Classification of Wildtype and Mutant Zebrafish Brains via Computational Method

Shuli Hu [1] , Wencong Li [1] , Dejia Tang [1] , Ji Young Yun [1]

1 Statistical and Data Sciences, Northampton, MA

Abstract

Classification of biological creatures' phenotypes has long been a field that scientists study at. In this project, we utilize support vector machine to distinguish structures of Zebrafish's brains by using data generated from landmark analysis (cited Morgan's paper). We create a tool for biologists to intuitively classify three-dimensional biological shapes into two groups, usually defined as wild type and mutant, and understand which part of the shapes have the most impact on the classification result. This project derives from Professor Barresi's biological image analysis research at Smith College.

Text based on plos sample manuscript, see   1
http://journals.plos.org/ploscompbiol/s/latex   2

Acknowledgements   3

Introduction   7

This project derives from Professor Barresi's biological image analysis research at Smith   8
College and provides a tool to classify the structures within zebrafish brains via support   9
vector machine. Our goal is to distinguish the wild and mutant types of zebrafish   10
brain's structures. Morgan, a student in Barresi Lab, used landmarks analysis to divide   11
the points in the three-dimentional images into small wedges and computed the   12
landmark, which is the most representative point, within each wedge. The image of   13
signals in a Zebrafish brain is shown in Figure 1. The shape is divided into 30 slices,   14
and each slice is further divided into 8 wedges. The landmark in each wedge is   15
calculated by taking the median distance of all points in each wedge, R. We use number   16
of points in each wedge and median R to run SVM models to do classifications.   17

Landmark Analysis 18

Programming languages used 19

Python 20

R 21

Git 22

Literature Review 23

Research in developmental biology has relied on the analysis of morphological 24
phenotypes through qualitative examination of maximum intensity projections that 25
surrender the power of three dimensional data. Statistical methods to analyze visual 26
data are needed, particularly to detect subtle phenotypes. 27

Morgan et al. (2018) have utilized the open source program, Ilastik, which employs a 28
training based machine learning, to eliminate the image noise. Then they preformed 29
principal component analysis to align commissures between samples, reducing 30
misalignment artifacts, and implemented a cylindrical coordinate system which 31
preserves image dimensionality normally lost in maximum intensity projection (MIP), 32
which facilitates presentation of the data, but sacrifices much of the complexity and 33
relational data contained in the image. Then they reduced the points identified by the 34
program as belonging to the structure to a set of landmark points that describe the 35
shape and distribution of signal corresponding to the structure. Finally, using the 36
landmark system, we are able to identify and quantify structural differences and 37
changes in signal distribution between wild type and mutant commissures. 38

Shape could be described by locating a finite number of points on each specimen 39
which are called landmarks. There are three basic types of landmarks: scientific, 40
mathematic and pseudo-landmarks. A scientific landmark is a point assigned by an 41
expert that corresponds between objects in some scientifically meaningful way, for 42
example the corner of an eye. Mathematical landmarks are points located on an object 43
according to some mathematical or geometrical property of the figure. Since it does not 44
assume a preference of one location to another, it is particularly useful in automated 45
morphological recognition and analysis for under-studied structure. Pseudo-landmarks 46
are constructed points on an object, located either around the outline or in between 47
scientific or mathematic landmarks. It is often used to approximate continuous curves 48
(Dryden and Mardia, 2016). This research has chosen to calculate an automatic set of 49
landmarks distributed across the structure in order to avoid introducing bias due to 50
expectations about where biological differences should emerge. 51

The SVM algorithm is a classification algorithm that provides state-of-the-art 52
performance in a wide variety of application domains, image classification. During the 53
past few years, SVM has been applied very broadly within the field of computational 54
biology especially in pattern recognition problems, including protein remote homology 55
detection, microarray gene expressions analysis, prediction of protein-protein 56
interactions, etc. 57

In 1999, Jaakkola et al. ushered in stage 4 of the development of homology detection 58
algorithms with a paper that garnered the "Best paper" award at the annual Intelligent 59
Systems for Molecular Biology conference. Their primary insight was that additional 60
accuracy can be obtained by modeling the difference between positive and negative 61
examples. Because the homology task required discriminating between related and 62
unrelated sequences, explicitly modeling the difference between these two sets of 63
sequences yields an extremely powerful method. The algorithm described in that paper 64
is called SVM-Fisher. 65

Data

We have 43 wildtypes samples and 35 mutant samples for training and testing. There are 152 landmarks for each sample, with each of them containing the following variables: number of points in that wedge median R (micro-meter): the median of the distances to the center of the slice of all the points in that wedge. alpha (micro-meter): distance from the center of the landmark to the midline theta (radian): the degree that shows the location of the wedge of a slice

We used the number of points and the median R to do classification via support vector machine. For missing 'median R' values due to absence of points in particular landmarks, we filled them with the median value of all the points in that landmark.

Variables

Number of points Median r Alpha Theta

Tidy Data

The original landmarks data is a wide table containing the sample index and all the columns holding information regarding the minimum and maximum values of Alpha and Theta, number of points, median r value, and the type of sample for a particular sample in each landmark. However, because all of such variables were joined by underscores in the variable names, such as `-14.29_-4.76_-0.79_0.0_50_pts` or `-14.29_-4.76_-0.79_0.0_50_r` and the value in each cell refers to the median r value or number of points, it was very difficult to see what each column actually represented. The ideal format of the data set was to have the sample index, minimum and maximum Alpha, minimum and maximum Theta, number of points, median r, and type of sample each be its own column. Hence, three key functions were used from the tidyr package: gather, separate, and spread. The gather function separated the dataset into key and value pairs for each index. The key was the column name containing all essential information connected by underscores and the value included the number of points or median r value. Then, the separate function separated the result from the gather function divided the column connected by underscore into 5 different columns, named as `min_alpha`, `max_alpha`, `min_theta`, `max_theta`, and `ptsOrR`. This was added to the result of the gather function that contained the index and value of each cell, either median R or number of points. Afterwards, the spread function widened the already wide table by expanding the `ptsOrR` column by creating two columns, each column representing median R and the number of points.

Missing Value

Samples with missing values are eliminated by Supporting vector machine. For wedges that do not have any point in it, `median r` cannot be calculated, which means that these sample will be eliminated when running SVM. Wedges without points have biologically meanings, and we should not ignore these wedges in our model. In order to keep the wedges in our model, we need to artificially pick a `median r` value to replace the missing ones. Supporting vector machine is sensitive to outliers, so we cannot pick an `r` value that could become outliers. We decided to calculate the mean of `median r` for the nth landmark of all 78 samples, and then we replace the missing `median r` values with the mean.

Supporting Vector Machine

SVM's have been proven to be a powerful algorithm for supervised clustering. During
the past few years, SVM has been applied very broadly within the field of
computational biology especially in pattern recognition problems. The goal of SVM is
to find a hyperplane $Y_i(b_0 + b_1 X_{i1} + b_2 X_{i2}) = 0$ that separates the nearest data as clean
as possible. The parameters bi are found by solving the optimization problem below.

- C: tuning parameter, toleration of violation.
- M: margin, distance of the closest points to the hyperplane.
- e_i : slack variable, correct/incorrect classification of a point.

What is SVM?

SVM's have been proven to be a powerful algorithm for supervised clustering. A good
separation is achieved by the hyperplane that has the largest distance to the nearest
training-data point of any class.

SVM's have been proven to be a powerful algorithm for supervised clustering.
During the past few years, SVM has been applied very broadly within the field of
computational biology especially in pattern recognition problems. The goal of SVM is
to find a hyperplane $Y_i(b_0 + b_1 X_{i1} + b_2 X_{i2}) = 0$ that separates the nearest data as clean
as possible. The parameters bi are found by solving the optimization problem below.

- C: tuning parameter, toleration of violation.
- M: margin, distance of the closest points to the hyperplane.
- e_i : slack variable, correct/incorrect classification of a point.

The optimization problem

Here is the function of the hyperplane:

$$f(x) = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2$$

if $f(x) = 0$, the observation is on the hyperplane.

$$y * (\beta_0 + \beta_1 * x_1 + \beta_2 * x_2)$$

is the perpendicular distance from the ith observation to the hyperplane. If it's >0, the
observation falls at the right side of the hyperplane and vice versa.

Workflow and User Interface

Step One: Data Processing and Modelling

This step is implemented using Python and packages including pandas, nump and
sklearn are required. Users would need to run and interact with the Python script
`svm.py` to build the model and pre-process the data if needed.

The script `svm.py` contains two components: a general-purpose
`svm_classification()` function that builds a SVM model to classify points for a
perticular landmark and a `main()` function that runs the `svm_classification()`
function for each landmark.

Input File 144

Input File 145

Input file must contain landmark data. Variables that are needed for classification are 146
required to be included in the input file. In our analysis, we used number of points in 147
each sub-section corresponding to each landmark of the 3D shape and the `median R` of 148
points in each wedge. ### Sample input file ### Output File 149

Step Two: Result Analyzation and Visualization 150

Testing 151

Cross-Validation 152

For our project, we have access to 43 wild-type samples and 35 mutant-type samples. 153
Due to this limited sample size, we dicided to use a leave-one-out cross validation 154
method to test our model. 155

For each testing sample, we built 152 SVMs for each landmark. For each SVM, we 156
used 10-fold cross validation to select a tuning parameter C value among 0.1, 1 and 10. 157
After we get precision scores and predictions of each landmark, we will present the 158
distribution of the landmarks' precision scores (Figure 3). The user would then be 159
allowed to set a threshold for certain precision scores to select out a subset of landmarks 160
that are considered significant. A majority vote would then be performed among the 161
selected landmarks to get an overall prediction for the sample. 162

Result 163

Visulization 164

() 165

Conclusion 166

The SVM model application on landmark data gives insightful analysis of: which 167
landmark (=which part of the brain) carries more information of the type. whether a 168
new sample is of mutant or wild type 169

Discussion 170

() 171

In the previous method random forest, the number of predictors p exceeds the 172
number of samples. Morgan applied PCA do reduce the dimension of the predictors. 173
The problem with dimension reduction is that it gives a linear combination of the 174
dimensions that are projected on those are kept. While the largest projections still 175
make sense, the minor projections are very random and thus difficult to interpret. 176
The SVM model also has its limitation in that it only considers one single landmark 177
at a time without considering the relationship across the landmarks of the whole sample. 178
Strength Limitation 179

Future Study 180

We will continue to work on: making the program more user friendly running more tests 181
to prove the accuracy of our model 182

Improvement: 183

Itenerating machine learning 184
    Instead of cross-validation, better results (what kind of better results) could be 185
achieved by using itenerating machine learning method. In iterative machine learning 186
we repeat the process of training and testing several times. At the first round the user 187
gives examples of objects belonging to some classes and the machine learning algorithm 188
is trained with this data. In the second round, the algorithm shows examples of objects 189
it thinks that belong to these classes. Now, the user merely adds objects to the 190
improved training set which the machine learning algorithm has put into a wrong class. 191
That is, the user only corrects the "misunderstandings" of the algorithm. In this way 192
we can concentrate on difficult examples of objects that are hard to classify or are for 193
some reason easily missed by humans. Such objects may lie close to the decision 194
boundaries or in the periphery in the multidimensional feature space. This iterative 195
process is continued until the machine learning algorithm does not make any mistakes 196
or the classification results do not improve anymore. 197
    Other models 198

Appendix Code 199

Support Vector Machine 200

Shiny App 201

References 202