Package 'jmSurface'

February 25, 2026

Title Semi-Parametric Association Surfaces for Joint Longitudinal-Survival Models

Version 0.1.0

Description Implements interpretable multi-biomarker fusion in joint longitudinal-survival models via semi-parametric association surfaces. Provides a two-stage estimation framework where Stage 1 fits mixed-effects longitudinal models and extracts Best Linear Unbiased Predictors ('BLUP's), and Stage 2 fits transition-specific penalized Cox models with tensor-product spline surfaces linking latent biomarker summaries to transition hazards. Supports multi-state disease processes with transition-specific surfaces, Restricted Maximum Likelihood ('REML') smoothing parameter selection, effective degrees of freedom ('EDF') diagnostics, dynamic prediction of transition probabilities, and three interpretability visualizations (surface plots, contour heatmaps, marginal effect slices).
Methods are described in Bhattacharjee (2025, under review).

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.0.0)

Imports nlme, survival, mgcv, stats, utils, graphics, grDevices

Suggests lme4, ggplot2, viridis, plotly, shiny, shinydashboard, dplyr, tidyr, testthat (>= 3.0.0)

NeedsCompilation no

Author Atanu Bhattacharjee [aut, cre] (ORCID: <https://orcid.org/0000-0002-5757-5513>)

Maintainer Atanu Bhattacharjee <atanustat@gmail.com>

Repository CRAN

Date/Publication 2026-02-25 10:20:02 UTC

Contents

`jmSurface-package` *jmSurface: Semi-Parametric Association Surfaces for Joint Models*

Description

The **jmSurface** package implements interpretable multi-biomarker fusion in joint longitudinal-survival models via semi-parametric association surfaces for multi-state disease processes.

Main functions

`jmSurf` Fit the two-stage joint model

`fit_longitudinal` Stage 1: fit longitudinal submodels

`fit_gam_cox` Stage 2: fit GAM-Cox with tensor-product surface

`edf_diagnostics` Extract EDF and complexity diagnostics

`dynPred` Dynamic prediction of transition probabilities

`plot_surface` 3D association surface visualization

`contour_heatmap` Contour heatmap of danger zones

`marginal_slices` Marginal effect slice plots

`simulate_jmSurface` Simulate multi-state data

`run_shiny_app` Launch interactive Shiny dashboard

Author(s)

Maintainer: Atanu Bhattacharjee <atanustat@gmail.com> (ORCID)

References

Bhattacharjee, A. (2025). Interpretable Multi-Biomarker Fusion in Joint Longitudinal-Survival Models via Semi-Parametric Association Surfaces.

`compute_blup_eta` *Compute BLUP-Based Latent Longitudinal Summaries*

Description

Extracts subject-specific BLUPs from fitted `lme` models and computes the latent trajectory $\eta_{ij}(t) = \hat{\beta}_{0j} + \hat{b}_{0ij} + (\hat{\beta}_{1j} + \hat{b}_{1ij})t$ at specified time points.

Usage

```
compute_blup_eta(lme_fits, patient_ids, times, markers = NULL)
```

Arguments

`lme_fits`	Named list of `nlme::lme` objects (from `fit_longitudinal`).
`patient_ids`	Numeric vector of patient IDs.
`times`	Numeric vector of evaluation times.
`markers`	Character vector of biomarker names. If `NULL`, uses names of `lme_fits`.

Value

Data frame with columns `patient_id`, `time`, and one `eta_*` column per biomarker.

`contour_heatmap` *Contour Heatmap of Association Surface*

Description

Produces a filled contour (heatmap) of the estimated association surface, identifying "danger zones" of elevated transition risk.

Usage

```
contour_heatmap(object, transition, n_grid = 50, col = NULL, main = NULL, ...)
```

Arguments

object	A "jmSurface" object.
transition	Character string specifying which transition.
n_grid	Integer grid resolution. Default 50.
col	Color palette function. Default hcl.colors(30, "YlOrRd", rev=TRUE).
main	Title. If NULL, auto-generated.
...	Additional arguments passed to filled.contour.

Value

Invisibly returns the prediction grid.

---

dynPred *Dynamic Prediction of Transition Probabilities*

Description

Computes personalized dynamic predictions of transition probabilities from a fitted jmSurface model. Given a patient's biomarker history up to a landmark time, projects the latent trajectories forward and integrates the transition-specific hazard to obtain cumulative transition probabilities.

Usage

```
dynPred(object, patient_id, landmark = 0, horizon = 3, n_points = 60)
```

Arguments

object	A "jmSurface" object from jmSurf.
patient_id	Integer patient identifier.
landmark	Numeric landmark time (predict from this time).
horizon	Numeric prediction horizon (predict this many years ahead).
n_points	Integer number of time points for the prediction grid. Default 60.

Details

The conditional transition probability is computed as:

$$\pi_i^{rs}(t_L, \Delta t) = 1 - \exp\left\{ -\int_{t_L}^{t_L+\Delta t} \lambda_i^{rs}(u|\hat{\eta}_i(u))du \right\}$$

where $\hat{\eta}_i(u)$ is the BLUP-projected trajectory and the integral is approximated via the Breslow estimator.

Value

Data frame with columns:

`time`	Absolute time points
`risk`	Cumulative transition probability
`hazard`	Instantaneous hazard at each time point
`transition`	Transition name
`to_state`	Target state
`patient_id`	Patient identifier
`landmark`	Landmark time used

`edf_diagnostics` *Effective Degrees of Freedom Diagnostics*

Description

Extracts EDF, deviance explained, and complexity diagnostics for each transition-specific association surface. EDF near 1 indicates linearity; EDF > 3 indicates substantial nonlinearity/interaction.

Usage

```
edf_diagnostics(object)
```

Arguments

`object`	A `"jmSurface"` object.

Details

The EDF is computed as

$$\text{EDF}_{rs} = \text{tr}\{(B'B + \lambda_{rs} S_{rs})^{-1} B'B\}$$

and represents the realized complexity of the association surface after REML-based penalization. Interpretation:

- `EDF ~ 1`: Surface effectively linear; standard parametric JM suffices
- `1 < EDF <= 3`: Moderate nonlinearity
- `EDF > 3`: Substantial nonlinearity and/or interaction effects

Value

Data frame with columns:

`transition`	Transition name
`edf`	Effective degrees of freedom of the surface smooth
`deviance_explained`	Proportion of deviance explained
`n_obs`	Number of observations
`n_events`	Number of events
`complexity`	Character label: "Linear", "Moderate", or "Nonlinear"
`p_value`	Approximate p-value for the smooth term

`fit_gam_cox` *Fit a GAM-Cox Model with Tensor-Product Spline Surface*

Description

Fits a penalized Cox proportional hazards model with a tensor-product spline surface for the latent biomarker summaries using `mgcv::gam` with `family = cox.ph()`.

Usage

```
fit_gam_cox(
  data,
  covariates = c("age_baseline", "sex"),
  k_marginal = c(5, 5),
  k_additive = 6,
  bs = "tp",
  method = "REML"
)
```

Arguments

`data`	Data frame from `.build_transition_data`, containing `time_in_state`, `status`, covariate columns, and `eta_*` columns.
`covariates`	Character vector of covariate names.
`k_marginal`	Integer vector of marginal basis dimensions. Default `c(5,5)`.
`k_additive`	Integer basis dimension for additive smooth of third biomarker. Default `6`.
`bs`	Spline basis type. Default `"tp"`.
`method`	Smoothing method. Default `"REML"`.

Value

A `mgcv::gam` object, or `NULL` on failure.

fit_longitudinal *Fit Longitudinal Mixed-Effects Models for Each Biomarker*

Description

Stage 1 of the two-stage estimation: fits a random intercept-slope model $y_{ij}(t) = (\beta_{0j} + b_{0ij}) + (\beta_{1j} + b_{1ij})t + \epsilon_{ij}(t)$ for each biomarker using `nlme::lme`.

Usage

```
fit_longitudinal(long_data, markers = NULL, verbose = TRUE)
```

Arguments

`long_data`	Data frame with columns `patient_id`, `visit_time_years`, `biomarker`, `value`.
`markers`	Character vector of biomarker names to fit. If `NULL`, all unique biomarkers in `long_data` are used.
`verbose`	Logical; print progress. Default `TRUE`.

Value

Named list of `nlme::lme` objects, one per biomarker. Failed fits are `NULL`.

jmSurf *Fit a Joint Longitudinal-Survival Model with Semi-Parametric Association Surfaces*

Description

Two-stage estimation framework for multi-state joint models with tensor-product spline association surfaces. Stage 1 fits mixed-effects longitudinal models and extracts BLUPs. Stage 2 fits transition-specific penalized Cox models with tensor-product spline surfaces via REML.

Usage

```
jmSurf(
  long_data,
  surv_data,
  transitions = NULL,
  covariates = c("age_baseline", "sex"),
  k_marginal = c(5, 5),
  k_additive = 6,
  bs = "tp",
  method = "REML",
  min_events = 10,
  verbose = TRUE
)
```

Arguments

long_data	Data frame of longitudinal biomarker measurements. Required columns: patient_id, visit_time_years, biomarker, value.
surv_data	Data frame of survival/transition events. Required columns: patient_id, start_time, stop_time, status, state_from, state_to, transition.
transitions	Character vector of transitions to model (e.g., "CKD -> CVD"). If NULL, all observed transitions are used.
covariates	Character vector of baseline covariate names present in surv_data. Default c("age_baseline", "sex").
k_marginal	Integer vector of length 1 or 2 giving marginal basis dimensions for the tensor-product spline. Default c(5, 5).
k_additive	Integer giving the basis dimension for the additive smooth of the third biomarker (if present). Default 6.
bs	Character string for the spline basis type. Default "tp" (thin-plate regression spline).
method	Smoothing parameter estimation method. Default "REML".
min_events	Integer minimum number of events required to fit a transition model. Default 10.
verbose	Logical; print progress messages. Default TRUE.

Details

The model for each transition (r, s) is:

$$\lambda_i^{rs}(t|\eta_i(t)) = \lambda_0^{rs}(t) \exp\{\gamma_{rs}' w_i + f_{rs}(\eta_i(t))\}$$

where f_{rs} is a semi-parametric association surface represented via tensor-product splines, and $\eta_i(t)$ are BLUP-based latent longitudinal summaries evaluated at the midpoint of each sojourn interval.

Value

An object of class "jmSurface" containing:

lme_fits	Named list of nlme::lme objects (one per biomarker)
gam_fits	Named list of mgcv::gam objects (one per transition)
eta_data	Named list of analysis data frames with latent summaries
transitions	Character vector of fitted transitions
biomarkers	Character vector of biomarker names
covariates	Character vector of covariate names used
edf	Named numeric vector of EDF values per transition
deviance_explained	Named numeric of deviance explained per transition
call	The matched call

References

Bhattacharjee, A. (2025). Interpretable Multi-Biomarker Fusion in Joint Longitudinal-Survival Models via Semi-Parametric Association Surfaces.

Bhattacharjee, A. (2024). jmBIG: Scalable Joint Models for Big Data.

Wood, S.N. (2017). Generalized Additive Models: An Introduction with R. Chapman & Hall/CRC.

Tsiatis, A.A. & Davidian, M. (2004). Joint modeling of longitudinal and time-to-event data: an overview. Statistica Sinica, 14, 809-834.

Examples

```
# Simulate data
sim <- simulate_jmSurface(n_patients = 300)

# Fit the joint model
fit <- jmSurf(
  long_data = sim$long_data,
  surv_data = sim$surv_data,
  covariates = c("age_baseline", "sex")
)

# Summary with EDF diagnostics
summary(fit)

# Dynamic prediction for patient 1
pred <- dynPred(fit, patient_id = 1, landmark = 2, horizon = 3)

# Visualize surfaces
plot_surface(fit, transition = "CKD -> CVD")
contour_heatmap(fit, transition = "CKD -> CVD")
marginal_slices(fit, transition = "CKD -> CVD")
```

---

load_example_data *Load Bundled Example Dataset*

Description

Convenience function to load the bundled CKD/CVD/Diabetes multi-state cohort data (2,000 patients, 3 biomarkers, 9 transitions) from the CSV files shipped with the package. Returns both the longitudinal biomarker data and survival event data in a list, ready for use with `jmSurf`.

The longitudinal data contains 68,112 biomarker measurements (eGFR, BNP, HbA1c) and the survival data contains 4,701 rows covering 9 transition types across CKD, CVD, Diabetes, and At-risk states.

Usage

```
load_example_data()
```

Value

A list with two data frames:

`long_data`	Longitudinal biomarker measurements (68,112 rows, 5 columns: `patient_id`, `visit_time_years`, `biomarker`, `value`, `unit`)
`surv_data`	Survival/transition events (4,701 rows, 17 columns including `patient_id`, `start_time`, `stop_time`, `status`, `state_from`, `state_to`, `transition`, `age_baseline`, `sex`, `bmi`)

Examples

```
dat <- load_example_data()
str(dat$long_data)
str(dat$surv_data)

# Fit a model directly
fit <- jmSurf(dat$long_data, dat$surv_data, verbose = FALSE)
summary(fit)
```

---

marginal_slices *Marginal Effect Slices*

Description

Produces marginal effect slice plots showing the effect of one biomarker on the log-hazard at fixed quantiles (Q25, Q50, Q75) of the other. Diverging slices indicate interaction; parallel slices indicate additivity.

Usage

```
marginal_slices(
  object,
  transition,
  n_points = 60,
  quantiles = c(0.25, 0.5, 0.75),
  colors = c("#264653", "#e76f51", "#2a9d8f"),
  main = NULL,
  ...
)
```

Arguments

`object`	A `"jmSurface"` object.
`transition`	Character string specifying which transition.
`n_points`	Integer number of evaluation points. Default `60`.

quantiles	Numeric vector of quantile probabilities for slicing. Default c(0.25, 0.50, 0.75).
colors	Character vector of colors for each slice. Default blue/orange/red.
main	Title. If NULL, auto-generated.
...	Additional arguments passed to plot.

Value

Invisibly returns the data frame of slice values.

---

plot.jmSurface *Plot method for jmSurface objects*

Description

Default plot dispatches to plot_surface for the first transition.

Usage

```
## S3 method for class 'jmSurface'
plot(x, transition = NULL, type = c("surface", "heatmap", "slices"), ...)
```

Arguments

x	A jmSurface object.
transition	Which transition to plot. Default: first.
type	One of "surface", "heatmap", "slices".
...	Additional arguments.

Value

Invisibly returns the prediction grid (a data frame) produced by the dispatched plotting function (plot_surface, contour_heatmap, or marginal_slices).

plot_surface *Plot Association Surface (3D or Perspective)*

Description

Produces a 3D perspective plot of the estimated semi-parametric association surface $\hat{f}_{rs}(\eta_1, \eta_2)$ for a specified transition.

Usage

```
plot_surface(
  object,
  transition,
  n_grid = 40,
  theta = -30,
  phi = 25,
  col = NULL,
  main = NULL,
  ...
)
```

Arguments

object	A "jmSurface" object.
transition	Character string specifying which transition to plot.
n_grid	Integer grid resolution. Default 40.
theta, phi	Viewing angles for persp. Defaults -30, 25.
col	Color palette. Default hcl.colors(50, "viridis").
main	Title. If NULL, auto-generated.
...	Additional arguments passed to persp.

Value

Invisibly returns the prediction grid with fitted values.

print.jmSurface *Print Method for jmSurface Objects*

Description

Brief overview of a fitted jmSurface model.

Usage

```
## S3 method for class 'jmSurface'
print(x, ...)
```

Arguments

x	A "jmSurface" object.
...	Ignored.

Value

The input object x, returned invisibly. Called for its side effect of printing a brief model overview to the console.

---

run_shiny_app *Launch the Interactive Shiny Application*

Description

Launches the interactive Shiny dashboard for personalized multi-state joint modeling with semi-parametric association surfaces. The app provides data upload, exploration, model fitting, personalized prediction, and surface visualization.

Usage

```
run_shiny_app(...)
```

Arguments

...	Additional arguments passed to shiny::runApp.

Details

The Shiny app requires additional packages: shiny, shinydashboard, shinyWidgets, ggplot2, viridis, plotly, dplyr, tidyr, DT, gridExtra.

Value

No return value, called for the side effect of launching an interactive Shiny application in the user's default web browser.

simulate_jmSurface *Simulate Multi-State Joint Modeling Data*

Description

Generates realistic simulated data for a multi-state chronic disease cohort (CKD/CVD/Diabetes) with three longitudinal biomarkers (eGFR, BNP, HbA1c) and bidirectional transitions. Includes demographic covariates and realistic biomarker trajectories.

Usage

```
simulate_jmSurface(
  n_patients = 500,
  max_followup = 15,
  max_events = 3,
  seed = 42
)
```

Arguments

n_patients	Integer number of patients. Default 500.
max_followup	Numeric maximum follow-up time in years. Default 15.
max_events	Integer maximum number of events per patient. Default 3.
seed	Integer random seed. Default 42.

Value

A list with two data frames:

long_data	Longitudinal biomarker measurements with columns patient_id, visit_time_years, biomarker, value, unit
surv_data	Survival/transition data with columns patient_id, start_time, stop_time, status, event_type, state_from, state_to, transition, entry_disease, age_baseline, sex, ethnicity, smoking, bmi

Examples

```
sim <- simulate_jmSurface(n_patients = 200, seed = 123)
head(sim$long_data)
table(sim$surv_data$transition)
```

summary.jmSurface *Summary Method for jmSurface Objects*

Description

Provides a comprehensive summary of the fitted joint model including longitudinal submodel parameters, transition-specific surface EDF and deviance explained, and model configuration.

Usage

```
## S3 method for class 'jmSurface'
summary(object, ...)
```

Arguments

object	A "jmSurface" object.
...	Ignored.

Value

Invisibly returns a list of summary components.

Index