# 2D-PCA for High Dimensional Reduction on Metagenome Classification

**Conference Paper** · October 2016

CITATIONS
0

READS
92

**4 authors**, including:

Toto Haryanto
University of Indonesia
**17** PUBLICATIONS **4** CITATIONS
SEE PROFILE

Wisnu Ananta Kusuma
Bogor Agricultural University
**41** PUBLICATIONS **29** CITATIONS
SEE PROFILE

Aziz Kustiyo
Bogor Agricultural University
**5** PUBLICATIONS **5** CITATIONS
SEE PROFILE

# 2D-PCA for High Dimensional Reduction on Metagenome Classification

Lani Kurniasari[1], Toto Haryanto[2*], Wisnu Ananta Kusuma[3] and Aziz Kustiyo[4]
[1]*Department of Computer Science, Bogor Agricultural University,16680,Indonesia, laninalks@gmail.com*
[2*]*Department of Computer Science, Bogor Agricultural University,16680, Indonesia, totoharyanto@.ipb.ac.id*
[3] *Department of Computer Science, Bogor Agricultural University, 16680, Indonesia, ananta@ipb.ac.id*
[4] *Department of Computer Science, Bogor Agricultural University,16680, Indonesia, azizku@ipb.ac.id*
(* corresponding author)

**Abstract**

Metagenome is a collection of microbes which the sample obtained directly from their natural environment. In metagenome analysis, binding is one process for classifying or cluster organism. Composition binning is one technique base on feature extraction before processed in the machine learning. K-mers as feature extraction widely known in bioinformatics research area especially metagenome analysis . Increasing k value in K-mers can produce high-dimensional feature and very time-consuming. Two-dimensional principal component analysis (2D-PCA) was conducted for dimension reduction in this research. The total of 214 microorganisms was obtained from NCBI at ftp://ftp.ncbi.nlm.nih.gov/genomes/Bacteria/all.fna.tar.gz and simulated by Metasim as DNA sequencer simulator software. K-Nearest Neighborhood (KNN) conducted for classification technique. The highest lapse time for 7-mers up to 304.70 seconds or 5.07 minute faster than without 2D-PCA for 3-NN classification process as the best result. The best result revealed from 3-NN with 99,51%, 93,75% and 99,89 for accuracy, sensitivity, and specificity respectively.

***Keywords*:** 2D-PCA, k-mers, KNN, metagenome analysis, metasim

## I. INTRODUCTION

Metagenome analysis still continuously to be conducted in bioinformatics research area. Metagenome is a collection of microbes which the sample obtained directly from their natural environment without prior culture process. Metagenome bining is necessary for many research fields in bioinformatics [1]. Metagenome fragment binning is necessary to differ among DNA organism in the environment. For instance, the project to discover genome biodiversity on Sargasso sea near Bermuda. The sample obtained from the sea of non-redundant sequences [2].

Basically, binning process can be conducted by two approach, homology based and composition based. Homology approach was conducted by sequence alignment while in composition approach require feature extraction and then supervise or unsupervised learning. Tools such BLAST and PSI-BLAST are example for homology based software [3].

Feature selection technique has essential role in bioinformatics research area. One of the mayor issue in bioinformatics application is large set dimensionality. Feature selection technique will be applied in bioinformatics application is casuistic depend on the motivation of researcher [4].

K-Mers is one the feature extraction based on the frequency of occurrence of DNA combination. McHardy [1] applied this technique to analyze metagenome. From research shown that 5-mers and support vector machine (SVM) revealed accuracy between 70 up to 90 percent depend on the length of the fragment.

The idea of k-mers is simple where the number of k will influence number of feature. In DNA sequence with four nucleotide adenine (A), cytosine (C), guanine (G) and thymine (T) can be formed $4^3$ or 64 feature combination when the value of $k$ is 3. Assignment more of the $k$ value are preferred because unique composition can be conducted. However, there thread-off is high dimensionality occurred extremely [5].

On the other hand, various technique for handling data with high dimensionality. Principal Component Analysis (PCA) is one of the well known as method to handle this problem. Commonly, PCA applied on image processing [6][7][8]. In bioinformatics application such protein classification, PCA also has been applied for reducing feature [9]. Comparing Independent Component Analysis (ICA) and PCA was proposed for Single Nucleotide Polymorphism (SNP) selection [10] and Gene Expression [11]

2D-PCA or two dimensional principal component introduced to answer the limitation of PCA. The representation image does not to be transformed to vector with 2D-PCA [12]. More than that, 2D-PCA has advantages from time efficiency than PCA. This method inspired us to apply 2D-PCA for

bioinformatics application on metagenome classification.

## II. RESEARCH METHODOLOGY

Systematically, this paper divided into four sections; data preprocessing, feature extraction, 2D-PCA dimension reduction and K-NN classification.

### *A. Data Preprocessing*

Whole genome data were accessed from National Center for Biotechnology Information (NCBI). This uniform resource locator (URL) at ftp://ftp.ncbi.nlm.nih.gov/genomes/Bacteria/all.fna.tar gz is source of the data. We have collected twenty five genus for 214 total organisms. List of genus data showed on Table 1.

TABLE 1: LIST OF THE GENUS DATA SET

| | |
|---|---|
| *Bacillus* | Pseudomonas |
| *Brucella* | Rickettsia |
| *Burkholderia* | Shewanella |
| *Campylobacter* | Shigella |
| *Chlamydophila* | Staphylococcus |
| *Chlorobium* | Streptococcus |
| *Clostridium* | Synechococcus |
| *Corynebacterium* | Thermotoga |
| *Francisella* | Vibrio |
| *Haemophilus* | Xanthomonas |
| *Helicobacter* | Yersinia |
| *Lactobacillus* | Mycoplasma |
| *Mycobacterium* | Pseudomonas |
| *Mycoplasma* | |

Metasim is familiar tools for generating reads or fragment based on whole genome data [13]. Output from Metasim is fastA format data type with various number reads depend on length of each read.

### *B. Feature Extraction*

K-mers feature extraction was performed in DNA sequence for calculating the frequency of occurrence of nucleotide. The output from Metasim then processed to obtain the feature of the sequences. The combination of appearance adenine (A), cytosine (C), guanine (G), and thymine (T) captured from the fragment obtained by metasim. Figure 1 is illustration of feature extraction process for 3-mers.

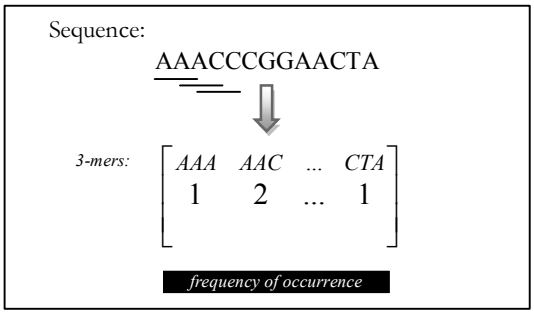



Figure 1. Illustration 3-mers feature extraction

Feature extraction process will reveal matrix with high dimensional feature depend on number of *k*. This study tried to determine various *k* start from 5, 6 and 7. This is reasonable for applying reduction dimension.

### *C. 2D-PCA reduction dimension*

2D-PCA as an extension of 1D PCA conducted for reduction dimension. Different from standard PCA that reduce the vector, 2D-PCA will reduce matrix. 2D-PCA widely used in the image process. Each fragment output from feature extraction produce vector of the frequency of occurrence adenine (A), cytosine (C), guanine (G) and thymine (T). A number of the feature are various depend on n value in n-mers used. For 5, 6 and 7-mers will produce 1025, 4096 and 16384 features respectively.

These numbers of features are too large to be processed in classification algorithm or machine learning. For this reason, 2D-PCA proposed to handle high dimensional data [12].

Let $X$ as vector transformation from random matrix $A$ $m \times n$. $Y$ is result of transformation with equation

$$Y = AX \tag{1}$$

In this research, A is the matrix of the feature from the n-mers frequency for each read. For example, if we have 7-mers, there are 16384 feature vector.

Thus, we have to reconstruct as matrix 128 x 128.

The value of X will be determined as good projection. To get good value of X as projection vector, covariance matrix of A can be traced with this equation

$$J(X) = tr(S_x) \tag{2}$$

When $j(X)$ are total of scatter of projected sample and $S_x$ is covariance matrix and $tr(S_x)$ denoted trace of $S_x$. The covariance matrix $S_x$ can be denoted in by

$$\begin{aligned} S_x &= E(Y - EY)(Y - EY)^T \\ &= E[AX - E(AX)][AX - E(AX)]^T \\ &= E[(A - EA)X][A - EA)X]^T \end{aligned} \tag{3}$$

Therefore,

$$tr(S_x) = X^T[E(A - EA)^T(A - EA)]X \tag{4}$$

Define $G_t$ as $n \times n$ covariance scatter matrix.

$$G_t = [E(A-EA)^T (A-EA)] \quad (5)$$

$G_t$ will be evaluated using matrix from training data. If we have M training reconstructed matrix and *jth* denoted as matrix $A_j$ = *(j=1,2,3, ... M)* and average of sample matrix denoted $\overline{A}$, then $G_t$ can be evaluated as

$$G_t = \frac{1}{M}\sum_{j=1}^{M}(A_j - \overline{A})^T (A_j - \overline{A}) \quad (6)$$

In the other word, equation (2) can be

$$j(X) = X^T G_t X \quad (7)$$

$G_t$ on the equation (7) called generalized total scatter criterion. After covariance of matrix A obtained, eigen value and eigen vector will be calculated. To obtained optimal value of X from $X_1, X_2, \ldots X_d$ interpret to orthonormal constrain and maximizing criterion *J(X)*.

$$\begin{cases} \{X_1,\ldots,X_d\} = \arg\max J(X) \\ X_i^T X_j = 0, i \neq j, i, j = 1,\ldots,d \end{cases} \quad (8)$$

Then, for given A as reconstructed matrix

$$Y_k = AX_k \text{ , } k=1,2,\ldots,d \quad (9)$$

$Y_1, Y_2, \ldots Y_d$ are projected feature vector called principal component (vector) of the reconstruction matrix A.

### D. KNN classification

Classification will be applied after 2D-PCA was finished. Transformation matrix output from 2D-PCA will be calculated. In 2D-PCA, column vector distance between testing set and training set were determined. For $B_i = \left[Y_1^{(i)}, Y_2^{(i)}, \ldots Y_d^{(i)}\right]$ dan $B_j = \left[Y_1^{(j)}, Y_2^{(j)}, \ldots Y_d^{(j)}\right]$. The distance between $B_i$ and $B_j$ when B is matrix training and testing data defined as

$$d(B_i, B_j) = \sum_{k=1}^{d} \left\| Y_k^{(i)} - Y_k^{(j)} \right\|_2$$

when

$$\left\| Y_k^{(i)} - Y_k^{(j)} \right\|_2 = \sqrt{\left(Y_k^{(1i)} - Y_k^{(1j)}\right)^2 + \ldots \left(Y_k^{(ni)} - Y_k^{(nj)}\right)^2}$$

The notation $\left\| Y_k^{(i)} - Y_k^{(j)} \right\|_2$ is distance between $Y_k^{(1i)}$ as training and $Y_k^{(1j)}$ as testing data.

## III. RESEARCH RESULT AND DISCUSSION

### A. Data Preprocessing

Metagenome Data were obtained from NCBI then simulated by Metasim software. Number of coverage for simulating this data is 10 and use 500 base pair (bp) for each read with FastA format. Number of read generated influenced by number of fragment for each organism. Figure 1 is an example of output from Metasim software from *Bacillus* genus.

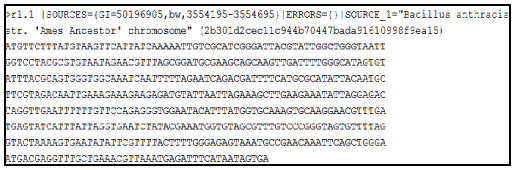



Figure 1: example of read from Metasim as DNA Simulator software

### B. Feature Extraction

n-mers feature extraction conducted for obtaining frequency of Adenine (A), Guanine (G), Cytosine and Thymine (T) combination in the nucleotide sequences. Number of feature in this experiment are 1024, 4096 and 16384 with number of n 5,6 and 7 respectively. For instance, if use n=5, there are AAAAA, AAAAC, AAAAG, AAAAT and etc. until TTTTT. The total of 214 data included training and testing extracted. Python programming language conducted for this process and Result of feature extraction is frequency of occurrence from A,C,G and T combinations. (Figure 2)

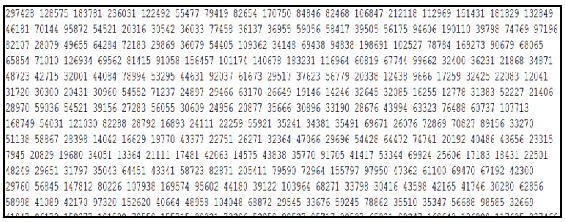



Figure 2: Capture of feature extraction with 5-mers

### C. 2D-PCA Dimension Reduction

The first step in 2D-PCA is constructed 2D matrix from n-mers feature extraction. Three matrices were constructed 32x32 for 5-mers, 64x64 for 6-mers and 128x128 for 7-mers. In principle component analysis, usually use covariance, however, to understand the degree of relation between features, the correlation was proposed after computing covariance. Equation (10) refers to the covariance formula.

$$r_{(x,y)} = \frac{Cov(x,y)}{\sigma_x . \sigma_y} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (10)$$

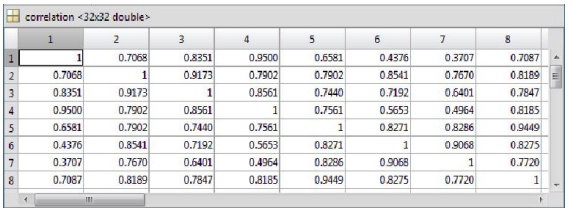
where :

$Cov(x, y)$ : covariance of the variable x and y

$\sigma_x$ : deviation standard of variable x

$\sigma_y$ : deviation standard of variable y

Figure 3 showed the example for correlation matrix constructed for 5-mers.




| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0.7068 | 0.8351 | 0.9500 | 0.6581 | 0.4376 | 0.3707 | 0.7087 |
| 2 | 0.7068 | 1 | 0.9173 | 0.7902 | 0.7902 | 0.8541 | 0.7670 | 0.8189 |
| 3 | 0.8351 | 0.9173 | 1 | 0.8561 | 0.7440 | 0.7192 | 0.6401 | 0.7847 |
| 4 | 0.9500 | 0.7902 | 0.8561 | 1 | 0.7561 | 0.5653 | 0.4964 | 0.8185 |
| 5 | 0.6581 | 0.7902 | 0.7440 | 0.7561 | 1 | 0.8271 | 0.8286 | 0.9449 |
| 6 | 0.4376 | 0.8541 | 0.7192 | 0.5653 | 0.8271 | 1 | 0.9068 | 0.8275 |
| 7 | 0.3707 | 0.7670 | 0.6401 | 0.4964 | 0.8286 | 0.9068 | 1 | 0.7720 |
| 8 | 0.7087 | 0.8189 | 0.7847 | 0.8185 | 0.9449 | 0.8275 | 0.7720 | 1 |

Figure 3: correlation matrix for 5-mers

After matrix constructed, vector Eigen, eigenvalue, proportion and cumulative proportion were calculated. Three types of principal starts from 88%, 95%, and 98%. The sequential step for implementing 2D-PCA in metagenome classification start from vector feature, generate 2D matrix construction, and PCA implementation. Table 2 show the matrix after 2D-PCA implemented for dimension reduction.

TABLE 2: MATRIX CONSTRUCTION WITH 2D-PCA

| vector feature | constructed matrix | Matrix 2D-PCA | | |
|---|---|---|---|---|
| | | 88% | 95% | 98% |
| 1024x1 | 32 x 32 | 32x4 | 32x9 | 32x16 |
| 4096x1 | 64 x 64 | 64x6 | 64x14 | 64x27 |
| 16384x1 | 128x128 | 128x10 | 128x26 | 128x53 |

*D. K-Nearest Neighbor Classification*

Matrix from 2D-PCA will become input for K-NN classification. The Euclidian distance between vectors of the column in training and testing data set were computed. Two value of k (3 and 5) nearest neighbor experimentally conducted in this research.

Before 2D-PCA applied, the performance of KNN has been measured. Table 4 showed accuracy, sensitivity and specificity 3-NN and 5-NN. 2D-PCA applied in three type of Principal Component (PC) and three type of k-mers value. These combinations run on 3-NN and 5-NN. Therefore, there is total of 18 combinations of performances.

To evaluate the performance classification, accuracy, sensitivity, and specificity were calculated. Table 5 illustrate the result of classification with three nearest neighbors. Overall, the performance of classifier increase in line with the percentage of principal component (PC). The best accuracy up to 99.51 percent and sensitivity 93.75 percent.

TABLE 4: PERFORMANCE OF CLASSIFICATION WITHOUT 2D-PCA

| | 3-NN | | |
|---|---|---|---|
| | accuracy | sensitivity | specificity |
| 5-mers | 98.83 | 84.38 | 99.79 |
| 6-mers | 99.12 | 87.5 | 99.89 |
| 7-mers | 99.12 | 87.5 | 99.89 |
| | 5-NN | | |
| | accuracy | sensitivity | specificity |
| 5-mers | 98.83 | 84.38 | 99.79 |
| 6-mers | 98.93 | 85.94 | 99.79 |
| 7-mers | 98.73 | 90.62 | 99.27 |

TABLE 5 PERFORMANCE OF 3-NN CLASSIFICATION

| vector feature | PC (%) | Parameter (%) | | |
|---|---|---|---|---|
| | | Accuracy | sensitivity | specificity |
| 5-mers | | 98.63 | 82.81 | 99.69 |
| 6-mers | 88 | 98.93 | 84.37 | 99.89 |
| 7-mers | | 99.02 | 85.94 | 99.89 |
| 5-mers | | 98.83 | 84.38 | 99.79 |
| 6-mers | 95 | 99.12 | 87.50 | 99.89 |
| 7-mers | | 99.32 | 90.62 | 99.89 |
| 5-mers | | 99.02 | 85.94 | 99.89 |
| 6-mers | 98 | 99.32 | 92.19 | 99.89 |
| 7-mers | | **99.51** | **93.75** | 99.89 |

Not only performance but also execution time have been covered. In dimension reduction, time execution is one of aspect could be considered. Table 6 showed running time for the classification process.

K-NN is lazy learning algorithm because all of the training set will be computed when classification process occurred. That is why important to evaluate running time. According to the table 6, the highest lapse time is 521.08 second or about 8,63 minute faster for 7-mers with 88 principal.

TABLE 6: RUNNING TIME COMPARISON OF 3-NN CLASSIFICATION

| Vector feature | PC (%) | running time (second) |
|---|---|---|
| without reduction | | 44.02 |
| 5-mers | 88 | 11.23 |
| | 95 | 18.03 |
| | 98 | 27.93 |
| without reduction | | 137.07 |
| 6-mers | 88 | 23.83 |
| | 95 | 45.76 |
| | 98 | 77.79 |
| without reduction | | 581.26 |
| 7-mers | 88 | 60.18 |
| | 95 | 132.94 |
| | 98 | 276.56 |

Beside three nearest neighbor, the research tried to evaluate five nearest neighbor in K-NN classification and Table 7 revealed the performance of classification.

According to the table 5 and 7, overall performance of 3-NN is better than 5-NN. Sensitivity in this research is a value describing the ability of classifier to classify organism appropriately. There are 64 organism from 16 genus were tested.

TABLE 7: PERFORMANCE OF 5-NN CLASSIFICATION

| vector feature | PC (%) | Parameter (%) | | |
|---|---|---|---|---|
| | | accuracy | sensitivity | specificity |
| 5-mers | 88 | 98.05 | 76.56 | 99.48 |
| 6-mers | | 98.63 | 82.81 | 99.69 |
| 7-mers | | 98.63 | 82.81 | 99.69 |
| 5-mers | 95 | 98.63 | 84.38 | 99.58 |
| 6-mers | | 98.73 | 84.37 | 99.69 |
| 7-mers | | 99.22 | 89.06 | 99.89 |
| 5-mers | 98 | 98.93 | 85.94 | 99.79 |
| 6-mers | | 99.32 | 90.62 | 99.89 |
| 7-mers | | 98.73 | 90.62 | 99.27 |

High sensitivity and accuracy are important in metagenome classification. Nevertheless, K-NN as lazy learning technique very depends on the number of the feature while determining the class target. Therefore, running time with 2D-PCA is also considered to evaluate part of training data as the model classifier. In this research running time with 3-NN and 5-NN are compared.

Composition based feature extraction is one of the key point in metagenome fragment binning. Usually, vector feature was conducted directly to machine learning or classifier. In this case, we have 16384 features and very time consuming if processed to K-NN classifier. 2D-PCA applied to solve the high dimension problem. Very long vector transformed two-dimensional matrix and then PCA was implemented. According to Table 6 and 8, showed that the implementation of 2D-PCA reduces significantly running time to the both scenario of 3-NN and 5-NN classification.

TABLE 8: RUNNING TIME COMPARISON OF 5-NN CLASSIFICATION

| vector feature | PC (%) | running time (second) |
|---|---|---|
| without reduction | | 48.07 |
| 5-mers | 88 | 12.28 |
| | 95 | 19.52 |
| | 98 | 31.36 |
| without reduction | | 151.39 |
| 6-mers | 88 | 24.32 |
| | 95 | 47.56 |
| | 98 | 91.80 |
| without reduction | | 601.57 |
| 7-mers | 88 | 60.74 |
| | 95 | 140.92 |
| | 98 | 281.52 |

Lapse time for the best classification result showed in Table 9. Lapse time means that classification process for 3-NN is 5,07 minute faster than without 2D-PCA. Meanwhile, using 5-NN take 7.67 minute faster than without 2D-PCA in classification process.

TABLE 9: LAPSE TIME COMPARISON FOR THE BEST RESULT

| | lapse time (sec) | accuracy | sensitivity | specificity |
|---|---|---|---|---|
| 3-NN | 304.70 | 99.51 | 93.75 | 99.89 |
| 5-NN | 460.65 | 99.32 | 90.62 | 99.89 |

## IV. CONCLUSION

2D-PCA has been proofed effectively reduce the dimension of the data and give the time efficiency for classification using k nearest neighbor (KNN). The influence of 2D-PCA to the accuracy performance are not significantly different than without 2D-PCA.


## ACKNOWLEDGEMENT

Thank you to Department of Computer Science, Bogor Agricultural University for Funding of this paper. To all member of Bioinformatics Research Group, thank you for insightful discussion.


## REFERENCES


[1] McHardy AC, Martín HG, Tsirigos A, Hugenholtz P, Rigoutsos I. Accurate phylogonetic classification of variabel-length DNA fragments. Nature Methods. 2007. 4(1):63–72. doi: 10.1038/nmeth976.

[2] Qunit R, Wanamaker S, Close TJ, Lonardi S. CLARK: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. BMC Genomic. 2015. 16(236):1-13. DOI 10.1186/s12864-015-1419-2.

[3] Altschul SF1, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Research, 1997.25(17): 3389–3402.

[4] Saeys Y, Inza I, Larranaga P. A review of feature selection techniques in bioinformatics [review]. Bioinformatics. *Gene Expression*. 2007 23(19): 1. doi:10.1093/ bioinformatics/btm344

[5] Wang R, Xu Y, Liu B. Recombination spot identification Based on gapped k-mers. Scientific Report. 6:23034. DOI: 10.1038/srep23934

[6] Rodríguez A, Peris JE, Redondo A, Shimada T, Peña L. Principal component analysis (PCA) of volatile terpene compounds dataset emitted by genetically modified sweet orange fruits and juices in which a D-limonene synthase was either up- or down-regulated vs. empty vector controls. 2016. Data in Brief 9:355–361. http://dx.doi.org/10.1016/j.dib.2016.09.003

[7] Morales S, Naranjo V, Angulo J, and Alcañiz M. Automatic Detection of Optic Disc Basedon PCA and Mathematical Morphology. IEEE Transaction on Medical Imaging. 2004. 32(4): 786-796.

[8] Landi A, Piaggi P, Pioggia G. Backpropagation-Based Non Linear PCA for Biomedical Applications: Proseeding of the 9th International Conference on Intelligent Systems Design and Application. 2010.
[9] Chaturvedi B, Patil N. A Novel Semi-supervised Approach for Protein Sequence Classification. IEEE Conference 2015.
[10] Nahlawi LI, Mousavi P. Single Nucleotide Polymorphism Selection using Independent Component Analysis: Proceeding of 32nd Annual International Conference of the IEEE EMBS Buenos Aires, Argentina, August 31 - September 4, 2010
[11] Costa JFP, Alonso H, and Roque L. A Weighted Principal Component Analysis and Its Application to Gene Expression Data. IEEE Transaction on Computational Biology and Bioinformatics. 2011. 8(1):246-252
[12] Yang J, Zhang D, Frangi AF, and Yang JY. Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition. IEEE Transaction on Pattern Analysis and Machine Intelligence.2004. 26(1): 131-137.
[13] Richter DC, Ott F, Auch AF, Schmid R, Huson DH MetaSim—A Sequencing Simulator for Genomics and Metagenomics. PLoS ONE.2008 3(10): e3373. doi:10.1371/journal.pone.0003373.