ACC Form

Juan C. Laria

This form documents the artifacts associated with the article (i.e., the data and code supporting the computational findings) and describes how to reproduce the findings.

Part 1: Data

- [ ] This paper does not involve analysis of external data (i.e., no data are used or the only data are generated by the authors via simulation in their code).
- [x] I certify that the author(s) of the manuscript have legitimate access to and permission to use the data used in this manuscript.

Abstract

The original dataset contains an expression set on diffuse large B-cell lymphoma. It accompanies the BioNet packages as example data.

Availability

- [x] Data **are** publicly available.
- [ ] Data **cannot be made** publicly available.

If the data are publicly available, see the *Publicly available data* section. Otherwise, see the *Non-publicly available data* section, below.

Publicly available data

- [ ] Data are available online at:
- [x] Data are available as part of the paper's supplementary material.
- [ ] Data are publicly available by request, following the process described here:
- [ ] Data are or will be made available through some other mechanism, described here:

Non-publicly available data

Description

To pre-process the data, we selected the genes for which individual Cox scores, obtained after fitting univariate Cox regression models, were more significant than a certain threshold. After removing missing values, the data were composed of 190 observations, 78 genetic features, and one clinical variable, which is a factor variable with several levels.

The original data can be loaded with

```
library(DLBCL)
data(exprLym)
```

Package `DLBCL` can be installed with

```
if (!requireNamespace("BiocManager", quietly = TRUE)){
    install.packages("BiocManager")
}
BiocManager::install("DLBCL")
```

Additional preprocessing to convert the data into standard `data.frame` format is optional.

```
data <- t(exprs(exprLym))
pdata <- pData(exprLym)
dlbcl <- merge(data, pdata, "row.names")
row.names(dlbcl) <- dlbcl$Row.names
dlbcl$Row.names <- NULL
dlbcl$StatusAtFollowUp <- NULL
dlbcl$Status <- factor(dlbcl$Status + 0)
dlbcl$time <- dlbcl$FollowUpYears
dlbcl$FollowUpYears <- NULL
```

We made the post-processed data frame available from the Supplementary Materials.

File format(s)

- [ ] CSV or other plain text.
- [x] Software-specific binary format (.Rda, Python pickle, etc.): pkcle
- [ ] Standardized binary format (e.g., netCDF, HDF5, etc.):
- [ ] Other (please specify):

Data dictionary

- [x] Provided by authors in the following file(s): `Section7_real/dlbcl_processed.RData`
- [ ] Data file(s) is(are) self-describing (e.g., netCDF files)
- [ ] Available at the following URL:

Additional Information (optional)

Part 2: Code

Abstract

We provide both the R package `glasp` and the R scripts to replicate the Figures and Tables in the paper. They are also available at the github repositories `jlaria/glasp` and `jlaria/glasp-code`, respectively.

Description

R package `glasp` can be installed directly from github with

```
devtools::install_github("jlaria/glasp", dependencies = TRUE)
```

Installing `glasp` along with other extra R packages should be enough to replicate the Figures and Section 7 of the paper (Application to right-censored survival data).

However, the simulations in Sections 4 and 5 require many dependencies (some of them were removed from CRAN recently). To avoid impossible dependencies and configuration issues, we wrapped everything in a docker image, that can be pulled with

```
docker run -it jlaria/glasp:0.0.1
```

Additionally, if you have `vscode`, `docker` and the `ms-vscode-remote.remote-containers` extension for `vscode`, you can open the cloned repository `jlaria/glasp-code` in a remote container and `vscode` will automatically install the required dependencies. Additional documentation can be found here.

Code format(s)

- [x] Script files
  - [x] R
  - [ ] Python
  - [ ] Matlab
  - [ ] Other:
- [x] Package
  - [x] R
  - [ ] Python
  - [ ] MATLAB toolbox
  - [ ] Other:
- [ ] Reproducible report
  - [ ] R Markdown
  - [ ] Jupyter notebook
  - [ ] Other:
- [ ] Shell script
- [x] Other (please specify):
  - [x] Dockerfile

Supporting software requirements

Version of primary software used

`R version 3.6.3`

Libraries and dependencies used by the code

Package `glasp_0.0.1` loads some extra libraries. A `sessionInfo()` reveals the dependencies.

```
library(glasp)
sessionInfo()
```

```
## R version 3.6.3 (2020-02-29)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Debian GNU/Linux 10 (buster)
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.8.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.8.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
```

```
## other attached packages:
## [1] glasp_0.0.1.9000
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.4.6      parsnip_0.1.1     pillar_1.4.4      compiler_3.6.3
##  [5] iterators_1.0.12  clues_0.6.2.2     tools_3.6.3       digest_0.6.25
##  [9] evaluate_0.14     lifecycle_0.2.0   tibble_3.0.1      nlme_3.1-144
## [13] lattice_0.20-40   mgcv_1.8-31       pkgconfig_2.0.3   rlang_0.4.6
## [17] foreach_1.5.0     Matrix_1.2-18     yaml_2.2.1        parallel_3.6.3
## [21] xfun_0.12         fda_5.1.4         dplyr_0.8.5       stringr_1.4.0
## [25] knitr_1.28        generics_0.0.2    vctrs_0.2.4       fda.usc_2.0.2
## [29] grid_3.6.3        tidyselect_1.0.0  glue_1.4.0        R6_2.4.1
## [33] rmarkdown_2.1     purrr_0.3.4       tidyr_1.0.3       magrittr_1.5
## [37] codetools_0.2-16  ellipsis_0.3.0    htmltools_0.4.0   MASS_7.3-51.5
## [41] splines_3.6.3     assertthat_0.2.1  stringi_1.4.6     doParallel_1.0.15
## [45] crayon_1.3.4
```

Supporting system/hardware requirements (optional)

The simulations were run using a standalone Spark cluster with several workstations as parallel backend. However, they can run locally as long as `sparklyr` is installed.

Parallelization used

- [ ] No parallel code used
- [ ] Multi-core parallelization on a single machine/node
  - Number of cores used:
- [x] Multi-machine/multi-node parallelization
  - Number of nodes and cores used: 3 nodes, 14 cores

License

- [ ] MIT License (default)
- [ ] BSD
- [x] GPL v3.0
- [ ] Creative Commons
- [ ] Other: (please specify below)

Additional information (optional)

Part 3: Reproducibility workflow

Scope

The provided workflow reproduces:

- [ ] Any numbers provided in text in the paper
- [x] All tables and figures in the paper
- [ ] Selected tables and figures in the paper, as explained and justified below:

Workflow

Format(s)

- [ ] Single master code file
- [ ] Wrapper (shell) script(s)

- [ ] Self-contained R Markdown file, Jupyter notebook, or other literate programming approach
- [ ] Text file (e.g., a readme-style file) that documents workflow
- [ ] Makefile
- [x] Other (more detail in *Instructions* below)

Instructions

To replicate the Figures, run the following bash commands, respectively, inside the top level directory `glasp-code`.

> Due to hardware specs, it is expected that the results vary a little from the ones described in the paper.

```
Rscript Figures/fig1.R
Rscript Figures/fig2.R
Rscript Section7_real/real_surv.R
```

To replicate the simulation Tables, use the following.

> The following simulations might take some time to finish, depending on the hardware. They will span to use all the cores in the system. Please, use with caution. It is recommended to run this inside the docker container provided.

```
Rscript Section4_linear/main.R
Rscript Section5_cox/surv.R
```

Expected run-time

Approximate time needed to reproduce the analyses on a standard desktop machine:

- [ ] $<$ 1 minute
- [ ] 1-10 minutes
- [ ] 10-60 minutes
- [x] 1-8 hours
- [ ] $>$ 8 hours
- [ ] Not feasible to run on a desktop machine, as described here:

Additional information (optional)

Notes (optional)