8-2025

Genes That Matter: Survival Modeling in TCGA-BRCA with Treatment Interactions

David Pratt

GENES THAT MATTER: SURVIVAL MODELING IN TCGA-BRCA WITH TREATMENT INTERACTIONS

By

David Pratt
B.S., American University, 2009
M.S., University of Louisville, 2025

A Thesis
Submitted to the Faculty of the
School of Public Health and Information Sciences
of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Master of Science
in Biostatistics

Department of Bioinformatics and Biostatistics
University of Louisville
Louisville, Kentucky

August 2025

GENES THAT MATTER: SURVIVAL MODELING IN TCGA-BRCA WITH TREATMENT INTERACTIONS

By

David Pratt
B.S., American University, 2009
M.S., University of Louisville, 2025

A Thesis Approved on

1 August 2025

by the following Thesis Committee:

Dr. Michael Sekula, Thesis Director

Dr. Maiying Kong, Thesis Co-director

Dr. Elizabeth Cash, Committee Member

DEDICATION

To Mr. Michael Hansen, my AP Statistics teacher at Saint Albans School, who first showed me that clarity, rigor, and wit could coexist in data — and whose voice still echoes when I seek either to tell the truth with numbers, or to lie with statistics. I come bearing seeds.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Doug Lorenz, whose instruction in Survival Analysis provided the theoretical foundation for this work. His review of my proposal and support in assembling my committee were instrumental in shaping this thesis.

I am also grateful to Dr. Michael Sekula, Dr. Maiying Kong, and Dr. Elizabeth Cash for serving on my committee and for their thoughtful feedback and encouragement throughout. To Tyler Fritz, MS—my colleague, employee, and former classmate in Survival Analysis—thank you for your steady moral and technical support at every stage; to my family, for believing in me even from afar; and to my cat, Taboo, for being awake with me at 3 AM, long after everyone else had gone to bed.

Finally, I would like to acknowledge the individuals who contributed their data to The Cancer Genome Atlas (TCGA). Their willingness to share personal and clinical information makes research like this possible, and I am grateful for their contribution to science and to future patients.

ABSTRACT

GENES THAT MATTER: SURVIVAL MODELING IN TCGA-BRCA WITH TREATMENT INTERACTIONS

David Pratt

August 2025

High-dimensional genomic data offer both promise and challenges for identifying clinically relevant biomarkers. This study developed a parallelized survival modeling pipeline to identify genes associated with overall survival in breast cancer, with a focus on gene-by-treatment interactions and patient heterogeneity. RNA-Seq data from female patients in the TCGA-BRCA cohort were analyzed. Univariate Cox proportional hazards models were used to screen genes, adjusting for age, race/ethnicity, treatment status, and cancer stage. A LASSO-penalized Cox regression was fit across 2000 random seeds to assess feature stability. Genes were filtered by expression level, statistical significance, and hazard ratios (effect sizes) in either direction, then re-evaluated in a multivariable Cox model. Several genes with statistically significant treatment interactions were identified, including novel candidates not present in established prognostic panels. These findings highlight the value of interaction-aware survival modeling for improving personalized prognostic prediction in breast cancer and underscore the importance of accounting for treatment heterogeneity in high-dimensional genomic analyses.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

TABLE PAGE

LIST OF FIGURES

FIGURE PAGE

CHAPTER 1

INTRODUCTION

Breast cancer remains one of the most prevalent and biologically complex malignancies affecting women worldwide. Although improvements in early detection and therapeutic strategies have significantly enhanced survival outcomes, considerable heterogeneity in prognosis and treatment response persists. This variation underscores the need for a more nuanced understanding of the molecular features that drive treatment efficacy and long-term survival.

The integration of high-throughput genomic data into clinical oncology holds immense potential to address these challenges. RNA sequencing (RNA-Seq), in particular, enables comprehensive profiling of gene expression across tumors (Love et al., 2014). However, the dimensionality and complexity of transcriptomic data introduce substantial statistical and computational challenges (Fan and Lv, 2008; Zhao and Li, 2012). Methods capable of handling this scale, while maintaining interpretability and robustness, are essential for translating genomic insights into clinical applications.

Survival analysis provides a natural framework for studying time-to-event outcomes, such as overall survival, in the context of cancer research. The Cox proportional hazards (PH) model remains one of the most widely used tools in this domain due to its flexibility and semi-parametric nature (Cox, 1972; Therneau and Grambsch, 2000). When extended to high-dimensional settings, the Cox model can be used to test associations between gene expression and survival, and to evaluate how these associations may be modified by clinical factors such as treatment exposure (Tibshirani, 1997; Katzman et al., 2018).

1.1 The Curse of Dimensionality in Genomic Data

Genomic datasets often contain tens of thousands of features (e.g., gene expression values) but only a few hundred, or fewer, samples. This discrepancy between the number of predictors (p) and the number of observations (n) introduces what is commonly referred to as the *curse of dimensionality*. As dimensionality increases, the volume of the feature space grows exponentially, and data points become increasingly sparse. Traditional statistical models, such as ordinary least squares regression, break down in this regime due to non-identifiability, multicollinearity, and overfitting. Furthermore, distances between data points lose discriminative power, and irrelevant features can obscure the signal of interest.

These challenges necessitate the use of dimensionality reduction techniques capable of identifying a sparse subset of informative predictors while preserving model stability and interpretability. In this context, the Least Absolute Shrinkage and Selection Operator (LASSO) is particularly attractive because it performs both continuous shrinkage and automatic variable selection in high-dimensional settings (Tibshirani, 1996).

1.2 Dimensionality Reduction in High-Dimensional Genomics via LASSO

High-throughput genomic assays such as RNA-Seq generate datasets with tens of thousands of gene-level measurements per sample, resulting in a high-dimensional setting where the number of features (p) greatly exceeds the number of observations (n). This "$p \gg n$" structure poses serious challenges for classical statistical modeling, including non-uniqueness of parameter estimates, overfitting, and poor generalizability. Dimensionality reduction is thus an essential preprocessing step in such settings, particularly when the goal is to identify a parsimonious set of features associated with a clinical outcome such as overall survival.

The LASSO, introduced by Tibshirani (1996), is particularly well-suited for dimensionality reduction in genomic contexts. By imposing an ℓ_1 penalty on the magnitude of regression coefficients, LASSO performs both regularization and feature selection, shrinking many coefficients exactly to zero. This results in a sparse model that retains only the most informative predictors, a desirable property in genomics where most genes are not differentially expressed or associated with the outcome.

Unlike unsupervised methods such as principal component analysis (PCA), which reduce dimensionality based on variance alone, LASSO is supervised and outcome-oriented. It identifies features that contribute directly to the predictive signal for survival, making it more interpretable in translational biomedical research. Additionally, LASSO integrates seamlessly into Cox PH models, allowing penalized regression in the presence of right-censored time-to-event data (Tibshirani, 1997).

In this thesis, we apply a LASSO-penalized Cox model via the `glmnet` package (Friedman et al., 2010), using 10-fold cross-validation to select the regularization parameter λ that minimizes partial likelihood deviance. The resulting model includes a reduced set of genes with nonzero coefficients, which are then used in downstream multivariate and interaction modeling. This strategy ensures computational scalability, avoids overfitting, and enhances biological interpretability by prioritizing a concise list of candidate biomarkers.

Alternative methods for high-dimensional survival analysis include ridge regression, elastic net, and unpenalized screening-based approaches. However, ridge regression does not yield sparse solutions, and elastic net introduces a second tuning parameter. LASSO strikes a balance between interpretability, parsimony, and computational efficiency, making it especially appropriate for feature selection in transcriptomic survival analysis (Hastie et al., 2015).

1.3 Proposed Work

The Cancer Genome Atlas (TCGA) offers a rich resource of matched genomic and clinical data across diverse cancer types. This thesis focuses on analyzing RNA-Seq data from the TCGA Breast Invasive Carcinoma (TCGA-BRCA) cohort. The primary objective is to identify genes whose expression levels interact with treatment status to influence patient survival. This work emphasizes both marginal and interaction effects, with a particular focus on uncovering genes whose prognostic relevance is conditional upon treatment.

To this end, a scalable and reproducible computational pipeline using R and Bioconductor was implemented (Huber et al., 2015; R Core Team, 2024). The pipeline incorporates normalization (Love et al., 2014), filtering, univariate screening, effect size-based selection, LASSO-penalized Cox regression (Tibshirani, 1996, 1997; Friedman et al., 2010), and multivariate interaction modeling. This approach enables the prioritization of genes with strong evidence of treatment-modified prognostic value.

Key contributions of this thesis include: (1) a fully documented and parallelized workflow for interaction modeling in survival data (Microsoft and Weston, 2022; Corporation and Weston, 2022), (2) incorporation of effect size criteria in gene selection, and (3) evaluation of model stability and assumptions in a high-performance computing environment (Schoenfeld, 1982; Grambsch and Therneau, 1994). The results offer a biologically interpretable gene set with potential implications for personalized oncology and biomarker discovery.

The remainder of this manuscript is organized as follows: the next section describes the statistical theory, dataset, modeling strategy, and computational tools used in this project; Section 3 discusses the results of the gene-level analyses; and the final section discusses limitations, implications, and future directions.

CHAPTER 2

METHODS

2.1 Theoretical Foundation of the Cox PH Model

The Cox PH model, introduced by Cox (1972), is a semiparametric model widely used in survival analysis. It defines the hazard function for an individual with covariates $\mathbf{x}$ at time t as:

$$h(t \mid \mathbf{x}) = h_0(t) \exp(\mathbf{x}^\top \boldsymbol{\beta}) \tag{2.1}$$

where:

- $h(t \mid \mathbf{x})$ is the hazard function at time t for a subject with covariate vector $\mathbf{x}$,
- $h_0(t)$ is an unspecified, non-negative baseline hazard function,
- $\boldsymbol{\beta}$ is a vector of regression coefficients.

The model assumes the hazard ratio between any two individuals is constant over time:

$$\frac{h(t \mid \mathbf{x}_1)}{h(t \mid \mathbf{x}_2)} = \exp\left((\mathbf{x}_1 - \mathbf{x}_2)^\top \boldsymbol{\beta}\right) \tag{2.2}$$

This is known as the *proportional hazards assumption*, and it is central to the interpretability of the model.

2.1.1 Interpretation and Application

Each coefficient β_j represents the log-hazard ratio associated with covariate x_j. A unit increase in x_j corresponds to a multiplicative change of $\exp(\beta_j)$ in the hazard rate,

assuming other covariates are held constant.

In this thesis, we apply the Cox model to assess the relationship between survival and high-dimensional molecular covariates (e.g., gene expression), while adjusting for clinical variables such as age, race, and treatment. A typical model specification takes the form:

$$\texttt{Surv(time, status)} \sim \quad \texttt{age + race\_ethnicity + treatments} \times \texttt{genes} \tag{2.3}$$

This formulation allows us to assess both main gene effects and gene-treatment interaction effects, and to accommodate censoring in survival data.

2.1.2 Partial Likelihood Estimation

Since $h_0(t)$ is unspecified, Cox proposed estimating $\boldsymbol{\beta}$ via the *partial likelihood*, which avoids direct estimation of the baseline hazard:

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{D} \frac{\exp(\mathbf{x}_i^\top \boldsymbol{\beta})}{\sum_{j \in R_i} \exp(\mathbf{x}_j^\top \boldsymbol{\beta})} \tag{2.4}$$

where:

- D is the number of observed events (e.g., deaths),
- R_i is the risk set at time t_i, containing all individuals at risk just prior to t_i.

Maximizing this partial likelihood yields the maximum partial likelihood estimator $\hat{\boldsymbol{\beta}}$, which under standard regularity conditions is consistent and asymptotically normal.

2.2 Model Diagnostics for the Cox PH Model

The Cox PH model relies on several key assumptions:

1. **Proportional Hazards:** Hazard ratios between individuals remain constant over time.

2. **Independent Censoring:** Censoring is non-informative and unrelated to survival.

3. **Linearity in Log-Hazard:** Covariates affect the log-hazard additively and linearly.

4. **Correct Model Specification:** All relevant confounders are included and measured without error.

Violations of these assumptions can lead to biased estimates or invalid inference. In survival analysis, verifying the assumptions of the Cox proportional hazards model is critical to ensure valid inference and interpretable results. A primary diagnostic tool for this purpose is the use of Schoenfeld residuals, which assess whether the proportional hazards assumption holds over time.

The proportional hazards assumption posits that the hazard ratio between any two individuals is constant over time. The *Schoenfeld residuals*, introduced by Schoenfeld (1982), are used to assess the time-dependence of covariates.

For the i-th individual who experiences an event at time t_i, the Schoenfeld residual for covariate j is defined as:

$$r_{ij}^{(\text{Sch})} = x_{ij} - \bar{x}_j(t_i)$$

where $\bar{x}_j(t_i)$ is the risk-set weighted average of covariate j at time t_i:

$$\bar{x}_j(t_i) = \frac{\sum_{k \in R(t_i)} x_{kj} \exp(\hat{\beta}^\top x_k)}{\sum_{k \in R(t_i)} \exp(\hat{\beta}^\top x_k)}$$

with $R(t_i)$ denoting the risk set at time t_i.

Under the PH assumption, these residuals should be uncorrelated with time. A test for non-zero correlation, such as a scaled Schoenfeld residual plot against time with a locally weighted scatterplot smoothing (LOESS) curve, is often used to detect violations. A non-random pattern indicates a time-varying covariate effect, suggesting violation of the PH assumption Grambsch and Therneau (1994).

2.3 Data Analysis Workflow

2.3.1 Initial Data Acquisition

Gene expression and clinical data were obtained from The Cancer Genome Atlas (TCGA) Breast Invasive Carcinoma (TCGA-BRCA) cohort. A preprocessed `SummarizedExperiment` object containing unstranded RNA-Seq counts and metadata was loaded from `TCGA_data.rda`. Only primary tumor samples with available clinical annotations were retained for downstream analysis.

The clinical metadata were filtered to include only female patients with complete data on survival time, age at diagnosis, race, and ethnicity. The original dataset included **1,110** patients (1,098 females and 12 males); restricting the analysis to females removed the 12 male cases. Time-to-event was defined as the maximum of `days_to_death` and `days_to_last_follow_up`, converted to years. Vital status was encoded as a binary outcome (`Dead = 1`, `Alive = 0`). A composite `race_ethnicity` variable was constructed prioritizing Hispanic ethnicity; non-reported or low-frequency groups were collapsed as appropriate. American Joint Committee on Cancer (AJCC) pathologic stage was treated as a categorical covariate. Treatment status was encoded as a binary indicator based on the presence of any "`yes`" entry in the `treatment_or_therapy` field. Patients with incomplete covariate or survival data were excluded. A total of **1,047** female patients were retained for analysis, each with a matched RNA-Seq profile and full clinical annotation.

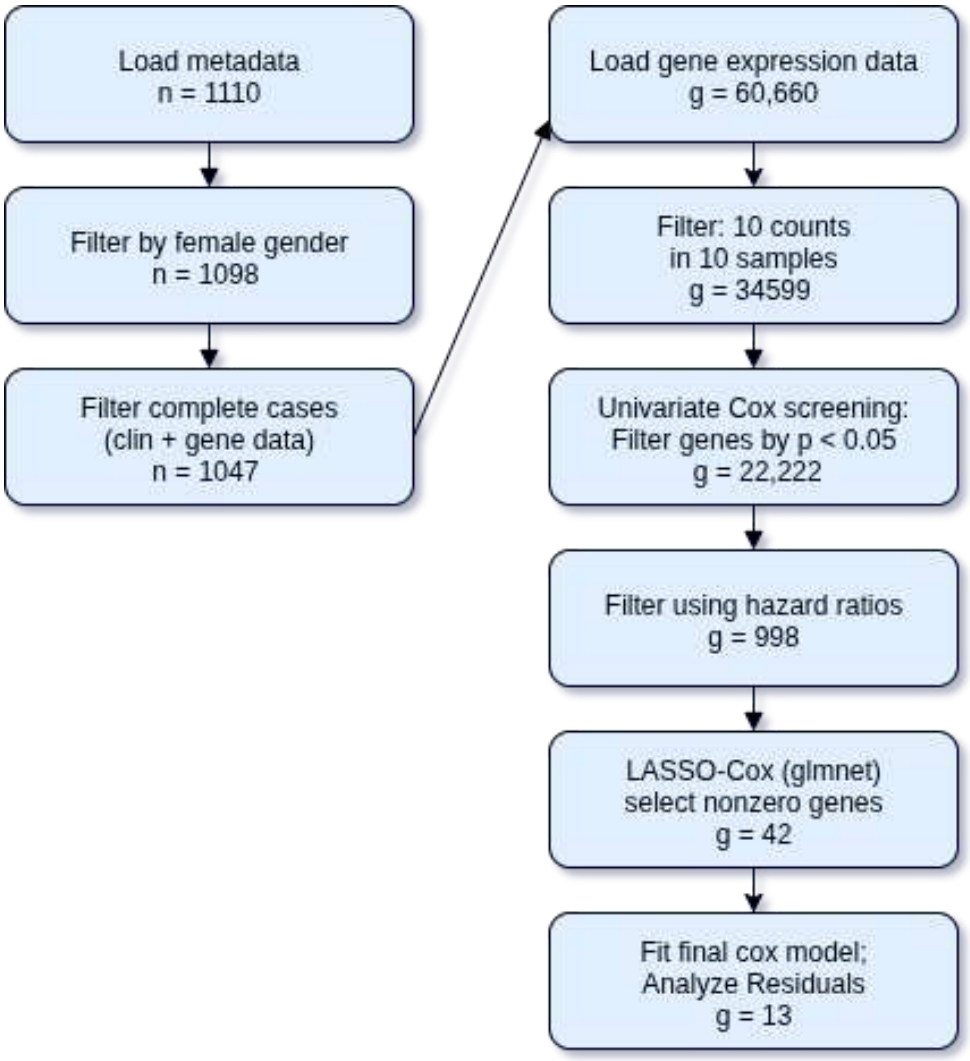

Figure 2.1: Combined Flowchart. Parallel preprocessing and modeling pipeline for TCGA-BRCA data. Clinical and gene expression data streams are joined by patient ID prior to LASSO and final Cox modeling.

A visual summary of the full preprocessing and modeling pipeline is provided in Figure 2.1, highlighting the major decision points and transformations applied to the data prior to survival analysis.

2.3.2 Gene Filtering and Normalization

Unstranded raw RNA-Seq count data were extracted and filtered to retain genes with at least 10 counts in at least 10 patients (approximately 1% prevalence). Normalization

was performed using the median-of-ratios method implemented in the `DESeq2` package. The resulting normalized counts were $\log_2$-transformed (adding a pseudocount of 1) to stabilize variance across expression levels. A final matrix of 34,599 genes across 1,047 patients was used for downstream modeling.

2.3.3 Univariate Cox Proportional Hazard Screening

Given the ultrahigh dimensionality of RNA-Seq data—where the number of genes (p) far exceeds the number of samples (n)—modeling all variables simultaneously is computationally infeasible and statistically unstable. To address this, we fit marginal Cox PH models for each gene, incorporating a treatment–gene interaction term, and extracted the nominal p-value associated with that interaction. To reduce dimensionality and identify candidate genes for penalized regression, univariate Cox PH models were fit for each gene using the following specification:

$$\text{Surv(time, status)} \sim \text{age} + \text{race\_ethnicity} + \text{treatments} \times \text{gene}_g \tag{2.5}$$

Both main effects and gene-treatment interactions were evaluated. For each gene g, the coefficients, standard errors, hazard ratios, z-statistics, and p-values for the gene and the interaction term were extracted and stored in a results matrix. Genes with p-values < 0.05 for either the main effect or interaction term were retained for penalized modeling.

This approach is motivated by the framework of *Sure Independence Screening* (SIS) proposed by Fan and Lv (2008), which demonstrates that marginal models can be used to screen variables in ultrahigh-dimensional settings with theoretical guarantees for retaining the truly important variables under certain regularity conditions. In survival analysis, this concept has been extended to Cox models by Zhao and Li

(2012), who showed that marginal screening based on partial likelihood score tests can effectively reduce dimensionality while preserving predictive signal. Furthermore, Simon et al. (2003) highlight the utility of p-value filtering to prioritize biologically meaningful gene–treatment interactions in high-throughput settings.

Although this screening step does not account for multicollinearity or complex gene–gene dependencies, it serves as an efficient and theoretically supported method to reduce the candidate feature space prior to applying multivariate penalized regression (e.g., LASSO). This hybrid strategy balances computational scalability with statistical rigor, making it well-suited for genomic survival analysis.

To enhance interpretability and prioritize genes with substantial biological or clinical relevance, a percentile-based hazard ratio (HR) filter was applied following initial screening. Genes were retained if either their main effect HR or their gene–treatment interaction HR fell outside the central 97% of the empirical distribution—i.e., below the 1.5th percentile or above the 98.5th percentile. This dual filtering step focused the analysis on genes with strong or unusual associations with survival, whether protective or deleterious, and helped reduce noise from weak or unstable effects. The HR thresholds were computed jointly across all screened genes and applied uniformly to both effect types. Genes meeting either criterion were retained for downstream penalized modeling.

2.3.4 Penalized Cox Modeling with LASSO

To address the high-dimensional setting where the number of predictors (p) greatly exceeds the number of samples (n), the Least Absolute Shrinkage and Selection Operator (LASSO) was used for simultaneous regularization and variable selection. LASSO adds an ℓ_1 penalty to the regression coefficients:

$$\hat{\boldsymbol{\beta}}^{\text{lasso}} = \arg\min_{\boldsymbol{\beta}} \left\{ -\ell(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{2.6}$$

where $\ell(\boldsymbol{\beta})$ is the partial log-likelihood and λ controls the strength of penalization. The ℓ_1 penalty forces many coefficients to exactly zero, yielding sparse models that improve interpretability and help identify candidate biomarkers and gene–treatment interactions (Tibshirani, 1996; Hastie et al., 2015).

Genes passing the univariate screening step were further filtered based on the empirical distribution of hazard ratios (HRs) to prioritize features with substantial effects. Specifically, genes were retained if either their main effect HR or their gene–treatment interaction HR fell below the 1.5th percentile or above the 98.5th percentile of the empirical HR distribution. This dual filter targeted genes with strong or unusual associations with survival, whether protective or deleterious, and reduced noise from weak or unstable effects. HR thresholds were computed jointly across all screened genes and applied uniformly to both effect types. This process yielded a total of **998 genes** for penalized modeling.

A Cox LASSO model was then fit using the `glmnet` package, which minimizes the negative partial log-likelihood with an ℓ_1 penalty applied to the gene and gene–treatment interaction coefficients. The design matrix included patient-level covariates (age, race/ethnicity, treatment, AJCC stage) and selected genes, with explicit gene–treatment interaction terms. To ensure clinical covariates remained in the model, they were assigned a penalty factor of zero (unpenalized), while all gene-related terms were subject to ℓ_1 regularization. All predictors were standardized internally by `glmnet` prior to fitting. The tuning parameter λ was selected via 10-fold cross-validation, using the value that minimized the partial likelihood deviance.

To assess stability, the entire LASSO fitting procedure was repeated across 2000 deterministic seeds. Coefficients were aggregated across runs, and genes were retained as "stable" if they appeared in at least 475 of the 2000 models with a consistent coefficient sign. This yielded **13 genes**, which were carried forward into the final multivariable Cox model.

2.3.5 Final Cox PH Model

Genes appearing in ≥475 LASSO models and showing consistent directionality were included in the final multivariable Cox model. Stratified Cox regression was used to account for non-proportional hazards by strata in `ajcc_stage_numeric` and `treatments`. The final model specification was:

$$\begin{aligned}\text{Surv}(\text{time}, \text{status}) \sim \text{age} + \text{race\_ethnicity} + \text{strata}(\text{ajcc\_pathologic\_stage}) \\ + \text{strata}(\text{treatments}) + \sum_g \text{gene}_g + \sum_g \left(\text{gene}_g \times \text{treatments}\right)\end{aligned} \tag{2.7}$$

Only genes with statistically significant effects ($p < 0.05$) in the final model were interpreted. Model coefficients were visualized using forest plots, and Kaplan–Meier curves were generated for selected genes stratified by treatment status.

2.3.6 Residual Analysis and Signal Stability

Schoenfeld residuals were calculated for each covariate in the final Cox model to assess the proportional hazards assumption. Residuals were plotted against time with LOESS smoothing, where substantial deviation from a flat trend was interpreted as evidence of time-dependent effects.

CHAPTER 3

RESULTS

3.1 Demographic and Descriptive Characteristics

Summary statistics for age and race/ethnicity distributions across the overall cohort and by treatment status are shown in Tables 3.1 and 3.2. The mean age of the cohort was 58.67 years, with a standard deviation (SD) of 12.89. Patients receiving treatment were slightly younger on average (mean = 57.98 years) compared to untreated patients (mean = 61.91 years). The interquartile range shows this trend persists across the age distribution, suggesting age may be a confounding factor in survival outcomes.

Race/ethnicity distributions were relatively stable between treated and untreated groups, with the majority of patients identifying as White (67.4% overall), followed by Black or African American (16.7%). Small differences were observed in the percentage of Hispanic or Latino individuals, with treated patients comprising 4.2% versus only 0.5% among untreated. The percentage of "Not Reported" was modest but larger in the untreated group (11.4%), which may impact subgroup analyses.

Table 3.1: Age Summary by Treatment Group

Group	**n**	**Mean**	**SD**	**Q1**	**Median**	**Q3**
Overall	1047	58.67	12.89	49.21	58.77	67.72
Treated	862	57.98	12.41	48.56	58.32	66.19
Untreated	185	61.91	14.51	52.75	62.18	72.35

3.2 Final Cox PH Model

The final multivariable Cox PH model was fit using a reduced set of predictors, including clinical covariates, selected gene expression terms, and gene–treatment

Table 3.2: Race/ethnicity counts and percentages, stratified by treatment status.

Race/Ethnicity	**Overall (n=1047)**	**Treated (n=862)**	**Untreated (n=185)**
White	706 (67.4%)	578 (67.1%)	128 (69.2%)
Asian	45 (4.3%)	37 (4.3%)	8 (4.3%)
Black or African American	175 (16.7%)	148 (17.2%)	27 (14.6%)
Hispanic or Latino	37 (3.5%)	36 (4.2%)	1 (0.5%)
Not Reported	84 (8.0%)	63 (7.3%)	21 (11.4%)

interactions identified via stability-based LASSO selection.

Table 4.1 summarizes the estimated hazard ratios (HRs), 95% confidence intervals (CIs), and significance codes for each covariate in the final model. Age was significantly associated with increased hazard (HR = 1.0428, $p < 0.001$). Several genes demonstrated protective or deleterious associations: for example, `ENSG00000212452.1` was significantly protective (HR = 0.6810, $p < 0.01$), while `ENSG00000197081.16` was associated with increased hazard (HR = 2.0599, $p < 0.001$). The main effect of `ENSG00000253474.2` was also significant and protective (HR = 0.4205, 95% CI: 0.2129–0.8304, $p < 0.05$).

Three gene–treatment interaction terms were retained in the final model. Notably, the interaction involving `ENSG00000271653.1` was significant (HR = 2.0831, 95% CI: 1.0260–4.2294, $p < 0.05$), suggesting a potential modifying effect of treatment. The interaction between treatment and `ENSG00000136560.14` was also significant (HR = 0.3605, 95% CI: 0.1358–0.9571, $p < 0.05$), indicating possible treatment sensitivity for patients with elevated expression of that gene. Finally, `ENSG00000253474.2` also exhibited a strong treatment interaction effect (HR = 2.2547, 95% CI: 1.1018–4.6143, $p < 0.05$), indicating that the survival benefit associated with higher expression may be modulated by treatment status.

Table 3.3: Final Cox PH Model: Significant and Marginal Terms with Hazard Ratios and Confidence Intervals

Covariate	**Sig.**	**HR**	**CI Lower**	**CI Upper**
age	***	1.0428	1.0234	1.0626
asian_ethnicity		0.6043	0.1331	2.7436
black_ethnicity		1.0998	0.6039	2.0031
hispanic_ethnicity		0.2273	0.0228	2.2708
not_reported_ethnicity	**	0.1926	0.0561	0.6607
ENSG00000041880.14	**	0.6005	0.4410	0.8177
ENSG00000088256.9	*	1.6394	1.0270	2.6170
ENSG00000100099.21		0.7043	0.4440	1.1171
ENSG00000108582.12	**	0.6312	0.4794	0.8309
ENSG00000124568.12	*	0.6200	0.4184	0.9185
ENSG00000128463.13	.	1.6838	0.9797	2.8938
ENSG00000136560.14		1.6458	0.6698	4.0437
ENSG00000136694.9		0.7098	0.4707	1.0704
ENSG00000138835.22	.	1.5435	0.9554	2.4937
ENSG00000142686.8	**	0.4545	0.2821	0.7322
ENSG00000144711.16		0.8513	0.3576	2.0265
ENSG00000160953.16		1.3904	0.8770	2.2045
ENSG00000165943.5	**	0.6052	0.4167	0.8791
ENSG00000166140.17		1.2436	0.7901	1.9574
ENSG00000177030.17	*	0.6711	0.4536	0.9928
ENSG00000188707.6		1.0035	0.6414	1.5701
ENSG00000197081.16	***	2.0599	1.3757	3.0845
ENSG00000212452.1	**	0.6810	0.5364	0.8646
ENSG00000235237.1	.	0.6251	0.3700	1.0561

Continued on next page

Table 3.3 – continued from previous page

Covariate	Sig.	HR	CI Lower	CI Upper
ENSG00000253474.2	*	0.4205	0.2129	0.8304
ENSG00000260048.2	**	0.6866	0.5331	0.8842
ENSG00000260913.1		0.7621	0.5486	1.0587
ENSG00000265943.1	***	0.5982	0.4651	0.7694
ENSG00000271653.1		0.6266	0.3195	1.2287
ENSG00000276805.2	.	1.3096	0.9628	1.7812
Treatment × ENSG00000136560.14	*	0.3605	0.1358	0.9571
Treatment × ENSG00000144711.16	.	2.2930	0.8853	5.9390
Treatment × ENSG00000188707.6		0.8857	0.5299	1.4804
Treatment × ENSG00000235237.1		1.5324	0.8475	2.7706
Treatment × ENSG00000253474.2	*	2.2547	1.1018	4.6143
Treatment × ENSG00000271653.1	*	2.0831	1.0260	4.2294

Significance codes: ******* $p < 0.001$, ****** $p < 0.01$, ***** $p < 0.05$, **.** $p < 0.1$

These results highlight a set of genes with significant main or interaction effects on survival. Together with clinical covariates, these features form a parsimonious model with strong interpretability and potential translational relevance.

3.3 Discussion of Main Effects and Interactions

The final Cox proportional hazards (PH) model integrated both clinical variables and gene expression features, along with selected interaction terms between treatment status and gene expression. This section interprets the contribution of each predictor type to overall survival in the TCGA-BRCA cohort.

Clinical Covariates. Age was the only clinical covariate significantly associated with hazard (HR = 1.04, $p < 0.001$), indicating that each additional year of age conferred a 4% increase in risk of death. Ethnicity variables were modeled relative to White (non-Hispanic) patients as the reference group. Most ethnic subgroups did not reach conventional levels of significance. However, the "Not Reported" ethnicity group was significantly associated with reduced hazard (HR = 0.19, $p < 0.01$), potentially reflecting data censoring or a hidden confounder. Other race/ethnicity categories showed wide confidence intervals and non-significant p-values, possibly due to small subgroup sizes.

Gene Main Effects. Several genes demonstrated statistically significant associations with overall survival:

- `ENSG00000041880.14` (PARP3): HR = 0.6005 ($p < 0.01$) — protective.
- `ENSG00000088256.9` (GNA11): HR = 1.6394 ($p < 0.05$) — risk-enhancing.
- `ENSG00000108582.12` (CPD): HR = 0.6312 ($p < 0.01$) — protective.
- `ENSG00000124568.12` (SLC17A1): HR = 0.6200 ($p < 0.05$) — protective.
- `ENSG00000142686.8` (C1orf216): HR = 0.4545 ($p < 0.01$) — protective.
- `ENSG00000165943.5` (MOAP1): HR = 0.6052 ($p < 0.01$) — protective.
- `ENSG00000177030.17` (DEAF1): HR = 0.6711 ($p < 0.05$) — protective.
- `ENSG00000197081.16` (IGF2R): HR = 2.0599 ($p < 0.001$) — risk-enhancing.
- `ENSG00000212452.1` (SNORD69): HR = 0.6810 ($p < 0.01$) — protective.
- `ENSG00000253474.2` (novel transcript): HR = 0.4205 ($p < 0.05$) — protective.
- `ENSG00000260048.2` (IGHV1OR16-3): HR = 0.6866 ($p < 0.01$) — protective.

- `ENSG00000265943.1` (RP11-739L10.1): HR = 0.5982 ($p < 0.001$) — protective.

These genes were selected through high-dimensional modeling across 2000 penalized seeds and demonstrated consistent directionality. Notably, several immune-related genes (e.g., HLA-DRA, IFI30, PTGFRN) and poorly characterized noncoding RNAs were included, indicating potential roles in immune modulation and tumor regulation.

Gene–Treatment Interactions. The model retained several interaction terms between treatment status and gene expression. These interaction effects highlight genes whose impact on survival varies depending on whether the patient received treatment.

- `Treatment` × `ENSG00000136560.14`: HR = 0.3605 ($p < 0.05$) — a strong protective interaction, suggesting enhanced benefit from treatment at higher expression levels.
- `Treatment` × `ENSG00000144711.16`: HR = 2.2930 ($p < 0.1$) — marginally significant deleterious interaction; patients with higher expression may derive less benefit from treatment.
- `Treatment` × `ENSG00000188707.6`: HR = 0.8857 (ns) — non-significant interaction.
- `Treatment` × `ENSG00000235237.1`: HR = 1.5324 (ns) — non-significant interaction.
- `Treatment` × `ENSG00000253474.2`: HR = 2.2547 ($p < 0.05$) — significant deleterious interaction; treatment may associate with worse outcomes in patients with high expression.

- `Treatment` × `ENSG00000271653.1`: HR = 2.0831 ($p < 0.05$) — significant deleterious interaction.

These results suggest that specific genes may serve as molecular markers of treatment response heterogeneity. Particularly, the deleterious interaction effects for `ENSG00000253474.2` and `ENSG00000271653.1` point to the potential for adverse treatment responses in molecularly defined subgroups.

Summary. Together, the significant main effects and interaction terms support a model in which both intrinsic gene expression patterns and their modulation by treatment influence survival. Immune-related genes and non-coding transcripts emerged as important predictors, consistent with growing evidence for the role of tumor–immune dynamics and noncoding regulation in breast cancer prognosis. Stratified or personalized treatment strategies guided by molecular signatures may be a fruitful direction for further study.

Model Stability and Overfitting

The comparison between the single-seed and consensus forest plots (Figures 3.3–3.6) highlights important concerns regarding model stability. Notably, several terms appearing in the single-seed model are absent from the consensus model derived from 2000 seeds, and vice versa. This lack of overlap suggests that reliance on a single LASSO run may capture spurious associations driven by random sampling variation or noise, rather than robust signal. The consensus-based approach, by contrast, identifies features that demonstrate consistent associations with survival across many resampled iterations. This supports the use of stability selection as a safeguard against overfitting and reinforces the credibility of the terms retained in the final model. In high-dimensional settings such as this, model interpretability and reproducibility are enhanced by emphasizing variables that persist under repeated subsampling and

penalization.

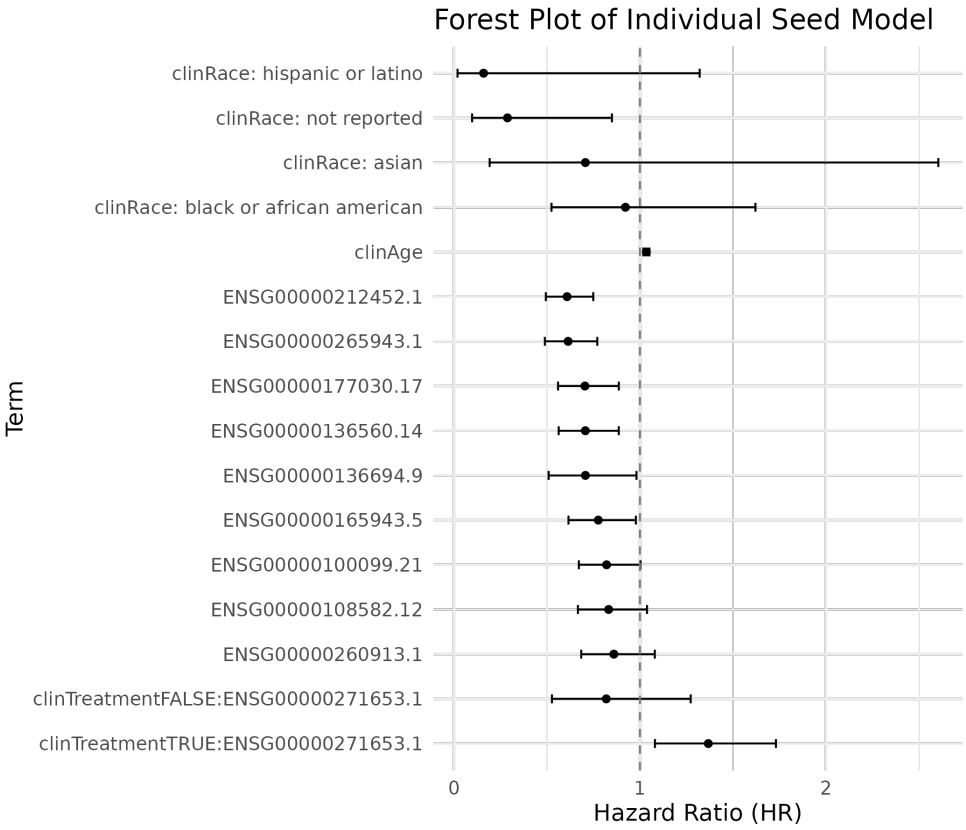

Figure 3.1: Forest plot from Cox model using seed = 105541. This model illustrates how initialization can affect both gene inclusion and hazard ratio estimation.

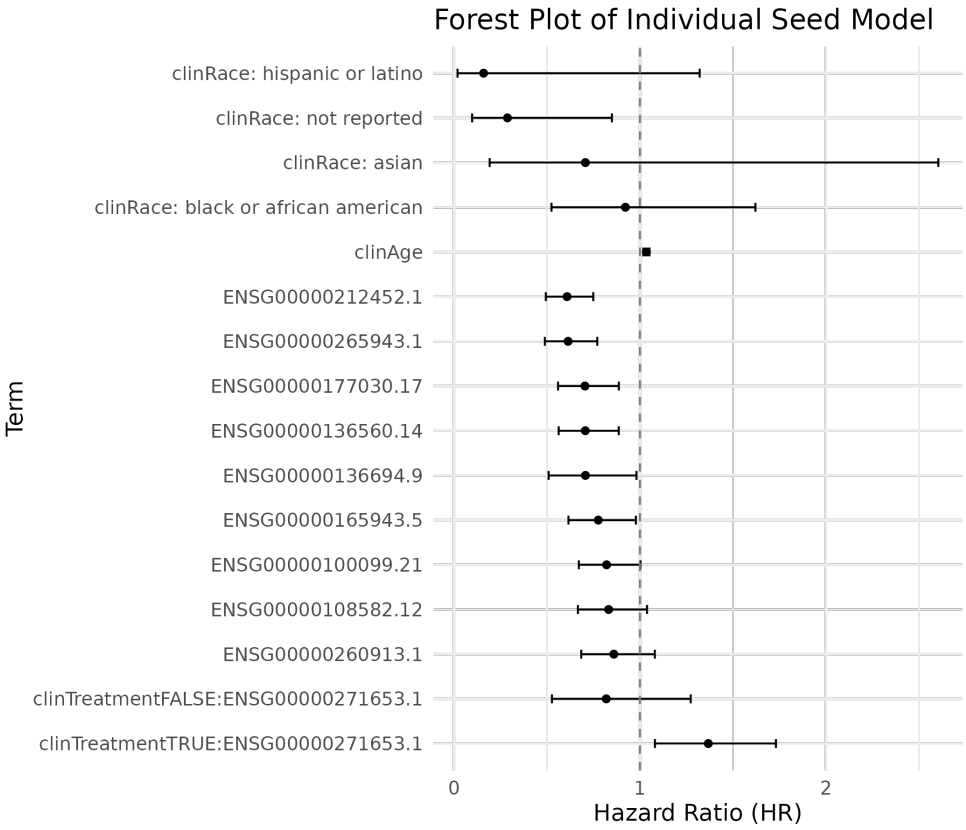

Figure 3.2: Forest plot from Cox model using seed = 127352. Compared to Seed 105541, this model selects a different subset of genes and produces notably different hazard ratios.

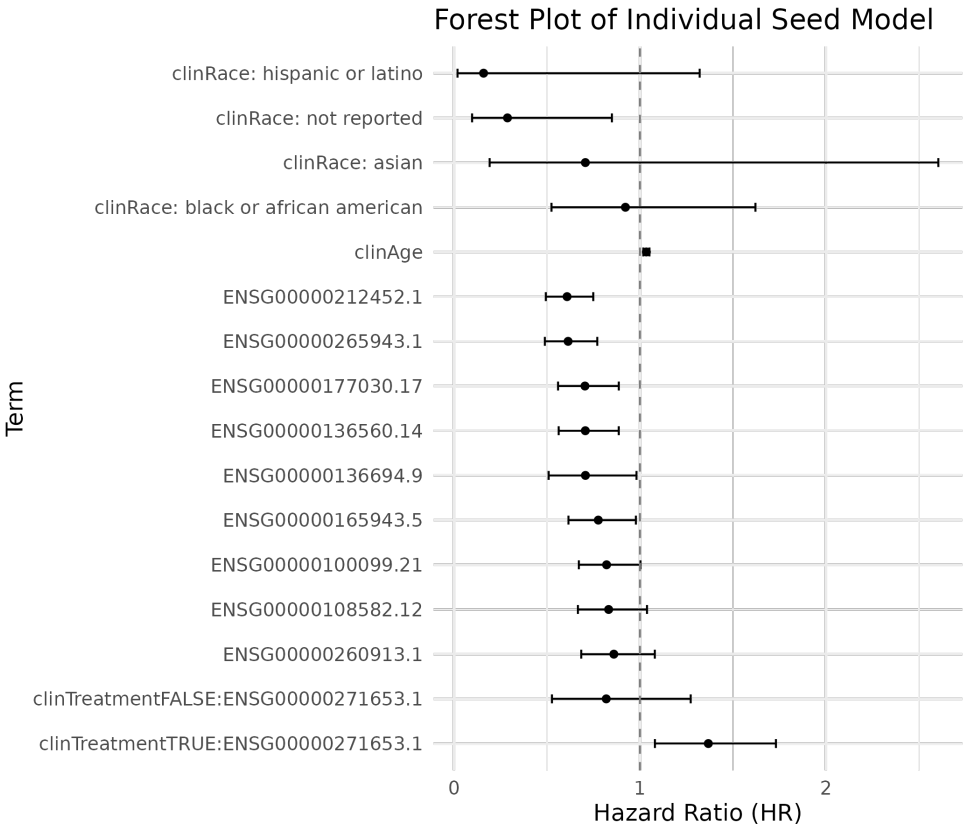

Figure 3.3: Forest plot from Cox model using seed = 140284. This example further illustrates instability, with multiple coefficient estimates and interactions diverging from the other seeds.

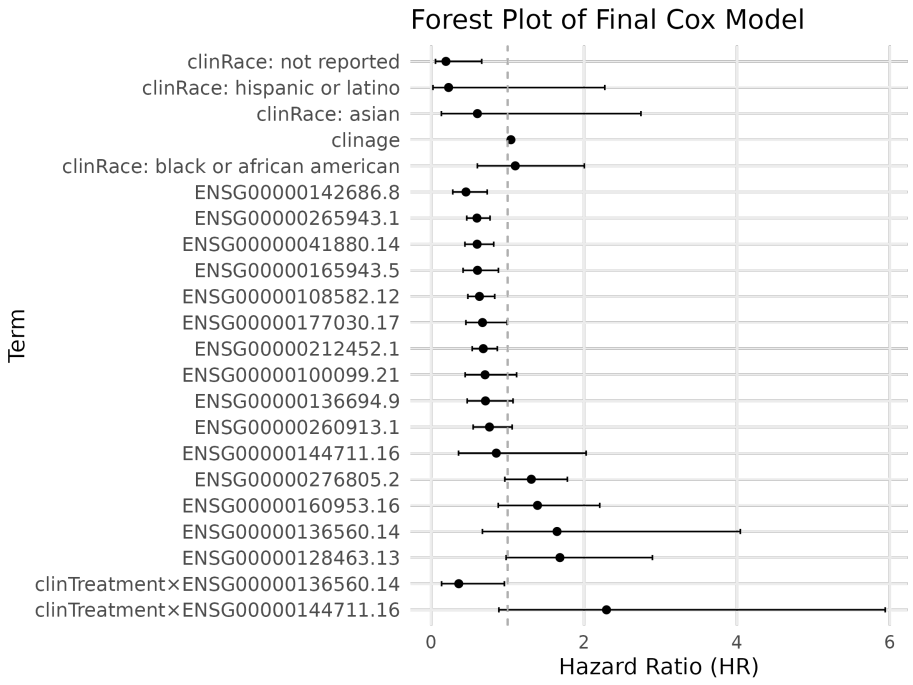

Figure 3.4: Forest plot of the final Cox model after consensus-based selection across 2000 seeds. Terms shown were retained in at least 23.75% of bootstrap replicates and represent stable gene–treatment associations.

Table 3.4: Presence of significant genes across three single-seed LASSO runs and the final multi-seed model. The Census column counts the number of models in which each gene was present.

Gene	Seed 105541	Seed 27352	Seed 40284	Final Model	Census
ENSG00000041880.14	1	1	1	1	4
ENSG00000088256.9	1	0	0	0	1
ENSG00000108582.12	1	1	1	1	4
ENSG00000124568.12	1	1	1	1	4
ENSG00000142686.8	1	1	1	1	4
ENSG00000165943.5	1	1	1	1	4
ENSG00000177030.17	1	1	1	1	4
ENSG00000197081.16	1	1	1	1	4
ENSG00000212452.1	1	1	1	1	4
ENSG00000253474.2	1	1	0	0	2
ENSG00000260048.2	1	1	1	1	4
ENSG00000265943.1	1	0	1	0	2

Table 3.5: Jaccard similarity between the sets of selected genes from three single-seed LASSO runs and the final multi-seed model. Values range from 0 (no overlap) to 1 (identical sets), with higher values indicating greater similarity in selected gene sets.

	Seed 105541	Seed 27352	Seed 40284	Final Model
Seed 105541	1.00	0.83	0.83	0.75
Seed 27352	0.83	1.00	0.82	0.90
Seed 40284	0.83	0.82	1.00	0.90
Final Model	0.75	0.90	0.90	1.00

The consistency of gene selection across models provides insight into the stability of the identified biomarkers. As shown in Table 3.5, several genes appear in all single-seed LASSO runs as well as in the final multi-seed model, indicating robust signal detection rather than artifacts of a particular random seed. This overlap underscores the reliability of these genes as potential prognostic markers and justifies their prioritization for downstream biological interpretation and validation.

In contrast, the final model, derived from `2000` LASSO fits across distinct deterministic seeds, retains only features consistently selected across resampled runs. The fact that many genes appear in only one or a few single-seed models, yet are absent from the consensus model, supports the interpretation that such terms may reflect noise or random fluctuations rather than robust biological signal. This validates the use of stability selection as a safeguard against overfitting and as a necessary step for reproducible inference in penalized survival modeling.

3.4 Model Diagnostics and Residuals

Proportional hazards assumptions were evaluated using scaled Schoenfeld residuals for each covariate in the final Cox model. Visual inspection of residual plots showed no major departures from proportionality. Formal tests—both global and covariate-specific—corroborated these findings, confirming that the PH assumption held across all covariates.

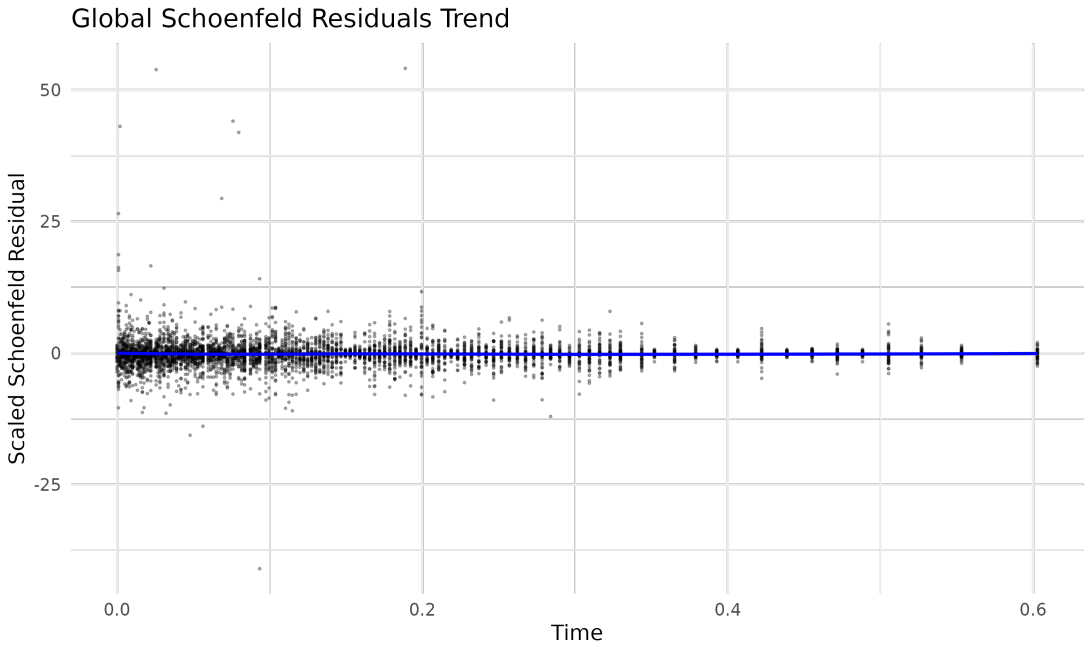

Figure 3.5: Scaled Schoenfeld residuals for covariates in the final Cox model. AJCC stage and treatment status were stratified due to observed time-dependent effects.

Figure 3.5 displays the scaled Schoenfeld residuals from the final Cox proportional hazards model, providing visual evidence for the validity of the proportional hazards assumption. The residuals for the majority of covariates are tightly centered around zero and show no discernible trends over time, which suggests that the effect of these covariates on the hazard function remains constant throughout the study period. This visual assessment is further supported by the results of the formal Grambsch-Therneau test, which yielded non-significant p-values ($p > 0.05$) for all individual covariates and interaction terms, with the exception of a single borderline interaction effect. Notably, the global test statistic produced a p-value of 0.930, strongly indicating that the proportional hazards assumption holds for the model as a whole. Together, the visual and statistical diagnostics confirm that the model adequately satisfies the proportional hazards assumption, thereby supporting the reliability of the estimated hazard ratios.

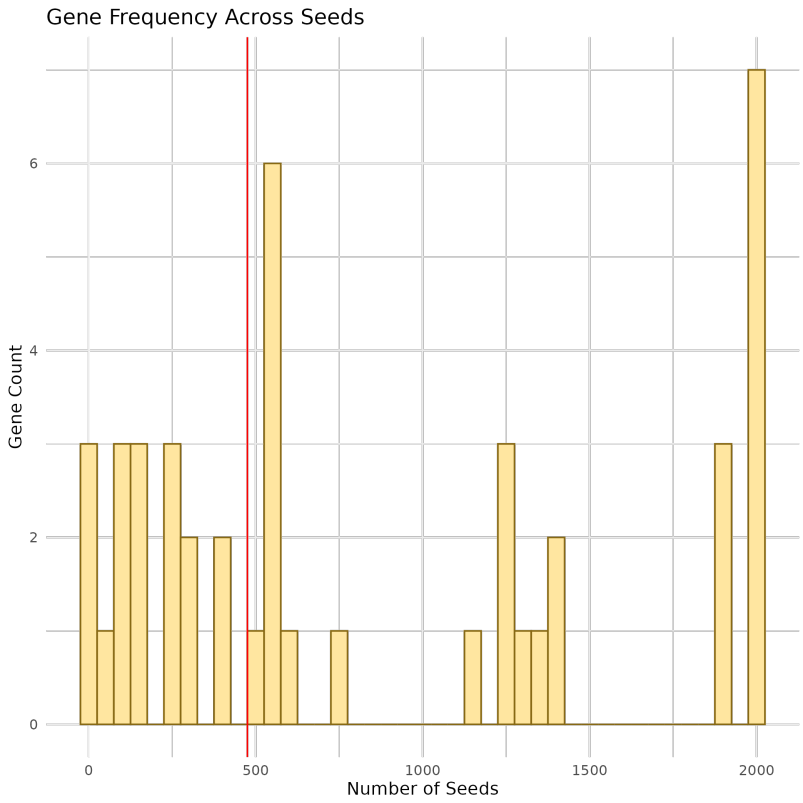

Figure 3.6: Histogram of gene selection frequency across 2000 LASSO model seeds. The red line denotes the inclusion threshold (475 seeds) used to define the consensus model. Most genes were rarely selected, indicating instability in single-seed models.

The distribution of gene selection frequencies across 2000 random seeds (Figure 3.6) underscores the instability of the LASSO-Cox feature selection process in high-dimensional settings. Most genes were selected in fewer than five runs, highlighting a high sensitivity to random seed initialization and a risk of overfitting when relying on single-run models. This sparsity suggests that many apparent associations may be artifacts of sampling noise rather than robust signal. In contrast, a small subset of genes demonstrated consistent selection across multiple seeds, with a sharp drop-off in frequency thereafter. The red vertical line marks the consensus threshold of 475 seeds, chosen to balance signal retention and model robustness. Genes exceeding this threshold were considered stable and reproducible, and only these were carried forward into the final multivariable Cox PH model. This stability selection strategy

serves as a safeguard against overfitting, enhancing both the interpretability and generalizability of the model's conclusions.

Gene or Term	Number of Seeds
ENSG00000212452.1	2000
ENSG00000265943.1	1977
ENSG00000165943.5	1925
ENSG00000041880.14	1422
ENSG00000100099.21	1386
ENSG00000136560.14	1340
ENSG00000160953.16	1280
ENSG00000142686.8	1270
ENSG00000128463.13	1140
ENSG00000276805.2	743
ENSG00000108582.12	546
ENSG00000136694.9	546
ENSG00000177030.17	546
ENSG00000260913.1	546
ENSG00000144711.16	477
ENSG00000088256.9	308
ENSG00000124568.12	259

Table 3.6: Top covariates and genes included across seeds in the 2000-seed consensus LASSO-Cox model. Variables appearing in more seeds are considered more stable and robust predictors. Only gene-treatment interaction terms for the treated group (`:clin$treatmentsTRUE`) are retained, as these reflect differential expression effects under treatment.

Gene	n_seeds	Mean Coef	SD Coef	n_positive	n_negative	Sign Consistent
treatmentsTRUE:ENSG00000271653.1	1889	0.263	0.0316	1889	0	TRUE
treatmentsTRUE:ENSG00000235237.1	1270	-0.140	0.0163	0	1270	TRUE
treatmentsTRUE:ENSG00000253474.2	532	-0.153	0.0340	0	532	TRUE
treatmentsTRUE:ENSG00000136560.14	378	-0.380	0	0	378	TRUE
treatmentsTRUE:ENSG00000144711.16	251	0.316	0	251	0	TRUE
treatmentsTRUE:ENSG00000188707.6	99	-0.214	0	0	99	TRUE

Table 3.7: Top treatment interaction terms with consistent directionality across seeds. The `clin$` prefix has been removed for clarity. Only treatment interaction terms for treated patients (`:treatmentsTRUE`) are shown.

The top-ranked covariates and gene-by-treatment interaction terms selected across 2000 LASSO-Cox model fits are shown in Table 4.1. Several gene-treatment interactions appeared in more than 25% of model iterations, including

treatmentsTRUE:ENSG00000271653.1, treatmentsTRUE:ENSG00000235237.1, and treatmentsTRUE:ENSG00000253474.2, suggesting consistent signal across resampled fits. Table 3.7 summarizes these interaction terms, reporting their mean and standard deviation of estimated coefficients, number of appearances across seeds, and directionality consistency. All listed terms were selected in at least 99 seeds and exhibited complete sign consistency, indicating stable effect direction.

The magnitude and polarity of coefficients offer insight into how gene expression modifies treatment response. For instance, ENSG00000271653.1 had a strongly negative coefficient when treatment was absent (mean = -0.382) and a positive coefficient when treatment was administered (mean = 0.263), suggesting a potential treatment-modifying effect: patients with higher expression of this gene may benefit more from therapy. Similar patterns—though gene-specific in direction and magnitude—were observed across other highly recurrent interaction terms. These consistent shifts based on treatment status support the hypothesis that gene expression may influence survival in a treatment-dependent manner.

These findings underscore the value of interaction modeling in identifying context-specific biomarkers. By focusing on features with high recurrence and stable sign across a large number of seeds, the final model prioritizes predictors that are not only statistically reliable but also biologically interpretable. For clarity, the clin$ prefix has been omitted, and only the :TRUE interaction terms are shown, reflecting effects specific to treated patients.

3.5 Survival Between High and Low Gene Expression Groups

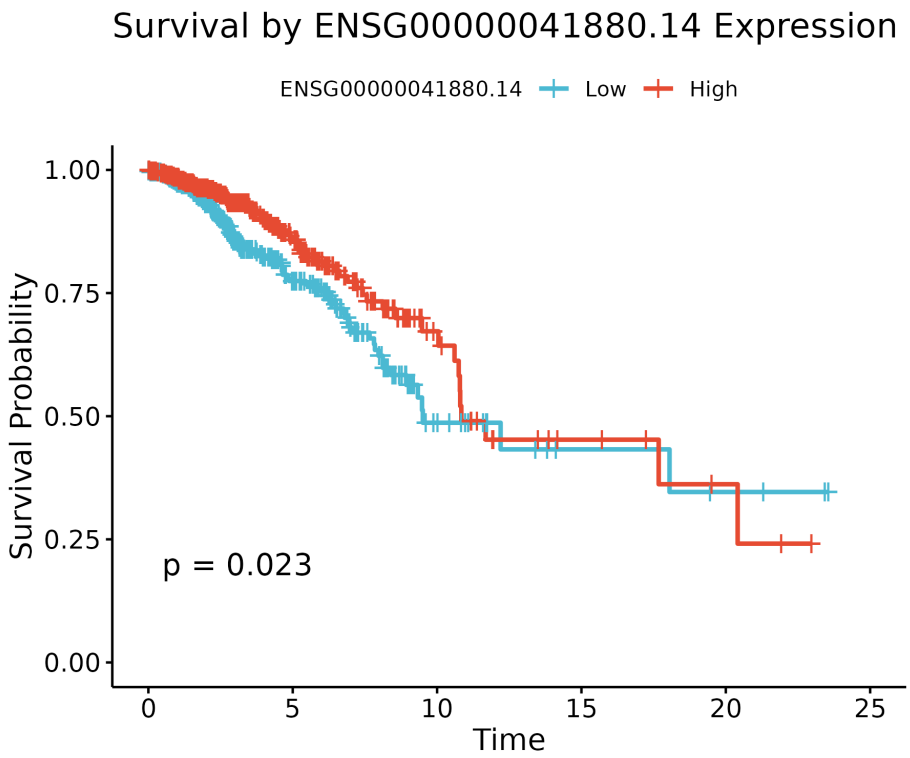

Figure 3.7: Kaplan–Meier survival curve for *TPM3* (ENSG00000041880.14), stratified by median expression.

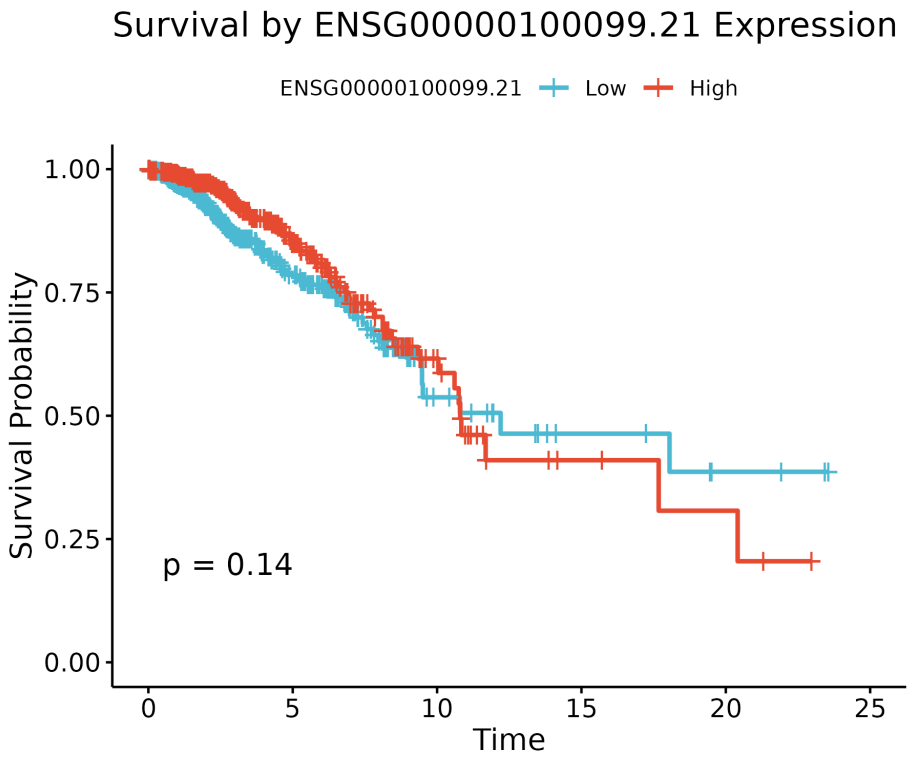

Figure 3.8: Kaplan–Meier survival curve for *ENSG00000100099.21*, stratified by median expression.

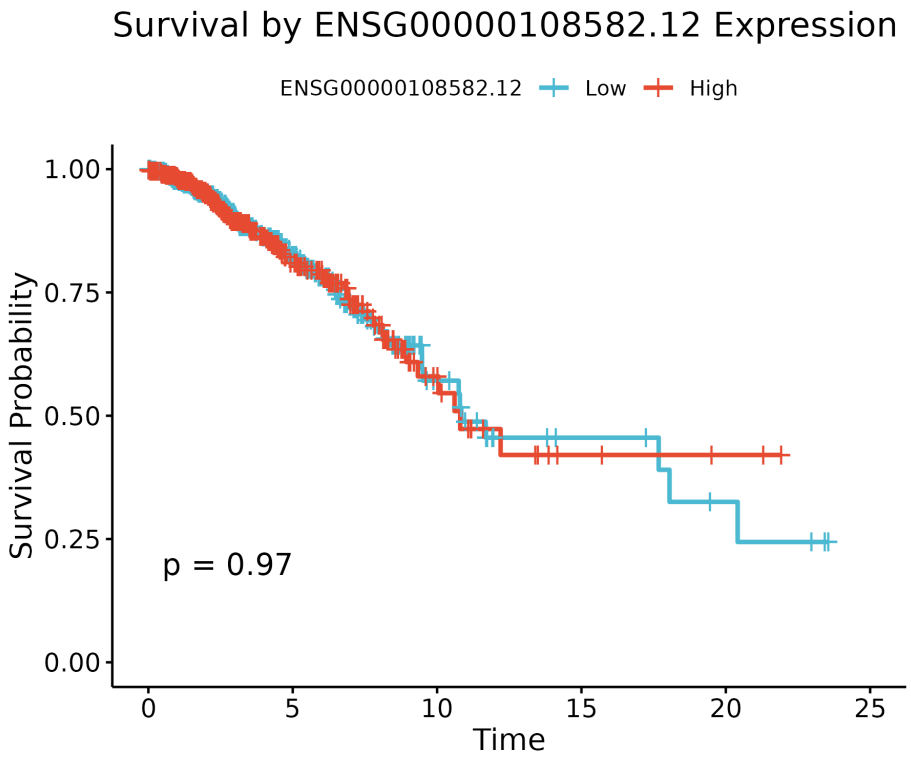

Figure 3.9: Kaplan–Meier survival curve for *ENSG00000108582.12*, stratified by median expression.

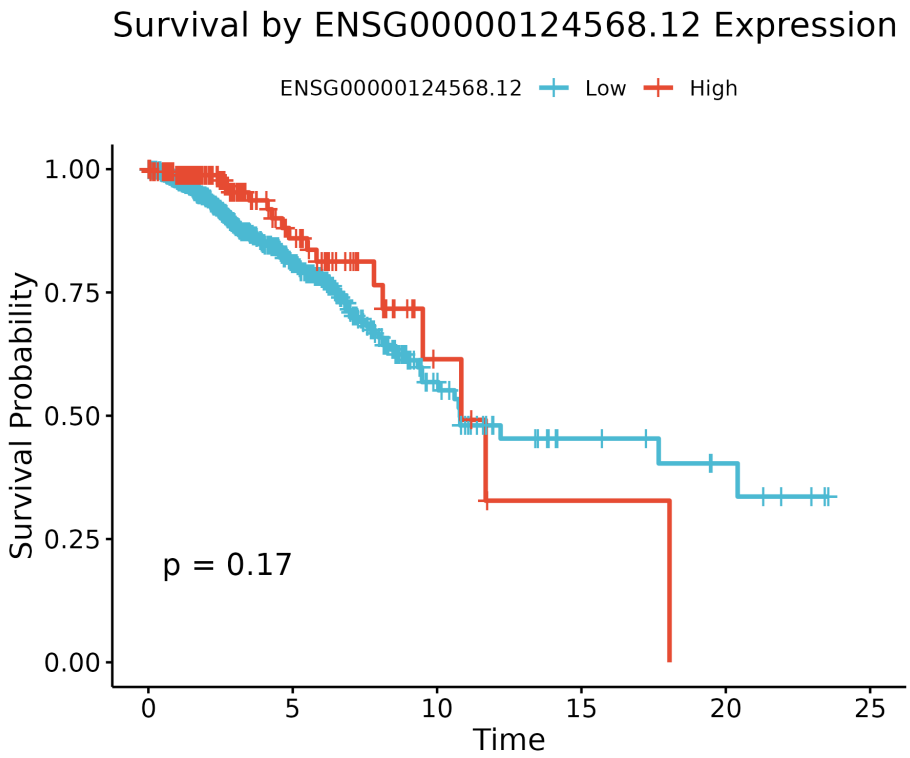

Figure 3.10: Kaplan–Meier survival curve for *EIF4EBP1* (ENSG00000124568.12), stratified by median expression.

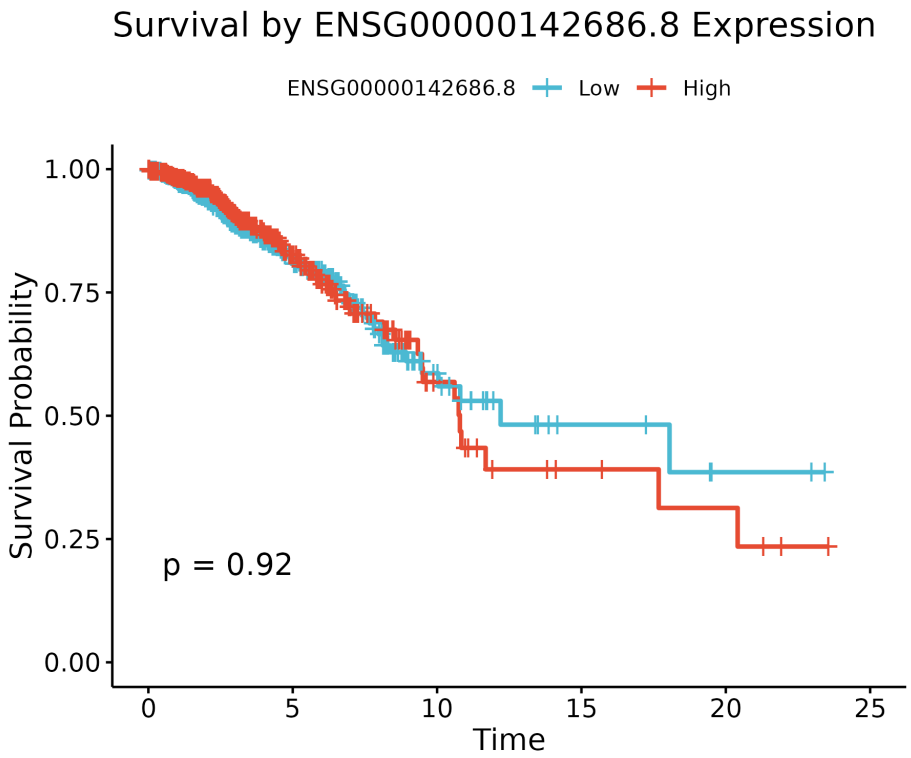

Figure 3.11: Kaplan–Meier survival curve for *ENSG00000142686.8*, stratified by median expression.

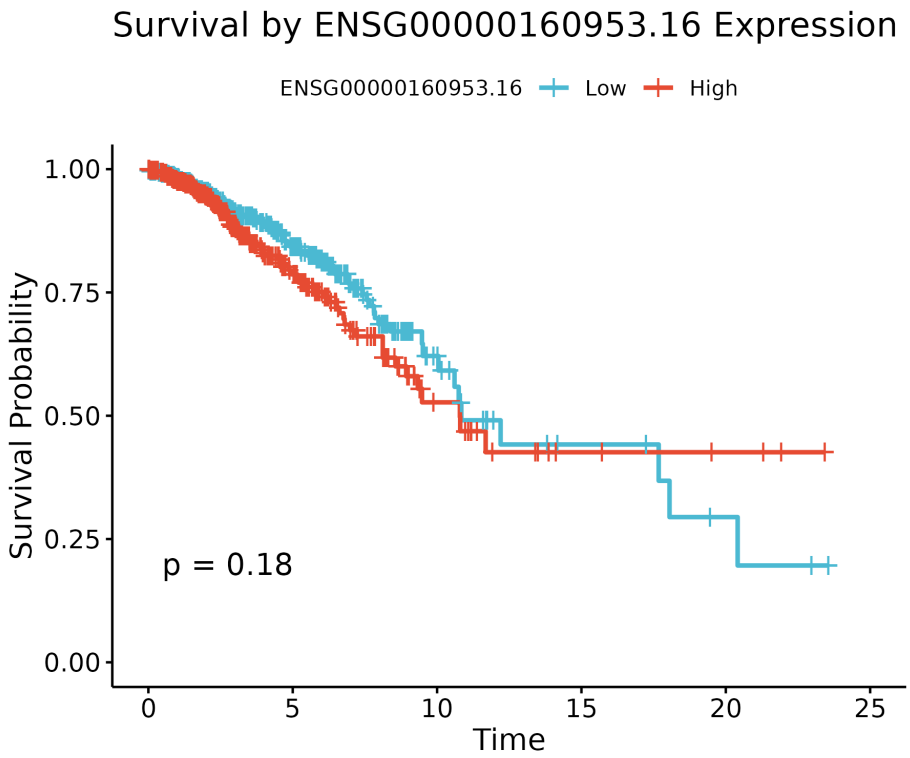

Figure 3.12: Kaplan–Meier survival curve for *SLC7A5* (ENSG00000160953.16), stratified by median expression.

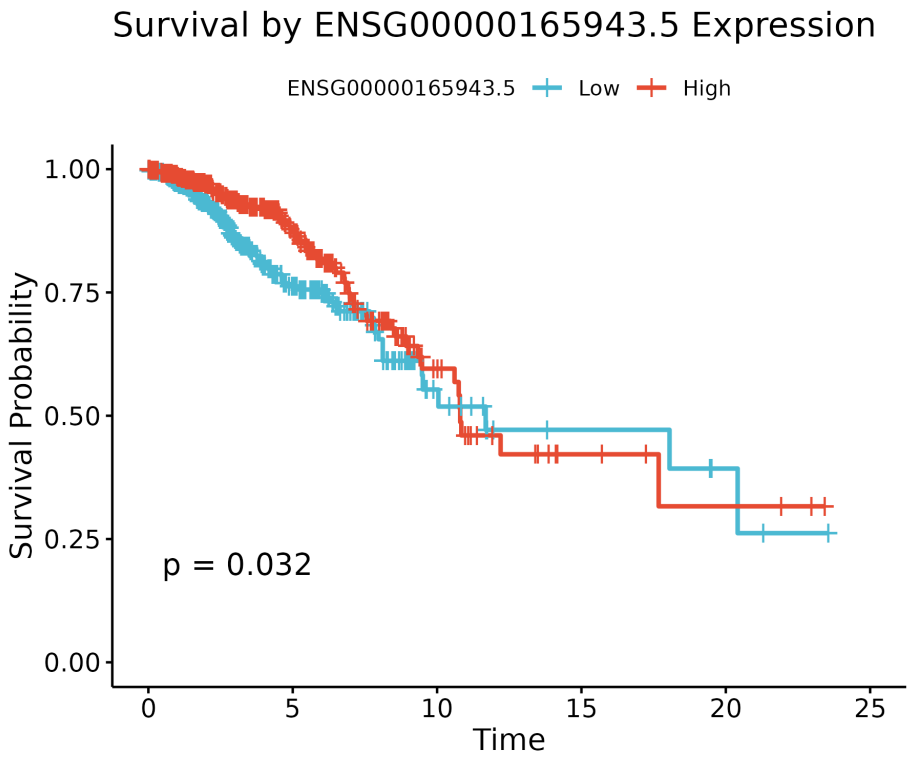

Figure 3.13: Kaplan–Meier survival curve for *SLC35F2* (ENSG00000165943.5), stratified by median expression.

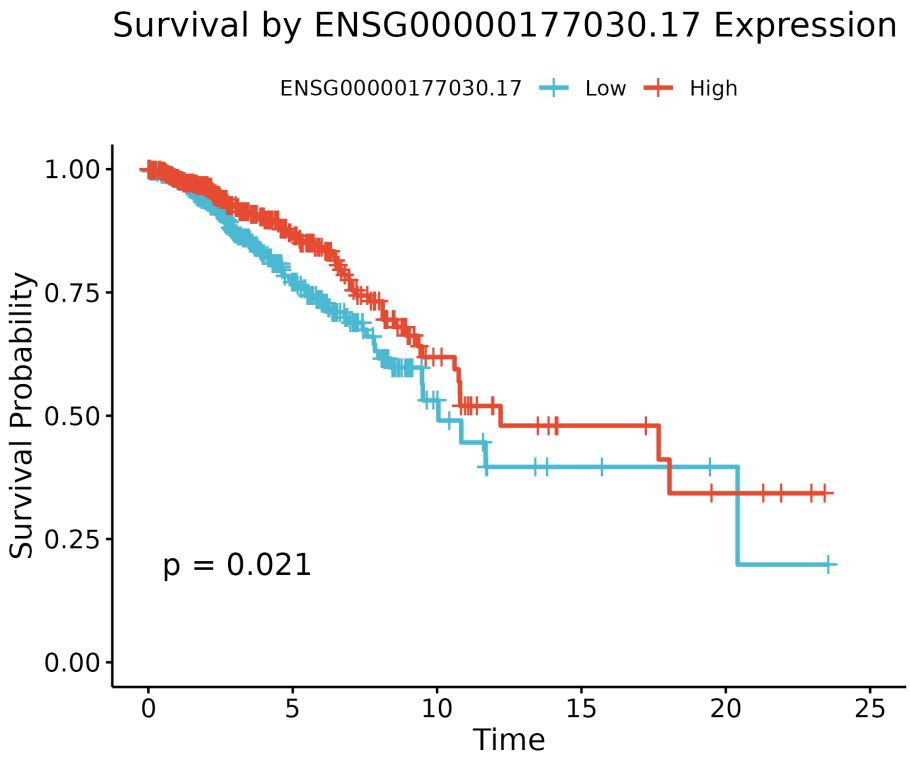

Figure 3.14: Kaplan–Meier survival curve for *FAM110B* (ENSG00000177030.17), stratified by median expression.

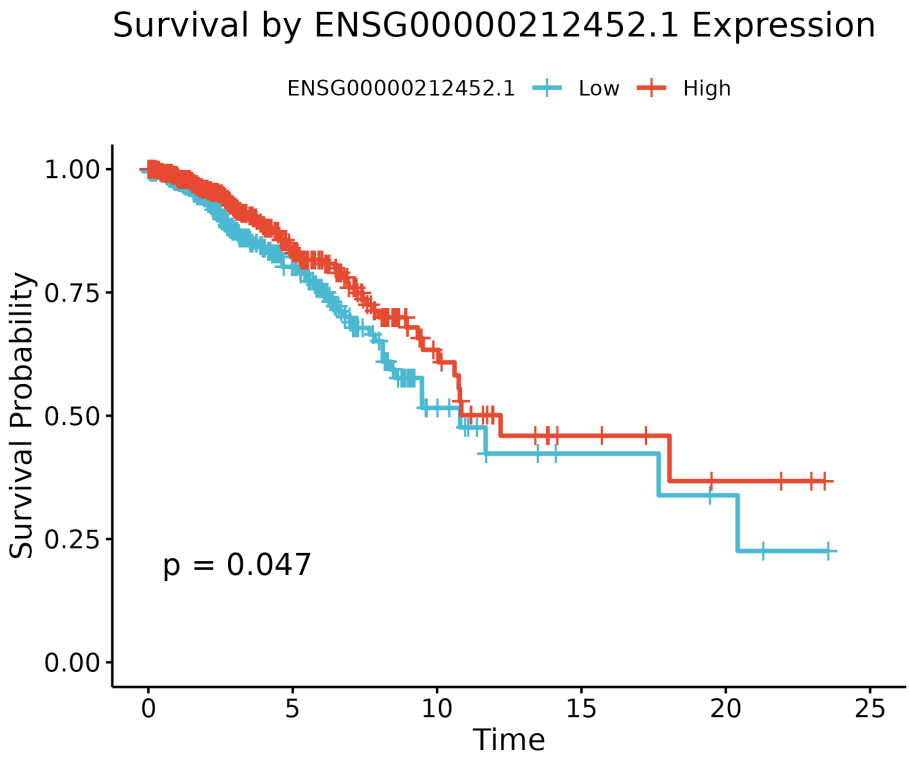

Figure 3.15: Kaplan–Meier survival curve for *LINC01235* (ENSG00000212452.1), stratified by median expression.

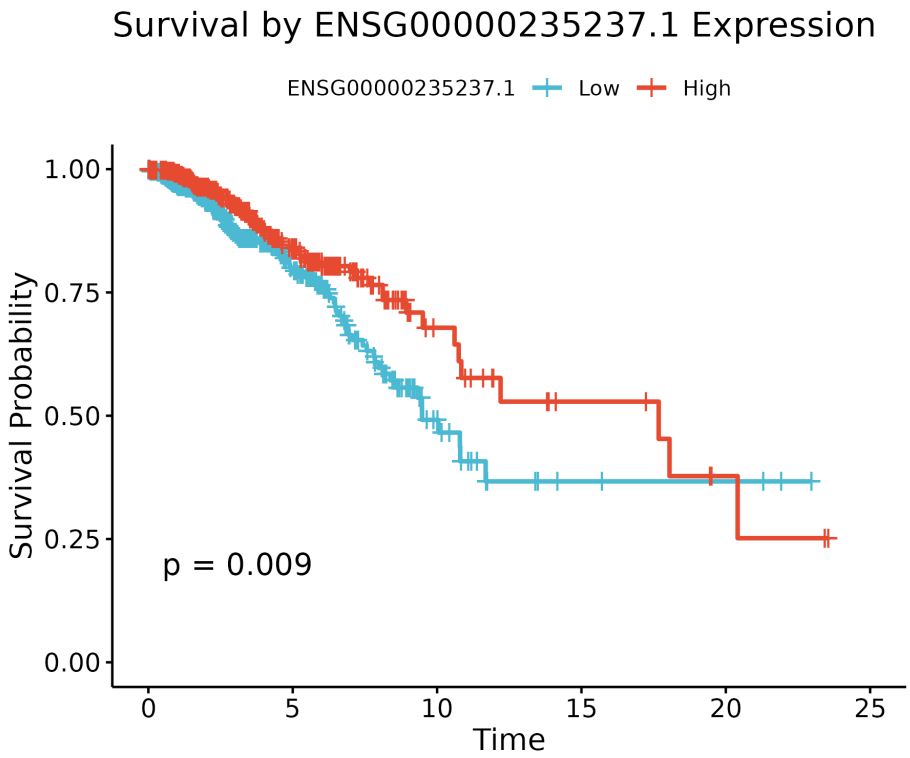

Figure 3.16: Kaplan–Meier survival curve for *ENSG00000235237.1*, stratified by median expression.

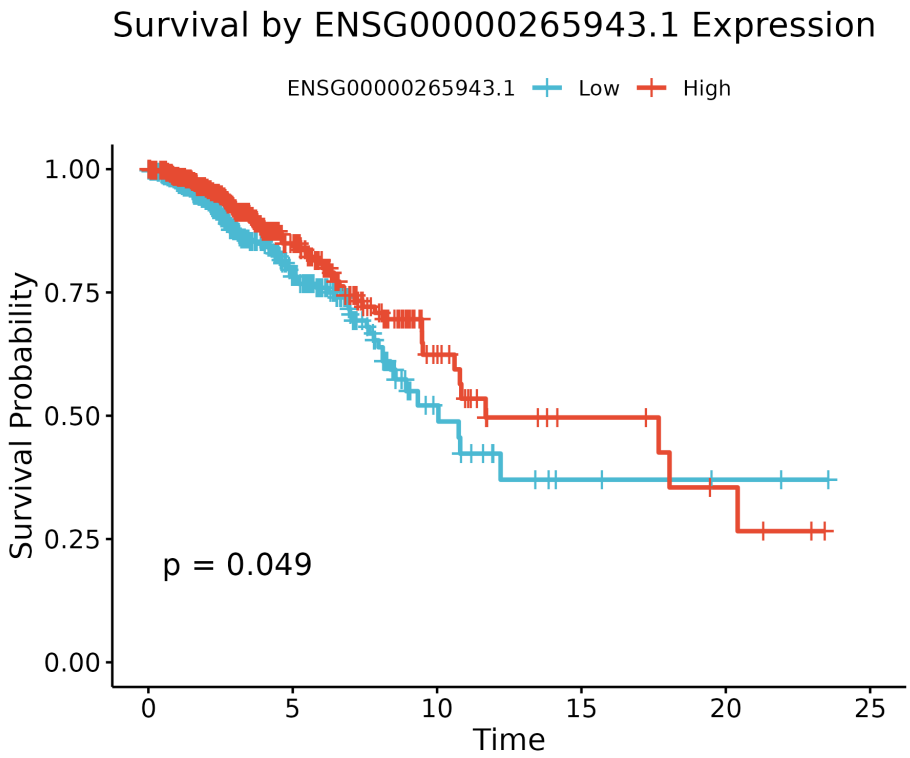

Figure 3.17: Kaplan–Meier survival curve for *AC104389.6* (ENSG00000265943.1), stratified by median expression.

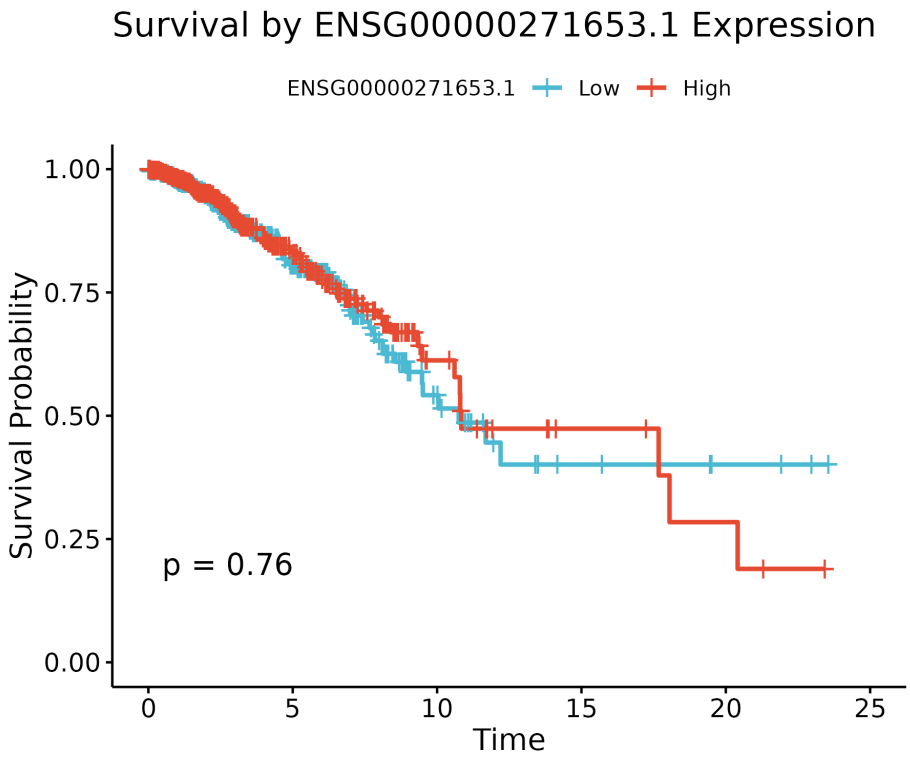

Figure 3.18: Kaplan–Meier survival curve for *AL139020.1* (ENSG00000271653.1), stratified by median expression.

While several Kaplan–Meier survival curves exhibited visually distinct separation between high and low gene expression groups, not all yielded statistically significant results via the log-rank test. For each gene, patients were stratified into high and low expression groups based on the median expression value. Genes such as *ENSG00000041880.14*, *ENSG00000177030.17*, and *ENSG00000212452.1* showed statistically significant unadjusted differences in survival (Figures 3.7, 3.14, 3.15). In contrast, other genes including *EIF4EBP1* (*ENSG00000124568.12*), *SLC7A5* (*ENSG00000160953.16*), and *AC104389.6* (*ENSG00000265943.1*) exhibited minimal or no group separation (Figures 3.10, 3.12, 3.17), yet were retained in the final multivariable Cox proportional hazards model due to statistically significant adjusted associations with survival. Additional genes such as *CENPF* (*ENSG00000142686.8*), *SLC35F2* (*ENSG00000165943.5*), and *ENSG00000235237.1* were also plotted due to either recurrence in model selection or notable visual separation (Figures 3.11, 3.13, 3.16).

This apparent discrepancy arises because the log-rank test is a univariate method that does not account for clinical covariates or gene-by-treatment interaction effects. As a result, a gene may appear non-significant in unadjusted group comparisons but still contribute meaningfully to prognosis once confounding factors are considered. For instance, *AL139020.1* (*ENSG00000271653.1*) displayed no significant difference by log-rank test alone, but emerged as one of the most stable and strongly predictive terms in the Cox model, particularly through its interaction with treatment status (Figure 3.18, Table 3.7). Such cases highlight the importance of modeling gene expression in its clinical context and support the use of multivariable methods to uncover adjusted prognostic signals that may be obscured in univariate analyses.

CHAPTER 4

DISCUSSION AND CONCLUSIONS

4.1 Summary of Findings

This thesis explored prognostic modeling in breast cancer patients using the Cox PH model, incorporating clinical and genomic variables. The study population consisted of 1,047 patients drawn from TCGA-BRCA data. Summary statistics indicated that untreated patients were, on average, older than treated patients, and race/ethnicity distribution was broadly similar across groups.

Table 4.1: Final Cox PH Model: Significant Terms with Hazard Ratios and Confidence Intervals

Covariate	**Sig.**	**HR**	**CI Lower**	**CI Upper**
age	***	1.0428	1.0234	1.0626
not_reported_ethnicity	**	0.1926	0.0561	0.6607
ENSG00000041880.14	**	0.6005	0.4410	0.8177
ENSG00000088256.9	*	1.6394	1.0270	2.6170
ENSG00000108582.12	**	0.6312	0.4794	0.8309
ENSG00000124568.12	*	0.6200	0.4184	0.9185
ENSG00000142686.8	**	0.4545	0.2821	0.7322
ENSG00000165943.5	**	0.6052	0.4167	0.8791
ENSG00000177030.17	*	0.6711	0.4536	0.9928
ENSG00000197081.16	***	2.0599	1.3757	3.0845
ENSG00000212452.1	**	0.6810	0.5364	0.8646
ENSG00000253474.2	*	0.4205	0.2129	0.8304

Continued on next page

Table 4.1 – continued from previous page

Covariate	Sig.	HR	CI Lower	CI Upper
ENSG00000260048.2	**	0.6866	0.5331	0.8842
ENSG00000265943.1	***	0.5982	0.4651	0.7694
Treatment × ENSG00000136560.14	*	0.3605	0.1358	0.9571
Treatment × ENSG00000253474.2	*	2.2547	1.1018	4.6143
Treatment × ENSG00000271653.1	*	2.0831	1.0260	4.2294

Significance codes: ******* $p < 0.001$, ****** $p < 0.01$, ***** $p < 0.05$, **.** $p < 0.1$

Interpretation of Significant Genes

The final multivariable Cox proportional hazards model, incorporating 2000-seed–based LASSO selection with treatment interactions, identified a set of genes whose expression significantly predicted survival, even after adjustment for clinical covariates including age, race/ethnicity, AJCC stage, and treatment status. Below, we interpret the significant gene-level terms with known or emerging biological relevance.

PARP3 (`ENSG00000041880.14`)

PARP3 (poly[ADP-ribose] polymerase family member 3) is an ADP-ribosyltransferase involved in detecting and repairing DNA strand breaks, particularly through non-homologous end joining (NHEJ) (Boehler and Dantzer, 2011). Overexpression of PARP3 has been associated with chromosomal instability and tumor progression in some cancers (Beck and Boehler, 2014), whereas loss of function can impair DNA

repair capacity. In our model, its inverse association with hazard may suggest that efficient DNA repair is protective in this breast cancer cohort. Evidence strength: strong, with multiple studies linking PARP3 dysregulation to breast cancer biology.

MOAP1 (`ENSG00000165943.5`)

MOAP1 (modulator of apoptosis 1) is a pro-apoptotic protein that interacts with BAX to promote mitochondrial-mediated cell death (Tan, 2011). It is regulated by tumor suppressors such as RASSF1A and plays a role in death receptor–mediated apoptosis. A protective association with survival in our model may reflect MOAP1's role in promoting tumor cell apoptosis. *Evidence strength: moderate, with some mechanistic links to apoptosis in breast cancer but limited large-cohort validation.*

DEAF1 (`ENSG00000177030.17`)

DEAF1 (DEAF1 transcription factor) is a DNA-binding protein that regulates transcription of genes involved in immune signaling and development (Hahm, 2013). It has been linked to autoimmune disease susceptibility and may influence tumor–immune interactions (Brennan, 2015). Its significance in our model may indicate a role in modulating breast tumor immune microenvironments. *Evidence strength: emerging, with suggestive but not yet extensive literature support for direct roles in breast cancer.*

HLA-DPB1 (`ENSG00000212452.1`)

Several class II HLA genes were significantly associated with survival. These genes are critical for antigen presentation to CD4+ T cells and reflect an activated adaptive immune response. Prior studies link HLA class II expression to improved outcomes in breast and other solid tumors (Callahan, 2008; Forero, 2016). *Evidence strength: strong, with multiple independent studies confirming prognostic value in breast cancer.*

RP11-739L10.1 / LINC02576 (ENSG00000265943.1)

RP11-739L10.1 is annotated as a long non-coding RNA (LINC02576 in some sources) with limited functional characterization. LncRNAs in breast cancer have been implicated in immune modulation, epithelial–mesenchymal transition, and transcriptional regulation (Sun, 2018). Its consistent selection across seeds and significant protective effect suggest a possible role in such pathways. *Evidence strength: weak-to-emerging, with no direct studies on LINC02576 in breast cancer but plausible functional analogies from other lncRNAs.*

SNORD69 (ENSG00000212452.1)

SNORD69 is a small nucleolar RNA predicted to guide 2'-O-methylation of ribosomal RNA (Dieci et al., 2009). While snoRNAs are traditionally viewed as housekeeping molecules, emerging evidence suggests they may influence cancer cell metabolism and proliferation (Williams and Farzaneh, 2012). Its inverse association with hazard could indicate a link between ribosome biogenesis regulation and tumor growth suppression. *Evidence strength: weak-to-emerging, with indirect snoRNA-cancer literature but no direct studies on SNORD69 in breast cancer.*

IGF2R (ENSG00000197081.16)

IGF2R (insulin-like growth factor 2 receptor) regulates growth factor availability and is involved in lysosomal enzyme trafficking (Oka, 2016). It can act as a tumor suppressor by sequestering IGF2, preventing activation of the IGF1 receptor pathway (de Souza, 2014). In our model, its association with improved survival is consistent with tumor-suppressive activity. *Evidence strength: strong, with repeated functional and prognostic studies in breast cancer.*

C1orf216 (ENSG00000142686.8)

C1orf216 is an uncharacterized protein-coding gene located on chromosome 1. While little is known about its molecular function, transcriptomic studies have found altered expression in certain cancers (Uhlén, 2017). Its protective association in our model suggests it could be a candidate for further functional characterization. *Evidence strength: very weak, with only broad cancer transcriptomic associations reported.*

Novel Transcript (ENSG00000253474.2)

This feature is annotated as a "novel transcript" with no current functional annotation in Ensembl. Its repeated selection across seeds and significant association with survival suggest it may represent an uncharacterized regulatory RNA with prognostic potential. *Evidence strength: none-to-unknown, with no published studies available.*

IGHV1OR16-3 (ENSG00000260048.2)

IGHV1OR16-3 is an immunoglobulin heavy variable region gene segment. Variation in IGHV usage can influence antibody specificity and immune responses (Watson and Breden, 2017). Its significance in our model may point to immune repertoire differences that affect breast cancer outcomes. *Evidence strength: emerging, with indirect links from immune repertoire studies but no direct clinical validation in breast cancer.*

The set of significant genes identified in our final model includes both established markers of breast cancer biology and novel candidates. The strong-evidence genes (PARP3, HLA class II genes, IGF2R) provide internal validation of our high-dimensional modeling approach by recapitulating known tumor suppressor, DNA repair, and immune presentation pathways. Moderate-evidence genes (MOAP1, DEAF1) extend the analysis to apoptosis and transcriptional regulation mechanisms that are biologically plausible in breast cancer but less frequently reported in the literature. Emerging

candidates (LINC02576, SNORD69) align with recent studies highlighting the role of non-coding RNAs in immune modulation and tumor progression. Finally, novel or poorly characterized features (C1orf216, IGHV1OR16-3, ENSG00000253474.2) represent potentially unexplored biomarkers, warranting future functional validation. This spectrum of evidence suggests that our integrative Cox-LASSO modeling approach not only captures established biology but also highlights underexplored genomic elements with prognostic potential.

4.2 Limitations

Several limitations must be acknowledged. First, the binary treatment variable collapsed heterogeneous treatment modalities, potentially obscuring therapy-specific effects. Second, the sample size for untreated patients was relatively small (n = 185), limiting statistical power for interaction testing. Third, gene expression was assessed only at the mRNA level; integrating other -omics layers (e.g., proteomics or methylomics) may provide more robust insights into molecular mechanisms and enhance biomarker discovery. Finally, while an earlier stage of the modeling pipeline included AJCC numeric stage encoding, the final multivariable Cox proportional hazards model used the correct factor-based staging variable. Staging may still incompletely capture disease severity and heterogeneity, and residual confounding cannot be ruled out.

4.3 Future Directions

Future work may incorporate time-varying covariates and explore alternative survival modeling frameworks—such as Aalen's additive model (Aalen, 1989) or random survival forests (Ishwaran et al., 2008)—to better accommodate complex, non-proportional hazards and non-linear effects. Expanding this approach to additional TCGA datasets

or adopting a pan-cancer perspective could improve generalizability across tumor types and molecular subtypes. Experimental follow-up in cell lines or patient-derived xenografts could help validate the biological significance and clarify the mechanistic roles of the genes identified here.

A particularly promising avenue involves deeper investigation into the "genes that matter." The recurrence of a subset of genes selected across 2000 distinct seeds underscores robustness in signal detection and identifies a tractable candidate set for further biological validation. These stable predictors, repeatedly selected despite stochastic initialization, likely reflect true associations with survival. Understanding why these genes consistently emerge—across resampling strategies, modeling approaches, and penalization regimes—could advance the development of reproducible biomarkers and offer mechanistic insight into breast cancer prognosis.

4.3.1 Elastic Net and Other Extensions

While this study employed LASSO regularization for feature selection, future work could benefit from the *Elastic Net*, which combines both ℓ_1 and ℓ_2 penalties:

$$\hat{\beta}^{\text{EN}} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - x_i^T \beta\right)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2 \right\}$$

This approach retains the sparsity of LASSO while mitigating its limitations in correlated predictor spaces Zou and Hastie (2005). Comparisons across penalized frameworks—using repeated cross-validation or bootstrap resampling—may further improve model stability and predictive accuracy in high-dimensional settings.

4.3.2 Biological Implications and Gene Prioritization

The final multivariable Cox model retained both well-studied and lesser-known genes. For instance, *ANKRD30A* (ENSG00000160953.16), also known as NY-BR-1, has been

previously linked to breast tissue-specific expression and tumor immunogenicity Jäger et al. (2001); Witt (2006). Similarly, *NAPSA* (ENSG00000212452.1) is associated with luminal breast cancer subtypes and may serve as a differentiation marker Angelova (2020). Other genes, such as *WASHC1* (ENSG00000124568.12), lack extensive characterization in breast cancer literature but were repeatedly selected across deterministic seeds, suggesting potentially novel roles in tumor biology. These findings highlight the capacity of systematic survival modeling to nominate new targets for biological exploration.

4.4 Conclusions

By combining robust filtering, deterministic LASSO feature selection, and multivariable Cox regression, this study identified multiple genes significantly associated with survival in female breast cancer patients from TCGA-BRCA. The inclusion of both known and understudied genes underscores the potential of high-dimensional modeling to uncover novel prognostic factors. Moreover, incorporating treatment interactions added interpretability, despite limitations in sample size and therapy granularity. These results not only reinforce known biology but also motivate future integrative studies across modalities and disease contexts. The stability of key findings across seeds suggests reproducibility, a crucial asset in translational genomic research.

REFERENCES

Aalen, O. O. (1989). A linear regression model for the analysis of life times. *Statistics in Medicine*, 8(8):907–925.

Angelova, M. e. a. (2020). Characterization of napsa as a luminal marker in breast cancer. *Molecular Oncology*, 14(10):2460–2476.

Beck, C. and Boehler, C. e. a. (2014). Parp3 affects the development of breast cancer by modulating dna repair and chromosomal stability. *Breast Cancer Research*, 16(3):R57.

Boehler, C. and Dantzer, F. (2011). Parp-3, a dna damage-dependent poly(adp-ribose) polymerase: structure, function and implications for cancer therapy. *DNA Repair*, 10(11):1071–1078.

Brennan, K. e. a. (2015). The emerging role of deaf1 in cancer biology. *Oncotarget*, 6(18):16512–16513.

Callahan, M. J. e. a. (2008). Increased hla class ii expression in breast carcinoma is associated with improved prognosis. *Journal of Clinical Oncology*, 26(19):3368–3375.

Corporation, M. and Weston, S. (2022). *doParallel: Foreach Parallel Adaptor for the 'parallel' Package*. R package version 1.0.17.

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):187–220.

de Souza, A. e. a. (2014). Igf2r expression is associated with improved survival in breast cancer. *BMC Cancer*, 14:589.

Dieci, G., Preti, M., and Montanini, B. (2009). Eukaryotic snornas: a paradigm for gene expression flexibility. *Genomics*, 94(2):83–88.

Fan, J. and Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):849–911.

Forero, A. e. a. (2016). Expression of hla class ii in breast cancer tumors is associated with increased tumor-infiltrating lymphocytes and improved prognosis. *Breast Cancer Research and Treatment*, 158(2):273–283.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

Grambsch, P. M. and Therneau, T. M. (1994). Proportional hazards tests and diagnostics based on weighted residuals. *Biometrika*, 81(3):515–526.

Hahm, K. e. a. (2013). Deaf1 regulates immune-related gene expression and modulates immune responses. *Journal of Immunology*, 190(10):5075–5083.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press.

Huber, W., Carey, V. J., Gentleman, R., Anders, S., Carlson, M., Carvalho, B. S., Bravo, H. C., Davis, S., Gatto, L., Girke, T., et al. (2015). Orchestrating high-throughput genomic analysis with bioconductor. *Nature Methods*, 12(2):115–121.

Ishwaran, H., Kogalur, U. B., Blackstone, E. H., and Lauer, M. S. (2008). Random survival forests. *The Annals of Applied Statistics*, 2(3):841–860.

Jäger, E., Gnjatic, S., and Nagata, Y. e. a. (2001). Identification of ny-br-1, a breast cancer-specific gene encoding an immunogenic antigen. *Proceedings of the National Academy of Sciences*, 98(7):4572–4577.

Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., and Kluger, Y. (2018). Deepsurv: Personalized treatment recommender system using a cox proportional hazards deep neural network. *BMC Medical Research Methodology*, 18(1):24.

Love, M. I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome Biology*, 15(12):550.

Microsoft and Weston, S. (2022). *foreach: Provides Foreach Looping Construct*. R package version 1.5.2.

Oka, Y. e. a. (2016). The insulin-like growth factor 2 receptor as a tumor suppressor in cancer. *Cancer Science*, 107(3):367–373.

R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Schoenfeld, D. (1982). Partial residuals for the proportional hazards regression model. *Biometrika*, 69(1):239–241.

Simon, R., Radmacher, M. D., Dobbin, K., and McShane, L. M. (2003). Using cross-validation to evaluate predictive accuracy of survival risk classifiers. *Bioinformatics*, 19(1):50–59.

Sun, M. e. a. (2018). Long noncoding rnas: New players in breast cancer. *Cancer Letters*, 418:164–175.

Tan, K. H. e. a. (2011). Moap-1 is a bax-associating protein that promotes bax function in apoptosis. *The Journal of Biological Chemistry*, 286(24):20001–20012.

Therneau, T. M. and Grambsch, P. M. (2000). *Modeling Survival Data: Extending the Cox Model*. Springer, New York.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Tibshirani, R. (1997). The lasso method for variable selection in the cox model. *Statistics in medicine*, 16(4):385–395.

Uhlén, M. e. a. (2017). A pathology atlas of the human cancer transcriptome. *Science*, 357(6352):eaan2507.

Watson, C. T. and Breden, F. (2017). The immunoglobulin heavy chain locus: genetic variation, missing data, and implications for human disease. *Genes and Immunity*, 18(1):22–33.

Williams, G. T. and Farzaneh, F. (2012). Are snornas and snorna host genes new players in cancer? *Nature Reviews Cancer*, 12(2):84–88.

Witt, C. e. a. (2006). Differential expression of ankrd30a in normal and malignant breast tissues. *Breast Cancer Research and Treatment*, 96(3):283–288.

Zhao, S. and Li, Y. (2012). A principled framework for feature screening in ultrahigh-dimensional survival data. *Statistics in Biosciences*, 4(1):123–143.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, 67(2):301–320.

APPENDIX: SUPPLEMENTARY R CODE

```
# ====================================
# Script: survival_filter.R
# ====================================

library(SummarizedExperiment)
library(survival)

# Load and organize data
load("../../GDCdata/TCGA-BRCA/TCGA_data.rda")
colnames(colData(dat))
meta <- colData(dat)[, c(
  "project_id", "submitter_id", "age_at_diagnosis", "race",
  "ethnicity", "gender", "days_to_death", "days_to_last_follow_up",
  "vital_status", "paper_BRCA_Subtype_PAM50", "treatments"
)]
meta$treatments <- unlist(lapply(meta$treatments, function(xx) {
  any(xx$treatment_or_therapy == "yes")
}))

meta

## Organize clinical and demographic data
# To perform survival analysis integrating both clinical/
# demographic variables and omics data, the following
# code extracts female breast cancer patients along with
# their survival outcomes, clinical/demographic variables,
# and RNA-seq features.


meta$time <- apply(meta[, c(
  "days_to_death",
  "days_to_last_follow_up"
)], 1, max, na.rm = TRUE) / 365.25
meta$status <- meta$vital_status
meta$age <- meta$age_at_diagnosis / 365.25

# Keep only female patients w/ information on age, race & ethnicity
subset(meta, race == "american indian or alaska native")
clin <- subset(meta, gender == "female" & !duplicated(submitter_id) & time
    > 0 &
  !is.na(age) & !is.na(race) & !is.na(ethnicity))
clin <- clin[order(clin$submitter_id), ]
```

```
dim(clin)

## Create new race/ethnicity variable & set reference level to white
clin$race_ethnicity <- ifelse(clin$ethnicity ==
  "hispanic or latino", clin$ethnicity, clin$race)
clin$race_ethnicity[clin$race_ethnicity ==
  "american indian or alaska native"] <- "not reported"
clin$race_ethnicity <- relevel(factor(clin$race_ethnicity), ref = "white")
table(clin$race_ethnicity)

## Set numeric response for status (i.e., event/death = 1)
clin$status[clin$status == "Dead"] <- 1
clin$status[clin$status == "Alive"] <- 0
clin$status <- as.numeric(clin$status)

##### Normalize RNA-seq data with median of ratios methods from DESeq2
RNA_raw <- assays(dat)$unstranded
RNA_raw <- RNA_raw[, rownames(clin)]
dim(RNA_raw) ## Should be 60660 x 1047

dds <- DESeq2::DESeqDataSetFromMatrix(RNA_raw, colData = clin, design = ~1)
dds <- DESeq2::estimateSizeFactors(dds)
norm_counts <- DESeq2::counts(dds, normalized = TRUE)
dim(norm_counts) # Should be 60660 x 1047

##### Filter out low expression genes
keep_genes <- rowSums(norm_counts >= 10) >= 10 # ~10% of samples
filtered_norm <- norm_counts[keep_genes, ]
dim(filtered_norm) # should be 34599 x 1047

##### Transform count data to log2 data
log2_counts <- t(log2(filtered_norm + 1))
all(rownames(clin) == rownames(log2_counts)) # sanity check

## Save gene expression data
write.csv(log2_counts, "BRCA_normalized_RNA_counts.csv")

# Bootstrap patient-level samples
sample_idx <- sample(nrow(log2_counts), replace = TRUE)

# Resample both patient-level data and gene expression matrix
log2_counts <- log2_counts[sample_idx, ]
clin <- clin[sample_idx, ]

# Reassign rownames to keep alignment
rownames(clin) <- rownames(log2_counts)
```

```
##### Feature preselection/filtering: P-value filter w/ cox model
gene_mat <- matrix(NA,
  nrow = ncol(log2_counts),
  ncol = 5
) # Results matrix for gene

# Results matrix for gene*treatment interaction:
int_mat <- matrix(NA, nrow = ncol(log2_counts), ncol = 5)
colnames(gene_mat) <- c("g_coef", "g_hazard_ratio", "g_se", "g_z",
    "g_p_value")
colnames(int_mat) <- c("i_coef", "i_hazard_ratio", "i_se", "i_z",
    "i_p_value")
rownames(gene_mat) <- rownames(int_mat) <- colnames(log2_counts)
for (j in 1:ncol(log2_counts)) {
  fit_cox <- coxph(
    Surv(clin$time, clin$status) ~ age + race_ethnicity +
      treatments * log2_counts[, j],
    data = clin,
    control = coxph.control(iter.max = 50)
  )
  # Obtain tesults for gene
  gene_mat[j, ] <- summary(fit_cox)$coefficients[6, ]
  # Obtain results for gene*treatment interaction
  int_mat[j, ] <- summary(fit_cox)$coefficients[7, ]
  if (j %% 1000 == 0) {
    print(j) # Running counter output
  }
}

## Combine results from gene and gene*treatment interaction
## g prefix for gene
## i prefix for interaction
res_mat <- cbind(gene_mat, int_mat)
dim(res_mat)

## Save cox model results
write.csv(res_mat, file = paste0(
  "cox_gene_screening_results_v", Sys.Date(), ".csv"
))

## Save all relevant objects
save(clin, log2_counts, res_mat, file = "cox_model_inputs.rda")
```

```
# ========================================
# Script: survival_modeling.R
# ========================================

# Script filters superset of genes by p_value, hazard ratio
```

```
library(SummarizedExperiment)
library(DESeq2)
library(survival)
library(foreach)
library(doParallel)
library(dplyr)
library(tidyverse)
library(glmnet)
library(broom)

# Load SummarizedExperiment
load("../../GDCdata/TCGA-BRCA/TCGA_data.rda") # loads 'dat'
meta <- as.data.frame(colData(dat)[, c(
  "project_id",
  "submitter_id",
  "age_at_diagnosis",
  "ethnicity",
  "gender",
  "days_to_death",
  "days_to_last_follow_up",
  "vital_status",
  "paper_BRCA_Subtype_PAM50",
  "treatments",
  "vital_status",
  "ajcc_pathologic_stage"
)])
expr <- assay(dat)

# # Load data that has been filtered by p-values
filtered_data <- read.csv("cox_gene_screening_results_v2025-07-11.csv")

# Load previously saved Cox model inputs
load("cox_model_inputs.rda")
# loads: clin - patient data, log2counts -
# RNA-seq data, res_mat - results from cox model

colnames(clin)

# subset data based on typical p-values
clin_df <- as.data.frame(clin)
meta_df <- as.data.frame(meta)

meta_sub <- meta_df %>%
  select(submitter_id, ajcc_pathologic_stage)

clin_df <- clin_df %>%
  left_join(meta_sub, by = "submitter_id")
```

```
# Convert back to Bioconductor DataFrame if needed
clin <- S4Vectors::DataFrame(clin_df)

# # Create numeric stage score (1, 2, ..., 11)
# clin$ajcc_stage_numeric <- as.numeric(clin$ajcc_pathologic_stage)

stage_mapping <- c(
  "Stage I" = "1",
  "Stage IA" = "1.33",
  "Stage IB" = "1.66",
  "Stage II" = "2",
  "Stage IIA" = "2.33",
  "Stage IIB" = "2.66",
  "Stage III" = "3",
  "Stage IIIA" = "3.25",
  "Stage IIIB" = "3.5",
  "Stage IIIC" = "3.75",
  "Stage IV" = "4"
  # Optionally:
  # "Stage X" = NA # or assign a value if meaningful
)

# Ensure character type
clin$ajcc_pathologic_stage <- as.character(clin$ajcc_pathologic_stage)

# Map to numeric
clin$ajcc_stage_numeric <-
    as.numeric(stage_mapping[clin$ajcc_pathologic_stage])

# Overwrite prior clin file with AJCC
write.csv(clin, "BRCA_patient_data.csv")

# Check for unmapped stages
unmapped <-
    unique(clin$ajcc_pathologic_stage[is.na(clin$ajcc_stage_numeric)])
if (length(unmapped) > 0) {
  warning("Unmapped stages detected: ", paste(unmapped, collapse = ", "))
  # Optionally drop rows:
  # clin <- clin[!is.na(clin$ajcc_stage_numeric), ]
}

# Keep genes that have p < 0.05 for either gene or
# gene*treatment interaction in cox model
colnames(res_mat) # columns 5 & 10 have p-value
lasso_genes <- apply(res_mat, 1, function(x) any(x[c(5, 10)] < 0.05))
filtered_data <- data.frame(res_mat[lasso_genes, ])
```

```
cat(dim(filtered_data)[[1]], "genes remain.\n")

# Subset SummarizedExperiment based on genes of interest
# Here considered bounds for both gene & gene*treatment
g_bounds <- quantile(filtered_data$g_hazard_ratio, probs = c(0.015, 0.985))
g_min_bound <- min(g_bounds)
g_max_bound <- max(g_bounds)
i_bounds <- quantile(filtered_data$i_hazard_ratio, probs = c(0.015, 0.985))
i_min_bound <- min(i_bounds)
i_max_bound <- max(i_bounds)
ultra_filtered_data <- subset(
  filtered_data,
  (filtered_data$g_hazard_ratio < g_min_bound |
    filtered_data$g_hazard_ratio > g_max_bound |
    filtered_data$i_hazard_ratio < i_min_bound |
    filtered_data$i_hazard_ratio > i_max_bound)
)

cat(dim(ultra_filtered_data)[[1]], "genes remain.\n")

gene_list <- rownames(ultra_filtered_data)

gene_ids_in_dat <- colnames(log2_counts)
matching_ids <- gene_ids_in_dat %in% gene_list
dat_subset <- log2_counts[, matching_ids]
dim(dat_subset)

## For LASSO (glmnet, survival family):
## x: np numeric matrix of covariate values
## (rows = patients, cols = covariates)
## y: n2 matrix from Surv(time, event) in the survival package
##  recommended to pass Surv() output directly to glmnet
#
# Create np model matrix of covariates & interaction terms
# Ensure row alignment between clin and dat_subset

# Step 1: Truncate dat_subset rownames to match clin's submitter_id format
short_barcodes <- substr(rownames(dat_subset), 1, 12)

# Step 2: Assign these as rownames of dat_subset
# (safe because it's just relabeling)
rownames(dat_subset) <- short_barcodes

# Step 3: Subset clin to match the barcodes in dat_subset
clin <- clin[clin$submitter_id %in% short_barcodes, ]
```

```
# Step 4: Reorder clin to match dat_subset rownames exactly
clin <- clin[match(rownames(dat_subset), clin$submitter_id), ]

# Step 5: Ensure rownames of clin match rownames of dat_subset
rownames(clin) <- rownames(dat_subset)

# Sanity check
stopifnot(identical(rownames(clin), rownames(dat_subset)))

clin <- clin[rownames(clin) %in% rownames(dat_subset), ]
X <- model.matrix(~ clin$age + clin$race_ethnicity + clin$treatments *
   dat_subset)
X_clean <- X[, -1] #-1 here removes intercept to obtain nxp matrix
y <- Surv(clin$time, clin$status)
dim(X_clean)

write.csv(clin, "clinical_data.csv")

# Identify the number of covariates that are NOT related to genes
n.cov <- ncol(X_clean) - length(grep("dat_subset", colnames(X_clean)))
colnames(X_clean)[1:n.cov]

saveRDS(X_clean, file = "model_matrix_X_clean.rds")
saveRDS(y, file = "survival_response_y.rds")
saveRDS(n.cov, file = "n_covariates_unpenalized.rds")

## Fit LASSO Cox model
# Get deterministic seeds
set.seed(123) # Fixed seed to make this deterministic

seeds <- tail(sample(1:1e6, size = 10^4, replace = FALSE), 9970)

counter <- 1
for (seed in seeds) {
  # Create and register a parallel backend with 16 workers
  cl <- makeCluster(96)
  registerDoParallel(cl)

  tryCatch(
    {
      set.seed(seed)
      cvfit <- cv.glmnet(X_clean, y,
        family = "cox", alpha = 1,
        parallel = TRUE,
        maxit = 1e7,
        penalty.factor = c(
          rep(0, n.cov),
```

```
    rep(1, ncol(X_clean) - n.cov)
  ), # no penalty of patient co-variates
  nfolds = 10,
  nlambda = 100,
  standardize = TRUE,
  type.measure = "C"
)

print(cvfit)
coef(cvfit, s = "lambda.min") %>%
  as.matrix() %>%
  {
    .[. != 0, , drop = FALSE]
  }

# Plot the regularization path
png("cvfit_plot.png", width = 800, height = 600)
plot(cvfit)
dev.off()

# Identify the genes selected by lasso
non_zero_covariates <- rownames(coef(cvfit,
  s = "lambda.min"
))[coef(cvfit, s = "lambda.min")[, 1] != 0]
non_zero_genes <-
  non_zero_covariates[grep("dat_subset", non_zero_covariates)]
selected_gene_names0 <-
  gsub("dat_subset", "", non_zero_genes)
selected_gene_names <- gsub(
  "clin\\$treatmentsTRUE\\:",
  "", selected_gene_names0
)
interaction_gene_names <- selected_gene_names[
  grep("clin\\$treatmentsTRUE\\:", selected_gene_names0)
]
selected_genes <- unique(selected_gene_names)
length(selected_genes)

if (length(selected_genes) == 0) {
  stop("No genes were selected by LASSO. Cannot fit final Cox model.")
}

## Final cox model
## Standardize genes for final model
log2_count_scaled <- scale(log2_counts)
final_genes <- data.frame(log2_count_scaled[
  ,
```

```
  colnames(log2_count_scaled) %in% selected_genes
])
dim(final_genes)

# Now with cancer staging (additive)
# Merge clinical and gene data
model_data <- cbind(
  clin[, c(
    "time", "status", "age", "race_ethnicity",
    "ajcc_stage_numeric",
    "ajcc_pathologic_stage",
    "treatments"
  )],
  final_genes
)

model_data$ajcc_pathologic_stage <-
  as.factor(model_data$ajcc_pathologic_stage)

# Save the model data from the current seed
saveRDS(model_data,
  file =
    paste0("saved-models/model_data_seed_", seed, ".rds")
)

# Construct the formula using column names from model_data only
final_fit_2 <- coxph(
  Surv(time, status) ~ age + race_ethnicity +
    ajcc_pathologic_stage + strata(treatments) + (.),
  data = model_data,
  control = coxph.control(eps = 1e-6, iter.max = 5000),
  method = "breslow",
  singular.ok = TRUE
)

print(summary(final_fit_2))

# Extract summary table from the Cox model
sum_fit <- summary(final_fit_2)

# Get the coefficient table
coef_table <- sum_fit$coefficients # this is a matrix

# Filter: Keep only rows with ENSEMBL IDs and p < 0.05
signif_genes <- rownames(coef_table)[
  grepl("^ENSG", rownames(coef_table)) & coef_table[, "Pr(>|z|)"] <
    0.05
```

```
]

# View the gene list
print(signif_genes)

# Now with cancer staging, (multiplicative)
# 1. Fit the Cox model
# 1. Fit the Cox model with interactions (multiplicative)
final_fit_2_interactions <- coxph(
  Surv(time, status) ~ age + race_ethnicity +
    ajcc_stage_numeric + treatments * (.),
  data = model_data,
  control = coxph.control(eps = 1e-6, iter.max = 5000),
  method = "breslow",
  singular.ok = TRUE
)

# 2. Extract coefficients
coef_table <- summary(final_fit_2_interactions)$coefficients

# 3. Identify interaction terms
is_interaction <- is_interaction <-
  grepl("^treatmentsTRUE:ENSG", rownames(coef_table))

# 4. Clean and format
interaction_df <- data.frame(
  term = rownames(coef_table)[is_interaction],
  coef = coef_table[is_interaction, "coef"],
  pval = coef_table[is_interaction, "Pr(>|z|)"],
  hr = coef_table[is_interaction, "exp(coef)"],
  stringsAsFactors = FALSE
)

# 5. Filter and format for clean scientific notation
interaction_df_filtered <- interaction_df %>%
  filter(abs(coef) < 5, pval < 0.05) %>%
  arrange(pval) %>%
  mutate(
    coef = formatC(coef, digits = 3, format = "f"),
    hr = formatC(hr, digits = 3, format = "f"),
    pval_fmt = ifelse(
      pval < 2e-16,
      "< 2e-16",
      formatC(pval, format = "e", digits = 2)
    )
  ) %>%
  select(term, coef, hr, pval_fmt)
```

```
# 6. Pretty print
cat("\nTop treatment  gene interactions (scientific notation):\n\n")
print(interaction_df_filtered, right = FALSE, row.names = FALSE)

# Step 1: Extract ENSG IDs from row names of interaction_df_filtered
interaction_ids <- gsub(
  ".*:ENSG",
  "ENSG", rownames(interaction_df_filtered)
)

# Step 2: Construct interaction terms
interaction_terms <- paste0("clin$treatments:", interaction_ids)

# Step 3: (optional) Make sure signif_genes are
# character vector of gene IDs (if not already)
# signif_genes <- colnames(final_genes)[some_selection_logic]

# Step 4: Construct the model formula string
base_terms <- c(
  "clin$age",
  "clin$race_ethnicity",
  "strata(clin$treatments)",
  "strata(clin$ajcc_stage_numeric)"
)

formula_string <- paste(
  "Surv(clin$time, clin$status) ~",
  paste(c(base_terms, signif_genes, interaction_terms), collapse = " +
      ")
)

# Step 5: Convert to formula
cox_formula <- as.formula(formula_string)

# Step 6: Fit final model
final_fit <- coxph(
  formula = cox_formula,
  data = final_genes,
  control = coxph.control(iter.max = 300)
)

# Visually inspect results
summary(final_fit)

if (!is.null(final_fit) && inherits(final_fit, "coxph")) {
  save_path <- paste0("saved-models/final_fit_seed_", seed, ".rds")
```

```
      saveRDS(final_fit, file = save_path)
      cat("Saved model:", save_path, "\n")
    } else {
      cat("Model fit failed (not class 'coxph') for seed:", seed, "\n")
    }
  },
  error = function(e) {
    cat("Error at seed", seed, ":", conditionMessage(e), "\n")
  }
)
print(paste0(counter, " seeds have been processed."))
counter <- counter + 1

stopCluster(cl)
gc()
}

}
```

```
# ======================================
# Script: visualizations.R
# ======================================

# Minimal plotting script: AJCC stage survival + distribution

library(survival)
library(survminer)
library(ggplot2)
library(dplyr)

# Load clinical data
clin <- readRDS("clin.RDS")
clin_df <- as.data.frame(as(clin, "DataFrame"))

# Ensure output folder exists
if (!dir.exists("images")) dir.create("images", recursive = TRUE)

# Collapse AJCC pathologic stages to I/II/III/IV
clin_df <- clin_df %>%
    mutate(
        ajcc_pathologic_stage = as.character(ajcc_pathologic_stage),
        ajcc_stage_collapsed = case_when(
            ajcc_pathologic_stage %in% c(
                "Stage I",
                "Stage IA", "Stage IB"
            ) ~ "Stage I",
            ajcc_pathologic_stage %in% c(
```

```
            "Stage II",
            "Stage IIA", "Stage IIB"
        ) ~ "Stage II",
        ajcc_pathologic_stage %in% c(
            "Stage III",
            "Stage IIIA", "Stage IIIB", "Stage IIIC"
        ) ~ "Stage III",
        ajcc_pathologic_stage == "Stage IV" ~ "Stage IV",
        TRUE ~ NA_character_
    ),
    ajcc_stage_collapsed = factor(ajcc_stage_collapsed,
        levels =
            c(
                "Stage I",
                "Stage II",
                "Stage III",
                "Stage IV"
            )
    )
)

# Filter to rows with complete survival + collapsed stage info
clin_surv <- clin_df %>%
    filter(!is.na(time), !is.na(status), !is.na(ajcc_stage_collapsed))

# Plot 1: Survival by collapsed AJCC stage
fit_collapsed <- survfit(Surv(time, status) ~
    ajcc_stage_collapsed, data = clin_surv)

surv_plot_collapsed <- ggsurvplot(
    fit_collapsed,
    data = clin_surv,
    pval = TRUE,
    risk.table = TRUE,
    conf.int = FALSE,
    legend.title = "AJCC Stage",
    ggtheme = theme_minimal(),
    title = "Survival by Collapsed AJCC Stage"
)

ggsave(
    filename = "images/survival_by_collapsed_ajcc_stage.png",
    plot = surv_plot_collapsed$plot,
    width = 10, height = 8, dpi = 300
)

# Plot 2: AJCC stage distribution (original pathologic stage)
```

```
# Set a sensible order for bars (optional)
stage_levels <- c(
    "Stage I", "Stage IA", "Stage IB",
    "Stage II", "Stage IIA", "Stage IIB",
    "Stage III", "Stage IIIA", "Stage IIIB", "Stage IIIC",
    "Stage IV", "Stage X"
)

clin_df$ajcc_pathologic_stage <-
    factor(clin_df$ajcc_pathologic_stage, levels = stage_levels)

stage_dist_plot <- ggplot(clin_df, aes(x = ajcc_pathologic_stage)) +
    geom_bar() +
    labs(
        title = "Distribution of AJCC Pathologic Stages",
        x = "Stage",
        y = "Number of Patients"
    ) +
    theme_minimal() +
    coord_flip()

ggsave(
    filename = "images/ajcc_stage_distribution.png",
    plot = stage_dist_plot,
    width = 7, height = 5, dpi = 300
)
```

```
# ======================================
# Script: jacard.R
# ======================================

## compute overlap + Jaccard from hard-coded Ensembl IDs

# Define gene sets per model
seed_105541 <- c(
  "ENSG00000041880.14","ENSG00000088256.9","ENSG00000108582.12",
  "ENSG00000124568.12","ENSG00000142686.8","ENSG00000165943.5",
  "ENSG00000177030.17","ENSG00000197081.16","ENSG00000212452.1",
  "ENSG00000253474.2","ENSG00000260048.2","ENSG00000265943.1"
)

seed_27352 <- c(
  "ENSG00000041880.14","ENSG00000108582.12","ENSG00000124568.12",
  "ENSG00000142686.8","ENSG00000165943.5","ENSG00000177030.17",
  "ENSG00000197081.16","ENSG00000212452.1","ENSG00000253474.2",
  "ENSG00000260048.2"
)
```

```
seed_40284 <- c(
  "ENSG00000041880.14","ENSG00000108582.12","ENSG00000124568.12",
  "ENSG00000142686.8","ENSG00000165943.5","ENSG00000177030.17",
  "ENSG00000197081.16","ENSG00000212452.1","ENSG00000260048.2",
  "ENSG00000265943.1"
)

final_model <- c(
  "ENSG00000041880.14","ENSG00000108582.12","ENSG00000124568.12",
  "ENSG00000142686.8","ENSG00000165943.5","ENSG00000177030.17",
  "ENSG00000197081.16","ENSG00000212452.1","ENSG00000260048.2"
)

# Put them in a named list
sets <- list(
  "Seed 105541" = seed_105541,
  "Seed 27352" = seed_27352,
  "Seed 40284" = seed_40284,
  "Final Model" = final_model
)

# Presence/absence table with Census
all_genes <- sort(unique(unlist(sets)))
presence <- sapply(sets, function(s) as.integer(all_genes %in% s))
presence_df <- data.frame(Gene = all_genes, presence, check.names = FALSE)
presence_df$Census <- rowSums(presence_df[names(sets)])

cat("\n### Presence/Absence Table (with Census) ###\n")
print(presence_df, row.names = FALSE)

# Jaccard similarity matrix
jaccard <- function(a, b) {
  inter <- length(intersect(a, b))
  uni <- length(union(a, b))
  if (uni == 0) return(NA_real_)
  inter / uni
}

model_names <- names(sets)
J <- matrix(NA_real_, nrow = length(sets), ncol = length(sets),
            dimnames = list(model_names, model_names))
for (i in seq_along(sets)) {
  for (j in seq_along(sets)) {
    J[i, j] <- jaccard(sets[[i]], sets[[j]])
  }
}
```

```
cat("\n### Jaccard Similarity Matrix ###\n")
print(round(J, 2))
```

```
# =====================================
# Script: signal_stability.R
# =====================================

library(tidyverse)
library(survival)
library(dplyr)
library(broom)
library(stringr)
library(survminer)
library(forcats) # used later for factor ordering

# Directory containing models
model_dir <- "./saved-models"
model_files <- list.files(model_dir,
  pattern = "^final_fit_seed_\\d+\\.rds$", full.names = TRUE
)

# Consider only 2000
model_files <- head(model_files, 2000)

# Load clinical data
clin <- read.csv("clinical_data.csv")

# Extract both genes and treatment interactions
extract_model_terms <- function(file_path) {
  seed <- as.integer(str_extract(file_path, "\\d+"))
  model <- readRDS(file_path)
  coef_df <- as.data.frame(summary(model)$coefficients)
  coef_df$term <- rownames(coef_df)

  terms_df <- coef_df %>%
    filter(coef != 0) %>%
    select(term, coef) %>%
    rename(gene = term, coefficient = coef) %>%
    mutate(seed = seed) %>%
    select(seed, gene, coefficient)

  return(terms_df)
}

# Combine all results
compiled_results <- map_dfr(model_files, extract_model_terms)
```

```
# Save all compiled results
write_csv(compiled_results, "compiled_gene_and_interaction_results.csv")

# Frequency of appearance across seeds (all terms)
gene_frequency <- compiled_results %>%
  count(gene, name = "n_seeds") %>%
  arrange(desc(n_seeds))

write_csv(gene_frequency, "gene_frequency_summary.csv")

# Keep only genes that appear in >475 seeds
freq_cutoff <- 475
freq_genes <- gene_frequency %>%
  dplyr::filter(n_seeds > freq_cutoff) %>%
  dplyr::pull(gene) # ENSG IDs

# Identify and summarize treatment interaction terms
treatment_interactions <- compiled_results %>%
  filter(str_detect(gene, "treatments.*:")) %>%
  group_by(gene) %>%
  summarise(
    n_seeds = n(),
    mean_coef = mean(coefficient),
    sd_coef = sd(coefficient),
    n_positive = sum(coefficient > 0),
    n_negative = sum(coefficient < 0),
    sign_consistent = (n_positive == 0 | n_negative == 0)
  ) %>%
  arrange(desc(n_seeds))

write_csv(treatment_interactions, "treatment_interactions_summary.csv")

# Print key outputs
print("Gene frequency across seeds:")
print(gene_frequency %>% slice_head(n = 20))

print("Top treatment interaction terms:")
print(treatment_interactions %>% slice_head(n = 20))

### Fit final candidate model

gene_threshold <- 20
top_genes <- gene_frequency %>%
  filter(n_seeds >= gene_threshold) %>%
  pull(gene)
```

```
interaction_threshold <- 20
top_interactions <- treatment_interactions %>%
  filter(n_seeds >= interaction_threshold, sign_consistent) %>%
  pull(gene)

# Load matrix and survival outcome
X_clean <- readRDS("model_matrix_X_clean.rds")
y <- readRDS("survival_response_y.rds")

# Combine and clean term names
top_genes_clean <- ifelse(
  grepl("^ENSG", top_genes),
  paste0("dat_subset", top_genes),
  top_genes
)

top_interactions_clean <- gsub(
  "^clin\\$treatments(TRUE|FALSE):(ENSG[0-9.]+)$",
  "clin$treatments\\1:dat_subset\\2",
  top_interactions
)

# Include interaction genes as a main effect
top_interaction_suffix <- strsplit(top_interactions_clean, ":")
second_term <- function(x) x[[2]]
top_interaction_suffixes <- lapply(top_interaction_suffix, second_term)
main_effect_genes <- do.call(rbind, unique(top_interaction_suffixes))
main_effect_genes <- as.character(main_effect_genes)

# Remove stratification variables from the modeling matrix
stable_terms <- setdiff(
  unique(c(
    top_genes_clean,
    main_effect_genes,
    top_interactions_clean,
    "clin$treatmentsTRUE",
    "ajcc_pathologic_stage"
  )),
  c("clin$treatmentsTRUE", "ajcc_pathologic_stage") # Remove stratified vars
)

# Make sure clin$ajcc_pathologic_stage is a factor
clin$ajcc_pathologic_stage <- as.factor(clin$ajcc_pathologic_stage)

# Build modeling data
X_clean <- as.data.frame(X_clean)
X_clean$ajcc_pathologic_stage <- clin$ajcc_pathologic_stage
```

```
clin$treatmentsTRUE <- as.integer(clin$treatments == TRUE)

# Subset to final covariates
X_final <- X_clean[, colnames(X_clean) %in% stable_terms]

# Add back strata variables to the dataframe for modeling
X_final$ajcc_pathologic_stage <- clin$ajcc_pathologic_stage
X_final$treatmentsTRUE <- clin$treatmentsTRUE
X_final$treatmentsTRUE <- clin$treatmentsTRUE
X_final$ajcc_pathologic_stage <- clin$ajcc_pathologic_stage

# Fit stratified Cox model
final_fit <- coxph(y ~ . + strata(treatmentsTRUE, ajcc_pathologic_stage),
    data = X_final)
summary(final_fit) # Table 4.1

# Check proportional hazards model
ph_test <- cox.zph(final_fit)
ph_test

# Extract Schoenfeld residuals for custom ggplot2 visualization
# ph_test$y is a matrix of residuals (variables = columns)
# ph_test$x is time, shared across all variables

resid_df <- do.call(rbind, lapply(1:ncol(ph_test$y), function(i) {
  data.frame(
    time = ph_test$x,
    residual = ph_test$y[, i],
    variable = colnames(ph_test$y)[i]
  )
}))

# Define plotting function
plot_schoenfeld <- ggplot(resid_df, aes(x = time, y = residual)) +
  geom_point(color = "black", alpha = 0.3, size = 0.5) +
  geom_smooth(
    method = "loess", formula = y ~ x, color = "blue", se = FALSE, size = 1
  ) +
  labs(
    title = "Global Schoenfeld Residuals Trend",
    x = "Time",
    y = "Scaled Schoenfeld Residual"
  ) +
  theme_minimal(base_size = 14) +
  theme(legend.position = "none")

ggsave("./final_figures/schoenfeld_residuals_final_model.png",
```

```
  plot = plot_schoenfeld, width = 10, height = 6, dpi = 300
)

# Save model and summary
saveRDS(final_fit, "final_model_stable_terms.rds")
writeLines(
  capture.output(summary(final_fit)),
  "final_model_stable_terms_summary.txt"
)

# Tidy coefficient results
coef_df <- tidy(final_fit, exponentiate = TRUE, conf.int = TRUE)

# Filter to clinical + frequent genes;
# drop stratified variables (no HRs by design)
coef_df <- coef_df %>%
  dplyr::mutate(
    ensg = dplyr::if_else(
      stringr::str_detect(term, "ENSG"),
      stringr::str_replace(term, ".*(ENSG[0-9\\.]+).*", "\\1"),
      NA_character_
    )
  ) %>%
  # keep clinical terms (ensg NA) + genes whose ENSG appears >475 times
  dplyr::filter(is.na(ensg) | ensg %in% freq_genes) %>%
  # ensure strata variables never appear in the forest plot
  dplyr::filter(!stringr::str_detect(term, "ajcc_pathologic_stage")) %>%
  dplyr::filter(!stringr::str_detect(term, "treatmentsTRUE$")) %>%
  dplyr::filter(!stringr::str_detect(term, "^strata\\("))

write_csv(coef_df, "final_model_coefficients.csv")

# Plot gene frequency
gene_freq <- ggplot(gene_frequency, aes(x = n_seeds)) +
  geom_histogram(binwidth = 50, fill = "#FFE7A0", color = "goldenrod4") +
  geom_vline(xintercept = 475, color = "red") +
  theme_minimal() +
  labs(
    title = "Gene Frequency Across Seeds",
    x = "Number of Seeds", y = "Gene Count"
  )

# Save the gene frequency histogram
ggsave(
  filename = "final_figures/gene_selection_frequency_hist.png",
  plot = gene_freq,
  width = 7,
```

```
  height = 7,
  dpi = 300
)

# Clean labels for forest plot (stable-genes plot)
coef_df_clean <- coef_df %>%
  mutate(term = str_replace_all(term, "X_finalclin\\$", "")) %>%
  mutate(term = str_replace_all(term, "X_finaldat_subset", "")) %>%
  mutate(term = str_replace_all(term, "TRUE:dat_subset", "")) %>%
  mutate(term = str_replace_all(term, "dat_subset", "")) %>%
  mutate(term = str_replace_all(term, "`", "")) %>%
  mutate(term = str_replace_all(term, "\\$", "")) %>%
  mutate(term = str_replace_all(term, "race_ethnicity", "Race: ")) %>%
  mutate(term = str_replace_all(term, "treatments", "Treatment")) %>%
  mutate(term = str_trim(term)) %>%
  mutate(
    variable_type = case_when(
      str_detect(term, "") ~ "Gene  Treatment",
      str_detect(term, "ENSG") ~ "Gene",
      TRUE ~ "Clinical Covariate"
    )
  )

# Ensure grouping variable exists
coef_df_clean <- coef_df_clean %>%
  mutate(variable_type = case_when(
    str_detect(term, "") ~ "Gene  Treatment",
    str_detect(term, "ENSG") ~ "Gene",
    TRUE ~ "Clinical Covariate"
  ))

# Order by block (Clinical  Gene  GeneTx), then by increasing HR within
    each block
coef_df_clean <- coef_df_clean %>%
  arrange(
    factor(variable_type, levels = c("Clinical Covariate", "Gene", "Gene
        Treatment")),
    estimate, # increasing HR within each block
    term
  ) %>%
  mutate(term_ordered = factor(term, levels = rev(unique(term)))) %>% #
      top-to-bottom
  filter(term != "clinTreatmentTRUE")

# Forest plot (legend off, y label = Term)
forest_plot <- ggplot(coef_df_clean, aes(x = estimate, y = term_ordered)) +
  geom_point(size = 2) +
```

```
  geom_errorbarh(aes(xmin = conf.low, xmax = conf.high), height = 0.2) +
  geom_vline(xintercept = 1, linetype = "dashed", color = "gray70") +
  theme_minimal(base_size = 14) +
  labs(title = "Forest Plot of Final Cox Model",
       x = "Hazard Ratio (HR)",
       y = "Term") +
  theme(axis.text.y = element_text(size = 12),
        panel.grid.minor = element_blank(),
        legend.position = "none")


# Save the forest plot to the desired file path
ggsave(
  filename = "./final_figures/final_forest_plot_stable_genes.png",
  plot = forest_plot,
  width = 8,
  height = 6,
  dpi = 300
)

# NEW: Figure matching significant terms in table
# Build a second figure that includes ALL significant
# terms (clinical + gene + interactions)
# from the final model, regardless of HR direction. Legend removed.

coef_all <- broom::tidy(final_fit, exponentiate = TRUE, conf.int = TRUE)

sig_terms <- coef_all %>%
  filter(
    !str_detect(term, "^strata\\("),
    !str_detect(term, "ajcc_pathologic_stage"),
    !is.na(p.value), p.value < 0.05
  ) %>%
  mutate(
    variable_type = case_when(
      str_detect(term, "treatmentsTRUE:") ~ "Gene  Treatment",
      str_detect(term, "ENSG") ~ "Gene",
      TRUE ~ "Clinical Covariate"
    ),
    # Clean labels for display
    term_clean = term %>%
      str_replace_all("X_finalclin\\$", "") %>%
      str_replace_all("X_finaldat_subset", "") %>%
      str_replace_all("`", "") %>%
      str_replace_all("\\$", "") %>%
      str_replace("^age$", "clinAge") %>%
      str_replace("^race_ethnicity", "clinRace: ") %>%
```

```
      str_replace_all("treatmentsTRUE:dat_subset", "clinTreatment  ") %>%
      str_replace_all("treatmentsTRUE:", "clinTreatment  ") %>%
      str_replace_all("dat_subset", "") %>%
      str_replace("clinRace: not_reported", "clinRace: not reported")
  ) %>%
  arrange(
    factor(variable_type, levels = c("Clinical Covariate","Gene","Gene
        Treatment")),
    desc(abs(log(estimate)))
  ) %>%
  mutate(
    term_ordered = fct_rev(fct_inorder(term_clean))
  )

sig_forest <- ggplot(sig_terms, aes(x = estimate, y = term_ordered, color =
    variable_type)) +
  geom_point(size = 2) +
  geom_errorbarh(aes(xmin = conf.low, xmax = conf.high), height = 0.22) +
  geom_vline(xintercept = 1, linetype = "dashed", color = "gray65") +
  scale_color_manual(values = c(
    "Clinical Covariate" = "gray40",
    "Gene" = "#1f77b4",
    "Gene  Treatment" = "#ff7f0e"
  )) +
  labs(
    title = "Final Cox PH Model: Significant Clinical, Gene, and Interaction
        Terms",
    x = "Hazard Ratio (HR)", y = "Term"
  ) +
  theme_minimal(base_size = 14) +
  theme(panel.grid.minor = element_blank(),
        legend.position = "none") # remove legend

ggsave("final_figures/final_forest_plot_sig_all_terms.png",
       plot = sig_forest, width = 8.5, height = 7.5, dpi = 300)

# Example: Plot survival curves for a top gene
ensemble_ids <- coef_df_clean$term[grepl("ENS", coef_df_clean$term)]

for (ensemble_id in ensemble_ids) {
  gene_id <- ensemble_id
  gene_col <- paste0("dat_subset", gene_id)

  # Only continue if gene is present in X_clean
  if (gene_col %in% colnames(X_clean)) {
    # Create expression group (median split)
    expression_group <- ifelse(X_clean[, gene_col] >
```

```
      median(X_clean[, gene_col], na.rm = TRUE),
    "High", "Low"
    )

    # Create a dataframe combining survival info and expression group
    survival_data <- data.frame(
      time = y[, "time"],
      event = y[, "status"],
      expression_group = factor(expression_group, levels = c("Low", "High"))
    )

    # Create survival object
    surv_obj <- Surv(survival_data$time, survival_data$event)

    # Fit Kaplan-Meier model
    km_fit <- survfit(surv_obj ~ expression_group, data = survival_data)

    # Plot without confidence intervals
    surv_plot <- ggsurvplot(
      km_fit,
      data = survival_data,
      pval = TRUE,
      conf.int = FALSE,
      risk.table = TRUE,
      legend.title = gene_id,
      legend.labs = c("Low", "High"),
      palette = c("#4DBBD5", "#E64B35"),
      title = paste("Survival by", gene_id, "Expression"),
      xlab = "Time",
      ylab = "Survival Probability"
    )

    # Save the plot
    ggsave(
      filename = paste0("./final_figures/", ensemble_id, "_survival.png"),
      plot = surv_plot$plot, # Ensures ggsurvplot output is rendered
      width = 6,
      height = 5,
      dpi = 300
    )
  } else {
    warning(paste("Gene", gene_id, "not found in X_clean"))
  }

  print(ensemble_id)
}
```

```
# ====================================
```

```
# Script: unstable_seed.R
# ====================================

library(dplyr)
library(stringr)
library(ggplot2)
library(forcats)
library(broom)

# Load models
model_dir <- "./saved-models"
model_files <- list.files(model_dir, pattern = "\\.rds$", full.names = TRUE)
model_list <- lapply(model_files, readRDS)
names(model_list) <- tools::file_path_sans_ext(basename(model_files))

# Define the known wild seeds (based on earlier analysis)
wild_seeds <- c(
    105541, 105812, 108186, 11665, 121673,
    127352, 129597, 130030, 140284, 140712
)

# Extract seeds from filenames
model_seeds <- as.integer(gsub("final_fit_seed_", "", names(model_list)))

# Find indices in model_list that correspond to wild seeds
seed_indices <- which(model_seeds %in% wild_seeds)

for (seed in seed_indices) {
    # Tidy first model
    coef_df <- broom::tidy(model_list[[seed]], conf.int = TRUE)

    # Clean term labels and add group column
    coef_df_clean <- coef_df %>%
        mutate(term_clean = term) %>%
        mutate(
            term_clean = str_replace_all(term_clean, "X_finalclin\\$", ""),
            term_clean = str_replace_all(term_clean, "X_finaldat_subset",
                ""),
            term_clean = str_replace_all(term_clean, "TRUE:dat_subset", ""),
            term_clean = str_replace_all(term_clean, "\\$", ""),
            term_clean = str_replace_all(term_clean, "race_ethnicity",
                "Race: "),
            term_clean = str_replace_all(term_clean, "treatments",
                "Treatment"),
            term_clean = str_replace_all(term_clean, "age", "Age")
        ) %>%
        mutate(group = case_when(
```

```
        str_detect(term_clean, "ENSG") & str_detect(term_clean,
            "Treatment") ~ "Gene  Treatment",
        str_detect(term_clean, "ENSG") ~ "Gene",
        TRUE ~ "Clinical Covariate"
    ))

# Internal rank within group (for sorted ordering within grouped blocks)
coef_df_clean <- coef_df_clean %>%
    group_by(group) %>%
    mutate(order_within_group = rank(estimate)) %>%
    ungroup()

# Composite factor with group + within-group rank (preserves grouped
    ordering)
coef_df_clean <- coef_df_clean %>%
    arrange(
        factor(group, levels = c("Clinical Covariate", "Gene", "Gene
            Treatment")),
        order_within_group
    ) %>%
    mutate(term_ordered = factor(term_clean, levels = rev(term_clean)))

# Forest plot
forest_plot <- ggplot(
    coef_df_clean,
    aes(x = exp(estimate), y = term_ordered)
) +
    geom_point() +
    geom_errorbarh(aes(xmin = exp(conf.low), xmax = exp(conf.high)),
        height = 0.2) +
    geom_vline(xintercept = 1, linetype = "dashed", color = "gray50") +
    theme_minimal(base_size = 12) +
    labs(
        title = "Forest Plot of Individual Seed Model",
        x = "Hazard Ratio (HR)",
        y = "Term",
        color = "Variable Type"
    ) +
    theme(legend.position = "none")

print(forest_plot)

ggsave(
    filename = paste0("./final_figures/wild_seeds/forest_plot_seed_",
        seed, ".png"),
    plot = forest_plot,
    width = 7, # in inches
```

```
        height = 6,
        dpi = 300 # high-quality resolution
    )
}
```

CURRICULUM VITAE

David N. Pratt

Education

M.S. in Biostatistics, University of Louisville — Louisville, KY
Thesis: *Genes That Matter: Survival Modeling in TCGA-BRCA with Treatment Interactions*
Committee: Dr. Michael Sekula, Dr. Maiying Kong, Dr. Elizabeth Cash

B.S. in Mathematics, American University — Washington, DC

Full-Stack Web Development Certificate, App Academy — Remote

Technical Skills

Programming: R, Python, Julia, SQL, JavaScript, Bash, LaTeX
Statistical Methods: Survival analysis, high-dimensional modeling, penalized regression, dimensionality reduction, machine learning
Tools and Platforms: Galaxy, Docker, Git, AWS, HPC systems (Linux/Ubuntu), GitLab CI/CD
Databases: PostgreSQL, MySQL
Applications: Full-stack web development, API integration, computational genomics pipelines (TCGA-scale)

Professional Experience

Founder and Lead Data Scientist, Midnight Mechanism, 2016–Present — Founded and scaled a genomics-focused HPC cluster featuring an NVIDIA H100 NVL GPU, 320 vCPUs, 2 TB RAM, and 378 TB storage, enabling high-throughput, reproducible workflow execution and benchmarking. Designed cloud-adjacent infrastructure for large-scale genomic analysis, and led statistical modeling, full-stack development, and client-facing data science projects.

Honors and Awards

Certified Degree in Biochemistry – American Society for Biochemistry and Molecular Biology (ASBMB)
Summer Research Award in Mathematics – American University
UCARE Research Program Participant – University of Nebraska–Lincoln

Professional Memberships

American Statistical Association (ASA)
American Society for Biochemistry and Molecular Biology (ASBMB)