Published in final edited form as:

Methods Mol Biol. 2009 ; 520: 259–272. doi:10.1007/978-1-60327-811-9_19.

Developing Classifiers for the Detection of Cancer Using Multianalytes

Adi Laurentiu Tarca, Sorin Draghici, and Roberto Romero

SUMMARY

The development of a successful classifier from multiple predictors (analytes) is a multistage process complicated typically by the paucity of the data samples when compared to the number of available predictors. Choosing an adequate validation strategy is key for drawing sound conclusions about the usefulness of the classifier. Other important decisions have to be made regarding the type of prediction model to be used and training algorithm, as well as the way in which the markers are selected. This chapter described the principles of the classifier development and underlines the most common pitfalls. A simulated dataset is used to illustrate the main concepts involved in supervised classification.

Keywords

Multiple predictors; Supervised classification; Feature/marker selection

1. INTRODUCTION

Current high-throughput technologies empowered life scientists with the ability of measuring the level of thousands of analytes (genes, proteins, antibodies, etc.) in a given biological sample. One of the major goals in these types of studies is to accurately classify a group of samples based on the measured expression profiles (1–4). This type of application is also known as *class prediction*, *classification* or *supervised learning*.

The classification process starts with a collection of samples with known outcome for which the expression level of several analytes is available. Let $x_{i,j}$ denote the expression level of the j th analyte (feature) in the i th sample and y_i the outcome. Even though the following discussion will imply that the outcome is a two-level factor, i.e., $y_i = 1$ for disease and $y_i = 0$ for healthy samples, most of the principles to be described further can be used when the outcome is a continuous variable (e.g., a survival time) or a multiclass problem (e.g., disease 1, disease 2, ..., disease k). After suitable data preprocessing, the next step is to decide which validation strategy will be used to assess the classifier's performance in an unbiased way. A straightforward strategy for instance is to randomly split the samples into two disjunct sets called *training* and *test* sets. The training data will be used to learn the association between expression levels and the outcome, while the test data will be used to assess the classifier's generalization ability. The next step is to choose a type of prediction model, e.g., a logistic regression model or a neural network model. Then, a reduced set of most predictive analytes must be selected. Using the selected predictors, the model's internal parameters are then tuned (optimized) so that the predicted outcome matches as much as possible the desired outcome for the samples in the training set. This process is called learning or training. The classifier development is illustrated in Fig. 1.

Later in this chapter, we will use the term *model* to denote the mathematical equation that associates the analyte expression profile of the i th sample, $\mathbf{x}_i$, to a predicted outcome $\hat{y}_i$ and

the term of *classification* to the entire procedure used to select markers, train the model, and compute the predicted outcome.

The rest of this chapter is organized as follows: First we briefly discuss issues related to data preprocessing and then we introduce classifier performance measures (e.g., accuracy) as well as practical methods to estimate them. Then, a brief introduction to various types of classification models will be given, followed by an introduction to methods for marker selection.

2. DATA PREPROCESSING

Regardless of the way they are measured, the analyte concentration levels cover usually a wide range of values, showing typically a skewed distribution. A simple method to improve the normality of the distribution in such cases is to use a logarithmic transformation. Most of the classifiers would benefit from such data transformation resulting by improving the prediction accuracy. In addition, when using a high-throughput platform to measure analyte expression levels, the resulting data may contain systematic biases. Hence the measured levels of the analytes have to be normalized in some way in order to make them comparable between the samples. For example, the probe intensities for a particular microarray can be much higher on average than those of the other arrays. More details about the most common normalization procedures related to microarrays, which are probably the most common current multianalyte platform, can be found in the literature (5,6). Caution should be taken not to choose any preprocessing method that makes use of the sample outcome since the group information may be either unavailable or prohibited to use.

3. CLASSIFIER FIGURES OF MERIT

The result of a classification process can be summarized in a contingency table, also called *confusion matrix*. The confusion matrix contrasts the predicted class labels of the samples $\hat{y}_i$ with the true class labels y_i as shown in Table 1. An example confusion matrix computed for a set of 60 healthy and 40 diseased samples is shown in Table 2.

The *classification rate* or *accuracy* of the prediction is defined as the average number of successfully classified samples, i.e., the sum of diagonal elements of the confusion matrix, divided by the total number of samples. In the example mentioned earlier, *Amniotic cavity* = $(30 + 40)/(40 + 60) = 70\%$. The *sensitivity* of the classifier is defined as the fraction of positive (disease) samples correctly classified: *Sen* = $(30)/(30 + 10) = 75\%$, while the *specificity* is the fraction of negative (healthy) samples correctly classified: *Spec* = $(40)/(40 + 20) = 66.7\%$. The sensitive characterizes the ability of the classifier to classify the diseased samples as such, while the specificity characterizes the ability to classify the healthy samples as such. Both are needed in a real-world application. For instance a classifier that predicts disease all the time will have a sensitivity of 100%, but zero specificity. Such a classifier would of course be useless in any application. Similarly, a high specificity can be obtained by biasing the classifiers prediction toward a healthy outcome which will in turn reduce the sensitivity. High sensitivity and specificity together imply high accuracy. The reverse is not true, however. For instance, a classifier that predicts disease all the time can yield an accuracy of 100% if there are no healthy samples in the set used to evaluate its performance. Other useful performance indices are the *positive predicted value* (PPV) and *negative predicted values* (NPV). The PPV represents the fraction of true positives among all positive predicted samples, while the NPV represents the fraction of true negatives among all negative predicted samples.

Most of the classification models do not produce directly the class label $\hat{y}_i$ from the vector of input values $\mathbf{x}_i$, but rather a continuous output Ψ_i which ranges in a certain interval, e.g.,

from 0 to 1. When Ψ_i is less than a given threshold, e.g., $T = 0.5$, then the sample I is classified as healthy ($\hat{y}_i = 0$), otherwise as disease ($\hat{y}_i = 1$). For a given dataset, using a threshold T lower than 0.5 will cause the prediction to be biased toward the disease class, and therefore improve the sensitivity to the detriment of specificity. The sensitivity and the specificity obtained when applying a trained model on a set of samples depend therefore on the choice of the threshold T . A receiver-operating characteristic (ROC) curve can be obtained by plotting sensitivity against 1-specificity values obtained by varying T from 0 to 1. The area under the resulting ROC curve (AUC) gives a complete picture of the performance of a trained classifier on a given dataset. The AUC ranges from 0 to 1, with 1 being the most fortunate situation when maximum sensitivity is obtained for all values of specificity.

4. VALIDATION STRATEGIES

In the following discussion, we will use the term *performance index* to refer to any of the aforementioned goodness measure: accuracy, sensitivity and specificity, AUC, or other. As shown in Fig. 1 the classifier performance should be evaluated by applying the trained model on a set of samples that were not used in the training procedure. Using the training data themselves to compute the performance indices would result in an optimistically biased estimate (7). This is because the model may *overfit* the training samples and recall them perfectly without necessarily being able to generalize well on new samples. A straightforward approach to avoid this common pitfall is to use a *hold-out* procedure in which all available samples are split at random into two parts. The first half is used to train the classifier (the *training set*), while the remaining half is used to assess the error (the *test set*). It is important that the training and the test set receive amounts of samples from each of the two classes (healthy or disease) that are proportional with the ratio between the classes in the expected application. This property is called stratification. A dataset split that respects the proportion is said to be well stratified or balanced.

The hold-out procedure is simple and easily understood as a correct method to use when it comes to validating a classifier. However, it has the disadvantage of using the data in a suboptimal way. This is because the classifier was neither trained nor tested on all available samples, but only a subset of them. The more samples used in training, the better the model will be; while the more samples used for testing, the more accurate is the performance estimate. An alternative classifier validation strategy is the use of *leave-one-out* crossvalidation method (LOO) in which the classifier is developed n times using $(n-1)$ samples and tested on the remaining sample. The n test results obtained in this way can be arranged into a confusion matrix, and the performance indices computed as described earlier. Although the estimate of the error obtained with the LOO procedure gives low bias, it may show high variance (8). A good trade-off between bias and variance may be obtained by using *N-fold crossvalidation* in which the dataset is split into $(n-m)$ training samples and m test samples ($N = n/m$). A threefold crossvalidation procedure is illustrated in Fig. 2.

5. PREDICITON MODELS

At the core of the classification process there is a prediction model M . This can usually be expressed as a set of mathematical expressions. The model takes as input the levels of d analytes observed for a given sample i , $\mathbf{x}_i$, and produces an output Ψ_i . The model M has typically a series of internal parameters $\mathbf{w}$ that are determined during the training process. The general prediction relation can be written as:

$$M(\mathbf{x}_i, \mathbf{w}) = \Psi_i \quad (1)$$

The class to which the sample I will be assigned depends on the magnitude of the output Ψ_i and a custom threshold T :

$$\hat{y}_i = \begin{cases} \text{Disease} & \text{if } \psi_i > T \\ \text{Healthy} & \text{otherwise} \end{cases} \quad (2)$$

The classifiers used in real applications include logistic regression and linear discriminants (9), weighted voting methods (10), support vector machines (11), and neural networks (12, 13). The differences among these types of classifiers include their complexity and ability to handle nonlinear class boundaries, as well as the way in which the parameters in the models are estimated. It is outside of the scope of this chapter to provide full details about various types of classifiers. More details can be found in the literature (14). However, in order to illustrate important concepts we will briefly introduce the logistic regression model. This model can be written as:

$$M(\mathbf{x}_i, \mathbf{w}) = \frac{1}{1 + \exp(-(w_0 + w_1 x_{i,1} + w_2 x_{i,2} + \dots + w_d x_{i,d}))} \quad (3)$$

It basically computes the weighted sum of the features plus an off-set (w_0) and transforms the result via the logistic function $f(z) = 1/[1 + \exp(-z)]$. The training of this classifier involves the estimation of the parameters $\mathbf{w}$ on the set of training samples. Once this step is completed via a sum of squared errors minimization, the model can be used to predict the class of any new sample. The prediction involves applying Eq. 2 with a threshold value, for instance, $T = 0.5$. Actually the value of Ψ_i for the logistic regression model is equivalent with the probability that the sample I belongs to the disease group. This interpretation holds if the class labels for diseased samples in the training set were set to 1, while those of the healthy samples were set to 0. The remainder of this chapter will illustrate the use of this classifier using a simulated dataset. In this simulation we consider we have 100 samples in each of the two groups (disease and healthy) and that we have measured ten analytes for each sample. We design the first two analytes to be somehow useful in predicting the class of samples, while the other eight to play the role of noise, i.e., no predictive value. The values of the predictor x_1 in the healthy group are drawn from a normal distribution with mean 0 and standard deviation of 1, while for the disease group the mean is set to 2. The values of the predictor x_2 in the healthy group are drawn from a normal distribution with mean -1 and standard deviation of 1, while for the disease group the mean is set to 0. The values of the remaining eight analytes in both groups are drawn from a normal distribution with mean 0 and standard deviation of 1, therefore deeming them not useful in the classification process. This dataset is shown in Fig. 3.

We split at random the simulated dataset into two equal and balanced parts simulating a hold-out validation strategy. Both the training and the test set have therefore 50 healthy and 50 diseased samples. Using the training dataset we train a logistic regression model, i.e., we estimate the coefficients $\mathbf{w}$ in Eq. 3 using the *glm* function in the R statistical language (<http://www.r-project.org>). These resulting coefficients are shown in Table 3 along with their significance p-value. The analyte x_1 appears to be the most significant followed by x_2 . This is exactly what we would have expected given the design of the simulate dataset. To test the ability of the trained model to generalize on unseen data, we apply the classifier to the simulated test set. The resulting confusion matrix is shown in Table 4. The performance indices on the test samples are: Amniotic cavity = 86%, Sen = 88%, Spec = 84%, and AUC = 91.3%. However, if we estimate the same performance indices of the trained model on the training data themselves, slightly optimistic values are obtained: Amniotic cavity = 88%,

Sen = 88%, Spec = 90%, AUC = 94.5%. The ROC curves for the training and test sets are shown in Fig. 4. These performance indices show that, as expected, evaluating the performance of the classifier on the training data will provide an optimistic view of its capabilities. Some readers might think that the differences between the two series of performance estimates are not too large. This is only because there were enough samples to well train the model (ten samples per analyte), and because the logistic regression model is robust to overfitting. With fewer data samples and more complex models the differences would have been much larger. Figure 5 shows the decision regions produced by the logistic regression model trained on the simulated dataset based only on the predictive features x_1 and x_2 . For comparison, the decision boundary of a three-layered neural network model with two hidden units is shown there, as well. It can be seen in this figure that two different prediction models produce different decision boundaries. However, since the number of data samples is not very large, both models produced the same accuracy on the test set. The same neural network model type, trained using a regularization procedure, provided 2% better accuracy than the logistic regression model on the same test dataset.

6. BIOMARKER/FEATURE SELECTION

Feature selection is a process in which one selects a subset containing ideally the most predictive features among all the measured ones. The feature selection step is very important in the classifier development because of several reasons, as follows. Firstly, several classifiers models (e.g., the logistic regression used earlier) cannot be trained when the number of samples is less than the number of features. Even though there are classification models, such as neural networks or support vector machines, that do not require more samples than features, it is always a good idea to have a number of features at least comparable if not lower than the number of available samples. Ideally, a ratio of ten samples per feature would provide solid and trustworthy models. Secondly, using uninformative features in the model together with the useful ones introduces noise and may, in fact, reduce the accuracy of the classifier. In the simulate dataset shown in Fig. 3, dropping the eight noisy feature would increase the *AUC* on the test set from 91.3 to 94.7%. This is possible because the more the number of features there are in the model, the fewer degrees of freedom are needed in order to estimate the parameters in the model. As a literature example, a classifier able to distinguish between two types of leukemia (ALL and AML) was built by Golub et al. (10) using 50 marker genes. Subsequently, it was shown that the same prediction accuracy can be obtained with only two or three of them (15). A third reason why few is better than more when it comes to inputs in a model is that a reduced number of features are easier to interpret in conjunction with the biology of the experiment under the study. Also, a simpler model reduced costs in a clinical setting (16) and the prediction time. However, the dimensionality reduction is not a goal per se, but only a means to improve the chances of obtaining a robust predictor that has good generalization ability.

In the context of the overall strategy for classifier development illustrated in Fig. 1, the feature selection step brings two challenges. The first challenge is to decide how to assess (measure) the goodness of a given subset of feature, and the second challenge is how to search for the best subset of feature in the explosive combinatorial space. A handy solution to the first challenge is to use the accuracy or *AUC* of the classifier using the given subset of features when trained and tested on the training set. The advantage of doing this is that no additional data are required and the model is trained with the full training set. The disadvantage of this is that there is no independent validation that the features are useful, and the truly best features may not necessarily be found. Nevertheless, the classifier development strategy as a whole is validated because the final test is done on an independent dataset (see Fig. 1). A second possibility to address the first challenge above is to use a hold-

out or crossvalidation procedure and to internally validate the choice of markers on the training data itself. The disadvantages of this are that there are less data to train the model (and therefore it may not predict well on the test dataset) and that the procedure is time consuming.

Given that a choice has been made on how to measure the goodness of a given subset of features, the second challenge, i.e., how to search for the best combination of them, has to be addressed. The only way to find the optimal combination that maximizes the goodness measure is to test all possibilities. For example, searching the best subset of markers out of ten available features would require to train all ten models of one input, all 45 models of two inputs, ..., 1 model of ten inputs, for a total of 1,023 models. However, if the number of available features is 100, this would require testing all 1.26×10^{30} possible combinations, which is not feasible in the lifetime of any scientist, using any available computer. Even if one would know in advance which the optimal subset size is, for example knowing that the best set contains 20 markers, finding the best set of 20 markers out of 100 is not tractable either.

A straightforward but clearly suboptimal solution to finding the best combination of d markers out of p available ones is to select the best single d features. This can be done by ranking the features based on a measure of the distribution dissimilarity among the sample groups. In this category of *filter* methods we can include the t -statistic (10) and Wilcoxon-rank-based U statistic. Another simple and informative measure of the predictive power of a single feature is the area under the ROC curve constructed in this case using a threshold classifier. We have applied these three filter methods to our simulated dataset given in Fig. 3. The results are shown in Table 5. It can be seen that all three filter methods ranked the features as expected: x_1 comes first followed by x_2 , and the remaining eight noisy features received much worse scores than the first 2.

A second feasible suboptimal solution to finding the best combination of d markers out of p available ones is to firstly reduce drastically the value of p by a filtering approach, to as low as a few hundred potentially interesting features. Then a *forward inclusion* or a *backward deletion* can be applied. With the forward inclusion the best single feature is selected first, then all combinations of the selected feature with the remaining ones are tested and the best one is retained. The procedure continues until d features are selected. The backward deletion starts with a large model of p inputs. One feature is removed at the time from the model, and the one that produced the minimum loss or the maximum gain in accuracy (compared with the full model) is definitely removed. The procedure is reiterated until there are only d features left in the model. More details on feature selection methods and classification can be found in the literature (9, 17, 18).

7. CONCLUSION

Developing a successful classifier requires an appropriate number of samples, ideally larger than the number of features used in the prediction model. Array-based technologies typically screen tens of thousands of analytes; therefore, selection of a reduced number of markers is required. The marker selection process together with the model training should be done on a separate dataset than the set on which the performance is measured. Crossvalidation methods provide a good compromise between the amount of data used to train the classifier and the reliability of the performance estimates.

References

1. Alizadeh A, et al. Distinct type of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*. 2000; 403:503–510. [PubMed: 10676951]

2. Perou, Jeffrey S, van der Rijni M, Rees C, Eisen M, Ross D, Pergamenschikov A, Williams C, Zhu S, et al. Distinctive gene expression patterns in human mammary epithelial cells and breast cancers. *Proc Natl Acad Sci USA*. 1999; 96(16):9212–9217. [PubMed: 10430922]
3. Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, Levine AJ. Broad patterns of gene expression revealed by clustering of tumor and normal colon tissues probed by nucleotide arrays. *Proc Natl Acad Sci*. 1999; 96:6745–6750. [PubMed: 10359783]
4. Ross T, Scherf U, Eisen MB, Perou CM, Rees C, Spellman P, Iyer V, Jeffrey SS, de Rijn MV, Waltham M, Pergamenschikov A, Lee JC, Lashkari D, Shalon D, Myers TG, Weinstein JN, Botstein D, Brown PO. Systematic variation in gene expression patterns in human cancer cell lines. *Nat Genet*. 2000; 24(3):227–235. [PubMed: 10700174]
5. Draghici, S. *Data Analysis Tools for DNA Microarrays*. Chapman and Hall/CRC; Boca Raton, FL: 2003.
6. Tarca AL, Romero R, Draghici S. Analysis of microarray experiments of gene expression profiling. *Am J Obstet Gynecol*. 2006; 195(2):373–88. [PubMed: 16890548]
7. Bradley E. Estimating the error rate of a prediction rule: improvement on cross-validation. *J Am Stat Assoc*. 1983; 78:316–331.
8. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*. Springer; 2001.
9. Dudoit S, Fridlyand J, Speed T. Comparison of discrimination methods for the sion data. *J Am Stat Assoc*. 2002; 97(457):77–87.
10. Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Collier H, Lo H, Downing JR, Caligiuri MA, Bloomfield C, Lander E. Molecular classification of cancer: class discovery and class predication by gene expression monitoring. *Science*. 1999; 286:531–537. [PubMed: 10521349]
11. Ramaswamy S, Tamayo P, Rifkin R, Mukherjee S, Yeang C, Angelo M, Ladd C, Reich M, Latulippe E, Mesirov J, Poggio T, Gerald W, Loda M, Lander E, Golub T. Multiclass cancer diagnosis using tumor gene expression signatures. *Proc Natl Acad Sci USA*. 2001; 98(26):15149–15154. [PubMed: 11742071]
12. Chatterjee M, Mohapatra S, Ionan A, Bawa G, Ali-Fehmi R, Wang X, Nowak J, Ye B, Nahhas FA, Lu K, Witkin SS, Fishman D, Munkarah A, Morris R, Levin NK, Shirley NN, Tromp G, Abrams J, Draghici S, Tainsky MA. Diagnostic markers of ovarian cancer by high-throughput antigen cloning and detection on arrays. *Cancer Res*. 2006; 66(2):1181–1190. [PubMed: 16424057]
13. Lin HS, Talwar HS, Tarca AL, Ionan A, Chatterjee M, Ye B, Wojciechowski J, Mohapatra S, Basson MD, Yoo GH, Peshek B, Lonardo F, Pan CJG, Folbe AJ, Draghici S, Abrams J, Tainsky MA. Autoantibody approach for serum-based detection of head and neck cancer. *Cancer Epidemiol Biomarkers Prev*. 2007; 16(11):2396–405. [PubMed: 18006929]
14. Tarca AL, Carey VJ, wen Chen X, Romero R, Draghici S. Machine learning and its applications to biology. *PLoS Comput Biol*. 2007; 3(6):e116. [PubMed: 17604446]
15. Somorjai RL, Dolenko B, Baumgartner R. Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions. *Bioinformatics*. 2003; 19(12): 1484–1491. [PubMed: 12912828]
16. Wang Y, Makedon F, Ford J, Pearlman J. Hykgene: a hybrid approach for selecting marker genes for phenotype classification using microarray gene expression data. *Bioinformatics*. 2005; 21(8): 1530–1537. [PubMed: 15585531]
17. Rogers, S.; Williams, R.; Campbell, C. *Bioinformatics Using Computational Intelligence Paradigms*, chapter Class Prediction with Microarray Datasets. Springer; Berlin: 2005. p. 119-142.
18. Tarca AL, Grandjean BPA, Larachi F. Feature selection methods for multiphase reactors data classification. *Ind Eng Chem Res*. 2005; 44(4):1073–1084.

Figure 1. Multianalyte classifier development. Data are preprocessed and then split into a training and a testing set. Using the training data, the most predictive analytes are identified and a model is trained. The resulting model is used to predict the samples membership in the test dataset and compute an unbiased estimate of classifier's performance.

Figure 2.

A 3-fold crossvalidation procedure applied on six healthy and six diseased samples dataset.

Figure 3.

A simulated dataset of 100 samples per group and ten analytes. The first analytes are deemed predicted about the class, while the remaining eight are noise.

Figure 4.
ROC curves for a 10-analyte logistic regression classifier.

Figure 5. Decision boundaries for two two-analyte classifiers. A logistic regression model (*left panel*) and a neural network model (*right panel*) are trained on the same data. The decision region for the simulated disease and healthy samples is shown in black and red, respectively. The simulated test data samples used to compute the classifier's performance are shown in *blue* (*filled circles* = healthy, *empty rectangles* = diseased).

Table 1

A generic confusion matrix for a 2-class problem

	True	
	Disease	Healthy
Disease	True positives	False positives
Predicted		
Healthy	False negatives	True negatives

Table 2

An example of confusion matrix in a 2-class classification problem

	True	
	Disease	Healthy
Disease	30	20
Predicted		
Healthy	10	40

Table 3

Coefficients of the logistic regression model for the dataset illustrated in Fig. 3

	Estimate	Std. error	z value	Pr(> z)
Intercept	-1.08	0.544	-1.989	0.0467
x_1	2.02	0.466	4.338	0.0000
x_2	0.95	0.363	2.609	0.0091
x_3	-0.02	0.362	-0.048	0.9615
x_4	-0.29	0.384	-0.757	0.4492
x_5	0.14	0.306	0.460	0.6458
x_6	-0.10	0.353	-0.281	0.7785
x_7	-0.04	0.314	-0.123	0.9019
x_8	-0.04	0.276	-0.150	0.8807
x_9	0.52	0.359	1.448	0.1476
x_{10}	-0.57	0.340	-1.673	0.0943

Table 4

Confusion matrix obtained with the logistic regression model on the simulate test set

	True	
	Disease	Healthy
Disease	44	8
Predicted		
Healthy	6	42

Table 5

Ranking of the features in the simulated dataset shown in Fig. 1

Feature	<i>P</i> (<i>t</i> -test)	<i>P</i> (U-test)	AUC
x_1	4.07-16	8.11e-13	0.914
x_2	4.52e-07	1.03e-06	0.783
x_3	0.5581	0.6969	0.523
x_4	0.5622	0.6566	0.474
x_5	0.4849	0.8442	0.509
x_6	0.8651	0.9917	0.501
x_7	0.6414	0.7801	0.485
x_8	0.7564	0.8227	0.487
x_9	0.6936	0.6666	0.473
x_{10}	0.2291	0.3574	0.446