

Bayesian parametric models for survival prediction in medical applications

Overlap of posterior distributions

Iwan Paolucci, PhD

7/3/23

Table of contents

Setup	2
Load data	2
Functions	3
WHAS	4
GBCS	5
PBC	7
ACTG	9

Setup

```
library(Cairo)
library(ggplot2)
library(ggpubr)
library(ggsci)
library(dplyr)
```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

filter, lag

The following objects are masked from 'package:base':

intersect, setdiff, setequal, union

```
library(stringr)
```

```
knitr::opts_chunk$set(dev.args = list(png = list(type = "cairo")),
  fig.path='out/figs/ovi/', dev = c('pdf', 'png'), dpi = 300)
```

Load data

```
load(file = 'data/ovi.rds')
```

```
data$experiment_lbl <- toupper(data$experiment_lbl)
```

```
data$model_lbl <- factor(data$model_lbl, labels = c('BPS Exponential', 'BPS Weibull'))
```

Functions

```
plot_ovi <- function(.data, experiment_str, model_str){

  data.grouped <- .data %>%
    filter(experiment == experiment_str) %>%
    mutate(
      var_lbl = factor(str_sub(var, 8, -1)),
      experiment_lbl = toupper(experiment_lbl)
    ) %>%
    group_by(experiment_lbl, model_lbl, var_lbl, partition) %>%
    summarise(
      n = n(),
      median = median(ovi),
      median_lower = wilcox.test(ovi, conf.int = TRUE)$conf.int[1],
      median_upper = wilcox.test(ovi, conf.int = TRUE)$conf.int[2],
      model = first(model)
    )

  data_filtered <- data.grouped %>% filter(model == model_str)

  plt.all <- ggline(data = data_filtered,
    x = 'partition', y = 'median', color = 'var_lbl', group = 'var_lbl',
    facet.by = c('experiment_lbl', 'model_lbl'),
    add.params = list(color = "var_lbl", size = 1, width = 0.1),
    palette = 'hls', size = 1, plot_type = 'b', alpha = 0.5) +
    geom_ribbon(data = data_filtered, aes(ymin = median_lower, ymax = median_upper,
      fill = var_lbl), alpha=0.2, outline.type = 'bot

  plt.all <- ggpar(plt.all, ylab = 'OVI', xlab = 'Partition', legend.title = "Variable")

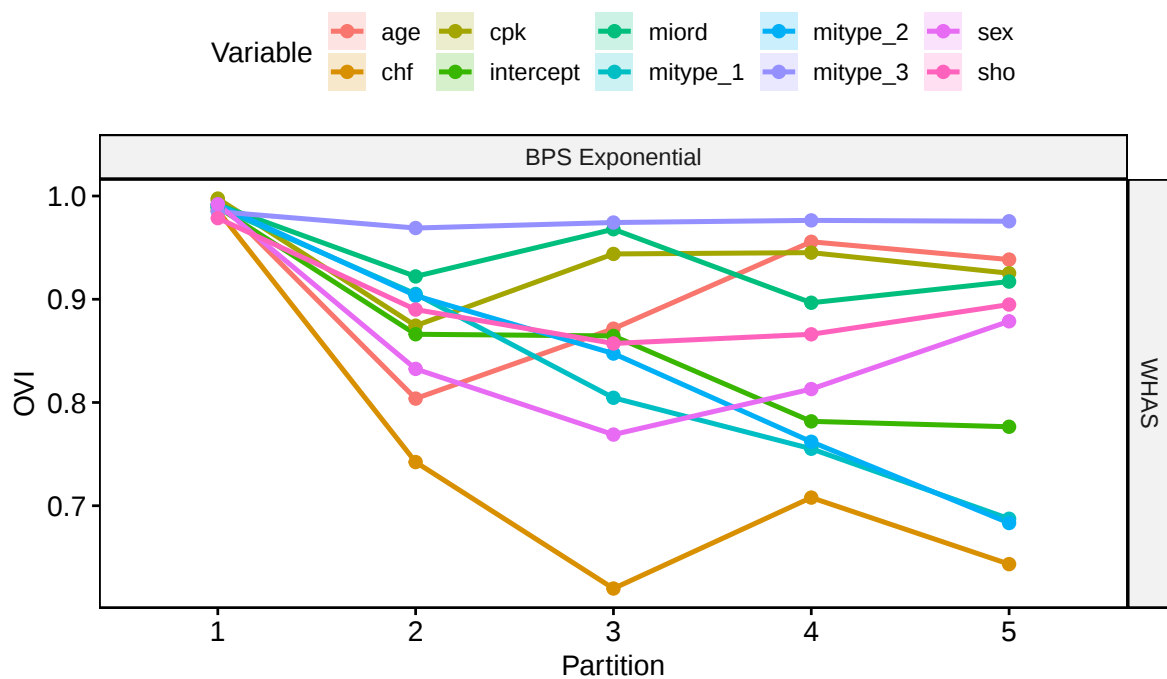
  return (plt.all)
}
```

WHAS

```
data.whas <- data %>%
  filter(!var %in% c('lambda_yrgrp_1975 & 1978', 'lambda_yrgrp_1981 & 1984', 'lambda_yrgrp_1985 & 1988'))

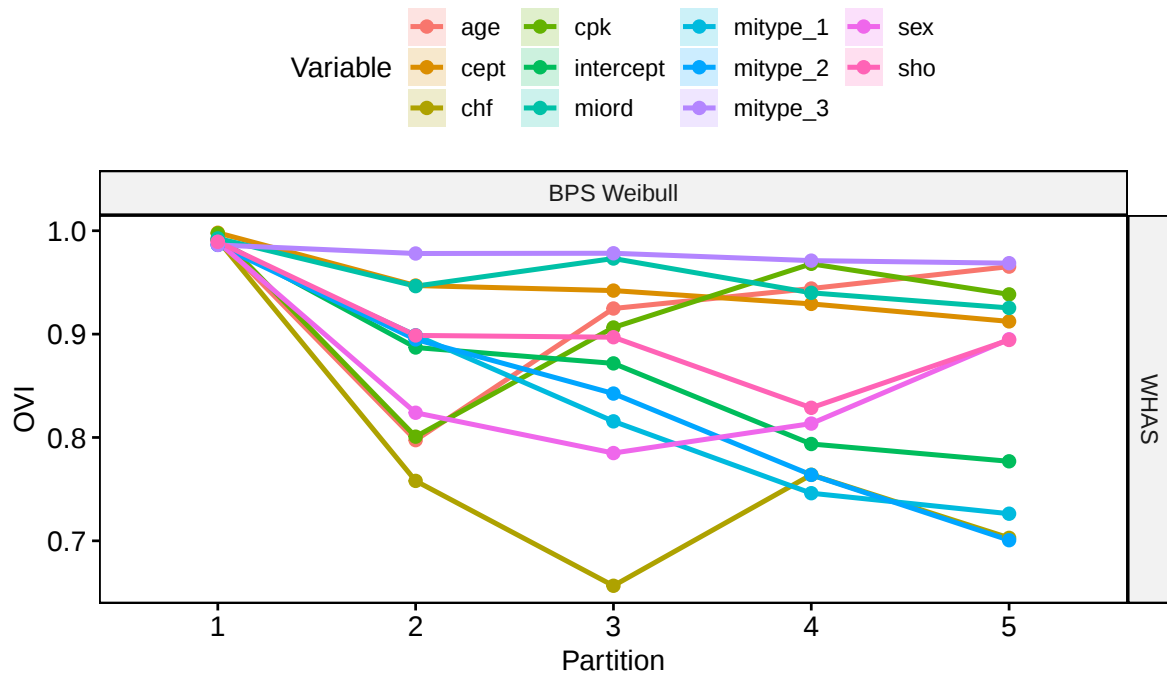
plot_ovi(data.whas, 'whas', 'pm_exp')
```

``summarise()`` has grouped output by 'experiment_lbl', 'model_lbl', 'var_lbl'. You can override using the ``.groups`` argument.



```
plot_ovi(data.whas, 'whas', 'pm_wb')
```

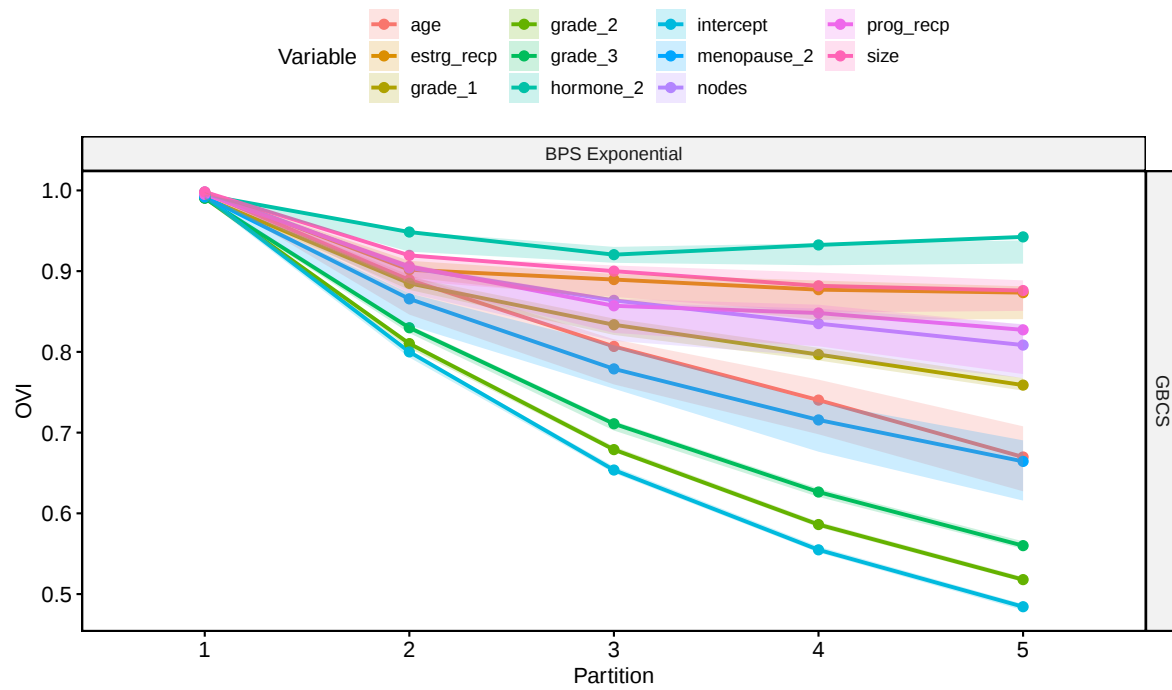
``summarise()`` has grouped output by 'experiment_lbl', 'model_lbl', 'var_lbl'. You can override using the ``.groups`` argument.



GBCS

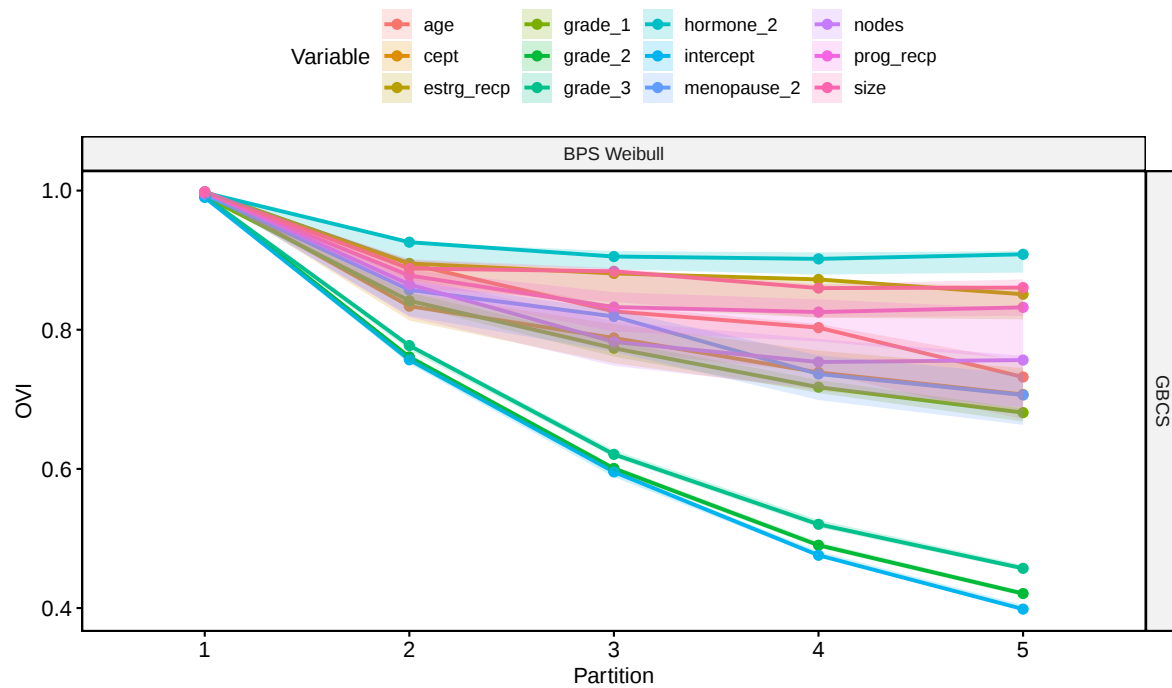
```
plot_ovi(data, 'gbcs', 'pm_exp')
```

`summarise()` has grouped output by 'experiment_lbl', 'model_lbl', 'var_lbl'. You can override using the `.groups` argument.



```
plot_ovi(data, 'gbcs', 'pm_wb')
```

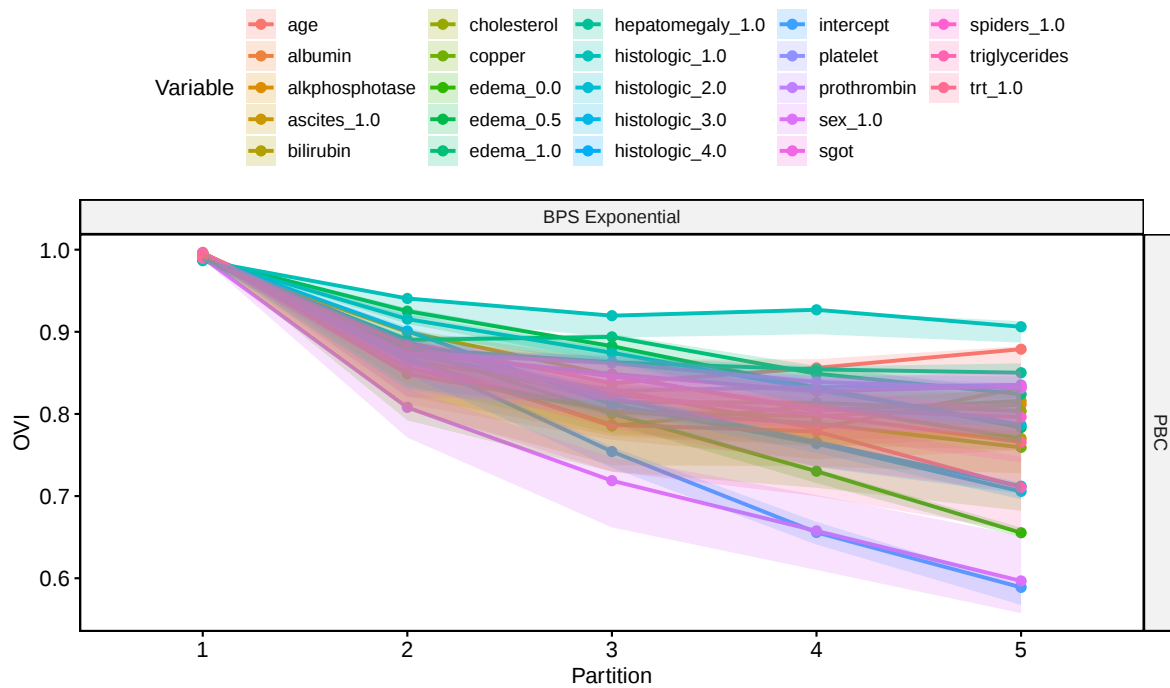
`summarise()` has grouped output by 'experiment_lbl', 'model_lbl', 'var_lbl'. You can override using the `.groups` argument.



PBC

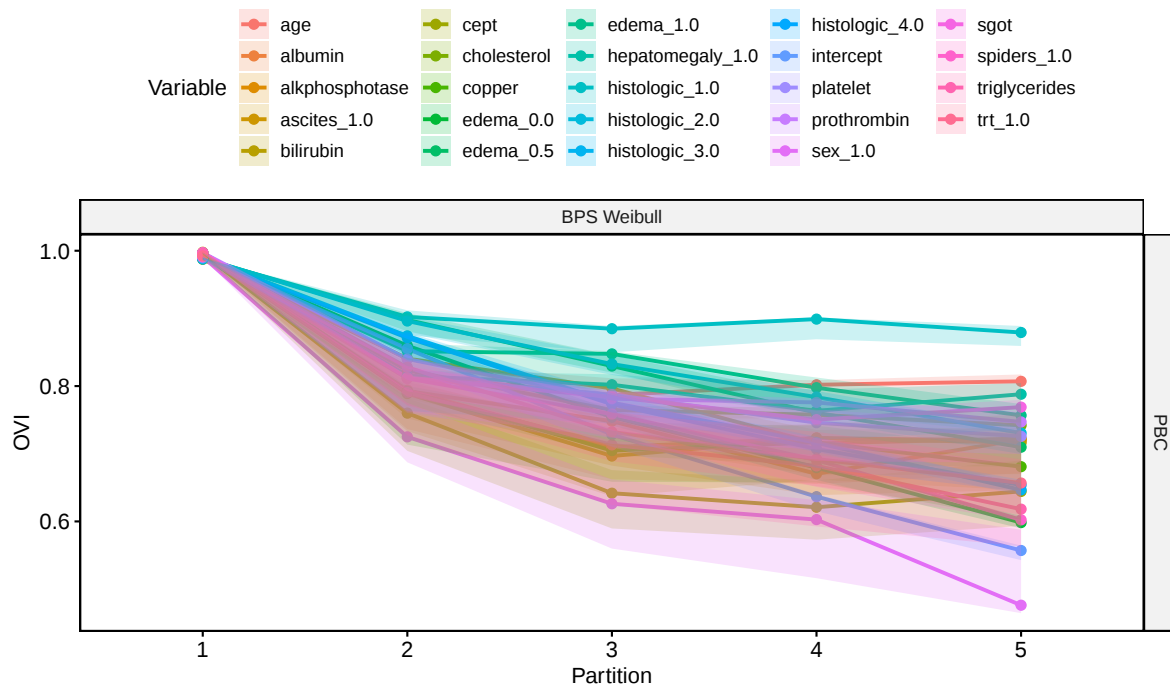
```
plot_ovi(data, 'pbc', 'pm_exp')
```

``summarise()`` has grouped output by 'experiment_lbl', 'model_lbl', 'var_lbl'. You can override using the ``.groups`` argument.



```
plot_ovi(data, 'pbc', 'pm_wb')
```

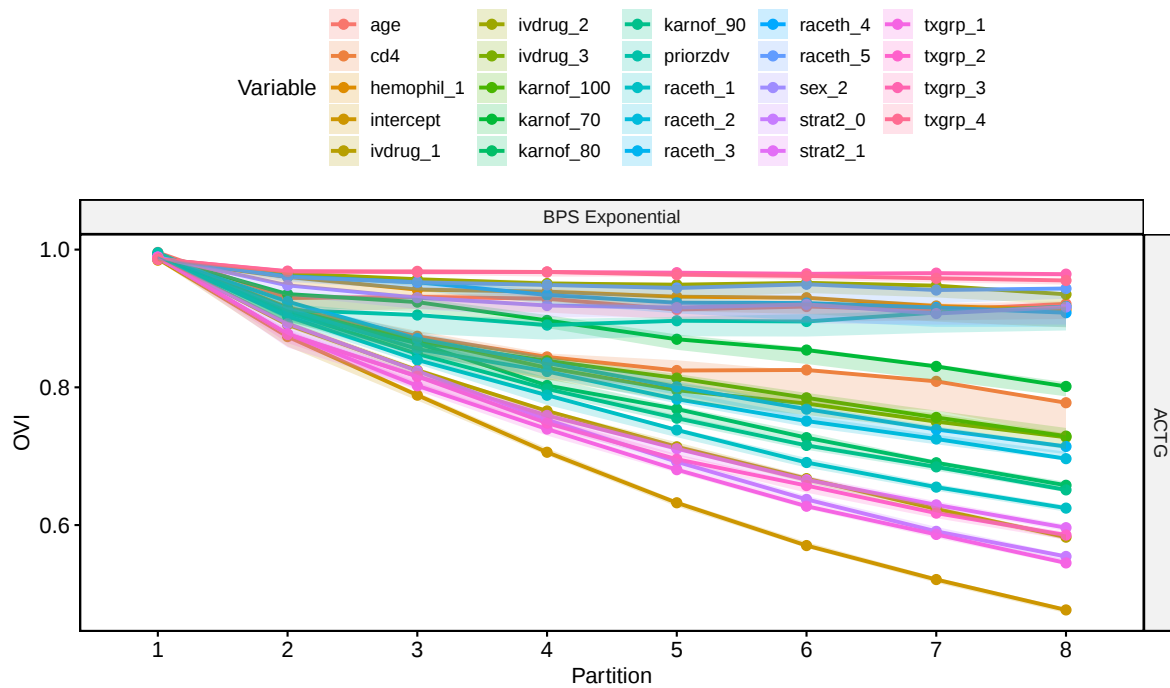
`summarise()` has grouped output by 'experiment_lbl', 'model_lbl', 'var_lbl'. You can override using the `.groups` argument.



ACTG

```
plot_ovi(data, 'actg', 'pm_exp')
```

``summarise()`` has grouped output by 'experiment_lbl', 'model_lbl', 'var_lbl'. You can override using the ``.groups`` argument.



```
plot_ovi(data, 'actg', 'pm_wb')
```

``summarise()`` has grouped output by `'experiment_lbl'`, `'model_lbl'`, `'var_lbl'`. You can override using the ``.groups`` argument.

