Nurbek Saparkhojayev and Dale R. Thompson, “Matching Electronic Fingerprints of RFID Tags using the Hotelling's Algorithm”, in *Proc. IEEE Sensors Applications Symposium (SAS)*, New Orleans, Louisiana, Feb. 17-19, 2009, pp. 19-24.

SAS 2009 – IEEE Sensors Applications Symposium
New Orleans, LA, USA - February 17-19, 2009

MATCHING ELECTRONIC FINGERPRINTS OF RFID TAGS USING THE HOTELLING'S ALGORITHM

Nurbek Saparkhojayev and Dale R. Thompson, *Senior Member, IEEE*

Abstract— **Matching algorithms, called classifiers, determine if a previously enrolled instance matches an observed instance based on some rules. They return a decision, which consists of three possible answers: match, non-match, and unclassified. A classifier assigns a class label to a sample and then checks the new instance with a sample one. Or, the classifier is trained with example instances so that it learns what class label should be applied to future unknown instances. Classifiers are based on statistical, probabilistic, and decision rules. In applying classifiers, the most important issue is finding the matching rates. Two important rates are the false acceptance rate (FAR) and the false rejection rate (FRR). In this work, we determine the FAR and FRR for the Hotelling's two-sample T^2 algorithm applied to the application of matching electronic fingerprints of radio frequency identification (RFID) tags in the presence of simulated noise. The algorithm is found to be a robust classifier for this application.**

I. INTRODUCTION

IN radio frequency identification (RFID) systems, a tag with a unique serial number is attached to an object and is queried over a wireless channel by a reader. The unique serial number of the tag indexes a database that can contain more information about the object such as what it is and where it has been. RFID has become popular in the supply chain to identify and track items. In addition, it is being used in contactless identification cards such as passports and credit cards.

The unique serial number of the tag identifies the object. However, in some versions of RFID it is easy to create a duplicate tag that has the same serial number. Such a counterfeit tag could be attached to a different object to trick the RFID system into believing that the object is the same as the original object.

It is becoming more common to counterfeit RFID tags, so that people lose their confidentiality and security [1]. In 2005, researchers at Johns Hopkins University showed that an inexpensive toolkit built with minimum customized hardware can brute-force cryptographic keys from one of the most widely sold RFID tags [2].

One way to prevent counterfeiting of RFID tags is to create an electronic fingerprint of a tag, store it, and later verify against the stored fingerprint. An electronic fingerprint of a tag consists of a vector of numbers representing different features of the tag based on amplitude, frequency, phase, and timing. Such electronic fingerprints have been constructed for Bluetooth, cellular phones, and WiFi.

Given that an electronic fingerprint of a tag is created, an algorithm that compares the observed and enrolled tags to determine if there is a match with high probability is required. These algorithms are called classifiers.

Assume that there is a large database of tag fingerprints. There are two possible ways to use a classifier algorithm. The general problem is stated as follows. Given an unknown tag fingerprint, determine if it matches any of the enrolled fingerprints in the database. In this case, the classifier algorithm checks the observed tag with all enrolled tags, and based on some rules gives a decision, which consists of three possible answers: match, non-match, and unclassified. This is also known as the identification problem. The second problem, which is less complex, is stated as follows. Given a tag that claims to be an enrolled tag, determine if the observed tag's fingerprint matches the enrolled fingerprint. In this case, there are two possible answers: match or non-match. This is sometimes called the authentication problem. The focus of this work will be on the authentication problem, although in some systems the difference between identification and authentication is difficult to determine.

In both problems, there is inherent noise in the measurements. However, in the authentication problem the classifier will return either a match or non-match. Nevertheless, the classifier may falsely determine a match even though it is the incorrect fingerprint. In addition, the classifier may falsely reject a fingerprint that is the correct fingerprint because of noise. The false acceptance rate (FAR) is the probability that a false identity claim will be accepted and the false rejection rate (FRR) is the probability that a true identity claim is falsely rejected. In this work, FAR and FRR are the metrics for determining how well the classifier performs matching. The main focus of this work will be on how noise affects the ability of the classifier to determine a match or non-match.

Classifiers have been used in many fields of science and society. Built using a variety of mathematical techniques, they can be viewed as decision systems which accept values of some features or characteristics of a situation or test as input and produce as an output a discrete label related to the input values. Classifiers are also used to identify tags. Classifiers determine if the observed fingerprint matches the enrolled fingerprint at the verification time. Methods performing the matching consist of traditional matched filters used in communication systems [3], statistical

Nurbek Saparkhojayev is with the Computer Science and Computer Engineering Department, University of Arkansas, Fayetteville, AR 72701. E-mail: nursp81@gmail.com.

Dale. R. Thompson is with the Computer Science and Computer Engineering Department, University of Arkansas, Fayetteville, AR 72701. E-mail: d.r.thompson@ieee.org.

9781-4244-2787-1/09/$25.00 ©2009 IEEE

classifier algorithms [4], and data mining classifiers such as Bayesian [5]. There are three general classification methods:

- Statistical method: Decisions are made based on mathematic calculations and three possible answers are: match, non-match, and unclassified. Common classifiers include the Bayesian, k-nearest neighbor classifier, and Hotelling's two-sample T^2 algorithm.
- Machine learning: A method based on decision rules, decision trees, and decision tables. Decision trees are well-known classifiers in the field of statistics. A decision tree represents a collection of rules, which are organized in a hierarchical fashion, that implement a decision structure. The most popular one is called decision tree classifier.
- Neural networks: Hierarchical classifier that combines the properties of feed forward neural networks and a decision tree [6] type structure.

The Hotelling's two-sample T^2 algorithm is a traditional statistical classifier that extends the Student's-t statistic to multiple variables. It uses a vector of deviations between the observed and the enrolled means and the pooled sample covariance matrix to determine a match [7]. This classifier is based on hypothesis testing. It was proposed by Harold Hotelling in 1947 and for that reason is named Hotelling's two-sample T^2 algorithm [7]. The work in [8] deals with the area of the monitoring and controlling the process in multivariate quality with the Hotelling's two-sample T^2 algorithm and design. The Hotelling's two-sample T^2 algorithm was used in [9] for discovering differential expression in microarrays.

The Hotelling's two-sample T^2 algorithm is shown in Eqn. 1 [10]. The higher the T^2 value, the more distant is the observation from the mean, which means the probability of matching is less.

$$T^2 = \frac{n_1 n_2}{n_1+n_2}(\overline{y_1} - \overline{y_2})' S_{pl}^{-1}(\overline{y_1} - \overline{y_2}) \quad (1)$$

where n_1 and n_2 are the number of samples from the first and second sample groups, $\overline{y_1}$ and $\overline{y_2}$ are vectors of means, and S_{pl} is the pooled sample covariance matrix given by the following.

$$S_{pl} = \left(\frac{1}{n_1+n_2-2}\right)[(n_1-1)S_1 + (n_2-1)S_2] \quad (2)$$

where S_1 and S_2 are the sample covariance matrices of the first and second sample groups. It is well known that the T^2 statistic can be transformed into the F statistic with degrees of freedom p and $n_1 + n_2 - p - 1$ by the following.

$$\left[\frac{n_1+n_2-p-1}{(n_1+n_2-2)p}\right] T^2 = F_\alpha(p, n_1+n_2-p-1) \quad (3)$$

II. Experiment

Synthetic tag fingerprints were created using random number generator functions. The pseudo-code for creating the synthetic tag fingerprints is shown in Fig. 1. First, by using the uniform distribution, we create ten random tag fingerprints with four features on each of them having a range of [2, 8]. After the creating the ten tag template fingerprints with four features, we calculate the mean and standard deviation of each feature across all tags. Then, each feature is normalized. The purpose of the normalization is for each feature to have a mean of zero and standard deviation of one. Therefore, we should have positive and negative numbers with a mean of zero. Next, we create forty copies of each template, so that there are ten tags with forty different measurements. Twenty of the measurements represent twenty measurements taken during enrollment. The remaining twenty measurements represent the twenty measurements taken during verification. Finally, we add the simulated noise with the specified mean value and standard deviation value.

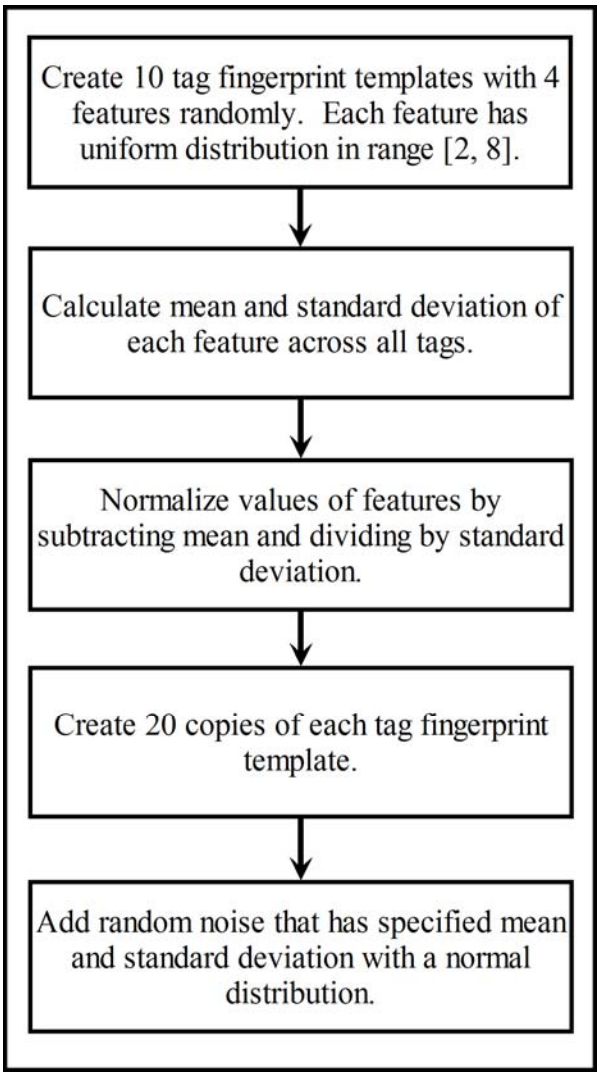

Fig. 1 – The pseudo-code for creating synthetic tag fingerprints.

The Hotelling's two-sample T^2 algorithm is tested for two different cases.

Case 1: We assume there are ten different tag fingerprints that each has four features. For each tag fingerprint, we simulate forty measurements in a noisy environment that has noise with a mean value of zero and a specified standard deviation. For each tag, we compare its fingerprint with the remaining nine tag fingerprints. The result has two possible answers: match or non-match. This is performed at noise levels with five different standard deviations. If the result of comparing a given tag with the other nine tags is a match, then this is a false match. The false acceptance rate (FAR) is calculated by dividing the number of false matches by the total number of matches performed.

Case 2: In the second case, we compare a set of fingerprints for a particular tag that has simulated noise with mean zero and a low standard deviation against sets of simulated tag fingerprints that have different noise means and the same standard deviation to see if the fingerprints are falsely rejected. We are investigating how much noise that the Hotelling's two-sample T^2 algorithm can tolerate before falsely rejecting the correct fingerprint with noise.

A. Case 1: FAR

In the first case, one tag fingerprint was compared against all other nine tag fingerprints to check the probability of being falsely accepted. Tag 0 was chosen and this tag was compared against tags with numbers in the range [1, 9]. The total number of comparisons is nine, because there are ten simulated tag fingerprints and we do not compare tag 0 with itself. For tags with noise that have mean of zero and standard deviation equal to 0.5, the T^2 statistic values were calculated using Eqn. 1, and compared with the transformed F statistic obtained using Eqn. 3. The F statistic is the same for all tag comparisons because we have the same number of measurements and the same number of features. For $p = 4$ and $n_1 = n_2 = 20$, the F statistic value is equal to 3.18 for 4 (p = 4) and 35 (n_1+n_2-p-1) degrees of freedom at $\alpha = 0.025$. The next step compares the T^2 statistic values with the transformed F statistic values independently, and calculates the average FAR for the distinct standard deviations.

The following formula from [10] is used to determine if the enrolled fingerprint matches the simulated observed fingerprint.

$$T^2 > \left[\frac{(n_1+n_2-2)p}{n_1+n_2-p-1}\right] F_\alpha(p, n_1+n_2-p-1) \qquad (4)$$

If T^2 is greater than the right-hand side of the equation, it is assumed that the tags are different. Otherwise, there is not enough evidence to support that they are different. In other words, if T^2 is smaller, then it is assumed that the observed fingerprint is the same as the enrolled fingerprint. For $p = 4$, $n_1 = 20$, $n_2 = 20$, and $\alpha = 0.025$, the right-hand side of Eqn. 4 is 13.81.

The results of Case 1 are shown in Table 1. The fingerprint for tag 0 is compared with all other tag fingerprints with zero mean noise and standard deviation values of 0.5, 0.75, 1.00, 1.25, and 1.50. Recall that the higher the value of T^2 means that there is more evidence that the fingerprints are different. The processing time for the algorithm to run in Mathematica is shown in Table 2.

Table 1. Case 1: Comparison of the fingerprint of tag 0 with all other tag fingerprints.

Comparison of tag 0 with tag#	T^2 for std. dev. 0.5	T^2 for std. dev. 0.75	T^2 for std. dev. 1.0	T^2 for std. dev. 1.25	T^2 for std. dev. 1.5
1	230.85	127.28	73.85	49.06	35.76
2	611.34	262.72	145.38	91.55	62.74
3	129.35	86.51	50.91	34.30	25.08
4	218.95	65.28	38.89	26.61	19.87
5	23.13	28.10	16.86	11.96	9.51
6	122.02	134.23	84.94	61.22	47.66
7	109.31	31.58	15.84	11.13	8.11
8	403.14	145.35	82.72	53.72	38.02
9	242.76	105.05	50.98	27.80	16.31

Table 2. Case 1: Processing time for finding the T^2 value.

Tag 0 to Tag#	Time for std. dev. 0.5 (s)	Time for std. dev. 0.75 (s)	Time for std. dev. 1.0 (s)	Time for std. dev. 1.25 (s)	Time for std. dev. 1.5 (s)
1	0.234	0.219	0.250	0.094	0.266
2	0.265	0.213	0.266	0.078	0.296
3	0.250	0.234	0.311	0.187	0.250
4	0.234	0.249	0.218	0.249	0.250
5	0.296	0.296	0.251	0.265	0.280
6	0.358	0.142	0.234	0.327	0.266
7	0.233	0.210	0.187	0.358	0.328
8	0.311	0.321	0.296	0.223	0.280
9	0.312	0.235	0.250	0.110	0.282

As seen in Table 1, the greater the standard deviation of the noise, the smaller the Hotelling's T^2 value, which means that it is harder to distinguish the different tags. In other words, the larger the standard deviation of noise the higher the probability that the observed fingerprint will look like tag 0. However, the running time of the Hotelling's two-sample T^2 algorithm is relatively constant as seen in Table 2.

The results of matching the fingerprint of tag 0 against all other tag fingerprints are shown in Table 3 for standard deviation of 0.5. No tag fingerprints match. Therefore, in the presence of noise with mean of 0 and standard deviation of 0.5, the Hotelling's two-sample T^2 algorithm is able to determine that all tags are different. The false acceptance rate is 0 %.

Table 3. Matching the fingerprint of tag 0 against other tag fingerprints with noise of mean 0 and standard deviation equal to 0.50.

Tag0 against Tag#	T^2 and $\sigma = 0.5$	$\left[\frac{(38)4}{35}\right] F_{\alpha=0.025}(4,35)$	Match
1	230.85	13.81	No
2	611.34	13.81	No
3	129.35	13.81	No
4	218.95	13.81	No
5	23.13	13.81	No
6	122.02	13.81	No
7	109.31	13.81	No
8	403.14	13.81	No
9	242.76	13.81	No

The results of matching the fingerprint of tag 0 against all other tag fingerprints are shown in Table 4 for standard deviation of 0.75. No tag fingerprints match. Therefore, in the presence of noise with mean of 0 and standard deviation of 0.75, the Hotelling's two-sample T^2 algorithm is able to determine that all tags are different. The false acceptance rate is 0%.

Table 4. Matching the fingerprint of tag 0 against other tag fingerprints with noise of mean 0 and standard deviation equal to 0.75.

Tag0 against Tag#	T^2 and $\sigma = 0.75$	$\left[\frac{(38)4}{35}\right] F_{\alpha=0.025}(4,35)$	Match
1	127.28	13.81	No
2	262.72	13.81	No
3	86.51	13.81	No
4	65.28	13.81	No
5	28.10	13.81	No
6	134.23	13.81	No
7	31.58	13.81	No
8	145.35	13.81	No
9	105.05	13.81	No

The results of matching the fingerprint of tag 0 against all other tag fingerprints are shown in Table 5 for standard deviation of 1.00. No tag fingerprints match. Therefore, in the presence of noise with mean of 0 and standard deviation of 1.00, the Hotelling's two-sample T^2 algorithm is able to determine that all tags are different. The false acceptance rate is 0%.

Table 5. Matching the fingerprint of tag 0 against other tag fingerprints with noise of mean 0 and standard deviation equal to 1.00.

Tag0 against Tag#	T^2 and $\sigma = 1.0$	$\left[\frac{(38)4}{35}\right] F_{\alpha=0.025}(4,35)$	Match
1	73.85	13.81	No
2	145.38	13.81	No
3	50.91	13.81	No
4	38.89	13.81	No
5	16.86	13.81	No
6	84.94	13.81	No
7	15.84	13.81	No
8	82.72	13.81	No
9	50.98	13.81	No

The results of matching the fingerprint of tag 0 against all other tag fingerprints are shown in Table 6 for standard deviation of 1.25. Two tag fingerprints match. Therefore, in the presence of noise with mean of 0 and standard deviation of 1.25, the Hotelling's two-sample T^2 algorithm falsely determines that two of the nine fingerprints match. The false acceptance rate is 22%.

Table 6. Matching the fingerprint of tag 0 against other tag fingerprints with noise of mean 0 and standard deviation equal to 1.25.

Tag0 against Tag#	T^2 and $\sigma = 1.25$	$\left[\frac{(38)4}{35}\right] F_{\alpha=0.025}(4,35)$	Match
1	49.06	13.81	No
2	91.55	13.81	No
3	34.30	13.81	No
4	26.61	13.81	No
5	11.96	13.81	Yes
6	61.22	13.81	No
7	11.13	13.81	Yes
8	53.72	13.81	No
9	27.80	13.81	No

The results of matching the fingerprint of tag 0 against all other tag fingerprints are shown in Table 7 for standard deviation of 1.50. Two tag fingerprints match. Therefore, in the presence of noise with mean of 0 and standard deviation of 1.50, the Hotelling's two-sample T^2 algorithm falsely determines that two of the nine fingerprints match. The false acceptance rate is 22%.

Table 7. Matching the fingerprint of tag 0 against other tag fingerprints with noise of mean 0 and standard deviation equal to 1.50.

Tag0 against Tag#	T^2 and σ = 1.5	$\left[\frac{(38)4}{35}\right] F_{\alpha=0.025}(4,35)$	Match
1	35.76	13.81	No
2	62.74	13.81	No
3	25.08	13.81	No
4	19.87	13.81	No
5	9.51	13.81	Yes
6	47.66	13.81	No
7	8.11	13.81	Yes
8	38.02	13.81	No
9	16.31	13.81	No

The false acceptance rates are shown in Table 8. It is possible for tag fingerprint 0 to be falsely accepted as one of the enrolled tag fingerprints when there is high noise level. From the Table 8, it is seen that FAR was 0% until the noise level standard deviation was increased to 1.25. The more noise we put into tag fingerprint, the easier it will be for this tag fingerprint to be falsely accepted by the Hotelling's two-sample T^2 algorithm as the enrolled tag fingerprint.

Table 8. FAR for noise levels with zero mean and five standard deviation values

Standard deviations, σ	False Acceptance Rate (FAR)
0.50	0%
0.75	0%
1.00	0%
1.25	22%
1.50	22%

B. Case 2: FRR

In the second approach, a single tag fingerprint with a standard deviation 1.50 was compared against itself at noise levels with different means. We had four different mean values of 0.25, 0.50, 0.75, and 1.00, and for each of them we created five different tag fingerprints by changing the seed. Also, we had one original copy of tag fingerprint for each mean value. Hence, the total number of comparisons was six. We started comparing tag fingerprints at a noise level with the mean value of 0.25 and we stopped when the mean value was 1.00. However, we did not change the standard deviation value. The average matching rates are shown in Table 9. The average false rejection rates are shown in Table 10.

Table 9. Matching rates of tag0's fingerprint against itself in noise levels with varying means and standard deviation 1.50

Noise level mean	T^2 values for six different random fingerprints	$\left[\frac{(38)4}{35}\right] F_{\alpha=0.025}(4,35)$	Average matching rate, %
0.25	2.00, 5.93, 3.27, 3.27, 4.92, 6.16	13.81	100
0.50	7.98, 12.61, 4.29, 8.77, 12.77, 7.96	13.81	100
0.75	17.97, 22.71, 7.29, 17.45, 25.11, 12.29	13.81	33
1.00	31.90, 36.25, 12.25, 29.29, 41.95, 19.17	13.81	17

Table 10. FRR of tag0's fingerprint against itself in noise levels with varying means and standard deviation 1.50 averaged over six runs.

Mean	False Rejection Rate (FRR)
0.25	0%
0.50	0%
0.75	67%
1.00	83%

Increasing the mean of the noise increases the rate at which the Hotelling's two-sample T^2 algorithm falsely rejects the fingerprint. This was the expected results. At a mean greater than 0.50, the algorithm starts to falsely reject fingerprints.

III. FUTURE WORK

While this work analyzes the Hotelling's two-sample T^2 algorithm for identifying tags, there are several other classifier algorithms. Therefore, future work to find and analyze other classifier algorithms on the application of electronic fingerprinting is needed. In addition, combinations of this algorithm and others need to be tested. The performance of the Hotelling's two-sample T^2 algorithm across a much larger set of parameters needs to be determined. In addition, it is possible to implement the Hotelling's two-sample T^2 algorithm in hardware level. Moreover, this algorithm can be implemented efficiently using different programming language. Furthermore, we can

implement this algorithm much more efficiently if we use parallel programming approach, which can multiply matrices in parallel way. Finally, real tag fingerprints need to be tested with the algorithm.

IV. Conclusion

The Hotelling's two-sample T^2 algorithm is robust in environments with noise. The covariance matrix captures the effects of noise so that the enrolled and observed tag fingerprints can be compared fairly using standard statistical hypothesis testing. The Hotelling's two-sample T^2 algorithm works well when the normalized noise level has a mean of zero and standard deviation between 0.50 and 1.00. Above a standard deviation of 1.00, the FAR increases. In addition, when the enrolled and observed tag fingerprints are measured in the same noise environment, the algorithm always correctly matches the enrolled and observed tag fingerprints.

The computation time of the Hotelling's two-sample T^2 algorithm appears to be constant. However, the runs were restricted to a fingerprint of four features, which generates a relatively small vector. If faster processing is required, the algorithm is very short and should lend itself to hardware implementation and/or parallel processing.

Acknowledgment

This material is based upon work supported by the National Science Foundation, Cyber Trust area, under Grant No. CNS-0716578.

References

[1] D.S. Kim, T-H. Shin, and J.S. Park, "A security framework in RFID multi-domain system," in Second International Conference on Availability, Reliability, and Security, 2007.

[2] T. Phillips, T. Karygiannis, and R. Kuhn, "Security standards for the RFID market", in IEEE Security & Privacy, 2005.

[3] R. Gerdes, T. E. Daniels, M. Mina and S. Russell, "Device identification via analog signal fingerprinting: A matched filter approach," in *Proc. 13th Annual Network and Distributed System Security Symposium*, San Diego, California, Feb 2006.

[4] J. Hall, M. Barbeau and E. Kranakis, "Detecting rogue devices in Bluetooth networks using radio frequency fingerprinting," in *Proc. IASTED Int'l Conf. Communications and Computer Networks*, Lima, Peru, Oct. 2006.

[5] J. Hall, M. Barbeau and E. Kranakis, "Enhancing intrusion detection in wireless networks using radio frequency fingerprinting," in *Proc. Communications, Internet and Information Technology (CIIT)*, St. Thomas, US Virgin Islands, November 22-24, 2004.

[6] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees.* Wadsworth: 1984.

[7] NIST/SEMATECH e-Handbook of Statistical Methods, National Institute of Standards and Technology, 2007. Available: http://www.itl.nist.gov/div898/handbook.

[8] L.M. Cheng, Y.Away, M.K.hasan, and N.M>Yusof, "Real-time Control Chart for Multivariate Statistical Process Control System," in *Proc.* Of the Joint Conference on Informatics and Research on Women in ICT (RWICT) 2004, 28-30 July 2004, Putra World Trade Center, Kuala Lumpur, Malaysia.

[9] Y. Lu, P. Liu, P. Xiao, and H. Deng, "Hotelling's T^2 multivariate profiling for detecting differential expression in microarrays," in Bioinformatics, vol.21, no.14, pp 3105-3113, 2005.

[10] A. C. Rencher, *Methods of Multivariate Analysis*, New York, NY: John Wiley & Sons, 1995.