Accelerated Oblique Random Survival Forests

`{aorsf}`: Like ORSF, but A

Lukas Burk

What is `{aorsf}`?

- Improvement on `{obliqueRSF}` (*Oblique Random Survival Forests*) by the original author
- `{obliqueRSF}` is orders of magnitude slower than other RSF implementations, did not scale to larger datasets
- `{obliqueRSF}` explicitly states it's superseded by `{aorsf}`
- -> More of a software improvement rather than a new method

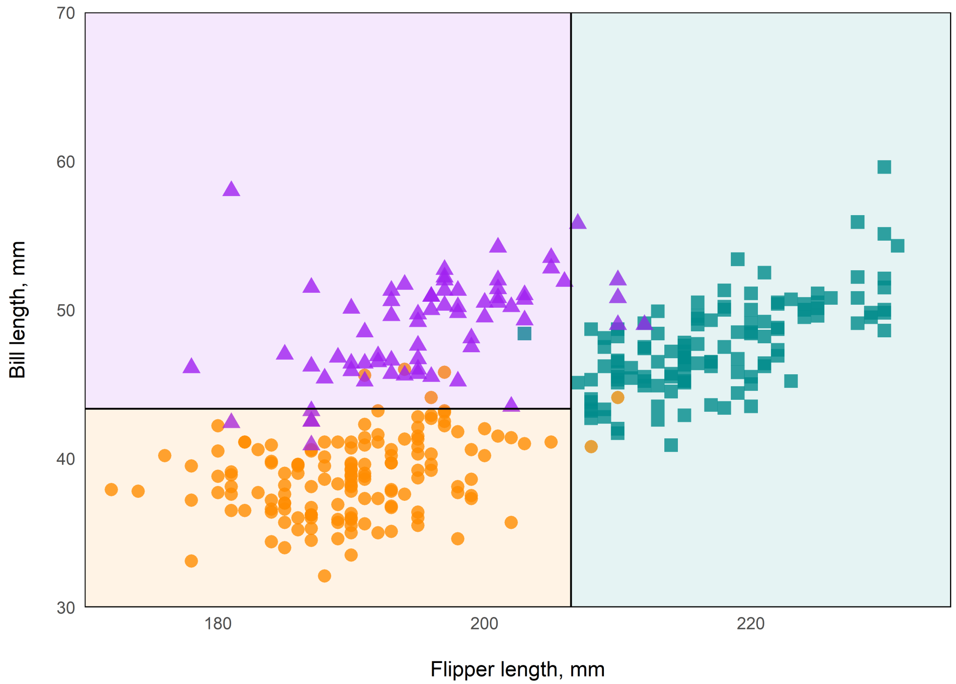
Why is it interesting?

- Extensive benchmark shows improved speed and good predictive accuracy
- Includes multiple interpretability methods
- Underwent rOpenSci software peer Rreview, reviewers included Marvin and me
- Author proactively submitted learner to `mlr3extralearners` and integrates with `tidymodels` framework

A Quick Overview of Oblique RSF

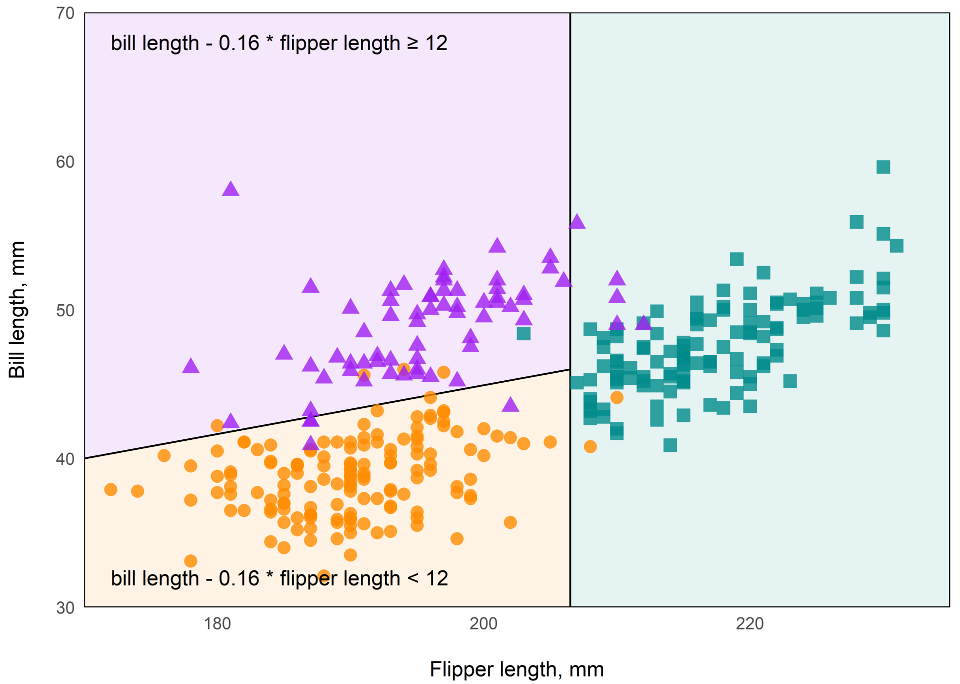
Regular splitting

Orthogonal decision boundaries



Oblique splitting

Linear but not orthogonal



oblique (*adj*): Slanted, diagonal

Accelerating ORSF

`{obliqueRSF}`

- Oblique splits are based on linear combinations of predictors
- Splits based on Cox regression model in (non-leaf) nodes
- Uses `glmnet` to identify linear combinations
- Adds flexibility, increases computational cost

`{aorsf}`

- `{aorsf}` applies Newton-Raphson scoring to partial Cox-likelihood
- Fastest (default) version: Only *one* NR iteration
- Also available:
  - Using NR until convergence
  - Using `{glmnet}`, similar to `{obliqueRSF}`

Interface

```r
library(aorsf)

fit <- orsf(data = pbc_orsf, # Built-in variant of survival::pbc
            formula = Surv(time, status) ~ . - id,
            oobag_pred_horizon = 365.25 * 5)
fit
```

```
---------- Oblique random survival forest

     Linear combinations: Accelerated
          N observations: 276
                N events: 111
                 N trees: 500
      N predictors total: 17
   N predictors per node: 5
 Average leaves per tree: 24
Min observations in leaf: 5
      Min events in leaf: 1
          OOB stat value: 0.84
           OOB stat type: Harrell's C-statistic
     Variable importance: anova

-----------------------------------------
```

Variable importance methods

`importance = "anova"` (default)

`orsf_vi_anova(fit)`

> A **p-value** is computed for each coefficient **in each linear combination of variables** in each decision tree.

> Importance for an individual predictor variable is the **proportion of times a p-value for its coefficient is < 0.01**.

> This technique is **very efficient computationally**, but may not be as effective as permutation or negation in terms of selecting signal over noise variables.

Variable importance methods

```
importance = "negation"
orsf_vi_negate(fit)
```

> Each variable is assessed separately by **multiplying the variable's coefficients by -1** and then determining how much the model's performance changes.

> The worse the model's performance after negating coefficients for a given variable, the more important the variable.

> This technique is **promising b/c it does not require permutation and it emphasizes variables with larger coefficients** in linear combinations,

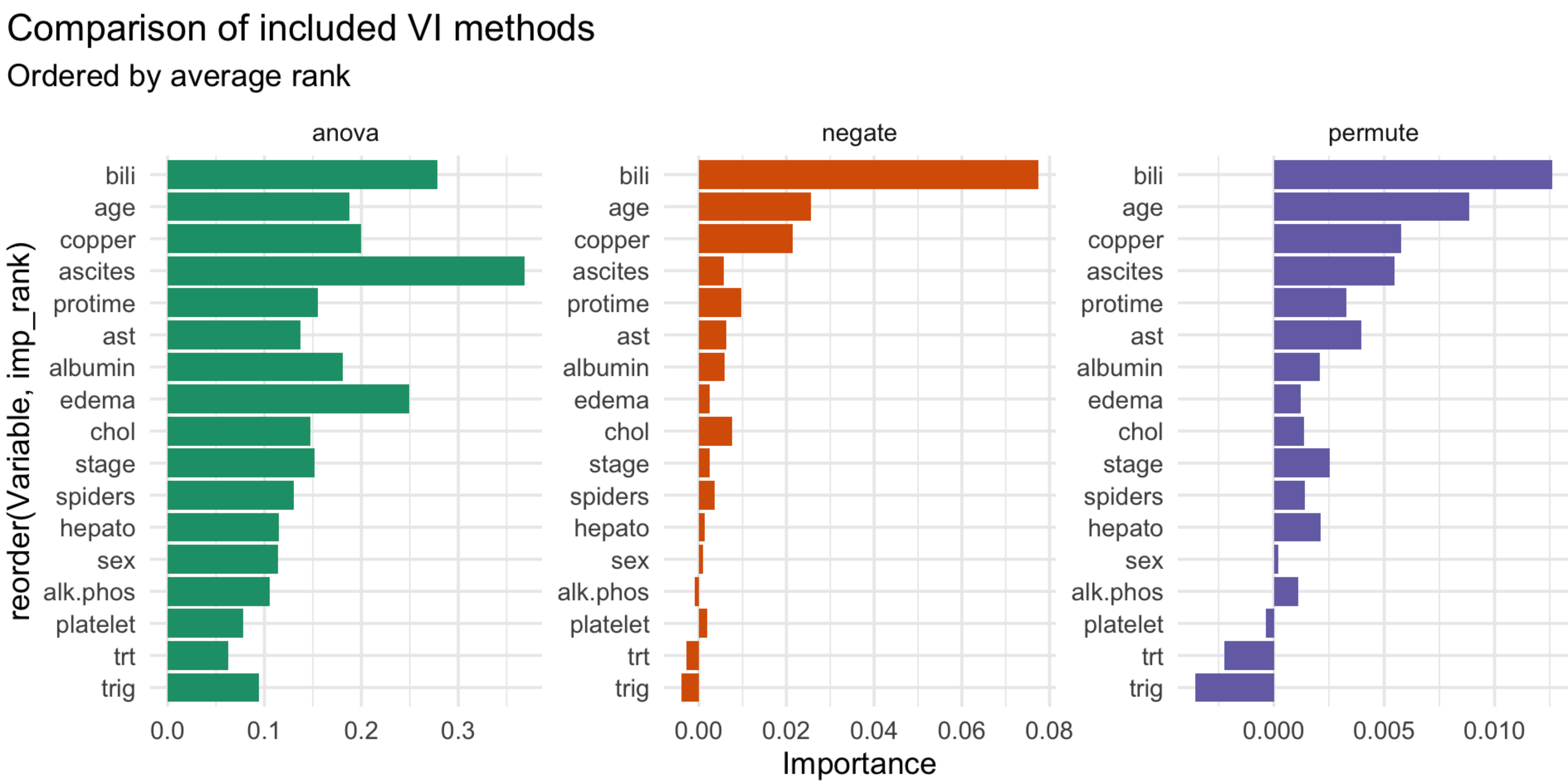
> but it is also **relatively new and hasn't been studied as much** as permutation importance.

Variable importance methods

```
importance = "permutation"
orsf_vi_permute(fit)
```

Standard permutation feature importance,
as known from other R(S)F implementations

Variable importance methods



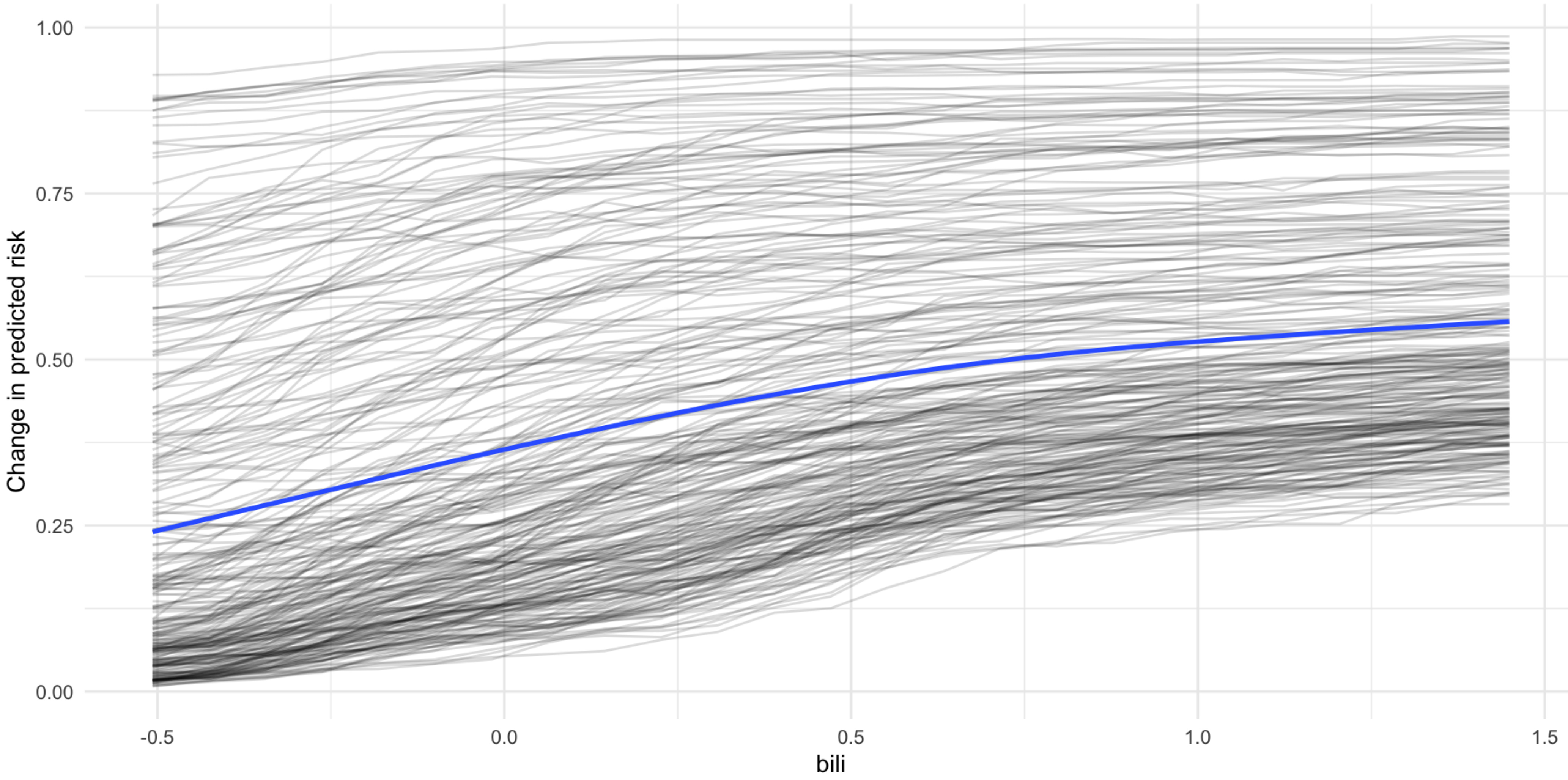
Interpretibility tools: Partial Dependence Plots

- Using either in-bag, OOB or test-data

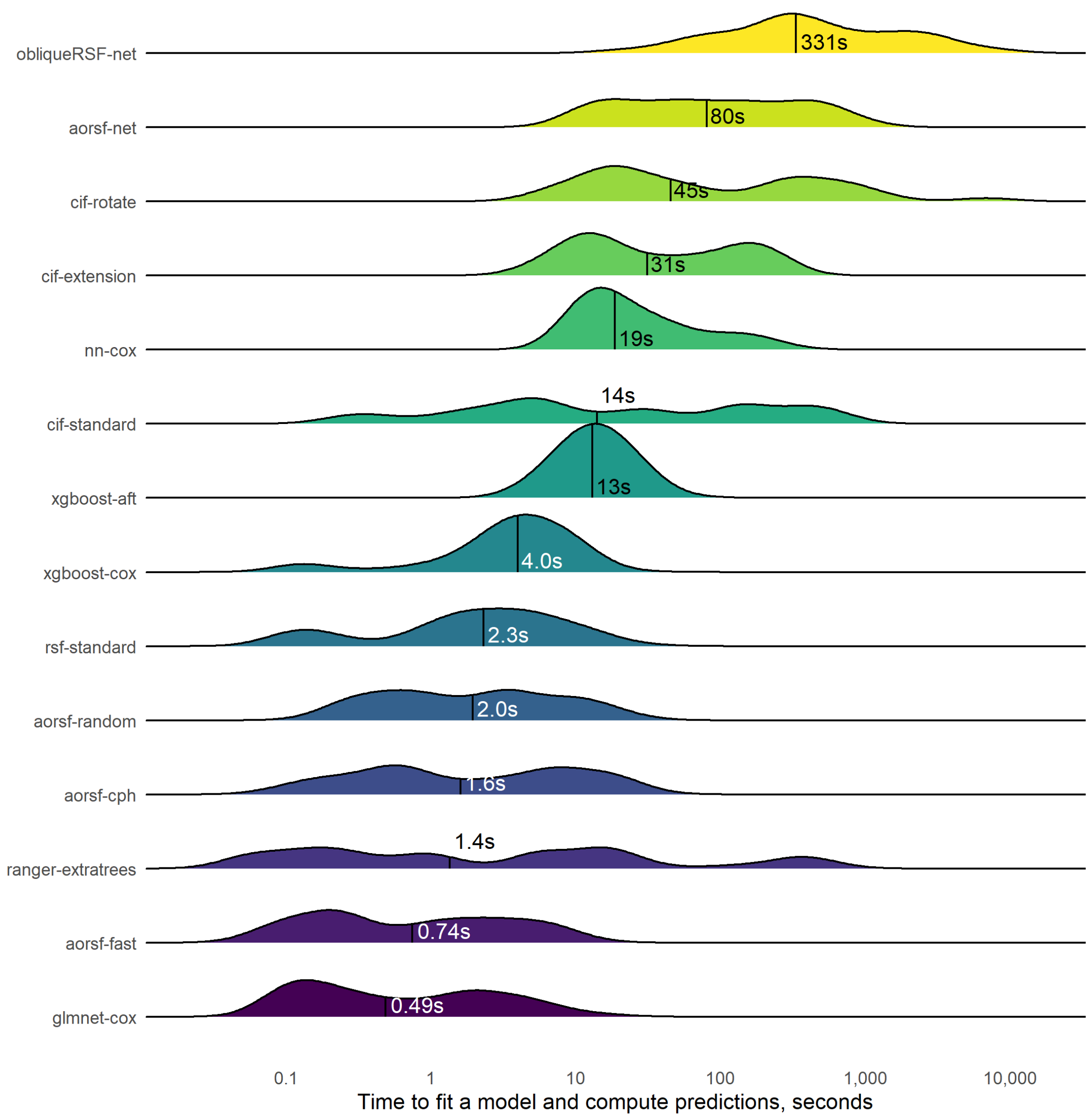
```
pd_sex_tv <- orsf_pd_oob(fit, pred_spec = list(sex = c("m", "f")),
                          pred_horizon = seq(365, 365*5))
```

Interpretibility tools: ICE curves

```r
pred_spec <- list(bili = seq(1, 10, length.out = 25))
ice_oob <- orsf_ice_oob(fit, pred_spec, boundary_checks = FALSE)
```



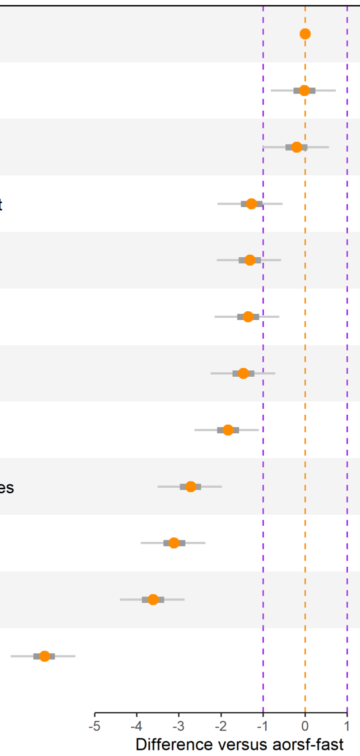
From Benchmark: Runtime



From Benchmark: Performance

- Evaluated on IPA, based on IBS over q25 and q75 of survival times.
- Uses Bayesian LMM: $\mathrm{IPA} = \hat{\gamma} \cdot \mathrm{learner} + (1|\mathrm{data/run})$

		Posterior probability		
Learner	Scaled integrated Brier score	Equivalence	Difference < 0	Difference < -1
aorsf-fast		---	---	---
aorsf-cph		0.99	0.51	0.01
aorsf-net		0.97	0.71	0.03
obliqueRSF-net		0.24	>.999	0.76
cif-standard		0.21	>.999	0.79
rsf-standard		0.18	>.999	0.82
cif-rotate		0.12	>.999	0.88
glmnet-cox		0.02	>.999	0.98
ranger-extratrees		0.00	>.999	>.999
cif-extension		0.00	>.999	>.999
aorsf-random		0.00	>.999	>.999
xgboost-cox		0.00	>.999	>.999



Further Reading

- Package website with great documentation and vignettes docs.ropensci.org/aorsf
- Author's slides on method and benchmark

Literature

- Original paper on ORSF: Jaeger et al. (2019): "Oblique random survival forests"
- Preprint on **arXiv**: Jaeger, Welden, Lenoir, Speiser, et al. (2022): "Accelerated and interpretable oblique random survival forests"
- Software paper on **JOSS**: Jaeger, Welden, Lenoir, and Pajewski (2022)

References

Jaeger, Byron C., D. Leann Long, Dustin M. Long, Mario Sims, Jeff M. Szychowski, Yuan-I Min, Leslie A. Mcclure, George Howard, and Noah Simon. 2019. "Oblique random survival forests." *The Annals of Applied Statistics* 13 (3): 1847–83. https://doi.org/10.1214/19-AOAS1261.

Jaeger, Byron C., Sawyer Welden, Kristin Lenoir, and Nicholas M. Pajewski. 2022. "Aorsf: An r Package for Supervised Learning Using the Oblique Random Survival Forest." *Journal of Open Source Software* 7 (77): 4705. https://doi.org/10.21105/joss.04705.

Jaeger, Byron C., Sawyer Welden, Kristin Lenoir, Jaime L. Speiser, Matthew W. Segar, Ambarish Pandey, and Nicholas M. Pajewski. 2022. "Accelerated and Interpretable Oblique Random Survival Forests." arXiv. https://doi.org/10.48550/ARXIV.2208.01129.