Project: Treatment Effect Comparison of Immunotherapy in Melanoma

Project: Unsupervised Discovery of Survival Phenotypes under Immunotherapy

1. Clinical motivation

Immunotherapy has transformed cancer treatment, but patients experience *heterogeneous benefit*:

- some patients achieve long-term survival
- others experience limited benefit

Understanding this heterogeneity is essential for precision medicine and treatment stratification in order to improve clinical decision-making.

Rather than defining subgroups manually, you may use a *data-driven approach.*

2. Statistical problem

For each patient i, we observe:

$$(T_i, \delta_i, X_i)$$

where:

- T_i: survival time
- δ_i: event indicator
- X_i: clinical covariates

Our goal is to estimate the individualized survival function:

$$S_i(t) = P(T_i > t \mid X_i)$$

and identify groups of patients with similar survival patterns.

3. Project objective

Identify *survival phenotypes* using:

- deep survival prediction (survdnn: https://cran.r-project.org/web/packages/survdnn/index.html)
- unsupervised clustering of survival curves (unsurv: https://github.com/ielbadisy/unsurv)
- clinical interpretation using RMST

These phenotypes represent distinct prognosis profiles.

4. Methodological pipeline

Step 1: Deep survival modeling (survdnn)

Train a neural survival model:

$$\hat{S}_i(t) = f_\theta(X_i)$$

Output: predicted survival curves for all patients.

Step 2: Unsupervised clustering (unsurv)

Cluster predicted survival curves:

$$C_i = \text{cluster}(\hat{S}_i)$$

This identifies K survival phenotypes.

Step 3: Clinical interpretation

Compare clusters using:

- clinical characteristics
- Kaplan-Meier curves
- response variables

This allows interpretation of phenotypes.

Step 4: Quantify survival benefit using RMST

Restricted Mean Survival Time:

$$RMST(\tau) = \int_0^\tau S(t)dt$$

Interpretation:

> Expected survival time up to horizon τ

Compare RMST across clusters.

Why deep survival models?

Classical Cox model:

$$\log h(t \mid X) = \beta^T X$$

assumes:

- linear effects
- proportional hazards

Deep models learn:

$$f_\theta(X)$$

allowing nonlinear effects and complex interactions.

4. Expected outputs

You will produce:

Model results

- trained SurvDNN model
- predicted survival curves

Clustering results

- cluster assignments
- cluster survival curves

Clinical interpretation

- Table 1 by cluster
- KM curves
- RMST comparison

5. Scientific questions

- Do distinct survival phenotypes exist?
- - Which cluster has best prognosis?

Which clinical variables explain differences?

Project: Unsupervised Discovery of Survival Phenotypes under Immunotherapy

1. Clinical motivation

Immunotherapy has transformed cancer treatment, but patients experience *heterogeneous benefit*:

- some patients achieve long-term survival
- others experience limited benefit

Understanding this heterogeneity is essential for precision medicine and treatment stratification in order to improve clinical decision-making.

Rather than defining subgroups manually, you may use a *data-driven approach.*

2. Statistical problem

For each patient i, we observe:

$$(T_i, \delta_i, X_i)$$

where:

- T_i: survival time
- δ_i: event indicator
- X_i: clinical covariates

Our goal is to estimate the individualized survival function:

$$S_i(t) = P(T_i > t \mid X_i)$$

and identify groups of patients with similar survival patterns.

3. Project objective

Identify *survival phenotypes* using:

- deep survival prediction (survdnn: https://cran.r-project.org/web/packages/survdnn/index.html)
- unsupervised clustering of survival curves (unsurv: https://github.com/ielbadisy/unsurv)
- clinical interpretation using RMST

These phenotypes represent distinct prognosis profiles.

4. Methodological pipeline

Step 1: Deep survival modeling (survdnn)

Train a neural survival model:

$$\hat{S}_i(t) = f_\theta(X_i)$$

Output: predicted survival curves for all patients.

Step 2: Unsupervised clustering (unsurv)

Cluster predicted survival curves:

$$C_i = \text{cluster}(\hat{S}_i)$$

This identifies K survival phenotypes.

Step 3: Clinical interpretation

Compare clusters using:

- clinical characteristics
- Kaplan–Meier curves
- response variables

This allows interpretation of phenotypes.

Step 4: Quantify survival benefit using RMST

Restricted Mean Survival Time:

$$RMST(\tau) = \int_0^\tau S(t)dt$$

Interpretation:

> Expected survival time up to horizon τ

Compare RMST across clusters.

Why deep survival models?

Classical Cox model:

$$\log h(t \mid X) = \beta^T X$$

assumes:

- linear effects
- proportional hazards

Deep models learn:

$$f_\theta(X)$$

allowing nonlinear effects and complex interactions.

4. Expected outputs

You will produce:

Model results

- trained SurvDNN model
- predicted survival curves

Clustering results

- cluster assignments
- cluster survival curves

Clinical interpretation

- Table 1 by cluster
- KM curves
- RMST comparison

5. Scientific questions

- Do distinct survival phenotypes exist?
- - Which cluster has best prognosis?

Which clinical variables explain differences?

Helpers

At the end after fitting the model, predicting invidual survivl curves and clustering patients you can compute RMST for more intuititve clinical interpretation.

```r
# 5) Dynamic RMST + tau*
# RMST_i(tau) =  _0^tau S_i(t) dt  (area under the predicted survival curve)
rmst_dynamic <- function(S_mat, time) {
  stopifnot(is.matrix(S_mat), length(time) == ncol(S_mat))
  stopifnot(all(diff(time) > 0))

  n <- nrow(S_mat)
  m <- ncol(S_mat)

  # Step 1) widths of each interval [t_j, t_{j+1}]
  dt <- diff(time)                         # length m-1

  # Step 2) trapezoid height per interval: (S(t_j) + S(t_{j+1})) / 2
  S_left  <- S_mat[, 1:(m - 1), drop = FALSE]
  S_right <- S_mat[, 2:m,       drop = FALSE]
  S_mid   <- (S_left + S_right) / 2        # n x (m-1)

  # Step 3) area increment per interval: height * width
  area_incr <- sweep(S_mid, 2, dt, `*`)    # multiply each column j by dt[j]

  # Step 4) cumulative sum of increments gives RMST at each grid point
  RMST_no0 <- t(apply(area_incr, 1, cumsum))  # n x (m-1)

  # Step 5) RMST at tau = 0 is 0 → prepend a zero column to align with `time`
  RMST <- cbind(0, RMST_no0)               # n x m
  colnames(RMST) <- time

  RMST
}
RMST_mat <- rmst_dynamic(S, times)

# Choose clinical horizon tau* (e.g., 24 months), but cap at observed max grid
tau_star <- min(24, max(times))

# Find exact grid match; if not present, use nearest grid point (explicit)
k_tau <- which(times == tau_star)
if (length(k_tau) != 1) {
  k_tau <- which.min(abs(times - tau_star))
  tau_star <- times[k_tau]
}

# Individual RMST at tau*
```

```r
melanoma2$RMST_tau <- RMST_mat[, k_tau]

aggregate(RMST_tau ~ cluster, data = melanoma2, FUN = mean)

# 7) Plots (mean + individual)
# Build a long data.frame: one row per (patient i, horizon tau_j)
n <- nrow(melanoma2)
m <- length(times)

rmst_long <- data.frame(
  id      = rep(seq_len(n), times = m),
  tau     = rep(times, each = n),
  rmst    = as.vector(RMST_mat),
  cluster = rep(melanoma2$cluster, times = m)
)

# Mean dynamic RMST per cluster (one mean curve per facet)
print(
  ggplot(rmst_long, aes(x = tau, y = rmst, group = cluster)) +
    stat_summary(fun = mean, geom = "line") +
    facet_wrap(~ cluster) +
    labs(x = "Tau", y = "RMST(tau)", title = "Mean dynamic RMST by cluster") +
    theme_minimal()
  )

# Individual RMST trajectories (many faint lines per facet)
print(
  ggplot(rmst_long, aes(x = tau, y = rmst, group = id)) +
    geom_line(alpha = 0.15) +
    facet_wrap(~ cluster) +
    labs(x = "Tau", y = "RMST(tau)", title = "Individual dynamic RMST curves by cluster") +
    theme_minimal()
)
```