="c57BL/6J")+3*sum(u$Genotype[1]=="rd10")
        i<-i+1
        }
        library(MASS)
        pc.cov1<-prcomp(b[,-width])
        pca.point.color = function(model.id) {
            if (model.id == 1) {
                return("black")
            } else if (model.id == 2) {
                return("red")
            } else if (model.id == 3) {
                return("blue")
            }
        }
        plot(pc.cov1$x[,1],pc.cov1$x[,2],col=sapply(b[,width], pca.point.color))
        summary(pc.cov1)
        cov<-as.matrix(pc.cov1$rotation[,1:2])
        score<-as.matrix(b[,-width])%*%cov
        predict<-as.matrix(b[,width])
        lda1<-lda(score,predict,CV=TRUE)
        error1<-sum(lda1$class!=predict)/length(b[,width])
        qda1<-qda(score,predict,CV=TRUE)
        error2<-sum(qda1$class!=predict)/length(b[,width])
```

```
#Using support vector machine to classify cells based on Number of neighbors
library(foreign)
setwd("C:/Users/JIE YU/Dropbox/Jie Yu/RPE/original")
a<-list.files()
length<-length(a)



min=6;
max=6;
for (x in a)
{u<-read.csv(x)
if (min(u$Neighbors_NumberOfNeighbors_0)<min)
{min=min(u$Neighbors_NumberOfNeighbors_0)}
if (max(u$Neighbors_NumberOfNeighbors_0)>max)
{max=max(u$Neighbors_NumberOfNeighbors_0)}
}


max<-max-3
width<-max-min+2
b<-matrix(1:length*width,length,width)
i<-1
```

```
for (x in a)
{u<-read.csv(x)
for (y in (min:max))
{b[i,y-min+1]<-
sum(u$Neighbors_NumberOfNeighbors_0==y)/length(u$Neighbors_NumberOfNeighbors_0)}
b[i,width]<-1*sum(u$Genotype[1]=="RPE65-/-
")*sum(u$Age[1]<400)+2*sum(u$Genotype[1]=="RPE65-/-
")*sum(u$Age[1]>=400)+3*sum(u$Genotype[1]=="c57BL/6J")*sum(u$Age[1]<70)+4*sum(u$Genotype[1
]=="c57BL/6J")*sum(u$Age[1]>=70)+5*sum(u$Genotype[1]=="rd10")*sum(u$Age[1]<70)+6*sum(u$Gen
otype[1]=="rd10")*sum(u$Age[1]>=70)
i<-i+1
}

b[,width]<-as.vector(b[,width])

library(MASS)
pc.cov1<-prcomp(b[,-13])
cov<-as.matrix(pc.cov1$rotation[,1:2])
score<-as.matrix(b[,-width])%*%cov



c<-matrix(1:length*3,length,3)
for (x in 1:length)
{c[x,1]=as.matrix(score[x,1])
```

```
c[x,2]=as.matrix(score[x,2])

c[x,3]=b[x,13]

}


c[,3]<-as.vector(c[,3])


width<-3


library(e1071)


c[,3]<-as.factor(c[,3])

width<-3

sum1<-0

times<-1000

for (x in 1:times)

{index<-1:length

testindex <- sample(index,2)

testset <- c[testindex,]

names(testset)<-c("V1","V2","V3")

testset<-as.data.frame(testset)




trainset <- c[-testindex,]
```

```
names(trainset)<-c("V1","V2","V3")

trainset<-as.data.frame(trainset)

trainset[,width]<-as.factor(trainset[,width])


model <- svm(V3~., data = trainset)

prediction <- predict(model, testset[,-width])

prediction<-as.data.frame(prediction)

sum1<-sum1+sum(prediction==testset[,width])/2

}

rate<-sum1/times

prediction

rate



model <- rpart(V3~., data = trainset)

model <- svm(formula=V3~., data = trainset)


index<-1:length


testindex <- sample(index, trunc(length(index)/3))

testset <- c[testindex,]

trainset <- c[-testindex,]

e<-as.matrix(lm(V2~V1,trainset)$coefficients)

f<-round(as.matrix(testset[,-2])%*%as.matrix(e[2,1])+as.matrix(e[1,1]*rep(1,34),34,1),digits=0)
```

```
sum(f+1==testset[,2])/34


# Using random forest to classify cells based on number of neighbors
library(foreign)
setwd("C:/Users/JIE YU/Dropbox/Jie Yu/RPE/original")
a<-list.files()
length<-length(a)



min=6;
max=6;
for (x in a)
{u<-read.csv(x)
if (min(u$Neighbors_NumberOfNeighbors_0)<min)
{min=min(u$Neighbors_NumberOfNeighbors_0)}
if (max(u$Neighbors_NumberOfNeighbors_0)>max)
{max=max(u$Neighbors_NumberOfNeighbors_0)}
}


max<-max-3
width<-max-min+2
b<-matrix(1:length*width,length,width)
i<-1
```

```
sumofindex<-0
for (x in a)
{u<-read.csv(x)
for (y in (min:max))
{b[i,y-min+1]<-
sum(u$Neighbors_NumberOfNeighbors_0==y)/length(u$Neighbors_NumberOfNeighbors_0)}
b[i,width]<-1*sum(u$Genotype[1]=="RPE65-/-
")*sum(u$Age[1]<400)+2*sum(u$Genotype[1]=="RPE65-/-
")*sum(u$Age[1]>=400)+3*sum(u$Genotype[1]=="c57BL/6J")*sum(u$Age[1]<70)+4*sum(u$Genotype[1
]=="c57BL/6J")*sum(u$Age[1]>=70)+5*sum(u$Genotype[1]=="rd10")*sum(u$Age[1]<70)+6*sum(u$Gen
otype[1]=="rd10")*sum(u$Age[1]>=70)
if (sum(u$Genotype[1]=="RPE65-/-")==1) {sumofindex<-sumofindex+1}
i<-i+1
}

length2<-length-sumofindex
h<-matrix(1:length2*width,length2,width)
j<-1
for (x in 1:length)
{ if (b[x,width]==3) {h[j,]<-b[x,]
 j<-j+1}
else if (b[x,width]==4) {h[j,]<-b[x,]
 j<-j+1}
else if (b[x,width]==5) {h[j,]<-b[x,]
```

```
j<-j+1}
else if (b[x,width]==6) {h[j,]<-b[x,]
j<-j+1}
}



h<-as.data.frame(h)
h$V13<-as.factor(h$V13)
b.rf<-randomForest(V13 ~., data=h,importance=TRUE,
                   proximity=TRUE)
print(b.rf)



b<-h
b<-as.data.frame(b)
sum3<-0
t3<-0
sum4<-0
t4<-0
```

```
sum5<-0

t5<-0

sum6<-0

t6<-0

sum7<-0

t7<-0

times<-1000

length<-dim(b)[1]

b[,width]<-as.factor(b[,width])

for (x in 1:times)

{index<-1:length

testindex <- sample(index,1)

c<-b

train<-c[-testindex,]

test<-c[testindex,]

b.rf<-randomForest(V13 ~., data=train,xtest=test[,-width],ytest=test[,width],importance=TRUE,

                   proximity=TRUE)

if (b.rf$test$predicted == 3) {

          sum3<-sum3+sum(b.rf$test$predicted==test[,width])

          t3<-t3+1

      } else if (b.rf$test$predicted == 4) {

          sum4<-sum4+sum(b.rf$test$predicted==test[,width])

          t4<-t4+1

      } else if (b.rf$test$predicted  == 5) {
```

```
        sum5<-sum5+sum(b.rf$test$predicted==test[,width])
        t5<-t5+1
    } else if (b.rf$test$predicted == 6) {
        sum6<-sum6+sum(b.rf$test$predicted==test[,width])
        t6<-t6+1
    }
if (b.rf$test$predicted == 5 || b.rf$test$predicted == 4) {
        sum7<-sum7+sum(test[,width]==5)+sum(test[,width]==4)
        t7<-t7+1
    }

}
rate3<-sum3/t3
rate4<-sum4/t4
rate5<-sum5/t5
rate6<-sum6/t6
rate7<-sum7/t7
rate3
rate4
rate5
rate6
rate7

# Using random forest to classify cells based on perimeter
```

```
library(foreign)
library(fda)
memory.limit(size=4095)
setwd("C:/Users/JIE YU/Dropbox/Jie Yu/RPE/original")
a<-list.files()
length<-length(a)

i<-1
sumofindex<-0
max<-0
min<-10000000
for (x in a)
{u<-read.csv(x)
if (max(density(u$AreaShape_Perimeter)$x)>max) {max<-
max(density(u$AreaShape_Perimeter)$x)}
if (min(density(u$AreaShape_Perimeter)$x)<min) {min<-
min(density(u$AreaShape_Perimeter)$x)}
}


d<-density(u$AreaShape_Perimeter)

b<-matrix(1:length,length,1)
```

```
b[1,1]<-1*sum(u$Genotype[1]=="RPE65-/-
")*sum(u$Age[1]<400)+2*sum(u$Genotype[1]=="RPE65-/-
")*sum(u$Age[1]>=400)+3*sum(u$Genotype[1]=="c57BL/6J")*sum(u$Age[1]<70)+4*sum(u$Genotype[1
]=="c57BL/6J")*sum(u$Age[1]>=70)+5*sum(u$Genotype[1]=="rd10")*sum(u$Age[1]<70)+6*sum(u$Gen
otype[1]=="rd10")*sum(u$Age[1]>=70)
i<-2

for (y in a)
{if (y!=x) {
u<-read.csv(y)
d$x<-cbind(d$x,density(u$AreaShape_Perimeter)$x)
d$y<-cbind(d$y,density(u$AreaShape_Perimeter)$y)
b[i,1]<-1*sum(u$Genotype[1]=="RPE65-/-
")*sum(u$Age[1]<400)+2*sum(u$Genotype[1]=="RPE65-/-
")*sum(u$Age[1]>=400)+3*sum(u$Genotype[1]=="c57BL/6J")*sum(u$Age[1]<70)+4*sum(u$Genotype[1
]=="c57BL/6J")*sum(u$Age[1]>=70)+5*sum(u$Genotype[1]=="rd10")*sum(u$Age[1]<70)+6*sum(u$Gen
otype[1]=="rd10")*sum(u$Age[1]>=70)
i<-i+1
}
}



datarange<-c(min,max)
bsp <- create.bspline.basis(datarange, 80)
```

```
f<-Data2fd(d$x,d$y,bsp)

g1<-pca.fd(f, nharm = 2)

g1<-cbind(g1$scores,b)

length<-dim(g1)[1]


sumofindex<-0

for (x in 1:length)

{if (g1[x,3]==1) {sumofindex<-sumofindex+1}

if (g1[x,3]==2) {sumofindex<-sumofindex+1}

}


length2<-length-sumofindex

width<-3

h<-matrix(1:length2*3,length2,3)

j<-1

for (x in 1:length)

{ if (g1[x,width]==3) {h[j,]<-g1[x,]

 j<-j+1}

else if (g1[x,width]==4) {h[j,]<-g1[x,]

 j<-j+1}

else if (g1[x,width]==5) {h[j,]<-g1[x,]

j<-j+1}

else if (g1[x,width]==6) {h[j,]<-g1[x,]

j<-j+1}
```

```
}



b<-h
b<-as.data.frame(b)
b[,width]<-as.factor(b[,width])
sum3<-0
t3<-0
sum4<-0
t4<-0
sum5<-0
t5<-0
sum6<-0
t6<-0
times<-1000
length<-dim(b)[1]
for (x in 1:times)
{index<-1:length
testindex <- sample(index,1)
c<-b
train<-c[-testindex,]
test<-c[testindex,]
b.rf<-randomForest(V3 ~., data=train,xtest=test[,-width],ytest=test[,width],importance=TRUE,
                   proximity=TRUE)
```

```
if (b.rf$test$predicted == 3) {
          sum3<-sum3+sum(b.rf$test$predicted==test[,width])
          t3<-t3+1
      } else if (b.rf$test$predicted == 4) {
          sum4<-sum4+sum(b.rf$test$predicted==test[,width])
          t4<-t4+1
      } else if (b.rf$test$predicted  == 5) {
          sum5<-sum5+sum(b.rf$test$predicted==test[,width])
          t5<-t5+1
      } else if (b.rf$test$predicted == 6) {
          sum6<-sum6+sum(b.rf$test$predicted==test[,width])
          t6<-t6+1
      }

}
rate3<-sum3/t3
rate4<-sum4/t4
rate5<-sum5/t5
rate6<-sum6/t6
rate3
rate4
rate5
rate6
```

```
# Using random forest to classify cells based on Orientation
library(foreign)
library(fda)
memory.limit(size=4095)
setwd("C:/Users/JIE YU/Dropbox/Jie Yu/RPE/original")
a<-list.files()
length<-length(a)

i<-1
sumofindex<-0
max<-0
min<-10000000
for (x in a)
{u<-read.csv(x)
if (max(density(u$AreaShape_Orientation)$x)>max) {max<-
max(density(u$AreaShape_Orientation)$x)}
if (min(density(u$AreaShape_Orientation)$x)<min) {min<-
min(density(u$AreaShape_Orientation)$x)}
}


d<-density(u$AreaShape_Orientation)

b<-matrix(1:length,length,1)
```

```
b[1,1]<-1*sum(u$Genotype[1]=="RPE65-/-
")*sum(u$Age[1]<400)+2*sum(u$Genotype[1]=="RPE65-/-
")*sum(u$Age[1]>=400)+3*sum(u$Genotype[1]=="c57BL/6J")*sum(u$Age[1]<70)+4*sum(u$Genotype[1
]=="c57BL/6J")*sum(u$Age[1]>=70)+5*sum(u$Genotype[1]=="rd10")*sum(u$Age[1]<70)+6*sum(u$Gen
otype[1]=="rd10")*sum(u$Age[1]>=70)
i<-2

for (y in a)
{if (y!=x) {
u<-read.csv(y)
d$x<-cbind(d$x,density(u$AreaShape_Orientation)$x)
d$y<-cbind(d$y,density(u$AreaShape_Orientation)$y)
b[i,1]<-1*sum(u$Genotype[1]=="RPE65-/-
")*sum(u$Age[1]<400)+2*sum(u$Genotype[1]=="RPE65-/-
")*sum(u$Age[1]>=400)+3*sum(u$Genotype[1]=="c57BL/6J")*sum(u$Age[1]<70)+4*sum(u$Genotype[1
]=="c57BL/6J")*sum(u$Age[1]>=70)+5*sum(u$Genotype[1]=="rd10")*sum(u$Age[1]<70)+6*sum(u$Gen
otype[1]=="rd10")*sum(u$Age[1]>=70)
i<-i+1
}
}



datarange<-c(min,max)
bsp <- create.bspline.basis(datarange, 800)
```

```
f<-Data2fd(d$x,d$y,bsp)

g1<-pca.fd(f, nharm = 2)

g1<-cbind(g1$scores,b)

length<-dim(g1)[1]


sumofindex<-0

for (x in 1:length)

{if (g1[x,3]==1) {sumofindex<-sumofindex+1}

if (g1[x,3]==2) {sumofindex<-sumofindex+1}

}


length2<-length-sumofindex

width<-3

h<-matrix(1:length2*3,length2,3)

j<-1

for (x in 1:length)

{ if (g1[x,width]==3) {h[j,]<-g1[x,]

 j<-j+1}

else if (g1[x,width]==4) {h[j,]<-g1[x,]

 j<-j+1}

else if (g1[x,width]==5) {h[j,]<-g1[x,]

j<-j+1}

else if (g1[x,width]==6) {h[j,]<-g1[x,]

j<-j+1}
```

```
}



b<-h
b<-as.data.frame(b)
b[,width]<-as.factor(b[,width])
sum3<-0
t3<-0
sum4<-0
t4<-0
sum5<-0
t5<-0
sum6<-0
t6<-0
times<-1000
length<-dim(b)[1]
for (x in 1:times)
{index<-1:length
testindex <- sample(index,1)
c<-b
train<-c[-testindex,]
test<-c[testindex,]
b.rf<-randomForest(V3 ~., data=train,xtest=test[,-width],ytest=test[,width],importance=TRUE,
                   proximity=TRUE)
```

```
if (b.rf$test$predicted == 3) {
          sum3<-sum3+sum(b.rf$test$predicted==test[,width])
          t3<-t3+1
      } else if (b.rf$test$predicted == 4) {
          sum4<-sum4+sum(b.rf$test$predicted==test[,width])
          t4<-t4+1
      } else if (b.rf$test$predicted  == 5) {
          sum5<-sum5+sum(b.rf$test$predicted==test[,width])
          t5<-t5+1
      } else if (b.rf$test$predicted == 6) {
          sum6<-sum6+sum(b.rf$test$predicted==test[,width])
          t6<-t6+1
      }

}
rate3<-sum3/t3
rate4<-sum4/t4
rate5<-sum5/t5
rate6<-sum6/t6
rate3
rate4
rate5
rate6
```

```
# Using random forest to classify cells based on Eccentricity
library(foreign)
library(fda)
memory.limit(size=4095)
setwd("C:/Users/JIE YU/Dropbox/Jie Yu/RPE/original")
a<-list.files()
length<-length(a)

i<-1
sumofindex<-0
max<-0
min<-10000000
for (x in a)
{u<-read.csv(x)
if (max(density(u$AreaShape_Eccentricity)$x)>max) {max<-
max(density(u$AreaShape_Eccentricity)$x)}
if (min(density(u$AreaShape_Eccentricity)$x)<min) {min<-
min(density(u$AreaShape_Eccentricity)$x)}
}

d<-density(u$AreaShape_Eccentricity)

b<-matrix(1:length,length,1)
```

```
	b[1,1]<-1*sum(u$Genotype[1]=="RPE65-/-
")*sum(u$Age[1]<400)+2*sum(u$Genotype[1]=="RPE65-/-
")*sum(u$Age[1]>=400)+3*sum(u$Genotype[1]=="c57BL/6J")*sum(u$Age[1]<70)+4*sum(u$Genotype[1
]=="c57BL/6J")*sum(u$Age[1]>=70)+5*sum(u$Genotype[1]=="rd10")*sum(u$Age[1]<70)+6*sum(u$Gen
otype[1]=="rd10")*sum(u$Age[1]>=70)
	i<-2

	for (y in a)
	{if (y!=x) {
	u<-read.csv(y)
	d$x<-cbind(d$x,density(u$AreaShape_Eccentricity)$x)
	d$y<-cbind(d$y,density(u$AreaShape_Eccentricity)$y)
	b[i,1]<-1*sum(u$Genotype[1]=="RPE65-/-
")*sum(u$Age[1]<400)+2*sum(u$Genotype[1]=="RPE65-/-
")*sum(u$Age[1]>=400)+3*sum(u$Genotype[1]=="c57BL/6J")*sum(u$Age[1]<70)+4*sum(u$Genotype[1
]=="c57BL/6J")*sum(u$Age[1]>=70)+5*sum(u$Genotype[1]=="rd10")*sum(u$Age[1]<70)+6*sum(u$Gen
otype[1]=="rd10")*sum(u$Age[1]>=70)
	i<-i+1
	}
	}



	datarange<-c(min,max)
	bsp <- create.bspline.basis(datarange, 800)
```

```
f<-Data2fd(d$x,d$y,bsp)

g1<-pca.fd(f, nharm = 2)

g1<-cbind(g1$scores,b)

length<-dim(g1)[1]


sumofindex<-0

for (x in 1:length)

{if (g1[x,3]==1) {sumofindex<-sumofindex+1}

if (g1[x,3]==2) {sumofindex<-sumofindex+1}

}


length2<-length-sumofindex

width<-3

h<-matrix(1:length2*3,length2,3)

j<-1

for (x in 1:length)

{ if (g1[x,width]==3) {h[j,]<-g1[x,]

 j<-j+1}

else if (g1[x,width]==4) {h[j,]<-g1[x,]

 j<-j+1}

else if (g1[x,width]==5) {h[j,]<-g1[x,]

j<-j+1}

else if (g1[x,width]==6) {h[j,]<-g1[x,]

j<-j+1}
```

```
}



b<-h
b<-as.data.frame(b)
b[,width]<-as.factor(b[,width])
sum3<-0
t3<-0
sum4<-0
t4<-0
sum5<-0
t5<-0
sum6<-0
t6<-0
times<-1000
length<-dim(b)[1]
for (x in 1:times)
{index<-1:length
testindex <- sample(index,1)
c<-b
train<-c[-testindex,]
test<-c[testindex,]
b.rf<-randomForest(V3 ~., data=train,xtest=test[,-width],ytest=test[,width],importance=TRUE,
                   proximity=TRUE)
```

```
if (b.rf$test$predicted == 3 && b.rf$test$votes[1]>0.85) {
          sum3<-sum3+sum(b.rf$test$predicted==test[,width])
          t3<-t3+1
      } else if (b.rf$test$predicted == 4 && b.rf$test$votes[2]>0.85) {
          sum4<-sum4+sum(b.rf$test$predicted==test[,width])
          t4<-t4+1
      } else if (b.rf$test$predicted  == 5  && b.rf$test$votes[3]>0.85) {
          sum5<-sum5+sum(b.rf$test$predicted==test[,width])
          t5<-t5+1
      } else if (b.rf$test$predicted == 6  && b.rf$test$votes[4]>0.85) {
          sum6<-sum6+sum(b.rf$test$predicted==test[,width])
          t6<-t6+1
      }

}
rate3<-sum3/t3
rate4<-sum4/t4
rate5<-sum5/t5
rate6<-sum6/t6
rate3
rate4
rate5
rate6
```

```
# Using random forest to classify cells based on solidity

library(foreign)

library(fda)

memory.limit(size=4095)

setwd("C:/Users/JIE YU/Dropbox/Jie Yu/RPE/original")

a<-list.files()

length<-length(a)


i<-1

sumofindex<-0

max<-0

min<-10000000

for (x in a)

{u<-read.csv(x)

if (max(density(u$AreaShape_Solidity)$x)>max) {max<-max(density(u$AreaShape_Solidity)$x)}

if (min(density(u$AreaShape_Solidity)$x)<min) {min<-min(density(u$AreaShape_Solidity)$x)}

}



d<-density(u$AreaShape_Solidity)


b<-matrix(1:length,length,1)

b[1,1]<-1*sum(u$Genotype[1]=="RPE65-/-
")*sum(u$Age[1]<400)+2*sum(u$Genotype[1]=="RPE65-/-
```

```
")*sum(u$Age[1]>=400)+3*sum(u$Genotype[1]=="c57BL/6J")*sum(u$Age[1]<70)+4*sum(u$Genotype[1
]=="c57BL/6J")*sum(u$Age[1]>=70)+5*sum(u$Genotype[1]=="rd10")*sum(u$Age[1]<70)+6*sum(u$Gen
otype[1]=="rd10")*sum(u$Age[1]>=70)
        i<-2

        for (y in a)
        {if (y!=x) {
        u<-read.csv(y)
        d$x<-cbind(d$x,density(u$AreaShape_Solidity)$x)
        d$y<-cbind(d$y,density(u$AreaShape_Solidity)$y)
        b[i,1]<-1*sum(u$Genotype[1]=="RPE65-/-
")*sum(u$Age[1]<400)+2*sum(u$Genotype[1]=="RPE65-/-
")*sum(u$Age[1]>=400)+3*sum(u$Genotype[1]=="c57BL/6J")*sum(u$Age[1]<70)+4*sum(u$Genotype[1
]=="c57BL/6J")*sum(u$Age[1]>=70)+5*sum(u$Genotype[1]=="rd10")*sum(u$Age[1]<70)+6*sum(u$Gen
otype[1]=="rd10")*sum(u$Age[1]>=70)
        i<-i+1
        }
        }


        datarange<-c(min,max)
        bsp <- create.bspline.basis(datarange, 800)
        f<-Data2fd(d$x,d$y,bsp)
        g1<-pca.fd(f, nharm = 2)
```

```
g1<-cbind(g1$scores,b)

length<-dim(g1)[1]


sumofindex<-0

for (x in 1:length)

{if (g1[x,3]==1) {sumofindex<-sumofindex+1}

if (g1[x,3]==2) {sumofindex<-sumofindex+1}

}


length2<-length-sumofindex

width<-3

h<-matrix(1:length2*3,length2,3)

j<-1

for (x in 1:length)

{ if (g1[x,width]==3) {h[j,]<-g1[x,]

 j<-j+1}

else if (g1[x,width]==4) {h[j,]<-g1[x,]

 j<-j+1}

else if (g1[x,width]==5) {h[j,]<-g1[x,]

j<-j+1}

else if (g1[x,width]==6) {h[j,]<-g1[x,]

j<-j+1}

}
```

```
b<-h
b<-as.data.frame(b)
b[,width]<-as.factor(b[,width])
sum3<-0
t3<-0
sum4<-0
t4<-0
sum5<-0
t5<-0
sum6<-0
t6<-0
times<-1000
length<-dim(b)[1]
for (x in 1:times)
{index<-1:length
testindex <- sample(index,1)
c<-b
train<-c[-testindex,]
test<-c[testindex,]
b.rf<-randomForest(V3 ~., data=train,xtest=test[,-width],ytest=test[,width],importance=TRUE,
                   proximity=TRUE)
if (b.rf$test$predicted == 3) {
        sum3<-sum3+sum(b.rf$test$predicted==test[,width])
```

```
        t3<-t3+1
      } else if (b.rf$test$predicted == 4) {
        sum4<-sum4+sum(b.rf$test$predicted==test[,width])
        t4<-t4+1
      } else if (b.rf$test$predicted  == 5) {
        sum5<-sum5+sum(b.rf$test$predicted==test[,width])
        t5<-t5+1
      } else if (b.rf$test$predicted == 6) {
        sum6<-sum6+sum(b.rf$test$predicted==test[,width])
        t6<-t6+1
      }

  }
  rate3<-sum3/t3
  rate4<-sum4/t4
  rate5<-sum5/t5
  rate6<-sum6/t6
  rate3
  rate4
  rate5
  rate6

  # Using random forest to classify cells based on the combination of eccentricity, solidity and
perimeter
```

```
X=read.csv("C:/Users/admin/Desktop/Dropbox/Jie Yu/RPE/R Code/Scores.csv",header=T)
b<-X
b<-as.data.frame(b)
width<-8
b[,width]<-as.factor(b[,width])
sum3<-0
t3<-0
sum4<-0
t4<-0
sum5<-0
t5<-0
sum6<-0
t6<-0
length<-dim(b)[1]
for (x in 1:length)
{
c<-b
train<-c[-x,]
test<-c[x,]
b.rf<-randomForest(V7 ~., data=train,xtest=test[,-width],ytest=test[,width],importance=TRUE,
                   proximity=TRUE)
if (b.rf$test$predicted == 3 && b.rf$test$votes[1]>0) {
        sum3<-sum3+sum(b.rf$test$predicted==test[,width])
        t3<-t3+1
```

```
      } else if (b.rf$test$predicted == 4 && b.rf$test$votes[2]>0) {
         sum4<-sum4+sum(b.rf$test$predicted==test[,width])
         t4<-t4+1
      } else if (b.rf$test$predicted  == 5  && b.rf$test$votes[3]>0) {
         sum5<-sum5+sum(b.rf$test$predicted==test[,width])
         t5<-t5+1
      } else if (b.rf$test$predicted == 6  && b.rf$test$votes[4]>0) {
         sum6<-sum6+sum(b.rf$test$predicted==test[,width])
         t6<-t6+1
      }

}
rate3<-sum3/t3
rate4<-sum4/t4
rate5<-sum5/t5
rate6<-sum6/t6
rate3
rate4
rate5
rate6
```