

**INVESTIGATION AND IMPLEMENTATION OF GENE SIGNATURE
DEVELOPMENT METHODS USING MICROARRAY DATA – A CASE STUDY
ON EARLY STATE NON-SMALL CELL LUNG CANCER**

by

Ruiqi Huang

BS, Biological Sciences, Xi'an Jiaotong – Liverpool University, China, 2012

Submitted to the Graduate Faculty of

Biostatistics

Graduate School of Public Health in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2015

UNIVERSITY OF PITTSBURGH
GRADUATE SCHOOL OF PUBLIC HEALTH

This thesis was presented

by

Ruiqi Huang

It was defended on

April 23, 2015

and approved by

Committee Chair

John Wilson, PhD, Assistant Professor, Biostatistics, Graduate School of Public Health,
University of Pittsburgh, Pittsburgh, PA

Committee Member

Shuguang Huang, PhD, Associate V.P of Department of Bioinformatics, Helomics Inc.
Pittsburgh, PA

Committee Member

Gong Tang, PhD, Associate Professor, Biostatistics, Graduate School of Public Health,
University of Pittsburgh, Pittsburgh, PA

Committee Member

Maria Brooks, PhD, Associate Professor, Department of Epidemiology, Graduate School of
Public Health, University of Pittsburgh, Pittsburgh, PA

Copyright © by Ruiqi Huang

2015

**INVESTIGATION AND IMPLEMENTATION OF GENE SIGNATURE DEVELOPMENT
METHODS USING MICROARRAY DATA
– A CASE STUDY ON EARLY STAGE NON-SMALL CELL LUNG CANCER**

Ruiqi Huang MS

University of Pittsburgh, 2015

ABSTRACT

Purpose

Gene signature development using microarrays started more than 15 years ago, yet there are still common mistakes made by researchers. The goal of this research is to investigate and implement gene signature methods using Affymetrix array data. It aims to establish a work flow with well-justified steps for gene signature development.

Public Health significance

The public health significance is to minimize NSCLC patients' risks of recurrence after surgical resection by identifying poor prognosis patients and suggesting that those who have high risk of recurrence to receive chemotherapy and/or supplemental treatments after surgery.

Methods

Gene expression data from 62 surgery samples of early stage un-treated NSCLC patients in JBR10 trial was used for training model development. Individual genes were selected using univariate Cox regression analysis, and then the gene set was summarized by principle components, which then served as the inputs to the Cox regression model. A multi-layer internal validation was conducted for model evaluation. The performance of the gene signature was evaluated by testing on two independent data sets.

Results

A signature of 88 genes was developed that can identify patients with significantly different survival prognosis (Hazard Ratio (**HR**): 11.5, 95% CI: 3.44 to 38.46, $P < 0.05$). The signature was successfully validated in independent datasets (CAN_DF (N=59): **HR**: 3.56, 95% CI: 1.38 to 9.19, $P < 0.05$; **HR**: 1.94, 95% CI: 0.89 to 4.21, $P = 0.088$; CAN_DF: **HR**, 95% CI, P ; **HR**, 95% CI, P ; UM (N=155): **HR**: 1.82, 95% CI: 1.09 to 3.03, $P < 0.05$; UM (N=176): **HR**: 1.95, 95% CI: 1.29 to 3.40, $P < 0.05$).

Conclusion

A work flow of gene signature development has been constructed, which is composed of preliminary gene filtering, individual gene selection, predictive model construction using supervised principle component analysis and further internal/external validation. . Using gene expression of 62 patients from Affymetrix array data in JBR.10 trials, an 88-gene signature for predicting a high likelihood of recurrence was obtained and validated in independent datasets.

TABLE OF CONTENTS

PREFACE.....	XII
1.0 INTRODUCTION.....	1
2.0 MATERIALS AND DATA	4
3.0 METHOD	5
3.1 CLINICAL DATA AND DESCRIPTIVE ANALYSIS.....	5
3.2 DATA PROCESSING	5
3.3 PRELIMINARY GENE FILTERING	5
3.4 INDIVIDUAL FEATURE SELECTION	6
3.5 PREDICTIVE MODEL BUILDING.....	6
3.6 INTERNAL VALIDATION	7
3.7 SIGNATURE VALIDATION ON INDEPENDENT MICROARRAY DATA SETS	8
4.0 RESULTS	9
4.1 PATIENT DEMOGRAPHICS.....	9
4.2 DERIVATION OF THE GENE EXPRESSION SIGNATURE.....	10
4.3 INTERNAL VALIDATION	13
4.4 VALIDATION OF PROGNOSTIC SIGNATURE	13
4.4.1 CAN_DF	14
4.4.2 UM.....	15
5.0 CONCLUSION AND DISCUSSION	18
APPENDIX A: SUPPLEMENTAL RESULTS UNDER COX ALONE STRATEGY	20

APPENDIX B: TRAINING AND TESTING RESULTS UNDER COX+ T TEST STRATEGY.....	24
APPENDIX C: R CODE	29
BIBLIOGRAPHY	37

LIST OF TABLES

Table 1 Demographic Feature of Patients in the Training and Validation Cohorts.....	9
Table 2 Cox Regression Results from Training Data, Risk Stratification Using Cutoff of 40 th and 50 th Percentile of Risk Score.....	12
Table 3 Risk Stratification in CAN_DF and UM datasets Using a 88-Gene Signature	14
Table 4 Cox Regression Results from CAN_DF Dataset, Risk Stratification by an 88-Gene Signature, Using 40 th and 50 th Percentile of Risk Scores as cutoffs from Training Data.....	14
Table 5 Cox Regression Results from UM Dataset, Risk Stratification by an 88-Gene Signature, Using Cutoff of 40 th and 50 th Percentile of Risk Scores from Training Data.....	16
Table 6 coefficient of each probes of 88-gene signature under 50 th percentile of risk score as cutoff	20
Table 7 Cox regression on risk groups defined by a 118 gene signature using cutoffs of 40 th and 50 th percentile of risk score as cutoffs (training)	21
Table 8 Risk Stratification in CAN_DF and UM datasets Using a 88-Gene Signature	21
Table 9 Cox regression on risk groups defined by a 118 gene signature using cutoffs of 40 th and 50 th percentile of risk score as cutoffs (CAN_DF)	22
Table 10 Cox regression on risk groups defined by a 118 gene signature using cutoffs of 40 th and 50 th percentile of risk score as cutoffs (UM)	23
Table 11 Cox regression results on risk groups from training data under Cox+ t test scenario, risk stratification using 40 th and 50 th percentile of risk score as cutoffs (training).....	24
Table 12 Cox regression results on risk groups from training data under Cox+ t test scenario, risk stratification using 40 th and 50 th percentile of risk score as cutoffs (CAN_DF).....	25

Table 13 Cox regression results on risk groups from training data under Cox+ t test scenario, risk stratification using 40th and 50th percentile of risk score as cutoffs (UM)..... 27

LIST OF FIGURES

Figure 1 Hazard Ratio from Principal Component Regression Model through an Iterative Process	11
Figure 2 Kaplan Meier Curves of Survival for High-Risk and Low-Risk Groups Assigned by 88-Gene Signature at Cutoffs of 40 th and 50 th percentile of Risk Score	12
Figure 3 Kaplan Meier Curves of Survival for High-Risk and Low-Risk Groups Assigned by 88-Gene Signature at Cutoff of 40 th and 50 th percentiles of Risk Score	15
Figure 4 Kaplan Meier Curves of Survival for High-Risk and Low-Risk groups assigned by 88-Gene Internal validation accessing modeling predict accuracy	16
Figure 5 Prognostic Signature Development and Validation Flowchart	17
Figure 6 Kaplan Meier Curves of survival for high- and low-risk groups assigned by a 118-gene signature using cutoffs of 40 th and 50 th percentile of risk scores as cutoffs (training)	21
Figure 7 Kaplan Meier Curves of survival for high- and low-risk groups assigned by a 118-gene signature using cutoffs of 40 th and 50 th percentile of risk scores as cutoffs (CAN_DF)	22
Figure 8 Kaplan Meier Curves of survival for high- and low-risk groups assigned by a 118-gene signature using cutoffs of 40 th and 50 th percentile of risk scores as cutoffs (UM)	23
Figure 9 Kaplan Meier Curves of survival for high- and low-risk groups assigned by a 88-gene and 118-gene signature using cutoffs of 40 th and 50 th percentile of risk scores as cutoffs under Cox+T test strategy	24
Figure 10 Kaplan Meier Curves of survival for high- and low-risk groups assigned by a 88-gene and 118-gene signature using cutoffs of 40 th and 50 th percentile of risk scores as cutoffs under Cox+T test strategy (CAN_DF)	26

Figure 11 Kaplan Meier Curves of survival for high- and low-risk groups assigned by a 88-gene and 118-gene signature using cutoffs of 40th and 50th percentile of risk scores as cutoffs under Cox+T test strategy (UM)..... 28

PREFACE

A special thanks goes to Dr. Huang for his excellent guidance on my thesis and supervising on my work at Helomics. I feel very thankful to have the opportunity for this almost one-year internship in learning and practicing statistical modeling.

I would like to thank Dr. John Wilson, the committee chair, and also my academic advisor for the reviewing of my thesis.

I am grateful to Dr. Tang and Dr. Brooks, who are willing to help and participate in my thesis committee. Thanks a lot for all your concerns and suggestions. Those really help me in improving my thesis.

1.0 INTRODUCTION

Gene signature development using microarrays started more than 15 years ago, yet there are still common mistakes made by researchers. Some of these mistakes are very basic. For example, the expression level, expression variation, and detectability (absent/present call) are often not considered in the selection of individual gene[1]. The primary goal of this thesis research project is to investigate and implement gene signature using Affymetrix array data. We aim to establish a work flow with well-justified steps for gene signature development. Generally, Cox proportional hazards regression was used for individual gene selection and principal components analysis of the gene sets were used for predictive model construction using Cox proportional hazards regression models. We have established a rigorous workflow for model evaluation that includes interval validation (e.g. leave-out-out cross validation, Bootstrapping) and validation on independent datasets.

The secondary goal is to compare different strategies in developing gene signature. In particular, we are interested in comparing different strategies for (i) individual feature selection and (ii) prediction model building after genes are selected. In surveying the literature for cancer gene signatures, most, if not all, choose the individual genes using Cox regression[2]. In this research, we explore and compare whether other strategies, such as t-test, can identify more informative genes. The results for the secondary goals are provided in appendix.

Lung cancer is the leading causes of cancer-related death in US and worldwide. Non-small cell lung cancer (NSCLC) defined as a disease in which malignant cells form in the tissues of the lung, occupies more than 85% of lung cancer cases[3]. Squamous cell carcinoma, large cell carcinoma and adenocarcinoma are three common types of non-small cell lung cancer named for the kinds of cell found in the cancer and how the cells look under a microscope. NSCLC are classified into 4 main stages (I-IV) by the TNM staging system, depending on the size of the tumor and, where the tumor is found [4]. Cancer stage is regarded as a main factor in guiding treatment. Surgical resection is the most recommended surgery procedure, yet five-year survival ranges from only 30%-60% percent among early-stage NSCLC patients[5]. 30% to 40% of stage I patients will relapse[6]. Nearly 50% of patients with stage I/ II non-small cell lung cancer (NSCLC) will die from recurrent disease despite surgical resection[7]. Though the current standard of treatment for patients with stage I NSCLC remains surgery alone, those poorer prognosis patients might benefit from ACT. Previous clinical trials[8] have determined that adjuvant vinorelbine plus cisplatin based chemotherapy (ACT) can prolong disease-free and overall survival among patients with completely resected in a range of IA-III A NSCLC [4]. However, the survival benefit for patient with stage IB is not significant[5, 9]. Very few patients with stage IA NSCLC have been enrolled in cisplatin-based chemotherapy. Some even observed a potential detrimental effect of using the chemotherapy on stage IA patients[8].

A challenge is the heterogeneity in recurrence rate among patients with the same lung cancer stage. This means that TNM staging incorrectly predicts the diseases recurrence and further suggest follow-up treatments. It's crucial to isolate a reliable molecular signature in tumors that could be used to identify those who are likely to develop recurrent disease and would thus benefit from adjuvant chemotherapy[10]. It's also assumed that a multiple gene signature might be

stronger than individual gene signature. Currently there is no consistent prognostic molecular marker for early stage cancer. Prognostic signatures in NSCLC with minimal overlap in their gene sets have been identified among previous studies[10-18]. A few of them have been subjects to independent validation[10, 12, 14, 16]. A recent study on identifying a 15-gene signature demonstrated its potential to identify high-risk patients among observation patients[1]. The testing results failed to predict the benefit from adjuvant chemotherapy as it proposed. Moreover, four of the fifteen genes in this signature are not always detectable (often called 'Absent' by Affymetrix MAS5 algorithm), indicating a less robust biomarker product.

The ultimate goal of personalized medicine is to limit chemotherapy intervention to those who will derive maximum benefit from it. In this study, it is hypothesized that if prognostic value of gene signature is achievable, risk stratification can be predicted for untreated NSCLC patient at stage I-II. Thus adjuvant chemotherapy can be applied to those who have higher risk and then 5-year survival of patient can be improved.

2.0 MATERIALS AND DATA

The training data set was composed of sixty-two gene expression profiles generated with U133A oligonucleotide microarrays by the National Cancer Institute of Canada Clinical Trials Group JBR.10[5, 8]. These were fresh frozen tumor tissues from early stage NSCLC patients who did not undergo chemotherapy. The microarray data were publically available at the National Center for Biotechnology Information Gene Expression Omnibus (GSE14814).

Candidate signatures were tested in two independent microarray data sets, which were all from the Consortium with support and collaboration of NCI investigators to develop and validate gene expression signatures of lung adenocarcinomas. The two testing datasets were a set of 176 samples from University of Michigan (UM) and a dataset of 83 samples from Dana-Farber Cancer Center (CAN_DF)[19].

3.0 METHOD

3.1 CLINICAL DATA AND DESCRIPTIVE ANALYSIS

The event of interest was defined as relapse free survival (free of lung cancer recurrence within 5-year follow-up), ie. survival time was calculated as $t = \text{Min}(\text{relapse-free follow-up time}, 5)$. Death from other causes was simply described as right censoring. Events occurred after 5 years were considered as a non-event; any non-event with less-than 5-year follow up time was considered censored.

3.2 DATA PROCESSING

Arrays of JBR10 were processed at two different times. Thereby data used here were from the original author in which the batch effect was removed by distance-weighted discrimination method. For all of the datasets, raw microarray data were normalized by the RMA method and then transformed to log₂ scale[20]. The Affymetrix MAS5 algorithm[21, 22] was used to evaluate the Absent/Present call for each probe set.

3.3 PRELIMINARY GENE FILTERING

From the entire 22283 probe sets, genes with low expression level and quality was filtered out. That is, we kept only probes with grade A annotation[23] and with high mean (greater than 25th

percentile), and high variation (standard deviation greater than 25th percentile), and probe sets with more than 50% Present call among sample for future analysis. The expressions for each gene were then standardized to z-score (centered and scaled) after primary gene filtering.

3.4 INDIVIDUAL FEATURE SELECTION

In order to preselect survival-related genes, univariate analyses were performed on each probe set using both two-sample t-tests (equal variance) and Cox regression models. Specifically, in t-tests, patients with events occurring before or after 5 years were assigned into two groups. Thereby those censored before 5 years belonged to neither of the groups (9 patients were identified) and were excluded from the t-test analysis. Probes with significant association with survival ($P < 0.05$) in either test were retained. Candidate genes were the top probes from either the results of Cox regression alone (main criterion) or from both Cox regression and t tests (alternative criterion). For the Cox + T-test approach, two-thirds of the genes were from Cox regression, and the remaining one-third are exclusively from t-test. These two gene selection methods were compared conditioning on the same number of genes used in the model.

3.5 PREDICTIVE MODEL BUILDING

Supervised Principal Component Regression Model (**SuperPC**)

To develop a gene signature for prediction patient's survival outcome, principle component analysis (PCA)[1] was applied to synthesize information from the candidate genes selected above

(Cox regression alone or Cox regression + T-test). The first six principal components (PC) with Eigen Values >1 were chosen as the inputs to Cox regression model. The risk scores derived from a Cox regression was a linear combination of the 6 PCs, which was then transformed into a linear combination of the individual genes. Taking the binary risk group as the covariate, univariate Cox regression analysis was conducted test the association between risk categories and survival outcome. Differences of survival distributions between two risk groups were studied by Kaplan-Meier product limit methods and log rank tests.

To determine the optimal number of probes for the signature, an iterative process was conducted to evaluate the model performance versus the number of probe sets included. Specifically, the process followed these steps: (1). Based on the gene rank from the Cox regression analysis, one probe each time was added to the candidate gene set (2). PCA of the gene set, and (3) Cox regression on the top 6 PCs. The regression model was then used to calculate the risk score for each patient. The Cox cutoff was predetermined to be the 50th percentile and 40th percentile of risk scores. (4) Calculate HR for high-risk group for each model and plot HR versus the number of genes. (5). Identify the area such that the model performance was good and also stable over a range of the number of genes.

3.6 INTERNAL VALIDATION

A multi-layer strategy of internal validation was conducted that includes bootstrapping and leave-one-out cross validation (LOOCV). Specifically, using Bootstrap sampling 40 times, each Bootstrap sample was composed of 62 random draws with replacement from the original 62 patients. Then for each of the 40 bootstrap samples, LOOCV was performed. As a result, there

were 62 LOOCV procedures within one Bootstrap. For one LOOCV, a SuperPC model was constructed for signature development based on gene expression of chosen 61 patients. Then the obtained cutoff was used to identify risk category of the left one patient by comparing to his/her risk score.

With a pre-determined number of genes to be used in the signature, the validation process was followed the same way for each Bootstrap sample (Individual Feature Selection -> PCA -> Cox Regression -> risk score formula and cutoff determination). This algorithm was then applied to the one sample left out to determine its risk category.

The accuracy for each gene signature was calculated to evaluate its performance on testing data. It was defined as the percentage of patients who had developed events within 5 years of follow-up who were predicted to be in the high-risk group plus the percentage of patients who were censored after 5 years and were predicted to be in the low-risk group.

3.7 SIGNATURE VALIDATION ON INDEPENDENT MICROARRAY DATA SETS

The 88-gene signature was tested for its performance on two independent published microarray data (CAN_DF, UM) described above. The risk scores were calculated according to the risk score formula obtained using the training data. When the risk scores were dichotomized at the values of 40th percentile and 50th percentile of risk scores from the training set, the 88-gene signature classified samples into high and low risk groups.

4.0 RESULTS

4.1 PATIENT DEMOGRAPHICS

Histology subtype, cancer stage, age, sex and event rate were described in the table below for both training and testing datasets. Generally, distributions of those geographic information seem to be somewhat distinct in different cohorts. However, due to the fact that this study mainly focused on exploring the prognostic value from genomic data, the independence from clinical information would not be accessed here. We considered the event rate (41.9%) for training dataset as a guide for cutoff selection. Thus 40th percentile of risk score could be used as the cutoff for risk stratification. However, a 50th percentile of risk score would be more preferable if we want to increase sensitivity in detecting poor prognosis patients.

Table 1 Demographic Feature of Patients in the Training and Validation Cohorts

Clinical factor	JBR.10 (N=62)	UM (N=176)	CAN_DF (N=83)
Pathological subtype			
Adenocarcinoma	32(51.6%)	176(100%)	83(100%)
Non-adenocarcinoma	30(49.4%)	0(0%)	0(0%)
ACT			
Treated	0(0%)	21(11.9%)	24(28.9%)
Untreated	62(100%)	155(88.1%)	59(71.1%)
Stage			
I	34(54.8%)	115(65.3%)	57(68.7%)
II	28(45.2%)	28(15.9%)	26(31.3%)
III	0(0%)	33(18.8%)	0(0%)
Age			
>=65	43(69.4%)	90(51.1%)	32(38.6%)
<65	19(30.6%)	86(48.9%)	51(61.4%)
Sex			
Male	44(71%)	98(55.7%)	46(55.4%)
Female	18(29%)	78(44.3%)	37(44.6%)

Starting from 22283 probes, as designed on the U133A Affymatrix platform, primary gene filtering kept 22277 with Grade A annotation. Then 14232 probes with mean and median greater than 25% of those from Grade A probes were retained. This was followed by filtering out probes with less than 50% Present calls, and 8910 probes were left. Using $P=0.05$ as the cutoff, 424 of the 8910 probe sets were significantly associated with survival in 62 patients. Using the same cutoff, 310 of 8190 were left from two-sample T test of 53 patients.

4.2 DERIVATION OF THE GENE EXPRESSION SIGNATURE

Figure 1, including the results of ranked probe sets from a number of 6 to 200, illustrated how HR changed at the two cutoff levels (40th percentile and 50th percentile) as increasing number of probes included in the risk score models.

A homogeneous trend was found between two different cutoff levels. As depicted, the hazard ratios obtained fluctuate significantly when numbers of probes included were small. It reached a local maximum at around 50, and sustained until around 100. The hazard ratios then decreased after probes >120.

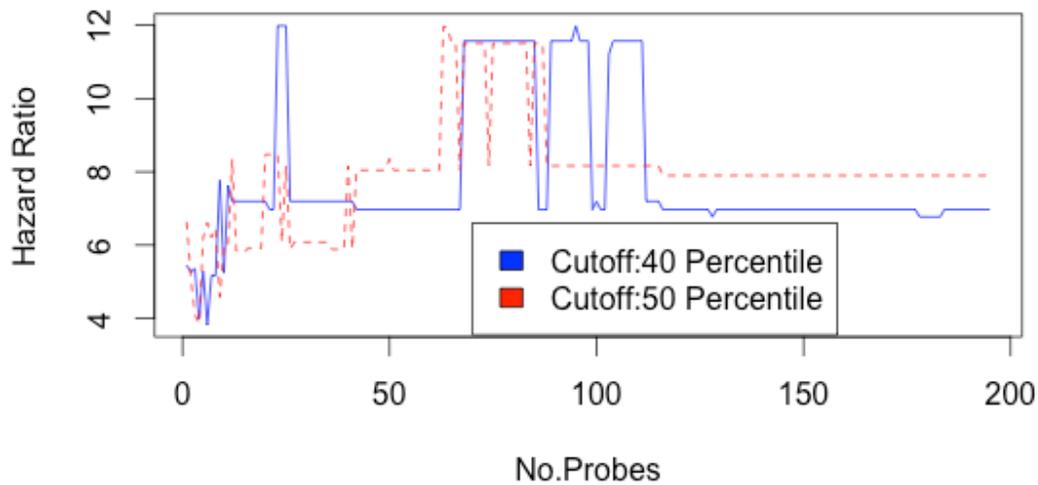


Figure 1 Hazard Ratio from Principal Component Regression Model through an Iterative Process

It was our primary interests to include a number of probes that provide relative high hazard ratios under this iterative procedure. In addition, it was assumed that additional contribution from T tests would help to improve signature performance especially when a fixed number of probes from the Cox-only scenario did not perform well. Moreover, choosing appropriate number of probes was also a matter of convenient comparison among different levels of cutoffs. Based on the above considerations, 88 and 118 were chosen as fixed number of probes to be considered in the multi-variable model under both Cox-only scenario and Cox+T test scenario. Considering levels of cutoff for each candidate signature, we eventually have 8 models (2 X 2 X 2) constructed under training dataset.

Under 4 models with different fixed number of probes included in the statistical model, both the 88-gene and the 118-gene signature were able to separate risk groups in combined stage I/II with a relative higher hazard ratio in a under Cox-only scenario in different cutoff level. (**Table 2:** Under the Cox-Only scenario: 88-genes: 40th Percentile of RS: HR: 11.6, 95%CI: 2.74 to 49.18,

P<0.001; 50th Percentile of RS: HR: 11.5, 95%CI, 3.44 to 38.46, P<0.001; 118-genes: 40th Percentile of RS: HR: 7.19, 95%CI: 2.15 to 24.05, P<0.001; 50th Percentile of RS: HR: 8.16, 95%CI, 2.8 to 23.75, P<0.001).

Table 2 Cox Regression Results from Training Data, Risk Stratification Using Cutoff of 40th and 50th

Percentile of Risk Score			
No. Probes Cutoff Level	HR	95CI%	Log-Rank Test P value
88			
40 th Percentile	11.6	(2.74,49.18)	2.76e-05
50 th Percentile	11.5	(3.44,38.46)	5.94e-07
118			
40 th Percentile	7.19	(2.15,24.05)	1.88e-04
50 th Percentile	8.16	(2.8,23.75)	4.95e-06

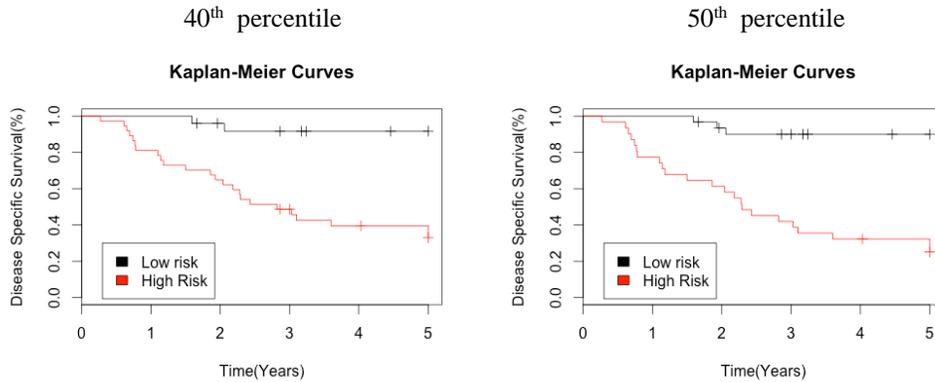


Figure 2 Kaplan Meier Curves of Survival for High-Risk and Low-Risk Groups Assigned by 88-Gene Signature at Cutoffs of 40th and 50th percentile of Risk Score

Results of the 118-gene signature and results under Cox plus T test scenario could be found in **appendix (Table 7 and Table 11)**. As mentioned, we observed same pattern of signature performance, in terms of values of hazard ratio, under models with two different cutoffs.

4.3 INTERNAL VALIDATION

A multi-layer internal validation was conducted onto the 88-gene /118-gene signatures with 2 different cutoffs in order to compare signature performance of these four models. The mean accuracy values, averaged by 40 accuracy values in Bootstraps for one model, were compared using T tests. As a result, there were no significant differences between the accuracies of signature of different number of probes using either 40th or 50th Percentile of risk score as cutoffs (T test of 88-Gene vs. 118-Gene: At 40th percentile: P=0.6; At 50th percentile: P=0.947), between the accuracies of same number of probes under different cutoffs (T test of 88-gene: At 40th Percentile of RS v.s. 50th Percentile of RS: P=0.062; T test of 118-gene: At 40th Percentile of RS v.s. 50th Percentile of RS: P=0.178;). It was also noticed that difference between the mean accuracy of 88-gene at 50th and 40th percentile levels was at a marginal level.

4.4 VALIDATION OF PROGNOSTIC SIGNATURE

The 88-gene signature was tested for its significance in three independent published microarray data sets. The cutoffs from the 40th and 50th percentile of risk score obtained in training sets were -0.505 and -0.154, which we used to classify patients from CAN_DF, UM and MSK datasets into low- and high- risk groups. In detail, we tested on no treatment subjects and all treatment status subject in CAN_DF (59 vs. 83) and UM dataset (155 vs. 176), regardless of their cancer stages. The numbers of patients assigned to each group in each validation set were shown in **Table 3**. Results of 118-gene signature could be found in **Appendix (Table 8)**.

Table 3 Risk Stratification in CAN_DF and UM datasets Using a 88-Gene Signature

Test Cohort	Event Rate	40 th Percentile of Risk Score		50 th Percentile of Risk score	
		High-risk	Low-risk	High-risk	Low-risk
CAN_DF (83)	29/83=34.9%	20	63	40	43
CAN_DF (59)	21/59=35.6%	15	40	30	29
UM (176)	75/176=42.6%	45	131	79	97
UM (155)	65/155=41.9%	42	113	71	84

4.4.1 CAN_DF

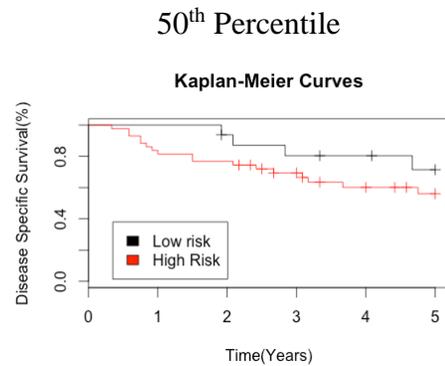
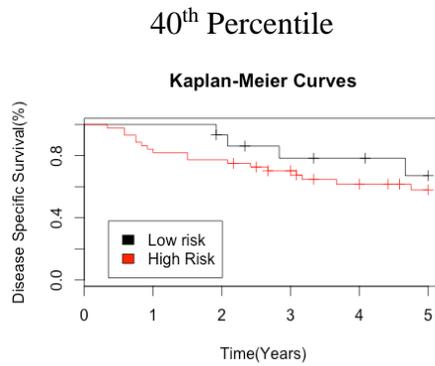
Overall, there was evidence of validation in applying purposed gene signatures in CAN_DF datasets, as shown in **Table 4, Table 9 and Table 12.**

The 88-gene signature (**Table 4**) under 50th percentile of risk scores as cutoff (N=59: HR: 3.56, 95%CI: 1.38 to 9.19, P<0.05;N=83: HR: 1.94, 95%CI: 0.89 to 4.21, P<0.088;) performed better compared with that under 40th percentile of risk scores as cutoff (N=59: HR: 1.63, 95%CI: 0.55 to 4.85, P=0.375;N=83: HR: 1.62, 95%CI: 0.62 to 4.27, P=0.324;), in terms of separating subjects into two risk groups with significantly different survival. Kaplan Meier Curves (**Figure 3**) illustrated how survival in different groups varied over 5 years.

Table 4 Cox Regression Results from CAN_DF Dataset, Risk Stratification by an 88-Gene Signature, Using 40th and 50th Percentile of Risk Scores as cutoffs from Training Data

No. Probes	HR	95CI%	Log Rank Test
Cutoff Level			P value
N=59			
40 th Percentile	1.63	(0.55,4.85)	0.375
50 th Percentile	3.56	(1.38,9.19)	0.005
N=83			
40 th Percentile	1.62	(0.62,4.27)	0.324
50 th Percentile	1.94	(0.89,4.21)	0.088

N=59



N=83

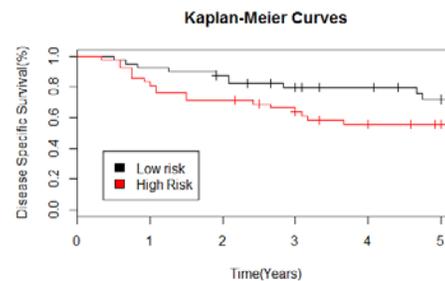
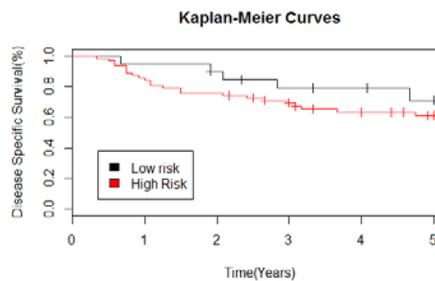


Figure 3 Kaplan Meier Curves of Survival for High-Risk and Low-Risk Groups Assigned by 88-Gene Signature at Cutoff of 40th and 50th percentiles of Risk Score

4.4.2 UM

Significant results (**Table 5**) were obtained in UM datasets at both cutoffs from the 88-gene signature (At cutoffs of 40th percentile of risk scores: N=155: HR=1.73, 95%CI; 0.94 to 3.18,P=0.07; N=176: HR=1.95, 95%CI; 1.07 to 3.55,P=0.0254; At cutoffs of 50th percentile of risk scores: N=155: HR=1.82, 95%CI; 1.09 to 3.03,P=0.0188; N=176: HR=2.09, 95%CI; 1.29 to 3.40,P=0.00243). Similarly, it appeared to perform better in the model using cutoffs of 50th percentile of risk scores. Kaplan Meier Cures in **Figure 4** could also estimate survival differences.

Table 5 Cox Regression Results from UM Dataset, Risk Stratification by an 88-Gene Signature, Using Cutoff of 40th and 50th Percentile of Risk Scores from Training Data

No. Probes Cutoff Level	HR	95CI%	Log Rank Test P value
N=155			
40 th Percentile	1.73	(0.94,3.18)	0.0745
50 th Percentile	1.82	(1.09,3.03)	0.0188
N=176			
40 th Percentile	1.95	(1.07,3.55)	0.0254
50 th Percentile	2.09	(1.29,3.40)	0.00243

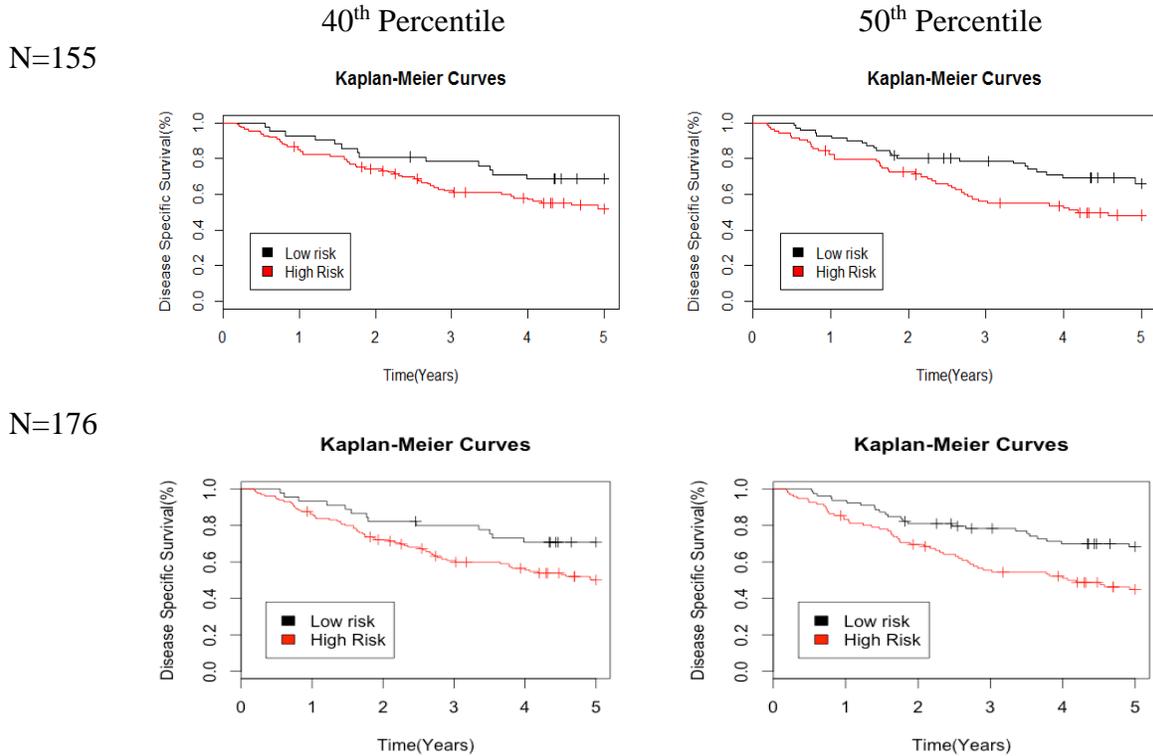


Figure 4 Kaplan Meier Curves of Survival for High-Risk and Low-Risk groups assigned by 88-Gene Internal validation accessing modeling predict accuracy

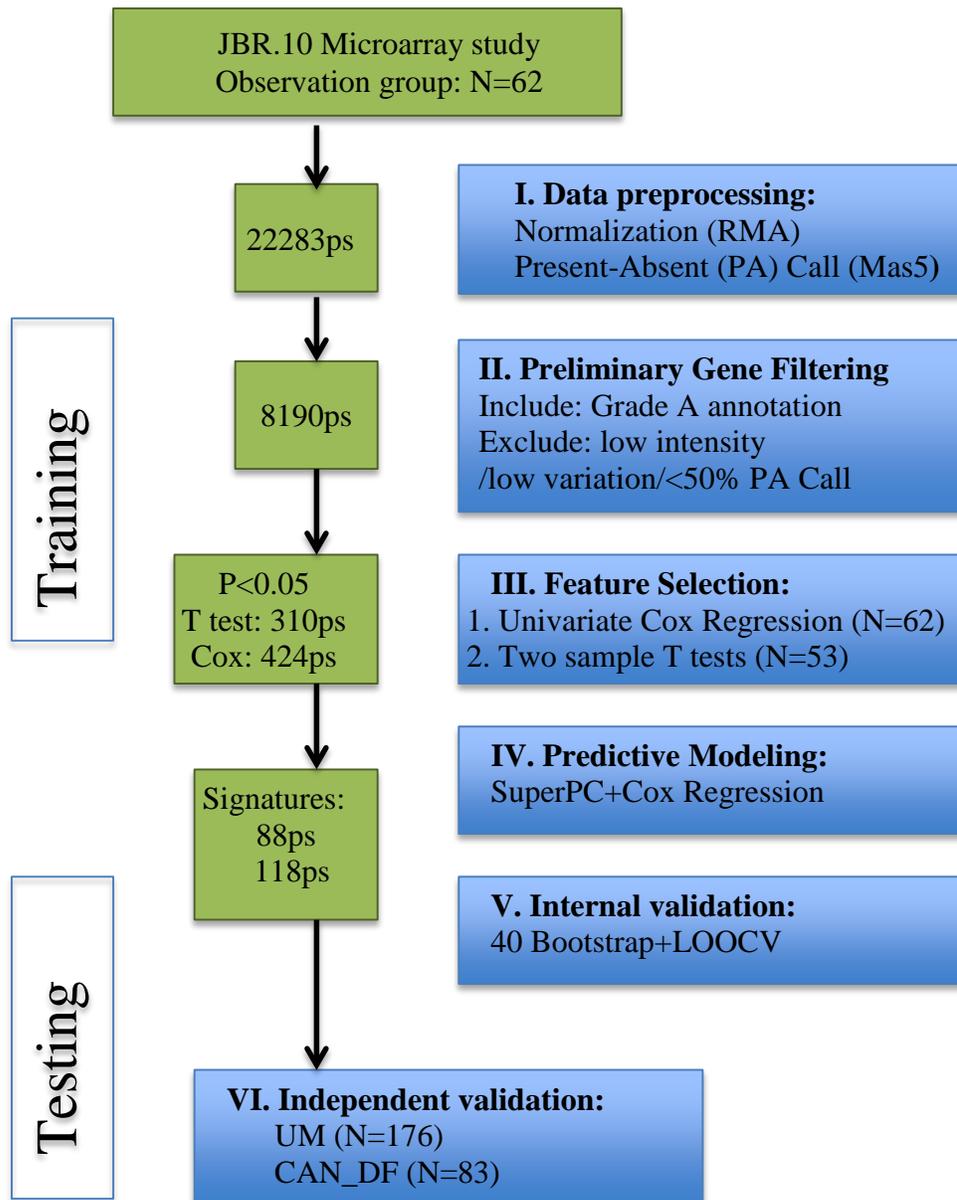


Figure 5 Prognostic Signature Development and Validation Flowchart

5.0 CONCLUSION AND DISCUSSION

A work flow of gene signature development composed of preliminary gene filtering, individual gene selection, predictive model construction using supervised principle component analysis and further internal/external validation, has been constructed. These steps ensured that the selected probes are of high qualities, in terms of variation and detectability, and differential expression among subjects, and also in terms of their association with survival time. Using the gene expression of 62 patients from the JBR.10 trial, an 88-gene signature was developed and then both validated internally and in 2 independent datasets. A 118-gene signature also worked successfully in predicting subjects (from training and testing datasets) into two risk groups with significantly different survival. Two cutoffs were considered, 50th percentile and 40th percentile of the risk scores. 40th percentile was considered because it is in line with the observed event rates in the 3 datasets. 50th percentile was considered because of the clinical reasons that a false negative is more costly than a false positive. There was no significant difference among different four models under Cox-alone criteria in terms of LOOCV. Taking potential additional contributions from T test into consideration, we compared hazard ratios from models with features selected by either Cox regression alone or Cox regression + T test. There was no clear evidence that Cox+ T test criteria would improve signature performance.

This study focused on implementing Principal Component Analysis into Cox regression to stratify patients into risk groups using their gene expression from Affymetrix array data. Other dimension reduction techniques, such as Partial Least Square and Support Vector Machine, could be used. It was our interest to explore the potential benefit of using T test in individual feature selection. However, T-test had its limitations in dealing with censored data. This thesis

project provided a workflow for gene signature development for risk stratification of various types of cancers that could be applied to microarrays. Further study on the candidate gene signatures could take into account other clinical covariates, such as cancer histology type, stage, sex and age. It is also of interest to assess the predictive value of the candidate gene signatures, which can be tested for whether certain chemotherapy (e.g. Adjuvant cisplatin/vinorelbine) could benefit early stage NSCLC patients, regarding their survival.

APPENDIX A: SUPPLEMENTAL RESULTS UNDER COX ALONE STRATEGY

Table 6 coefficient of each probes of 88-gene signature under 50th percentile of risk score as cutoff

#	Probe Set Name	Coefficient*	#	Probe Set Name	Coefficient*
1	217995_at	-0.0378	45	212527_at	-0.0193
2	203973_s_at	-0.0309	46	203939_at	-0.0009
3	212528_at	-0.0278	47	212124_at	-0.0364
4	203509_at	-0.0292	48	214853_s_at	-0.0363
5	208992_s_at	-0.0339	49	90610_at	-0.0211
6	221591_s_at	0.0328	50	212359_s_at	-0.0285
7	218768_at	0.0047	51	218358_at	-0.0206
8	201502_s_at	-0.0330	52	203273_s_at	0.0336
9	221718_s_at	-0.0378	53	38158_at	0.0294
10	202421_at	-0.0239	54	218880_at	-0.0265
11	201242_s_at	-0.0302	55	210247_at	0.0263
12	213923_at	-0.0357	56	208684_at	-0.0300
13	213603_s_at	-0.0332	57	200862_at	-0.0204
14	203028_s_at	-0.0220	58	204972_at	-0.0311
15	209536_s_at	-0.0354	59	200878_at	-0.0323
16	213260_at	0.0120	60	202891_at	-0.0186
17	202023_at	-0.0259	61	203637_s_at	-0.0230
18	214665_s_at	-0.0362	62	212531_at	-0.0211
19	212203_x_at	-0.0270	63	209546_s_at	-0.0241
20	203147_s_at	-0.0221	64	221495_s_at	-0.0302
21	212737_at	-0.0321	65	219363_s_at	0.0146
22	219259_at	-0.0222	66	218782_s_at	0.0234
23	204179_at	0.0180	67	219250_s_at	-0.0226
24	35820_at	-0.0283	68	211105_s_at	-0.0343
25	204521_at	0.0269	69	200086_s_at	-0.0151
26	213848_at	-0.0319	70	205241_at	-0.0234
27	217732_s_at	-0.0315	71	221269_s_at	-0.0307
28	222024_s_at	-0.0328	72	202679_at	-0.0321
29	219520_s_at	-0.0329	73	218231_at	-0.0290
30	200885_at	-0.0325	74	217767_at	-0.0279
31	202295_s_at	-0.0321	75	208690_s_at	-0.0345
32	208991_at	-0.0379	76	203140_at	-0.0354
33	203636_at	-0.0190	77	218574_s_at	-0.0285
34	205945_at	-0.0306	78	209605_at	-0.0260
35	210069_at	0.0195	79	205917_at	-0.0257
36	200918_s_at	-0.0254	80	211366_x_at	-0.0253
37	202814_s_at	-0.0069	81	203935_at	-0.0305
38	205189_s_at	0.0355	82	216565_x_at	-0.0228
39	201077_s_at	-0.0218	83	207843_x_at	-0.0196
40	201201_at	-0.0281	84	212420_at	-0.0195

Table 7 Cox regression on risk groups defined by a 118 gene signature using cutoffs of 40th and 50th percentile of risk score as cutoffs (training)

No. Probes Cutoff Level	HR	95CI%	Log-rank
118-Gene Sig			
40 th Percentile	7.19	(2.15,24.05)	1.88e-04
50 th Percentile	8.16	(2.8,23.75)	4.95e-06

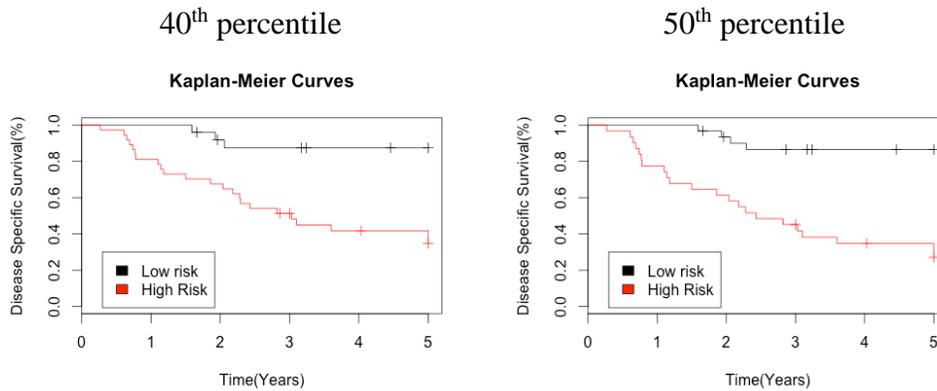


Figure 6 Kaplan Meier Curves of survival for high- and low-risk groups assigned by a 118-gene signature using cutoffs of 40th and 50th percentile of risk scores as cutoffs (training)

Table 8 Risk Stratification in CAN_DF and UM datasets Using a 88-Gene Signature

118	Event rate	40 th Percentile of Risk Score		50 th Percentile of Risk Score	
		High-risk	Low-risk	High-risk	Low-risk
CAN_DF (83)	29/83=34.9%	20	63	35	48
CAN_DF (59)	21/59=35.6%	14	45	27	32
UM (176)	75/176=42.6%	42	134	71	105
UM (155)	65/155=41.9%	37	118	64	91

Table 9 Cox regression on risk groups defined by a 118 gene signature using cutoffs of 40th and 50th percentile of risk score as cutoffs (CAN_DF)

No. Probes Cutoff Level	HR	95CI%	Log-rank
N=59			
118-Gene Sig			
40 th Percentile	1.54	(0.52,4.58)	0.437
50 th Percentile	2.75	(1.06,7.10)	0.029
N=83			
118-Gene Sig			
40 th Percentile	1.62	(0.62,4.27)	0.324
50 th Percentile	2.14	(0.94,4.86)	0.064

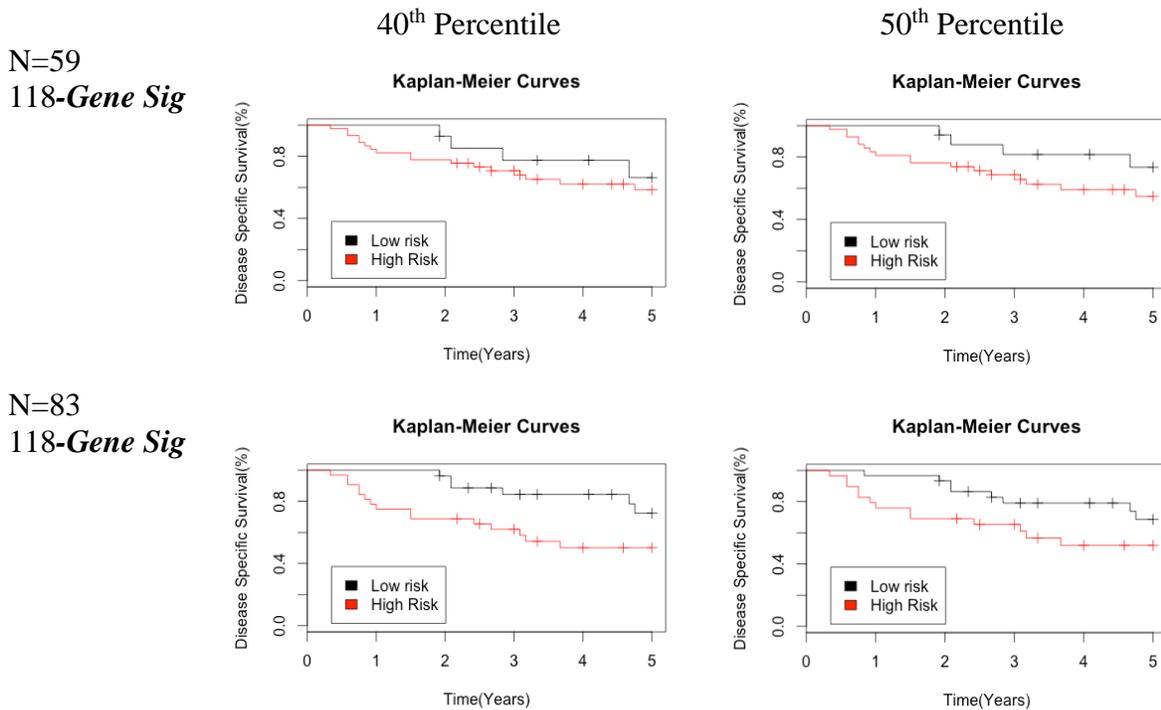


Figure 7 Kaplan Meier Curves of survival for high- and low-risk groups assigned by a 118-gene signature using cutoffs of 40th and 50th percentile of risk scores as cutoffs (CAN_DF)

Table 10 Cox regression on risk groups defined by a 118 gene signature using cutoffs of 40th and 50th percentile of risk score as cutoffs (UM)

No. Probes Cutoff Level	HR	95CI%	Log-rank
N=155			
118-Gene Sig			
40 th Percentile	1.36	(0.74,2.5)	0.324
50 th Percentile	1.87	(1.10,3.17)	0.018
N=176			
118-Gene Sig			
40 th Percentile	1.53	(0.86,2.73)	0.146
50 th Percentile	1.96	(1.19,3.22)	0.00675

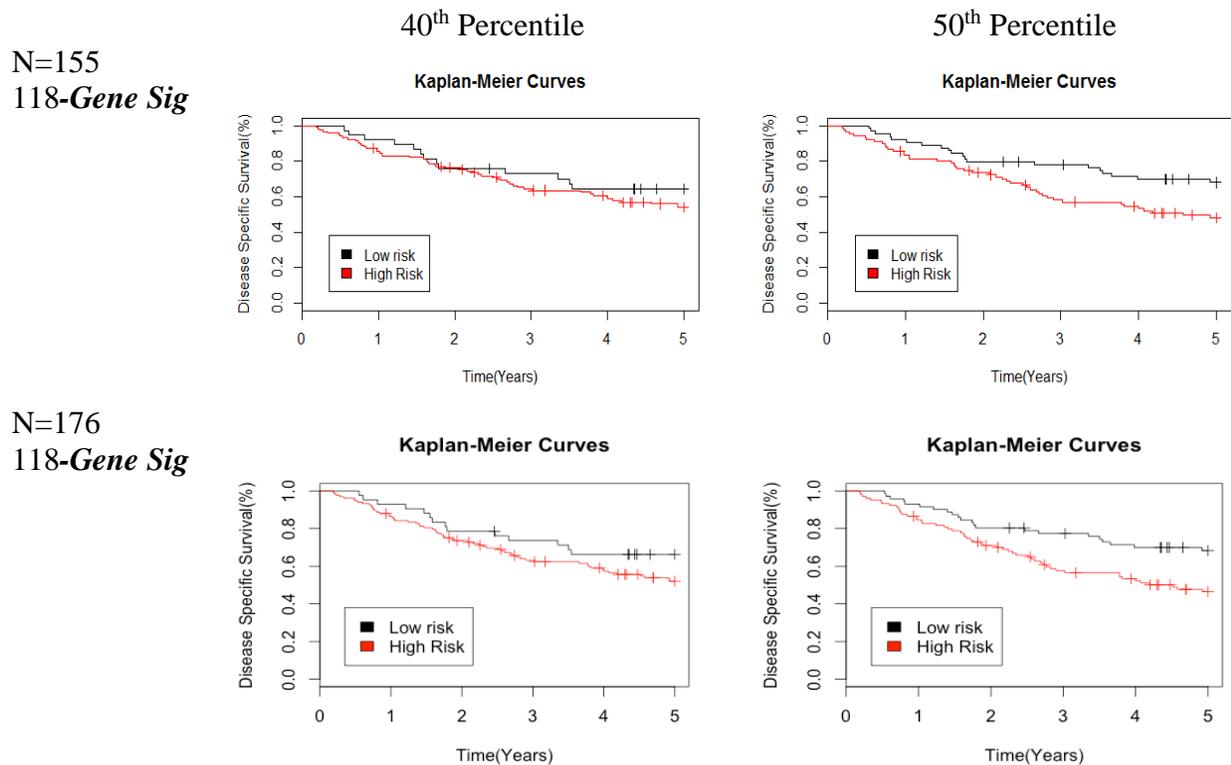


Figure 8 Kaplan Meier Curves of survival for high- and low-risk groups assigned by a 118-gene signature using cutoffs of 40th and 50th percentile of risk scores as cutoffs (UM)

APPENDIX B: TRAINING AND TESTING RESULTS UNDER COX+ T TEST STRATEGY

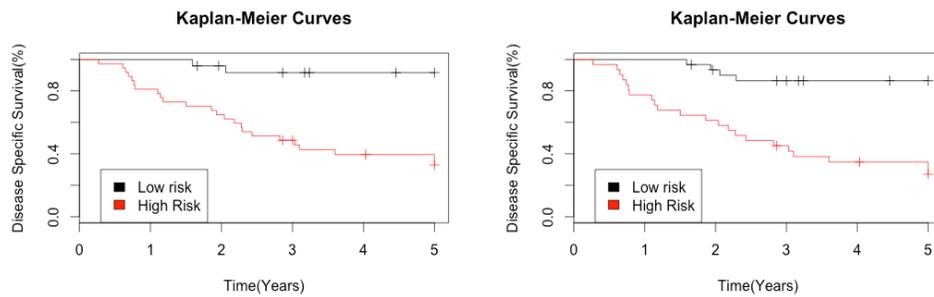
Table 11 Cox regression results on risk groups from training data under Cox+ t test scenario, risk stratification using 40th and 50th percentile of risk score as cutoffs (training)

No. Probes Cutoff Level	HR	95CI%	Log rank
<i>88-Gene Sig</i>			
40 th Percentile	11.6	(2.74,49.18)	2.76e-05
50 th Percentile	8.47	(2.90,24.75)	3.31e-06
<i>118-Gene Sig</i>			
40 th Percentile	11.6	(2.74,49.18)	2.76e-05
50 th Percentile	8.16	(2.80,23.75)	4.95e-06

88

40th Percentile

50th Percentile



118

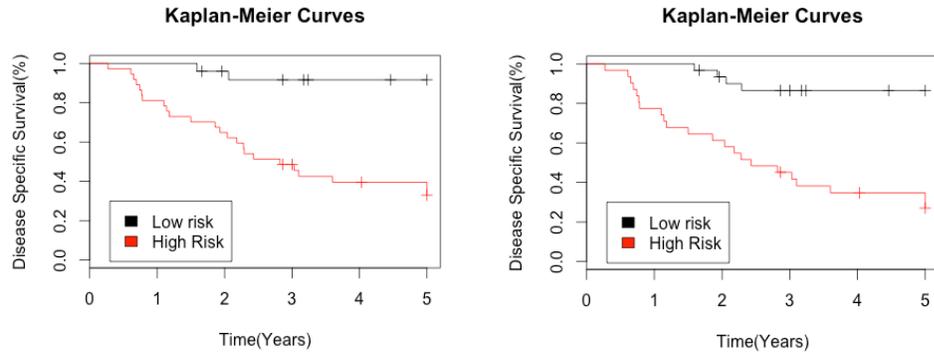


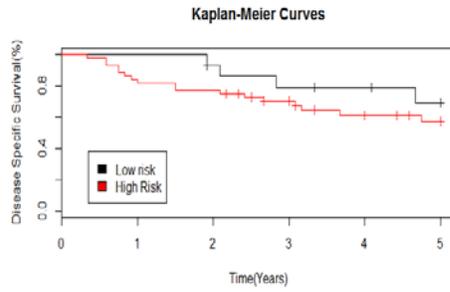
Figure 9 Kaplan Meier Curves of survival for high- and low-risk groups assigned by a 88-gene and 118-gene signature using cutoffs of 40th and 50th percentile of risk scores as cutoffs under Cox+T test strategy

Table 12 Cox regression results on risk groups from training data under Cox+ t test scenario, risk stratification using 40th and 50th percentile of risk score as cutoffs (CAN_DF)

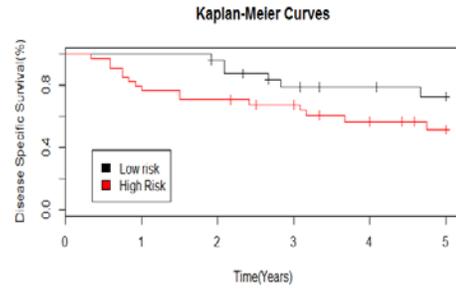
No. Probes	HR	95CI%	Log rank
Cutoff Level			
N=59			
88-Gene Sig			
40 th Percentile	1.73	(0.58,5.14)	0.319
50 th Percentile	2.27	(0.88,5.86)	0.0814
118-Gene Sig			
40 th Percentile	1.69	(0.5,5.74)	0.393
50 th Percentile	3.57	(1.2,10.64)	0.016
N=83			
88-Gene Sig			
40 th Percentile	1.42	(0.54,3.74)	0.481
50 th Percentile	1.76	(0.8,3.89)	0.159
118-Gene Sig			
40 th Percentile	1.54	(0.53,4.44)	0.42
50 th Percentile	1.89	(0.72,4.94)	0.124

N=59
88-*Gene Sig*

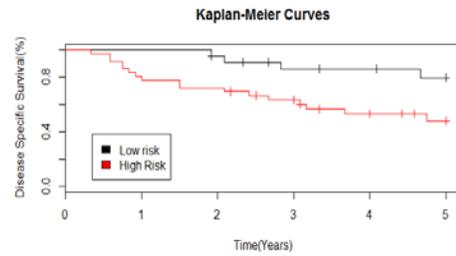
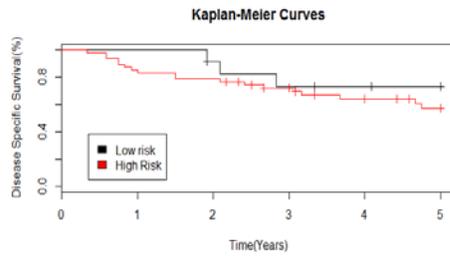
40th Percentile



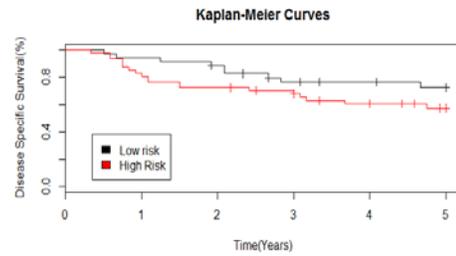
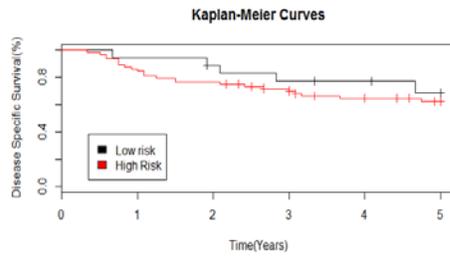
50th Percentile



118-*Gene Sig*



N=83
88-*Gene Sig*



118-*Gene Sig*

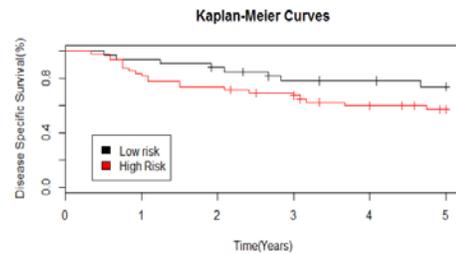
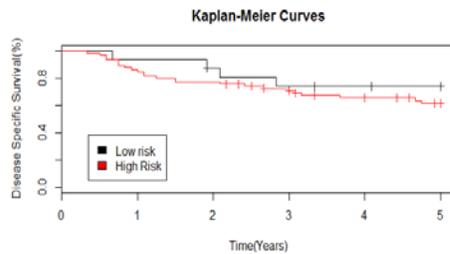


Figure 10 Kaplan Meier Curves of survival for high- and low-risk groups assigned by a 88-gene and 118-gene signature using cutoffs of 40th and 50th percentile of risk scores as cutoffs under Cox+T test strategy (CAN_DF)

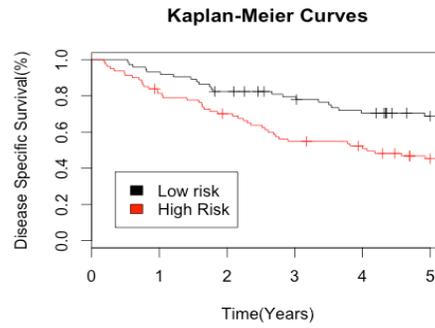
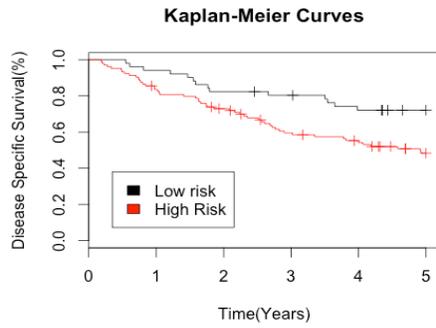
Table 13 Cox regression results on risk groups from training data under Cox+ t test scenario, risk stratification using 40th and 50th percentile of risk score as cutoffs (UM)

No. Probes	HR	95CI%	Log rank
Cutoff Level			
N=155			
88			
40 th Percentile	2.17	(1.2,3.92)	0.00853
50 th Percentile	2.14	(1.28,3.58)	0.00302
118			
40 th Percentile	1.98	(1.08,3.64)	0.0249
50 th Percentile	1.89	(1.14,3.15)	0.013
N=176			
88			
40 th Percentile	1.95	(1.07,3.55)	0.0254
50 th Percentile	2.09	(1.29,3.4)	0.00243
118			
40 th Percentile	1.53	(0.86,2.73)	0.146
50 th Percentile	1.96	(1.19,3.22)	0.00675

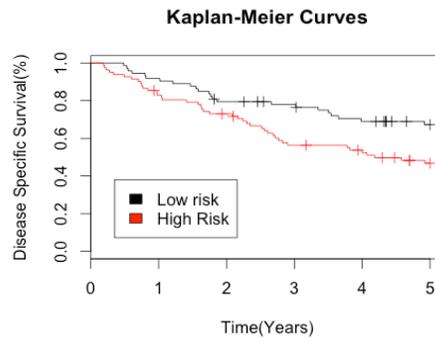
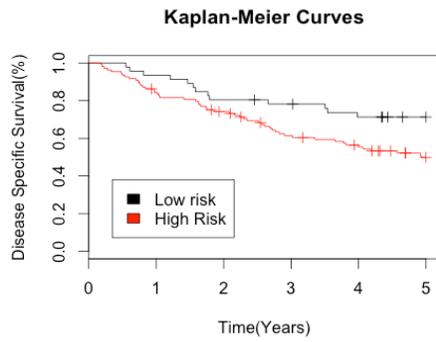
N=155
88

40th Percentile

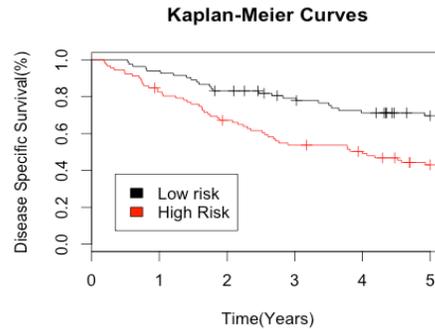
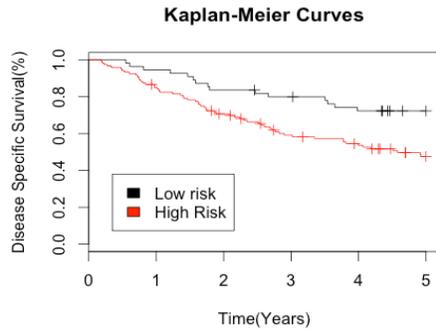
50th Percentile



118



N=176
88



118

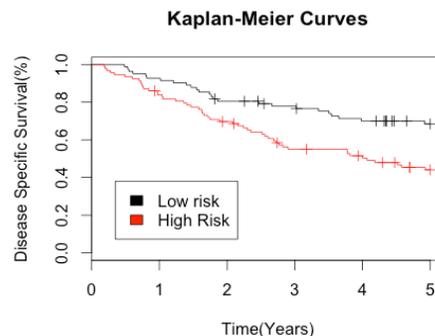
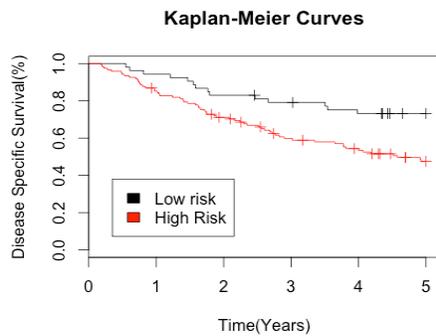


Figure 11 Kaplan Meier Curves of survival for high- and low-risk groups assigned by a 88-gene and 118-gene signature using cutoffs of 40th and 50th percentile of risk scores as cutoffs under Cox+T test strategy (UM)

APPENDIX C: R CODE

I. Data Preprocessing

```
GSE14814.cli=read.csv(paste(dir1,"GSE14814_clinical.csv", sep=""), row.names=1,check.names=F)
save(GSE14814.cli,file="GSE14814.cli.Rdata")
GSE14814.exp=(read.csv(paste(dir1,"GSE14814_expression.csv", sep=""), row.names=1,check.names=F))
save(GSE14814.exp,file="GSE14814.exp.Rdata")

dim(GSE14814.cli)#133:44
dim(GSE14814.exp)#22283

clinical.jbr10_133=GSE14814.cli[,c("characteristics_ch1.1","characteristics_ch1.2","characteristics_ch1.3",
    "characteristics_ch1.4","characteristics_ch1.5","characteristics_ch1.6",
    "characteristics_ch1.9","characteristics_ch1.10")]
colnames(clinical.jbr10_133)=c("trt","stage","age","sex","Cause","Histo","time","status")

#age
clinical.jbr10_133[, "age"]=as.numeric(substr(clinical.jbr10_133[, "age"],6,9))
clinical.jbr10_133=cbind(clinical.jbr10_133,ageI=clinical.jbr10_133[, "age"])
clinical.jbr10_133[, "ageI"]=as.factor(ifelse(clinical.jbr10_133[, "ageI"]<65,0,1))

#time
time=c(0)
for (i in 1:133){
  time[i]=as.numeric(strsplit(as.character(clinical.jbr10_133[i,"time"]),":")[1][2])
}
clinical.jbr10_133[, "time"]=time
clinical.jbr10_133=cbind(clinical.jbr10_133,time.5y=time)
clinical.jbr10_133[clinical.jbr10_133[, "time.5y"]>5,"time.5y"]=5

#sex
clinical.jbr10_133[, "sex"]=matrix(unlist(strsplit(as.vector(clinical.jbr10_133[, "sex"]), ":" , fixed = TRUE)),ncol=2,byrow=TRUE)[,2]
clinical.jbr10_133=cbind(clinical.jbr10_133,female=c(0))
clinical.jbr10_133[, "female"]=ifelse(clinical.jbr10_133[, "sex"]==" Female",1,0)
clinical.jbr10_133[, "sex"]=as.factor(clinical.jbr10_133[, "sex"])

#status
clinical.jbr10_133=cbind(clinical.jbr10_133,statusI=clinical.jbr10_133[, "status"])
clinical.jbr10_133[, "statusI"]=ifelse(clinical.jbr10_133[, "statusI"]=="DSS status: Alive",0,1)
save(clinical.jbr10_133,file="clinical.jbr10_133.Rdata")

summary(clinical.jbr10_133[,c("sex","stage","status","Histo","Cause","time","time.5y","age","ageI")])

#descriptive analysis
#if no treatment patients
clinical.jbr.untrt=subset(clinical.jbr10_133,clinical.jbr10_133[, "trt"]=="Post Surgical Treatment: OBS")#62
dim(clinical.jbr.untrt)#62:12
save(clinical.jbr.untrt,file="clinical.jbr.untrt.Rdata")

summary(clinical.jbr.untrt[,c("sex","stage","status","Histo","Cause","time","time.5y","age","ageI")])
evet_rate.jbr10=nrow(subset(clinical.jbr.untrt,! (clinical.jbr.untrt[, "status"]=="DSS status: Alive")))/nrow(clinical.jbr.untrt)
GSE14814.exp.untrt=GSE14814.exp[rownames(clinical.jbr.untrt),]
save(GSE14814.exp.untrt,file="GSE14814.exp.untrt.Rdata")
```

II. Preliminary Gene Filtering

```
#grade A
aa=read.delim("export.tsv", header = TRUE)
gradeA.probe=aa[[1]]#gene list in Grade A###22277

### mean
Mean.62=apply(jbr10.untrt.exp.gradeA,2,mean)
quantile(Mean.62,na.rm = TRUE)[2]#4.84
S1=names(Mean.62[Mean.62>quantile(Mean.62,na.rm = TRUE)[2]])#16723

### standard deviation
Sd.62=apply(jbr10.untrt.exp.gradeA,2,sd)
quantile(Sd.62,na.rm=TRUE)[2]#0.262
S2=names(Sd.62[Sd.62>quantile(Sd.62,na.rm=TRUE)[2]])#16723

name1=intersect(S1,S2)#14232
sum(is.na(name1))#1
```

```

name2=intersect(name1,gradeA.probe)
sum(is.na(name2))#0
S3exp=GSE14814.exp.untrt[,name2]#62:14231

#####pacall>50%#####
load("pacall133.Rdata")#pacall.exprs:22283:133

#dealing with name
colnames(pacall.exprs)[1:43]=substr(colnames(pacall.exprs)[1:43],1,10)
colnames(pacall.exprs)[44:133]=substr(colnames(pacall.exprs)[44:133],1,9)

PaCall62=pacall.exprs[colnames(S3exp),rownames(S3exp)]#14231:62

prop62=c(rep(0,nrow(PaCall62)))
for (i in 1:nrow(PaCall62)){
prop62[i]=sum(PaCall62[i,]== "P")/ncol(PaCall62)
}
names(prop62)=rownames(PaCall62)

#>50%
Pcut=0.5
sum(prop62>=Pcut)#8190
S4exp=S3exp[,names(prop62[prop62>=Pcut])]#62:8910
save(S4exp,file="S4exp.Rdata")

III. Feature selection
time.jbr=clinical.jbr.untrt[, "time.5y"]
names(time.jbr)=rownames(clinical.jbr.untrt)
event.jbr=clinical.jbr.untrt[, "status1"]
names(event.jbr)=rownames(clinical.jbr.untrt)
##univariate Cox regression
Cox_filter=function(data){
summary(coxph(Surv(time.jbr,event.jbr)~ data))$coef[, "Pr(>|z|)"]
}
cox_probe=apply(data, 2, Cox_filter)
names(cox_probe)=colnames(data)
save(cox_probe,file="cox_probe.Rdata")

o.cox.p=cox_probe[order(cox_probe),drop = FALSE]
o.cox.na=names(o.cox.p)
opv.fdr.unicox=data.frame(gene=o.cox.na, cox.pv=o.cox.p)
save(uni.cox,file="uni.cox.Rdata")
load("uni.cox.Rdata")

#### t-test--only 53
##I.(2-sample, equal variance)
sub1=subset(clinical.jbr.untrt,clinical.jbr.untrt[, "time.5y"]<5,select=c("time.5y", "status1"))
sub2=subset(sub1,sub1[, "status1"]==0)
exclude=rownames(sub2)
save(exclude,file="exclude.Rdata")
load("exclude.Rdata")

eventI=event.jbr
eventI=subset(eventI,!names(eventI)%in%exclude)

event0=eventI[eventI==0]#27
event1=eventI[eventI==1]#26

#test on equal variance
var.test(data[names(event0),],data[names(event1),])
#con: equal variance

ttest.P=c(0)
for (i in 1:ncol(data)){
ttest=t.test(data[names(event0),i], data[names(event1),i],
alternative =c("two.sided", "less", "greater"),
mu = 0, paired = FALSE, var.equal = TRUE,
conf.level = 0.95)
ttest.P[i]=ttest$p.value
}
names(ttest.P)=colnames(data)
save(ttest.P,file="ttest.P.Rdata")

fdr.ttest.P=subset(ttest.P,ttest.P<0.01)#48
fdr.ttest.P2=subset(ttest.P,ttest.P<0.05)#310

opv.fdr.unicox=fdr.unicox[order(fdr.unicox[, "pv"]),drop=FALSE]
o.ttestP=ttest.P[order(ttest.P),drop = FALSE]
save(o.ttestP,file="o.ttestP.Rdata")

```

```

sig.list_coxttest=function(ncox,nttest)
{
  list_u=union(cox.Na[1:ncox],ttest.Na[1:nttest])
  list_ucox=setdiff(cox.Na[1:ncox],ttest.Na[1:nttest])
  list_ut=setdiff(ttest.Na[1:nttest],cox.Na[1:ncox])
  list_inter=intersect(ttest.Na[1:nttest],cox.Na[1:ncox])
  return(list(list_u,list_ucox,list_ut,list_inter))
}

### T test for extreme observation
## rank survival time(concerning the original), pick top10 from the top/botton
##
time.ori=clinical.jbr.untrt[, "time"]
names(time.ori)=rownames(clinical.jbr.untrt)
time.order=names(time.ori[order(time.ori)])
grpS=time.order[1:10]
grpL=time.order[53:62]
var.test(data[grpS,],data[grpL,])

# ttest in extreme
extr.ttest=c(0)
for (i in 1:ncol(data)){
  extr.ttest[i]=t.test(data[grpS,i], data[grpL,i],
    alternative =c("two.sided", "less", "greater"),
    mu = 0, paired = FALSE, var.equal = TRUE,
    conf.level = 0.95)$p.value
}
names(extr.ttest)=colnames(data)

rank_compare=function(n,data,cox_probe,ttest.P){
  pv=data.frame(gene=colnames(data),coxP=cox_probe,ttest.P=ttest.P)
  sort1=pv[order(pv$coxP),]
  S200_cox=cbind(sort1[1:n,],rank.C=c(1:n))
  sort2=S200_cox[order(S200_cox$ttest.P),]
  S200_ttest=cbind(sort2,rank.T=c(1:n))
  plot(S200_ttest$rank.C,S200_ttest$rank.T,xlab="Ranks from Cox Model",
    ylab="Ranks from T Test",main="Rank Comparisons")
}
save(rank_compare,file="rank_compare.Rdata")

#2. cox and ttest in extreme observation
plot_10=rank_compare(10,data,cox_probe,extr.ttest)
plot_100=rank_compare(100,data,cox_probe,extr.ttest)
plot_500=rank_compare(500,data,cox_probe,extr.ttest)

#1.cox and ttest
plot_10=rank_compare(10,data,cox_probe,ttest.P)
plot_100=rank_compare(100,data,cox_probe,ttest.P)
plot_500=rank_compare(500,data,cox_probe,ttest.P)

```

IV. Supervised Principal Component-Cox Regression Model

```

findcoeff.new=function(x,list,time,event,nPCA,P){
  ##I. x manipulate into a n by p matrix, where n is #. of patients, P is #. of probes
  ##1. x is firstly Z score transformed
  X.S=scale(x,T,T)
  ##2. gene expression of gene list
  X.sig=X.S[,list]
  ##
  ##II. PCA
  pca=prcomp(X.sig,retx=T, center=T, scale=T)
  eigen=(pca$sdev^2)#find components that are >1
  #at this stage we choose 6
  #we can choose non-trivial components
  ###
  #nCom=6
  #PC=list(c(0))
  #for (i in 1:nCom){
  #PC[i]=pca$x[,i]
  #}
  #####
  pc1=pca$x[,1]
  pc2=pca$x[,2]
  pc3=pca$x[,3]
  pc4=pca$x[,4]
  pc5=pca$x[,5]
  pc6=pca$x[,6]
  #calculate coefficient from cox regression
  pcr=coxph(Surv(time,event)~pc1+pc2+pc3+pc4+pc5+pc6)
  coeff.pcr=pcr[[1]]
}

```

```

###
loading.pc=pca$rotation[,1:nPCA]
##risk score derived from PCA
RS.pca=c(0)
for (i in 1:nrow(X.sig)){
  RS.pca[i]= coeff.pc[1]*X.sig[i,]%%loading.pc[, 'PC1']
  + coeff.pc[2]*X.sig[i,]%%loading.pc[, 'PC2']
  + coeff.pc[3]*X.sig[i,]%%loading.pc[, 'PC3']
  + coeff.pc[4]*X.sig[i,]%%loading.pc[, 'PC4']
  + coeff.pc[5]*X.sig[i,]%%loading.pc[, 'PC5']
  + coeff.pc[6]*X.sig[i,]%%loading.pc[, 'PC6']
}
names(RS.pca)=rownames(X.sig)
cutoff=median(RS.pca)
grp=ifelse(RS.pca>=cutoff, 1, 0)
### survival for high-low risk group
surv=coxph(Surv(time,event)~grp)
plot(survfit(Surv(time,event)~grp),xlab="Time(Years)", ylab="Disease Specific Survival(%)",col=c("black","red"))
title("Kaplan-Meier Curves")
legend(0.3,0.3,c("Low risk", "High Risk"),c("black","red"))
### coeff for every probe
coeff.probe=c(0)
for (i in 1:nrow(loading.pc)){
  coeff.probe[i]=coeff.pc[1]*loading.pc[i,'PC1']
  +coeff.pc[2]*loading.pc[i,'PC2']
  +coeff.pc[3]*loading.pc[i,'PC3']
  +coeff.pc[4]*loading.pc[i,'PC4']
  +coeff.pc[5]*loading.pc[i,'PC5']
  +coeff.pc[6]*loading.pc[i,'PC6']
}
names(coeff.probe)=rownames(loading.pc)

RS.train=c(0)
for (i in 1 : nrow(X.sig))
{
  RS.train[i]=X.sig[i,] %% coeff.probe
}

names(RS.train)=rownames(X.sig)
hist(RS.train)
cut.train=quantile(RS.train,P)
grp.train=ifelse(RS.train>=cut.train, 1, 0)
cox.train=coxph(Surv(time,event)~grp.train)
logrank.train=survdiff(Surv(time,event) ~ grp.train)
plot.train=plot(survfit(Surv(time,event)~grp.train),xlab="Time(Years)", ylab="Disease Specific Survival(%)",col=c("black","red"))
title("Kaplan-Meier Curves")
legend(0.3,0.3,c("Low risk", "High Risk"),c("black","red"))
return(list(grp.train,coeff.probe,cut.train,cox.train,logrank.train))
}

```

V. Internal Validation (for cox alone)

```

LeaveOneOut=function(x){
  # x is n x p
  #I. subsamples
  s.exp=scale(x,T,T)
  ## II. generate subset sample list for loocv (62 subsamples with 61 patients for each)
  in.list=matrix(c(0),nrow(s.exp),nrow(s.exp)-1)#62 x 61
  for (i in 1:nrow(s.exp)){
    in.list[i,]=rownames(s.exp)[-i]
  }
  rownames(in.list)=c(1:nrow(s.exp))
  rownames(in.list)=paste("cv",rownames(in.list),sep="")

  out.list=as.matrix(c(rownames(s.exp)),npatient,1)
  rownames(out.list)=c(1:nrow(s.exp))
  rownames(out.list)=paste("cv",rownames(out.list),sep="")

  ## III. get the gene expression data/survival data for each subset(data frame as a whole)
  in.exp=data.frame(matrix(c(0),ncol(in.list),ncol(s.exp))#62 matrix:61 X P
  out.exp=matrix(c(0),nrow(s.exp),ncol(s.exp))#62 x p
  for (i in 1:nrow(s.exp)){
    in.exp[[i]]=s.exp[in.list[i,],]
    colnames(in.exp[[i]])=colnames(s.exp)
    out.exp[i,]=s.exp[out.list[i,],]
    rownames(out.exp)=as.vector(out.list)
    colnames(out.exp)=colnames(s.exp)
  }

  return(list(in.exp,out.exp))
}

```

```

}

#try on the 1st boot--62 loocv, calculate accuracy
boot_loocv_acc=function(boot.exp,i,nmatrix,npatient,oriProbe,nProbe,exclude){
  #for one boot
  loocv1.exprs=LeaveOneOut(boot.exp,i)
  loocv1.exp.61=loocv1.exprs[[1]]#data frame, 62 matrices, each 61 X 8910
  loocv1.exp.1=loocv1.exprs[[2]]#matrix

  time.b1=list(c(0))#62 list, each is a 61 length vector
  event.b1=list(c(0))
  coxP.b1=list(c(0))

  for(i in 1:nmatrix){
    time.b1[[i]]=time.jbr[rownames(loocv1.exp.61[[i]])]
    event.b1[[i]]=event.jbr[rownames(loocv1.exp.61[[i]])]
  }
  ##
  coxP.b1=list(list())
  for(i in 1:nmatrix){
    #coxP.b1[[i]]=apply(loocv1.exp.61[[i]],2,Cox_filter,time=time.b1[[i]],event=event.b1[[i]])
    coxP.b1[[i]]=apply(loocv1.exp.61[[i]],2,function(x) summary(coxph(Surv(time.b1[[i]],event.b1[[i]]~x)$coef[,"Pr(>|z|)"])
    cat("i=")
    cat(i)
    cat("\n")
  }
  ###
  coxP.b1<- matrix(unlist(coxP.b1), nrow=nmatrix,ncol=oriProbe, byrow = TRUE)
  rownames(coxP.b1)=c(1:nmatrix)
  rownames(coxP.b1)=paste("Loocv1_",rownames(coxP.b1),sep="")
  colnames(coxP.b1)=colnames(data)

  ###
  O.coxP.b1=list(c(0))
  lista88.b1=list(c(0))
  for(i in 1:nmatrix){
    O.coxP.b1[[i]]=coxP.b1[i,][order(coxP.b1[i,]),drop = FALSE]
    #pick top88
    lista88.b1[[i]]=O.coxP.b1[[i]][1:nProbe]
  }

  #SuperPC
  grp_b1_loocv=list(c(0))
  for(i in 1:nmatrix){
    grp_b1_loocv[[i]]=findcoeff.new(loocv1.exp.61[[i]],names(lista88.b1[[i]]),time.b1[[i]],event.b1[[i]],nPCA=6,P=0.5)
    #output:(list(grp.train,coeff.probe,cut.train,surv.train))
  }

  # risk stratification on testing # 1st cv
  loocv1.testexp=matrix(c(0),nmatrix,nProbe)
  loocv1.testRS=matrix(c(0),nmatrix,1)
  loocv1.testgrp=c(0)
  for (i in 1:nmatrix){
    loocv1.testexp[i,]=loocv1.exp.1[i,names(lista88.b1[[i]])]
    loocv1.testRS[i,]=matrix(c(loocv1.testexp[i,]),1,nProbe)%*%as.matrix(c(grp_b1_loocv[[i]][[2]]),nProbe,1)
    rownames(loocv1.testRS)=rownames(loocv1.exp.1)
    loocv1.testgrp[i]=ifelse(loocv1.testRS[i,]>=grp_b1_loocv[[i]][[3]],1,0)
  }
  names(loocv1.testgrp)=rownames(loocv1.exp.1)

  # accuracy
  event.loocv1=event.jbr[names(loocv1.testgrp)]
  loocv1.test_hit=c(0)
  irre=c(0)
  for(i in 1:nmatrix){
    if (names(loocv1.testgrp)[i] %in% exclude){
      loocv1.test_hit[i]="NA"
      irre[i]=1
    }
    else{
      loocv1.test_hit[i]=ifelse(loocv1.testgrp[i]==event.loocv1[i],1,0)
      irre[i]=0
    }
  }
}
#names(loocv1.test_hit)=c(1:nrow(exp.sig.ori))
#names(loocv1.test_hit)=paste("cv",names(loocv1.test_hit),sep="")
#names(loocv1.test_hit)=names(loocv1.testgrp)
loocv1_accu=sum(loocv1.test_hit,na.rm = T)/(nmatrix-sum(irre))
return(loocv1_accu)
}

```

```

acc=c(0)
nboot=40
for (j in 1:nboot){
  acc[j]=boot_loocv_acc(boot_62exp[[j]],oriProbe=8190,nmatrix=62,npatient=61,nProbe=88,exclude)
  cat("j=")
  cat(j)
  cat("\n")
}

#optimization concern for cox alone
#select top6 to top200 probe in cox test to find the one generate most largest HR
#at 0.5
hr.supPC50=c(0)
nPCA=6
for(i in nPCA:200){
  hr.supPC50[i]=exp(findcoeff.new(GSE14814.exp.untrt,cox.Na[1:i],time.jbr,event.jbr,nPCA,0.5)[[4]]$coef)
}
save(hr.supPC50,file="hr.supPC50.Rdata")
load("hr.supPC50.Rdata")
plot(c(6:200),hr.supPC50[6:200],xlab="#.of probes",ylab="Hazard Ratio", main="Probe Sets Optimization Using HR")

#at 0.4
hr.supPC40=c(0)
nPCA=6
for(i in nPCA:200){
  hr.supPC40[i]=exp(findcoeff.new(GSE14814.exp.untrt,cox.Na[1:i],time.jbr,event.jbr,nPCA,0.4)[[4]]$coef)
}
save(hr.supPC40,file="hr.supPC40.Rdata")
load("hr.supPC40.Rdata")
plot(c(6:200),hr.supPC40[6:200],xlab="#.of probes",ylab="Hazard Ratio", main="Probe Sets Optimization Using HR")

#####88a
rs.lista88_40=findcoeff.new(GSE14814.exp.untrt,lista88,time.jbr,event.jbr,nPCA=6,0.4)
rs.lista88_50=findcoeff.new(GSE14814.exp.untrt,lista88,time.jbr,event.jbr,nPCA=6,0.5)

#####88b
rs.listb88_40=findcoeff.new(GSE14814.exp.untrt,listb88[[1]],time.jbr,event.jbr,nPCA=6,0.4)
rs.listb88_50=findcoeff.new(GSE14814.exp.untrt,listb88[[1]],time.jbr,event.jbr,nPCA=6,0.5)

#####118a
rs.lista118_40=findcoeff.new(GSE14814.exp.untrt,lista118,time.jbr,event.jbr,nPCA=6,0.4)
rs.lista118_50=findcoeff.new(GSE14814.exp.untrt,lista118,time.jbr,event.jbr,nPCA=6,0.5)

#####118b
rs.listb118_40=findcoeff.new(GSE14814.exp.untrt,listb118[[1]],time.jbr,event.jbr,nPCA=6,0.4)
rs.listb118_50=findcoeff.new(GSE14814.exp.untrt,listb118[[1]],time.jbr,event.jbr,nPCA=6,0.5)

```

VI. External Validation

```

test.RS=function(test,list,time.t,event.t,coeff.probe,cut.train)
{
  ##test:n x p
  test.S=scale(test,T,T)
  test.sig=test.S[,list]
  RS.test=c(0)
  for (i in 1 : nrow(test.sig))
  {
    RS.test[i]=test.sig[i,] %>% coeff.probe
  }
  names(RS.test)=rownames(test.sig)
  hist(RS.test)
  grp.test=ifelse(RS.test>=cut.train,1,0)
  cox.test=coxph(Surv(time.t,event.t)~grp.test)
  logrank.test=survdiff(Surv(time.t,event.t) ~ grp.test)
  plot.test=plot(survfit(Surv(time.t,event.t)~grp.test),xlab="Time(Years)", ylab="Disease Specific Survival(%)",col=c("black","red"))
  title("Kaplan-Meier Curves")
  legend(0.35,0.38,c("Low risk", "High Risk"),c("black", "red"))
  return(list(RS.test,grp.test,cox.test,logrank.test))
}

###CAN_DF
CAN_DF83=t(read.csv(paste(dir1,"data.CAN_DF(83).csv", sep=""), row.names=1,check.names=F))
save(CAN_DF83,file="CAN_DF83.Rdata")#83:22296
load("CAN_DF83.Rdata")
CAN_DF83_2=CAN_DF83
#dds time(censored)
CAN_DF83_2[as.numeric(CAN_DF83_2[, "overall_survival_months"])>60,"death"]="Alive"
CAN_DF83_2[as.numeric(CAN_DF83_2[, "overall_survival_months"])>60,"overall_survival_months"]="60"
CAN_DF83_2[, "death"]=ifelse(CAN_DF83_2[, "death"]=="Alive", 0,1)

```

```

CAN_DF83_2[, "had_adjuvant_chemo"] = ifelse(CAN_DF83_2[, "had_adjuvant_chemo"] == "FALSE", 0, 1)
CAN_DF.sub = CAN_DF83_2[, c("histology", "had_adjuvant_chemo", "death", "overall_survival_months",
    "age", "gender", "stage.title")]
rownames(CAN_DF.sub) = rownames(CAN_DF83_2)
ageI = ifelse(as.numeric(CAN_DF.sub[, "age"]) > 65, 1, 0) # one age is missing
names(ageI) = rownames(CAN_DF.sub)
CAN_DF.sub = cbind(CAN_DF.sub, ageI)

# use the whole dataset (trt + untrt)
CAN_DF.exp.w = apply(CAN_DF83_2[, 14:22296], 2, as.numeric) # 83:22283
rownames(CAN_DF.exp.w) = rownames(CAN_DF83_2)

time.CAN_DF.w = as.numeric(CAN_DF.sub[, 4]) / 12
names(time.CAN_DF.w) = rownames(CAN_DF.sub)
event.CAN_DF.w = as.numeric(CAN_DF.sub[, 3])
names(event.CAN_DF.w) = rownames(CAN_DF.sub)

### cox
CAN83.a88_40 = test.RS(CAN_DF.exp.w, lista88, time.CAN_DF.w, event.CAN_DF.w, rs.lista88_40[[2]], rs.lista88_40[[3]])
CAN83.a88_50 = test.RS(CAN_DF.exp.w, lista88, time.CAN_DF.w, event.CAN_DF.w, rs.lista88_50[[2]], rs.lista88_50[[3]])
CAN83.a118_40 = test.RS(CAN_DF.exp.w, lista118, time.CAN_DF.w, event.CAN_DF.w, rs.lista118_40[[2]], rs.lista118_40[[3]])
CAN83.a118_50 = test.RS(CAN_DF.exp.w, lista118, time.CAN_DF.w, event.CAN_DF.w, rs.lista118_50[[2]], rs.lista118_50[[3]])
### cox + t test
CAN83.b88_40 = test.RS(CAN_DF.exp.w, listb88[[1]], time.CAN_DF.w, event.CAN_DF.w, rs.lista88_40[[2]], rs.lista88_40[[3]])
CAN83.b88_50 = test.RS(CAN_DF.exp.w, listb88[[1]], time.CAN_DF.w, event.CAN_DF.w, rs.lista88_50[[2]], rs.lista88_50[[3]])
CAN83.b118_40 = test.RS(CAN_DF.exp.w, listb118[[1]], time.CAN_DF.w, event.CAN_DF.w, rs.lista118_40[[2]], rs.lista118_40[[3]])
CAN83.b118_50 = test.RS(CAN_DF.exp.w, listb118[[1]], time.CAN_DF.w, event.CAN_DF.w, rs.lista118_50[[2]], rs.lista118_50[[3]])

# use only urt
CAN_DF.cli.untrt = subset(CAN_DF.sub, CAN_DF.sub[, 2] == "0") # 59
CAN_DF.exp.untrt = apply(CAN_DF83_2[rownames(CAN_DF.cli.untrt), 14:22296], 2, as.numeric) # 41:22283
rownames(CAN_DF.exp.untrt) = rownames(CAN_DF.cli.untrt)

time.CAN_DF.untrt = as.numeric(CAN_DF.cli.untrt[, 4]) / 12
names(time.CAN_DF.untrt) = rownames(CAN_DF.cli.untrt)
event.CAN_DF.untrt = as.numeric(CAN_DF.cli.untrt[, 3])
names(event.CAN_DF.untrt) = rownames(CAN_DF.cli.untrt)

CAN59.a88_40 = test.RS(CAN_DF.exp.untrt, lista88, time.CAN_DF.untrt, event.CAN_DF.untrt, rs.lista88_40[[2]], rs.lista88_40[[3]])
CAN59.a88_50 = test.RS(CAN_DF.exp.untrt, lista88, time.CAN_DF.untrt, event.CAN_DF.untrt, rs.lista88_50[[2]], rs.lista88_50[[3]])
CAN59.a118_40 = test.RS(CAN_DF.exp.untrt, lista118, time.CAN_DF.untrt, event.CAN_DF.untrt, rs.lista118_40[[2]], rs.lista118_40[[3]])
CAN59.a118_50 = test.RS(CAN_DF.exp.untrt, lista118, time.CAN_DF.untrt, event.CAN_DF.untrt, rs.lista118_50[[2]], rs.lista118_50[[3]])

CAN59.b88_40 = test.RS(CAN_DF.exp.untrt, listb88[[1]], time.CAN_DF.untrt, event.CAN_DF.untrt, rs.lista88_40[[2]], rs.lista88_40[[3]])
CAN59.b88_50 = test.RS(CAN_DF.exp.untrt, listb88[[1]], time.CAN_DF.untrt, event.CAN_DF.untrt, rs.lista88_50[[2]], rs.lista88_50[[3]])
CAN59.b118_40 = test.RS(CAN_DF.exp.untrt, listb118[[1]], time.CAN_DF.untrt, event.CAN_DF.untrt, rs.lista118_40[[2]], rs.lista118_40[[3]])
CAN59.b118_50 = test.RS(CAN_DF.exp.untrt, listb118[[1]], time.CAN_DF.untrt, event.CAN_DF.untrt, rs.lista118_50[[2]], rs.lista118_50[[3]])

# UM
# UM176 = t(read.csv(paste(dir1, "data.UM(176).csv", sep=""), row.names=1, check.names=F))
# save(UM176, file="UM176.Rdata") # 176:22296
load("UM176.Rdata")
# UM176[1:4, 1:13]
UM176_2 = UM176

UM176_2[, as.numeric(UM176_2[, "overall_survival_months"]) > 60, "death"] = "Alive"
UM176_2[, as.numeric(UM176_2[, "overall_survival_months"]) > 60, "overall_survival_months"] = "60"

UM176_2[, "death"] = ifelse(UM176_2[, "death"] == "Alive", 0, 1)
UM176_2[, "had_adjuvant_chemo"] = ifelse(UM176_2[, "had_adjuvant_chemo"] == "FALSE", 0, 1)

UM176.sub = UM176_2[, c("histology", "had_adjuvant_chemo", "death", "overall_survival_months", "age", "gender", "stage.title")]
rownames(UM176.sub) = rownames(UM176_2)

### using the whole sample
# um
save(UM176.sub, file="UM176.sub.Rdata")
load("UM176.sub.Rdata")

UM176.sub[1:4, ]
time.um.w = as.numeric(UM176.sub[, "overall_survival_months"]) / 12
names(time.um.w) = rownames(UM176.sub)
event.um.w = as.numeric(UM176.sub[, "death"])
names(event.um.w) = rownames(UM176.sub)

UM.exp.w = apply(UM176[, 14:22296], 2, as.numeric)
rownames(UM.exp.w) = rownames(UM176)

UM176.a88_40 = test.RS(UM.exp.w, lista88, time.um.w, event.um.w, rs.lista88_40[[2]], rs.lista88_40[[3]])
UM176.a88_50 = test.RS(UM.exp.w, lista88, time.um.w, event.um.w, rs.lista88_50[[2]], rs.lista88_50[[3]])

```

```

UM176.a118_40=test.RS(UM.exp.w,lista118,time.um.w,event.um.w,rs.lista118_40[[2]],rs.lista118_40[[3]])
UM176.a118_50=test.RS(UM.exp.w,lista118,time.um.w,event.um.w,rs.lista118_50[[2]],rs.lista118_50[[3]])

UM176.b88_40=test.RS(UM.exp.w,listb88[[1]],time.um.w,event.um.w,rs.listb88_40[[2]],rs.listb88_40[[3]])
UM176.b88_50=test.RS(UM.exp.w,listb88[[1]],time.um.w,event.um.w,rs.listb88_50[[2]],rs.listb88_50[[3]])
UM176.b118_40=test.RS(UM.exp.w,listb118[[1]],time.um.w,event.um.w,rs.listb118_40[[2]],rs.listb118_40[[3]])
UM176.b118_50=test.RS(UM.exp.w,listb118[[1]],time.um.w,event.um.w,rs.listb118_50[[2]],rs.listb118_50[[3]])

##using only untrt
UM.cli.nochemo=subset(UM176.sub,as.numeric(UM176.sub[,2])==0)#155:7
time.um.untrt=as.numeric(UM.cli.nochemo[, "overall_survival_months"])/12
names(time.um.untrt)=rownames(UM.cli.nochemo)
event.um.untrt=as.numeric(UM.cli.nochemo[, "death"])
names(event.um.untrt)=rownames(UM.cli.nochemo)
UM.exp.untrt=apply(UM176[rownames(UM.cli.nochemo),14:22296],2,as.numeric)
rownames(UM.exp.untrt)=rownames(UM.cli.nochemo)

UM155.a88_40=test.RS(UM.exp.untrt,lista88,time.um.untrt,event.um.untrt,rs.lista88_40[[2]],rs.lista88_40[[3]])
UM155.a88_50=test.RS(UM.exp.untrt,lista88,time.um.untrt,event.um.untrt,rs.lista88_50[[2]],rs.lista88_50[[3]])
UM155.a118_40=test.RS(UM.exp.untrt,lista118,time.um.untrt,event.um.untrt,rs.lista118_40[[2]],rs.lista118_40[[3]])
UM155.a118_50=test.RS(UM.exp.untrt,lista118,time.um.untrt,event.um.untrt,rs.lista118_50[[2]],rs.lista118_50[[3]])

UM155.b88_40=test.RS(UM.exp.untrt,listb88[[1]],time.um.untrt,event.um.untrt,rs.listb88_40[[2]],rs.listb88_40[[3]])
UM155.b88_50=test.RS(UM.exp.untrt,listb88[[1]],time.um.untrt,event.um.untrt,rs.listb88_50[[2]],rs.listb88_50[[3]])
UM155.b118_40=test.RS(UM.exp.untrt,listb118[[1]],time.um.untrt,event.um.untrt,rs.listb118_40[[2]],rs.listb118_40[[3]])
UM155.b118_50=test.RS(UM.exp.untrt,listb118[[1]],time.um.untrt,event.um.untrt,rs.listb118_50[[2]],rs.listb118_50[[3]])

####MSK###104
MSK104=t(read.csv(paste(dir1,"data.MSK(104).csv", sep=""), row.names=1,check.names=F))
save(MSK104,file="MSK104.Rdata")
load("MSK104.Rdata")
MSK104_2=MSK104
MSK104_2[as.numeric(MSK104_2[, "overall_survival_months"])>60,"death"]="Alive"
MSK104_2[as.numeric(MSK104_2[, "overall_survival_months"])>60,"overall_survival_months"]="60"
MSK104_2[, "death"]=ifelse(MSK104_2[, "death"]=="Alive", 0, 1)
MSK104_2[, "had_adjuvant_chemo"]=ifelse(MSK104_2[, "had_adjuvant_chemo"]=="FALSE",0,1)

MSK104_2[, "stage.title"]
MSK104.sub=MSK104_2[,c("histology", "had_adjuvant_chemo", "death", "overall_survival_months", "age", "gender", "stage.title")]
rownames(MSK104.sub)=rownames(MSK104_2)
###
#use the whole dataset
MSK.exp.w=apply(MSK104_2[,14:22296],2,as.numeric)#104:22283
rownames(MSK.exp.w)=rownames(MSK104.sub)
time.MSK.w=as.numeric(MSK104.sub[, "overall_survival_months"])/12
names(time.MSK.w)=rownames(MSK104.sub)
event.MSK.w=as.numeric(MSK104.sub[, "death"])
names(event.MSK.w)=rownames(MSK104.sub)

msk104.a88_40=test.RS(MSK.exp.w,lista88,time.MSK.w,event.MSK.w,rs.lista88_40[[2]],rs.lista88_40[[3]])
msk104.a88_50=test.RS(MSK.exp.w,lista88,time.MSK.w,event.MSK.w,rs.lista88_50[[2]],rs.lista88_50[[3]])
msk104.a118_40=test.RS(MSK.exp.w,lista118,time.MSK.w,event.MSK.w,rs.lista118_40[[2]],rs.lista118_40[[3]])
msk104.a118_50=test.RS(MSK.exp.w,lista118,time.MSK.w,event.MSK.w,rs.lista118_50[[2]],rs.lista118_50[[3]])
msk104.b88_40=test.RS(MSK.exp.w,listb88[[1]],time.MSK.w,event.MSK.w,rs.listb88_40[[2]],rs.listb88_40[[3]])
msk104.b88_50=test.RS(MSK.exp.w,listb88[[1]],time.MSK.w,event.MSK.w,rs.listb88_50[[2]],rs.listb88_50[[3]])
msk104.b118_40=test.RS(MSK.exp.w,listb118[[1]],time.MSK.w,event.MSK.w,rs.listb118_40[[2]],rs.listb118_40[[3]])
msk104.b118_50=test.RS(MSK.exp.w,listb118[[1]],time.MSK.w,event.MSK.w,rs.listb118_50[[2]],rs.listb118_50[[3]])

```

BIBLIOGRAPHY

1. Zhu, C.Q., et al., *Prognostic and predictive gene signature for adjuvant chemotherapy in resected non-small-cell lung cancer*. J Clin Oncol, 2010. **28**(29): p. 4417-24.
2. Kent, J. and J. O'Quigley, *Measures of dependence for censored survival data*. Biometrika, 1988. **75**(3): p. 525-534.
3. Der, S.D., et al., *Validation of a histology-independent prognostic gene signature for early-stage, non-small-cell lung cancer including stage IA patients*. J Thorac Oncol, 2014. **9**(1): p. 59-64.
4. Molina, J.R., et al., *Non-small cell lung cancer: epidemiology, risk factors, treatment, and survivorship*. Mayo Clin Proc, 2008. **83**(5): p. 584-94.
5. Winton, T., et al., *Vinorelbine plus cisplatin vs. observation in resected non-small-cell lung cancer*. N Engl J Med, 2005. **352**(25): p. 2589-97.
6. Nesbitt, J.C., et al., *Survival in early-stage non-small cell lung cancer*. Ann Thorac Surg, 1995. **60**(2): p. 466-72.
7. Scott, W.J., et al., *Treatment of non-small cell lung cancer stage I and stage II: ACCP evidence-based clinical practice guidelines (2nd edition)*. Chest, 2007. **132**(3 Suppl): p. 234S-242S.
8. Pisters, K.M., et al., *Cancer Care Ontario and American Society of Clinical Oncology adjuvant chemotherapy and adjuvant radiation therapy for stages I-IIIa resectable non small-cell lung cancer guideline*. J Clin Oncol, 2007. **25**(34): p. 5506-18.
9. Pignon, J.P., et al., *Lung adjuvant cisplatin evaluation: a pooled analysis by the LACE Collaborative Group*. J Clin Oncol, 2008. **26**(21): p. 3552-9.
10. Lu, Y., et al., *A gene expression signature predicts survival of patients with stage I non-small cell lung cancer*. PLoS Med, 2006. **3**(12): p. e467.
11. Beer, D.G., et al., *Gene-expression profiles predict survival of patients with lung adenocarcinoma*. Nat Med, 2002. **8**(8): p. 816-24.
12. Chen, H.Y., et al., *A five-gene signature and clinical outcome in non-small-cell lung cancer*. N Engl J Med, 2007. **356**(1): p. 11-20.
13. Potti, A., et al., *A genomic strategy to refine prognosis in early-stage non-small-cell lung cancer*. N Engl J Med, 2006. **355**(6): p. 570-80.
14. Raponi, M., et al., *Gene expression signatures for predicting prognosis of squamous cell and adenocarcinomas of the lung*. Cancer Res, 2006. **66**(15): p. 7466-72.
15. Wigle, D.A., et al., *Molecular profiling of non-small cell lung cancer and correlation with disease-free survival*. Cancer Res, 2002. **62**(11): p. 3005-8.
16. Bianchi, F., et al., *Survival prediction of stage I lung adenocarcinomas by expression of 10 genes*. J Clin Invest, 2007. **117**(11): p. 3436-44.
17. Sun, Z., D.A. Wigle, and P. Yang, *Non-overlapping and non-cell-type-specific gene expression signatures predict lung cancer survival*. J Clin Oncol, 2008. **26**(6): p. 877-83.
18. Lau, S.K., et al., *Three-gene prognostic classifier for early-stage non small-cell lung cancer*. J Clin Oncol, 2007. **25**(35): p. 5562-9.
19. Shedden, K., et al., *Gene expression-based survival prediction in lung adenocarcinoma: a multi-site, blinded validation study*. Nat Med, 2008. **14**(8): p. 822-7.

20. Irizarry, R.A., et al., *Exploration, normalization, and summaries of high density oligonucleotide array probe level data*. *Biostatistics*, 2003. **4**(2): p. 249-64.
21. Lim, W.K., et al., *Comparative analysis of microarray normalization procedures: effects on reverse engineering gene networks*. *Bioinformatics*, 2007. **23**(13): p. i282-8.
22. Pepper, S.D., et al., *The utility of MAS5 expression summary and detection call algorithms*. *BMC Bioinformatics*, 2007. **8**: p. 273.
23. *Affymetrix: Transcript assignment for NetAffx™ annotation, Affymetrix GeneChip IVT array white paper collection*. 2006 [cited 2006; Available from: http://media.affymetrix.com/support/technical/whitepapers/netaffxannot_whitepaper.pdf].