**Survival Rates in Ovarian Cancer Patients Based on Gene Expression Level**

Melissa Wilson

STAT 4893W

**Introduction**

Ovarian cancer is one of the deadliest cancers affecting women, accounting for about 152,000 deaths worldwide every year (Reid, Permuth, & Sellers, 2017). If caught in stage I, the 5-year survival is 92%. However, only 15% of women are diagnosed at this stage; most women are diagnosed with late-stage tumors, where the 5-year survival rate is 51% for stage III and 29% for stage IV (Torre et al., 2018). Therefore, improving early detection methods is important for minimizing the effects of this disease. One way to increase detection is to identify genes that are associated with the presence of tumors, called a genetic marker. These genes produce proteins in much higher or lower levels compared to unafflicted individuals, or have mutations not typically present in the normal population. Genetic markers may be identified by comparing survival rates to the expression level or sequence of these genes.

Survival analysis is a statistical technique widely used in biomedical research. “Survival” can mean time until death, but the technique may be used to model time until recurrence of the disease. In general, survival analysis can be used to model any “time until event” data, which can also be found in engineering fields where time until part or system failure is often of interest (Diamoutene, Barro, Somda, Noureddine, & Foguem, 2016). A major complication with this type of data is that the event may not occur over the time frame that the individual is followed. The study may end before the event occurs or the participant may drop out of the study. In this case, the observation is said to be censored – the time until event is unknown, but the survival is known to be at least as long as the length of time the individual was followed for. When analyzing this kind of data, it is important that censoring is independent of the event to avoid introducing bias. Censoring of events is one reason why ordinary linear regression is inappropriate in modeling this data (Leung, Elashoff, & Afifi, 1997).

Another reason linear regression is inadequate in modeling time to even data is that survival time isn’t normally distributed because time until event is always non-negative; parametric analysis methods typically assume that survival follows distributions such as Weibull, exponential, or log-normal. These parametric methods are more powerful, which is particularly useful when the sample size is small. However, determining the distribution is often difficult; the semiparametric Cox proportional hazards method is typically used due to its lack of required distribution assumptions and ability to analyze continuous variables and covariates (Schober & Vetter, 2018).

The primary goal of survival analysis is to estimate a survival function using both censored and uncensored data. The survival function S(t) is defined as

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

where T is a random variable denoting time until event occurs and t is time, both ranging between 0 and infinity. This function describes the probability that an individual survives past time t. At time t, the probability of survival is 1 (In & Lee, 2018). The hazard function h(t) can be derived as

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

which gives the instantaneous rate of event occurrence at time t, and is the negative derivative of the log of the survival function (Kleinbaum & Klein, 2012).

If no censoring exists within the data, S(t) can be estimated non-parametrically by 1 – F(t), where F(t) is the empirical distribution function of the data. If censoring does exist, the Kaplan-Meier method is a non-parametric way to estimate the survival function. Kaplan-Meier curves are often used to plot survival times of different groups (Kaplan & Meier, 1958). Log-rank tests can then be used to see if the Kaplan-Meier survival curves are significantly different between the groups (Schober & Vetter, 2018). The test statistic for the log-rank test is given as

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

where represents the sum of the observed number of outcomes in group j over time t, and represents the sum of the expected number of outcomes in group j over time t. Under the null hypothesis that the survival curves are the same in each group, the test statistic Z follows a chi-square distribution with degrees of freedom j-1. Since the log-rank test compares different Kaplan-Meier curves, it is best at comparing categorical variables with a discrete number of groups and thus is inadequate at describing the impact of continuous variables like gene expression, so it will not be used here.

Cox proportional hazards regression models can circumvent these limitations by modeling the hazard function instead of the survival function. This model estimates the hazard function definition given in Equation (2), and is given as

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

where X is a vector of p predictors, β is a vector of p parameters, and h0(t) is the baseline hazard function. This model assumes that the hazard ratio, defined as

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

between two individuals is independent of time. This implies that the hazard curves for each group should be proportional. When X2 is X1 + 1, the hazard ratio simplifies to eβ, so the coefficients can be interpreted as the change in the log of the hazard ratio given one unit increase in X (Cox, 1972). The proportional hazards assumption is tested by checking if the Schoenfeld residuals are constant over time in each predictor. Schoenfeld residuals are the difference between the actual and expected predictor values of an individual at the time they experience the event. If this assumption is violated, the interaction between the predictor and time is included in the model (Grambsch & Therneau, 1994).

**Methods and Materials**

The data used is a combination of data published in Denkert 2009 and Bonome 2008 (Bonome et al., 2008; Denkert et al., 2009). These datasets were chosen due to their public accessibility through the Gene Expression Omnibus, their inclusion of patient survival information, and their use of the same microarray platform. Gene expression in a sample is quantified using a microarray; different microarrays can detect different genes and their variants as well as report the expression level in different ways, so it is important that data are derived from the same microarray and platform to ensure that gene expression in both datasets are similarly distributed. Both studies used the Affymetrix Human Genome U133A Array with the GPL96 platform. Data was also restricted to patients with serous adenocarcinoma, a specific type of tumor affecting ovaries, in order to limit variability in survival time due to cancer type. This specific tumor type was chosen as it is the most common form of ovarian cancer and thus has more data available than other types. There is a total of 253 observations, 68 from the Denkert study and 185 from the Bonome study, with patients being followed for up to 13.6 years.

Both gene expression and clinical data were accessed using the GEOquery package in R (Sean & Meltzer, 2007). This package downloads data from the Gene Expression Omnibus and loads it into R, formatting them so both microarray and clinical data are accessible for further analysis. Batch effects, which are variation in gene expression due to collection of data at different times and by different people, were analyzed and adjusted for using the BatchQC package in R (Manimaran et al., 2016). The package alters the gene expression levels in the data before the model fitting based on empirical Bayes methods to standardize expression levels across the two batches; it does not introduce a random effect parameter into the model (Johnson, Li, & Rabinovic, 2007). Expression levels of BRCA1, HE4, P53, and RASSF1A were used as predictors in a Cox proportional hazards model due to previous literature indicating their potential as biomarkers (Agarwal et al., 2011). Thus, the Cox regression model becomes

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

The maximum likelihood estimates of these parameters are found by maximizing the likelihood function

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

where Lj is the likelihood of failing at time j and k is the total number of failure times. Since this function considers probabilities for subjects who failed so it doesn’t consider probabilities for censored observations, this is a partial likelihood. Censored participants are taken into account in the risk set R(t(j)) used to calculate these likelihoods, which is the set of individuals who can fail at time j, since those who were censored after time j were still at risk of failure at time j. The likelihood of failing at time j is dependent on being in the risk set, so censored observations still provide information about the likelihood of failing at a given time by increasing the risk set (Kleinbaum & Klein, 2012).

In order to test the proportional hazards assumption, scaled Schoenfeld residuals over time are calculated for each predictor. If the assumption holds, then the residuals are constant over time. The Schoenfeld residuals for time k is given as a vector whose length is the number of predictors and is defined as

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

where k is the event time, Z(k) is the vector of observed predictor values of the subject experiencing an event at time tk, and M(β, tk) is the conditional weighted expectation of the predictor vector at time t. The scaled Schoenfeld residuals are thus given as

|  |  |  |
| --- | --- | --- |
|  |  | (9) |

where V-1(β, tk) is the inverse weighted covariance matrix of the regression parameters (Grambsch & Therneau, 1994). The Schoenfeld residuals are plotted over time, with a separate plot being made for each predictor. If the proportional hazards assumption is met, the residuals should be flat over time. To test if the Schoenfeld residuals vary significantly with time, the test statistic for the jth predictor is given by

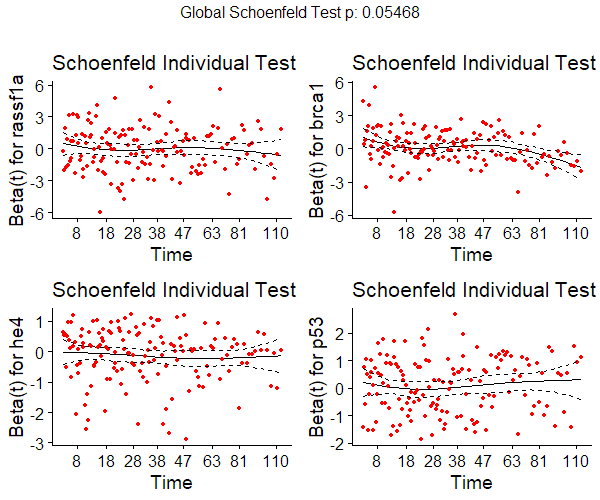
|  |  |  |
| --- | --- | --- |
|  |  | (10) |

where d is the total number of event times, g(tk) is a function of time, and is the mean of g(tk). Under the null hypothesis that the Schoenfeld residuals are independent of time, the test statistic follows a chi-square distribution with one degree of freedom. In this paper, the Kaplan-Meier product-limit estimator will be used for the function g(tk) in order to decrease the influence of outlier survival times (Park & Hendry, 2015). If a predictor violates the proportional hazards assumption, the Cox model can be extended to include the interaction between time and the predictor (Grambsch & Therneau, 1994).

The concordance index was used in order to compare the quality of different model fittings. It is the probability that the predicted survival from the model matches the observed survival. This is calculated by taking two observations at a time and classifying them as concordant or discordant. If a pair is concordant, that means that if subject i dies before subject j this matches the model prediction. A pair of observations is discordant if the subject who lived longer was expected to die first by the model. The concordance index is the proportion of pairs that were concordant. Random guessing would lead to a concordance index of 0.5; values higher than this indicate that the model is better than a random fit. The final model is selected based on having the highest concordance index of models tested (Raykar, Steck, Krishnapuram, Dehing-Oberije, & Lambin, 2008).

**Results**

Plots of scaled Schoenfeld residuals over time for each gene are shown in Figure 1. These plots illustrate the proportional hazard assumption – if the assumption is met, the residuals should be constant over time. The correlation between survival time and scaled Schoenfeld residuals was tested using the statistic described in Equation (10). As seen in the BRCA1 graph, the residuals decrease significantly over time (χ2 = 8.1, p = 0.004). All other genes satisfy the proportional hazards assumption.



**Figure 1.** Fit of Schoenfeld residuals for each gene over time. RASSF1A, HR4, and P53 have constant residuals over time, whereas BRCA1 has decreasing residuals.

The times were split into six month intervals in order to accommodate BRCA1 as a time-varying explanatory variable (Kartsonaki, 2016). Addition of a time turns the Cox model into

|  |  |  |
| --- | --- | --- |
|  |  | (11) |

where γ is the coefficient of the BRCA1 and time interaction. A summary of this model is given in Table 1. Negative coefficients indicate that the predictor decreases survival whereas positive coefficients indicate that the predictor increases survival. Only BRCA1 and its interaction with time are significant. The concordance index of this model is 0.576. Predictors were dropped from the model one at a time starting with those with the highest p-value until only significant predictors were left.

**Table 1.** Coefficients and significance of predictors in the time-varying model.

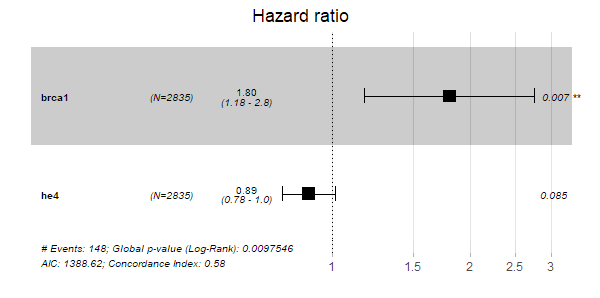
|  |  |  |  |
| --- | --- | --- | --- |
| Predictor | Coefficient | Z | P-value |
| RASSF1A | 0.025 | 0.15 | 0.88 |
| BRCA1 | 0.588 | 2.67 | 0.008 |
| HE4 | -0.121 | -1.81 | 0.07 |
| P53 | 0.085 | 1.03 | 0.30 |
| BRCA1:time | -0.012 | -2.60 | 0.009 |

Table 2 summarizes the final model. RASSF1A and P53 were dropped and HE4 was retained. The concordance index of this model is 0.578, which is an improvement over the model where only RASSF1A was dropped which had a concordance of 0.576. HE4 was retained despite it not being statistically significant at the 0.05 level since the model with only BRCA1 and its interaction with time had a concordance of 0.559.

**Table 2.** Coefficients and significance of predictors in the final model.

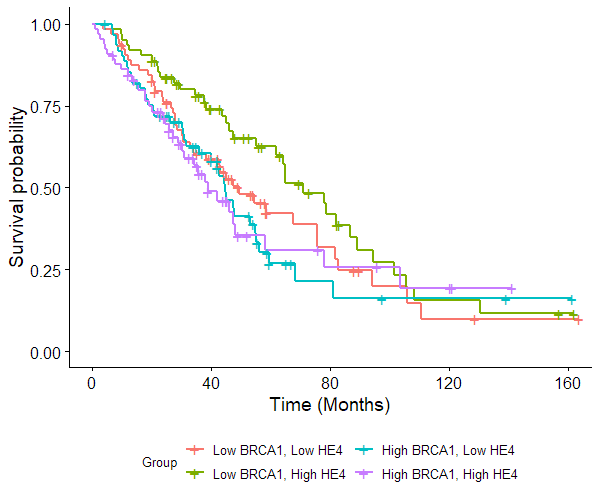
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Predictor | Coefficient | Exp(Coefficient) | Z | P-value |
| BRCA1 | 0.589 | 1.80 | 2.71 | 0.007 |
| HE4 | -0.115 | 0.89 | -1.72 | 0.09 |
| BRCA1:time | -0.011 | 0.99 | -2.61 | 0.009 |

A forest plot of the final model is shown in Figure 2. It illustrates the hazard ratio of each predictor, which is equivalent to the exponential of that predictor’s coefficient. Hazard ratios above one indicates a negative effect on survival and below one indicates a positive effect on survival with increasing values of the predictor. BRCA1 decrease survival probability whereas HE4 increases it. Specifically, a one unit increase in the expression level of HE4 is associated with an 11% decrease in the expected hazard, whereas a one unit increase in BRCA1’s expression is associated with an (80-0.01t)% increase in hazard, where t is time in months. BRCA1’s significance can be seen in its confidence interval not including one.



**Figure 2.** Forest plot showing the hazard ratios of BRCA1 and HE4 in the final model. The BRCA1 significantly affects survival, with increasing levels of BRCA1 increasing the hazard ratio. Increasing levels of HE4 lowers the hazard ratio. The function does not graph interactions, so the BRCA1:time interaction is not shown in this figure.

Kaplan-Meier curves are shown in Figure 3 below. Individuals were classified as either low or high for a given gene based on whether their expression level was below or above the median expression level for that gene, respectively. Each Kaplan-Meier curve in Figure 3 represents the survival of people based on their combination of BRCA1 and HE4. The group with the highest survival probability had low BRCA1 and high HE4, whereas the group with the lowest survival probability has high BRCA1 and low HE4. This agrees with the results of the Cox proportional hazards model, where HE4 was found to have a positive influence on survival and BRCA1 was found to have a negative influence. Individuals who had either high or low expression levels in both genes had intermediate survival probabilities. The violation of the proportional hazards assumption can also be seen in Figure 3 due to the curves crossing over, meaning that survival in each group is not proportional to each other. Specifically, you can see that high BRCA1 groups, those shown in purple and blue, cross over the low BRCA1 groups colored in red and green around month 100. This violates the proportional hazards assumption and is why BRCA1’s interaction with time is included in the final Cox model, as its impact on survival changes significantly over time – in the first 100 months, high BRCA1 is associated with lower survival probabilities, whereas after approximately 100 months high BRCA1 is associated with higher survival probabilities.



**Figure 3.** Kaplan-Meier curves of ovarian cancer patient survival based on expression level of BRCA1 and HE4. Median expression level was used as a cutoff point for classifying patients into high and low groups. Patients with low BRCA1 and high HE4 (green) had higher survival probabilities whereas patients with high BRCA1 and low HE4 (blue) had lower survival probabilities.

**Discussion**

The results show that BRCA1 and HE4 affect survival in ovarian cancer patients, with BRCA negatively affecting survival and HE4 positively affecting survival. This supports recent research identifying the significance and effect expression of these genes have on cancer survival, giving evidence for their potential as genetic markers in the prognosis of ovarian cancer (Luo et al., 2018; Tsibulak et al., 2018). This report illustrates the usefulness of survival analysis techniques in biomedical research when survival, relapse, or recurrence is an outcome of interest, where ordinary linear regression or logistic regression does not consider the skewed nature of survival times or the presence of censored observations.

The Cox method implemented in this study was chosen over parametric survival methods due to the lack of assumptions about the distribution of gene expression levels, widespread and long-term use in related literature, and well documented packages in R to facilitate its use. However, newer methods have been developed to address questions of survival and gene expression. Tabatabai argues that parametric survival models that assume the survival function follows a hypertabastic distribution, a probability distribution involving hyperbolic secant and cotangent functions, offers advantages over other survival methods by offering models with better fits, explicitly defined hazard and survival functions, and robustness to departures from an assumed distribution (Tabatabai, Eby, Nimeh, Li, & Singh, 2012). Another technique used in survival analysis that I did not discuss is the use of survival trees and random forests. These are non-parametric alternatives to Cox regression that split people into groups based on their observed predictor value. Some research suggests that survival tree based methods have similar predictive power as Cox-based methods when dealing with the highly-dimensional nature of gene expression data; survival random forest methods that build many trees using only a random sample of the data then averaging the models out in particular are often more accurate in studies that attempt to identify significant genes from a whole microarray of thousands of genes while controlling for the often small number of people sampled (Bonato et al., 2011; van Wieringen, Kun, Hampel, & Boulesteix, 2009).

**References**

Agarwal, A., Rein, B. J. D., Gupta, S., Dada, R., Safi, J., & Michener, C. (2011). Potential markers for detection and monitoring of ovarian cancer. *Journal of Oncology*, *2011*. https://doi.org/10.1155/2011/475983

Bonato, V., Baladandayuthapani, V., Broom, B. M., Sulman, E. P., Aldape, K. D., & Do, K. A. (2011). Bayesian ensemble methods for survival prediction in gene expression data. *Bioinformatics*, *27*(3), 359–367. https://doi.org/10.1093/bioinformatics/btq660

Bonome, T., Levine, D. A., Shih, J., Randonovich, M., Pise-Masison, C. A., Bogomolniy, F., … Birrer, M. J. (2008). A gene signature predicting for survival in suboptimally debulked patients with ovarian cancer. *Cancer Research*, *68*(13), 5478–5486. https://doi.org/10.1158/0008-5472.CAN-07-6595

Cox, D. R. (1972). Regression Models and Life-Tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, *34*(2), 187–220. https://doi.org/10.2307/2985181

Denkert, C., Budczies, J., Darb-Esfahani, S., Györffy, B., Sehouli, J., Könsgen, D., … Lage, H. (2009). A prognostic gene expression index in ovarian cancer-validation across different independent data sets. *The Journal of Pathology*, *218*(2), 273–280. https://doi.org/10.1002/path.2547

Diamoutene, A., Barro, D., Somda, M. A. S., Noureddine, F., & Foguem, B. K. (2016). SURVIVAL ANALYSIS IN LIVING AND ENGINEERING SCIENCES. *JP Journal of Biostatistics*, *13*(2), 223–238. https://doi.org/10.17654/bo013020223

Grambsch, P. M., & Therneau, T. M. (1994). Proportional Hazards Tests and Diagnostics Based on Weighted Residuals. *Biometrika*, *81*(3), 515. https://doi.org/10.2307/2337123

In, J., & Lee, D. K. (2018, June 1). Survival analysis: Part I - Analysis of time-to-event. *Korean Journal of Anesthesiology*. Korean Society of Anesthesiologists. https://doi.org/10.4097/kja.d.18.00067

Johnson, W. E., Li, C., & Rabinovic, A. (2007). Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, *8*(1), 118–127. https://doi.org/10.1093/biostatistics/kxj037

Kaplan, E. L., & Meier, P. (1958). *Nonparametric Estimation from Incomplete Observations*. *Source: Journal of the American Statistical Association* (Vol. 53).

Kartsonaki, C. (2016). Survival analysis. *Diagnostic Histopathology*, *22*(7), 263–270.

Kleinbaum, D. G., & Klein, M. (2012). *Survival Analysis: A Self-Learning Text* (Third). New York, NY: Springer. https://doi.org/10.1007/978-1-4419-6646-9

Leung, K.-M., Elashoff, R. M., & Afifi, A. A. (1997). CENSORING ISSUES IN SURVIVAL ANALYSIS. *Annual Review of Public Health*, *18*(1), 83–104. https://doi.org/10.1146/annurev.publhealth.18.1.83

Luo, C. H., Zhao, M., Chen, X. Y., Shahabi, S., Qiang, W., Zeng, L., … Zhou, H. H. (2018). HE4 and eIF3a expression correlates with surgical outcome and overall survival in ovarian cancer patients with secondary cytoreduction. *Journal of Cancer*, *9*(14), 2472–2479. https://doi.org/10.7150/jca.25184

Manimaran, S., Selby, H. M., Okrah, K., Ruberman, C., Leek, J. T., Quackenbush, J., … Johnson, W. E. (2016). BatchQC: interactive software for evaluating sample and batch effects in genomic data. *Bioinformatics*, *32*(24), 3836–3838. https://doi.org/10.1093/bioinformatics/btw538

Park, S., & Hendry, D. J. (2015). Reassessing Schoenfeld Residual Tests of Proportional Hazards in Political Science Event History Analyses. *American Journal of Political Science*, *59*(4), 1072–1087. https://doi.org/10.1111/ajps.12176

Raykar, V. C., Steck, H., Krishnapuram, B., Dehing-Oberije, C., & Lambin, P. (2008). On Ranking in Survival Analysis: Bounds on the Concordance Index. In *Advances in Neural Information Processing Systems*. MIT Press.

Reid, B. M., Permuth, J. B., & Sellers, T. A. (2017, March 1). Epidemiology of ovarian cancer: a review. *Cancer Biology and Medicine*. Cancer Biology and Medicine. https://doi.org/10.20892/j.issn.2095-3941.2016.0084

Schober, P., & Vetter, T. R. (2018). Survival Analysis and Interpretation of Time-to-Event Data. *Anesthesia & Analgesia*, *127*(3), 792–798. https://doi.org/10.1213/ANE.0000000000003653

Sean, D., & Meltzer, P. S. (2007). GEOquery: A bridge between the Gene Expression Omnibus (GEO) and BioConductor. *Bioinformatics*, *23*(14), 1846–1847. https://doi.org/10.1093/bioinformatics/btm254

Tabatabai, M. A., Eby, W. M., Nimeh, N., Li, H., & Singh, K. P. (2012). Clinical and multiple gene expression variables in survival analysis of breast cancer: Analysis with the hypertabastic survival model. *BMC Medical Genomics*, *5*. https://doi.org/10.1186/1755-8794-5-63

Torre, L. A., Trabert, B., DeSantis, C. E., Miller, K. D., Samimi, G., Runowicz, C. D., … Siegel, R. L. (2018). Ovarian cancer statistics, 2018. *CA: A Cancer Journal for Clinicians*, *68*(4), 284–296. https://doi.org/10.3322/caac.21456

Tsibulak, I., Wieser, V., Degasper, C., Shivalingaiah, G., Wenzel, S., Sprung, S., … Zeimet, A. G. (2018). BRCA1 and BRCA2 mRNA-expression prove to be of clinical impact in ovarian cancer. *British Journal of Cancer*, *119*(6), 683–692. https://doi.org/10.1038/s41416-018-0217-4

van Wieringen, W. N., Kun, D., Hampel, R., & Boulesteix, A. L. (2009). Survival prediction using gene expression data: A review and comparison. *Computational Statistics and Data Analysis*, *53*(5), 1590–1603. https://doi.org/10.1016/j.csda.2008.05.021

**Appendix**

|  |
| --- |
| > library(GEOquery)  > library(reshape2)  > library(survival)  > library(ggplot2)  > library(GGally)  > library(survMisc)  > library(limma)  > library(BatchQC)  > library(survminer)  > library(hgu133a.db)  > library(annotate)  > library(sva)  > library(coxphw)  > library(magrittr)  > library(dplyr)  Attaching package: ‘dplyr’  The following object is masked from ‘package:nlme’:  collapse  The following object is masked from ‘package:AnnotationDbi’:  select  The following objects are masked from ‘package:IRanges’:  collapse, desc, intersect, setdiff, slice, union  The following objects are masked from ‘package:S4Vectors’:  first, intersect, rename, setdiff, setequal, union  The following object is masked from ‘package:GGally’:  nasa  The following object is masked from ‘package:Biobase’:  combine  The following objects are masked from ‘package:BiocGenerics’:  combine, intersect, setdiff, union  The following objects are masked from ‘package:stats’:  filter, lag  The following objects are masked from ‘package:base’:  intersect, setdiff, setequal, union  > library(Greg)  >  >  > ########## getting data  >  >  >  > # get denkert data  > denkert.dat = getGEO("GSE14764", GSEMatrix =TRUE, getGPL=FALSE)  Found 1 file(s)  GSE14764\_series\_matrix.txt.gz  Using locally cached version: C:\Users\Public\Documents\Wondershare\CreatorTemp\Rtmp2vnaBu/GSE14764\_series\_matrix.txt.gz  Parsed with column specification:  cols(  .default = col\_double(),  ID\_REF = col\_character()  )  See spec(...) for full column specifications.  > denkert.dat = denkert.dat[[1]]  > denkert.exp = exprs(denkert.dat)  > denkert.phenotype = pData(denkert.dat)  >  > # get bonome data  > bonome.dat = getGEO("GSE26712", GSEMatrix =TRUE, getGPL=FALSE)  Found 1 file(s)  GSE26712\_series\_matrix.txt.gz  Using locally cached version: C:\Users\Public\Documents\Wondershare\CreatorTemp\Rtmp2vnaBu/GSE26712\_series\_matrix.txt.gz  Parsed with column specification:  cols(  .default = col\_double(),  ID\_REF = col\_character()  )  See spec(...) for full column specifications.  > bonome.dat = bonome.dat[[1]]  > bonome.exp = exprs(bonome.dat)  > bonome.phenotype = pData(bonome.dat)  > # get ride of normal cells  > bonome.phenotype = bonome.phenotype[11:195,]  > bonome.exp = bonome.exp[,row.names(bonome.phenotype)]  >  > # keep only neccessary phenotype data columns  > denkert.phenotype = denkert.phenotype[,c(2,26,39,40,41)]  > bonome.phenotype = bonome.phenotype[,c(2,24,38,40)]  >  > # keep only those with serous ovca  > denkert.phenotype = denkert.phenotype[denkert.phenotype[,3]=="serous ovca",]  > denkert.exp = denkert.exp[,row.names(denkert.phenotype)]  > # all data in bonome from serous tumors  >  > # is there batch effect? takes like 10 minutes to load  > batchQC(cbind(denkert.exp,bonome.exp),batch = c(rep(1,ncol(denkert.exp)),rep(2,ncol(bonome.exp)))) # yes  > all.expr = ComBat(cbind(denkert.exp,bonome.exp),batch = c(rep(1,ncol(denkert.exp)),rep(2,ncol(bonome.exp))))  Found2batches  Adjusting for0covariate(s) or covariate level(s)  Standardizing Data across genes  Fitting L/S model and finding priors  Finding parametric adjustments  Adjusting the Data  >  > # turn bonome's survival years into months and make into same format  > bonome.phenotype$survival.months = 12\*as.numeric(bonome.phenotype$`survival years:ch1`)  > denkert.phenotype$survival.months = as.numeric(denkert.phenotype$`overall survival time:ch1`)  > denkert.phenotype$survival.event = as.integer(denkert.phenotype$`overall survival event:ch1`)  > bonome.phenotype$survival.event = bonome.phenotype$`status:ch1`  > bonome.phenotype[bonome.phenotype$`status:ch1`=="AWD (alive with disease)",6] = 0  > bonome.phenotype[bonome.phenotype$`status:ch1`=="NED (no evidence of disease)",6] = 0  > bonome.phenotype[bonome.phenotype$`status:ch1`=="DOD (dead of disease)",6] = 1  >  > # combine phenotype data  > phenotype = rbind(denkert.phenotype[,c(1,2,6,7)],bonome.phenotype[,c(1,2,5,6)])  > phenotype$survival.event = as.integer(phenotype$survival.event)  >  > # need to map probe id to gene symbol  > x <- hgu133aSYMBOL  > mapped.probes = mappedkeys(x)  > genesym.probeid = as.data.frame(x[mapped.probes])  >  > # find probe id of genes of interest  > brca1 = genesym.probeid[genesym.probeid[,2]=="BRCA1",1]  > he4 = genesym.probeid[genesym.probeid[,2]=="WFDC2",1]  > p53 = genesym.probeid[genesym.probeid[,2]=="TP53",1]  > rassf1a = genesym.probeid[genesym.probeid[,2]=="RASSF1",1]  >  >  >  >  >  >  > ############## survival analysis  >  >  >  >  >  > # adds column to data.frame  > add.gene = function(dat, gene){  +  + # get name of gene  + gene.name = deparse(substitute(gene))  +  + # get gene expression level  + gene.expr = all.expr[gene[1],]  +  + # add new column  + dat = cbind(dat,gene.expr)  +  + # rename last column  + colnames(dat)[ncol(dat)] = gene.name  +  + return(dat)  + }  >  >  > # add genes  > ov = add.gene(phenotype, brca1)  > ov = add.gene(ov, he4)  > ov = add.gene(ov, p53)  > ov = add.gene(ov, rassf1a)  >  > # create the survival object  > survival = Surv(ov$survival.months,ov$survival.event)  >  > # create the cox regression model  > fit = coxph(survival~rassf1a+brca1+he4+p53,data = ov)  > summary(fit)  Call:  coxph(formula = survival ~ rassf1a + brca1 + he4 + p53, data = ov)  n= 253, number of events= 148  coef exp(coef) se(coef) z Pr(>|z|)  rassf1a 0.006478 1.006499 0.168765 0.038 0.9694  brca1 0.115503 1.122438 0.128706 0.897 0.3695  he4 -0.125887 0.881714 0.066110 -1.904 0.0569 .  p53 0.087618 1.091571 0.081899 1.070 0.2847  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  exp(coef) exp(-coef) lower .95 upper .95  rassf1a 1.0065 0.9935 0.7230 1.401  brca1 1.1224 0.8909 0.8722 1.444  he4 0.8817 1.1342 0.7746 1.004  p53 1.0916 0.9161 0.9297 1.282  Concordance= 0.556 (se = 0.027 )  Likelihood ratio test= 5.34 on 4 df, p=0.3  Wald test = 5.47 on 4 df, p=0.2  Score (logrank) test = 5.49 on 4 df, p=0.2    > # test proportional hazards assumption. tests if scaled Schoenfeld residuals ind of time  > test.ph = cox.zph(fit)  > test.ph # brca1 significant  rho chisq p  rassf1a -0.0541 0.446 0.50431  brca1 -0.2345 8.089 0.00445  he4 -0.0691 0.836 0.36067  p53 0.0684 0.701 0.40258  GLOBAL NA 9.271 0.05468  > ggcoxzph(test.ph) # decrease in est for brca1 as time went on  >  > # need to split times intervals so can have time and brca1 interaction  > ov.timesplit = timeSplitter(data = ov, by = 4,  + event\_var = "survival.event",  + event\_start\_status = 0,  + time\_var = "survival.months")  >  > # should be similar to previous model  > interval.fit = coxph(Surv(Start\_time, Stop\_time, survival.event)~rassf1a+brca1+he4+p53,data = ov.timesplit)  > summary(interval.fit)  Call:  coxph(formula = Surv(Start\_time, Stop\_time, survival.event) ~  rassf1a + brca1 + he4 + p53, data = ov.timesplit)  n= 2835, number of events= 148  coef exp(coef) se(coef) z Pr(>|z|)  rassf1a 0.006478 1.006499 0.168765 0.038 0.9694  brca1 0.115503 1.122438 0.128706 0.897 0.3695  he4 -0.125887 0.881714 0.066110 -1.904 0.0569 .  p53 0.087618 1.091571 0.081899 1.070 0.2847  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  exp(coef) exp(-coef) lower .95 upper .95  rassf1a 1.0065 0.9935 0.7230 1.401  brca1 1.1224 0.8909 0.8722 1.444  he4 0.8817 1.1342 0.7746 1.004  p53 1.0916 0.9161 0.9297 1.282  Concordance= 0.556 (se = 0.029 )  Likelihood ratio test= 5.34 on 4 df, p=0.3  Wald test = 5.47 on 4 df, p=0.2  Score (logrank) test = 5.49 on 4 df, p=0.2  >  > # add time interaction  > time.var.fit = coxph(Surv(Start\_time, Stop\_time, survival.event)~rassf1a+brca1+he4+p53+brca1:Start\_time,data = ov.timesplit)  > summary(time.var.fit)  Call:  coxph(formula = Surv(Start\_time, Stop\_time, survival.event) ~  rassf1a + brca1 + he4 + p53 + brca1:Start\_time, data = ov.timesplit)  n= 2835, number of events= 148  coef exp(coef) se(coef) z Pr(>|z|)  rassf1a 0.024846 1.025157 0.171766 0.145 0.88499  brca1 0.587803 1.800029 0.219990 2.672 0.00754 \*\*  he4 -0.120926 0.886099 0.066979 -1.805 0.07100 .  p53 0.085241 1.088979 0.082423 1.034 0.30105  brca1:Start\_time -0.011589 0.988478 0.004459 -2.599 0.00934 \*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  exp(coef) exp(-coef) lower .95 upper .95  rassf1a 1.0252 0.9755 0.7321 1.4355  brca1 1.8000 0.5555 1.1696 2.7704  he4 0.8861 1.1285 0.7771 1.0104  p53 1.0890 0.9183 0.9265 1.2799  brca1:Start\_time 0.9885 1.0117 0.9799 0.9972  Concordance= 0.576 (se = 0.029 )  Likelihood ratio test= 12.53 on 5 df, p=0.03  Wald test = 12.04 on 5 df, p=0.03  Score (logrank) test = 12.06 on 5 df, p=0.03  >  > # drop rassf1a  > time.2 = coxph(Surv(Start\_time, Stop\_time, survival.event)~brca1+he4+p53+brca1:Start\_time,data = ov.timesplit)  > summary(time.2)  Call:  coxph(formula = Surv(Start\_time, Stop\_time, survival.event) ~  brca1 + he4 + p53 + brca1:Start\_time, data = ov.timesplit)  n= 2835, number of events= 148  coef exp(coef) se(coef) z Pr(>|z|)  brca1 0.584958 1.794916 0.219232 2.668 0.00763 \*\*  he4 -0.120931 0.886095 0.067020 -1.804 0.07117 .  p53 0.086225 1.090051 0.082145 1.050 0.29387  brca1:Start\_time -0.011578 0.988489 0.004463 -2.594 0.00948 \*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  exp(coef) exp(-coef) lower .95 upper .95  brca1 1.7949 0.5571 1.1680 2.7584  he4 0.8861 1.1285 0.7770 1.0105  p53 1.0901 0.9174 0.9280 1.2805  brca1:Start\_time 0.9885 1.0116 0.9799 0.9972  Concordance= 0.576 (se = 0.029 )  Likelihood ratio test= 12.5 on 4 df, p=0.01  Wald test = 12.03 on 4 df, p=0.02  Score (logrank) test = 12.04 on 4 df, p=0.02  >  > # drop p53  > time.3 = coxph(Surv(Start\_time, Stop\_time, survival.event)~brca1+he4+brca1:Start\_time,data = ov.timesplit)  > summary(time.3)  Call:  coxph(formula = Surv(Start\_time, Stop\_time, survival.event) ~  brca1 + he4 + brca1:Start\_time, data = ov.timesplit)  n= 2835, number of events= 148  coef exp(coef) se(coef) z Pr(>|z|)  brca1 0.588791 1.801810 0.216772 2.716 0.00660 \*\*  he4 -0.115113 0.891265 0.066886 -1.721 0.08524 .  brca1:Start\_time -0.011413 0.988652 0.004376 -2.608 0.00911 \*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  exp(coef) exp(-coef) lower .95 upper .95  brca1 1.8018 0.555 1.1781 2.7557  he4 0.8913 1.122 0.7818 1.0161  brca1:Start\_time 0.9887 1.011 0.9802 0.9972  Concordance= 0.578 (se = 0.028 )  Likelihood ratio test= 11.4 on 3 df, p=0.01  Wald test = 11.19 on 3 df, p=0.01  Score (logrank) test = 11.19 on 3 df, p=0.01  >  > # drop he4  > time.4 = coxph(Surv(Start\_time, Stop\_time, survival.event)~brca1+brca1:Start\_time,data = ov.timesplit)  > summary(time.4)  Call:  coxph(formula = Surv(Start\_time, Stop\_time, survival.event) ~  brca1 + brca1:Start\_time, data = ov.timesplit)  n= 2835, number of events= 148  coef exp(coef) se(coef) z Pr(>|z|)  brca1 0.603610 1.828708 0.216506 2.788 0.00530 \*\*  brca1:Start\_time -0.011668 0.988400 0.004363 -2.674 0.00749 \*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  exp(coef) exp(-coef) lower .95 upper .95  brca1 1.8287 0.5468 1.196 2.7953  brca1:Start\_time 0.9884 1.0117 0.980 0.9969  Concordance= 0.559 (se = 0.028 )  Likelihood ratio test= 8.58 on 2 df, p=0.01  Wald test = 8.33 on 2 df, p=0.02  Score (logrank) test = 8.22 on 2 df, p=0.02  >  > # time 3 maximizes concordance, final model  > ggforest(time.3,data = ov.timesplit) # left decreased death, right increased death  Warning message:  Removed 1 rows containing missing values (geom\_errorbar).  >  >  > # split into different groups to graph. use original data  > ov$brca1.group = 0  > ov$he4.group = 0  > ov[which(ov$brca1<median(ov$brca1)),"brca1.group"] = "lower"  > ov[which(ov$brca1>=median(ov$brca1)),"brca1.group"] = "upper"  > ov[which(ov$he4<median(ov$he4)),"he4.group"] = "lower"  > ov[which(ov$he4>=median(ov$he4)),"he4.group"] = "upper"  > ov$brca1.group=as.factor(ov$brca1.group)  > ov$he4.group=as.factor(ov$he4.group)  >  > fit.km = survfit(Surv(ov$survival.months,ov$survival.event)~brca1.group + he4.group, data = ov)  > ggsurvplot(fit.km, legend = "bottom", legend.title = "Group",  + legend.labs = c("Low BRCA1, Low HR4", "Low BRCA1, High HE4", "High BRCA1, Low HE4", "High BRCA1, High HE4")) + guides(colour = guide\_legend(nrow = 2)) |
|  |
| |  | | --- | | > | |