dsSurvival: Privacy preserving survival models for federated individual patient meta-analysis in DataSHIELD

Soumya Banerjee [1]&, Ghislain N. Sofack [2,3]&, Thodoris Papakonstantinou[2,3], Demetris Avraam [4,5], Paul Burton [4], Daniela Zöller [2,3]∗, Tom R.P. Bishop [1]∗

1 Medical Research Council Epidemiology Unit, University of Cambridge School of Clinical Medicine, Cambridge, United Kingdom

2 Institute of Medical Biometry and Statistics, Faculty of Medicine and Medical Center – University of Freiburg, Germany

3 Freiburg Center for Data Analysis and Modelling, University of Freiburg, Germany

4 Population Health Sciences Institute, Newcastle University, Newcastle, United Kingdom

5 Department of Public Health, University of Copenhagen, Copenhagen, Denmark

& Equal first authors

∗ Equal senior authors

Corresponding author Tom.Bishop@mrc-epid.cam.ac.uk

Abstract

Objective Achieving sufficient statistical power in a survival analysis usually requires large amounts of data from different sites. Sensitivity of individual-level data, ethical and practical considerations regarding data sharing across institutions could be a potential challenge for achieving this added power. Hence we implemented a federated meta-analysis approach of survival models in DataSHIELD, where only anonymous aggregated data are shared across institutions, while simultaneously allowing for exploratory, interactive modelling.

In this case, meta-analysis techniques to combine analysis results from each site are a solution, but an analytic workflow involving local analysis undertaken at individual studies hinders exploration. Thus, the aim is to provide a framework for performing meta-analysis of Cox regression models across institutions without manual analysis steps for the data providers.

Results We introduce a package (dsSurvival) which allows privacy preserving meta-analysis of survival models, including the calculation of hazard ratios. Our tool can be of great use in biomedical research where there is a need for building survival models and there are privacy concerns about sharing data.

Keywords

Survival analysis, meta-analysis, federated analysis.

Introduction

Survival models are widely used in biomedical research for analyzing survival data [1]. These models help researchers compare the effect of exposures on mortality or other outcomes of interest [2]. The Cox proportional hazards model [3] is one of the most popular survival analysis models and was primarily developed to determine the importance of predictors in survival, by using covariate information to make individual predictions [4].

Achieving sufficient power in survival analysis usually requires large amounts of data from several sites or institutions. Multi-site analysis across studies with different population characteristics help us understand how diseases affect different populations and what it is about these populations that cause these differences. However, the number of cases at a single site is often rather small, making statistical analysis challenging. Also due to the sensitivity of individual-level biomedical data, ethical and practical considerations related to data transmission, and institutional policies, it may sometimes be difficult to share individual-level data [5].

In consortia, this issue is often addressed by manual analysis in each site, followed by a manual meta-analysis of the analysis results from the individual sites. This process is very time-consuming and error-prone, making exploratory analysis (e.g., for understanding different effect patterns in each site) impractical. As an alternative, the DataSHIELD framework can be used.

DataSHIELD is a framework that enables the remote and privacy preserving analysis of sensitive research data [6]. The framework is based on the programming language R [7] [8]. In each site, specifically requested aggregated anonymous analysis results can be requested, which are then combined in a central analysis server. The requirement is that the analysis be privacy preserving and be conducted across globally distributed cohorts.

We have implemented a meta-analysis approach based on the Cox-model in DataSHIELD using individual patient data that are distributed across several sites, without moving those data to a central site i.e. the individual-level data remain within each site and only non-disclosive aggregated data are shared. Our software package for DataSHIELD allows building of survival models and analyzing results in a federated privacy preserving fashion.

Remote federated meta-analysis allows the analysis to come to the data and enables multiple research groups to collate their data [7] [8]. This is an alternative to literature based meta-analysis since study variables and outcomes can be harmonised [9]. Our package offers considerable advantages over: (1) literature based meta-analysis, which suffers from publication bias as well as restricting the analytic end-points you may wish to use; (2) central pooling of data, which provides important governance challenges and can engender privacy risks; and (3) asking researchers in each location to do local analyses based on a shared analysis plan, which all too often demands numerous emails, with repeated reminders, to disseminate analytic protocols and return results for meta-analysis, which is typically time-consuming and can be error prone. Our tool can be of great use in domains where there is a need for building survival models and there are privacy concerns about sharing data.

Main text

Basics of survival analysis

Survival analysis can be used to analyze clinical data if there are records of patient mortality and time to event data. The key quantity is a survival function:

$$S(t) = Pr(T > t) \tag{1}$$

where $S(t)$ is the survival function, t is the current time, and T is a random variable denoting time of death. Pr() is the probability that the time of death is greater than time t i.e. the probability of surviving till time t.

The instantaneous hazard ($\lambda(t)$) is the probability of death occurring within time period [t, t + δ t] given survival till time t. This is related to the survival function as follows:

$$S(t) = \exp(-\int_0^t \lambda(z)dz) \tag{2}$$

The proportional hazards model assumes that the effect of covariates is proportional to the hazard. This is modelled as follows using the hazard function $\lambda(t)$:

$$\lambda(t|X_i) = \lambda_0(t)\exp(\sum_{j=1}^{p}\beta_j X_{ij}) \tag{3}$$

where $\lambda_0(t)$ is called the baseline hazard and $\lambda(t)$ is the hazard at time t. β denotes the parameters and X_{ij} is the jth covariate for the i th subject. We aim to meta-analyze these log hazard ratios.

Implementation

DataSHIELD operates on a distributed architecture that only allows restricted computation. DataSHIELD has a client-server architecture (Fig. 1). There are multiple servers located on separate sites and there is a single analysis client. Assign functions in DataSHIELD perform computation and ultimately create objects that persist on the servers and are not shared with the analysis client. These server-side objects can then be used for subsequent computations. Aggregate functions in DataSHIELD perform computation on each site, check for disclosure risks, and send aggregated results back to a client. The results do not persist on the servers, but can be saved on the client. This is shown in Fig. 1.

The communication between the client and server for the survival models is shown in Fig. 2 for an assign function (*ds.Surv()* to create survival objects on the servers) and aggregate function (*ds.coxphSLMA()* to perform a meta-analysis of Cox regression models). This shows an asynchronous mode of operation in DataSHIELD where multiple parties (sites) perform secure computation. The client-side package is called dsSurvivalClient and the server-side package is called dsSurvival.

The server-side package dsSurvival 1.0.0 contains the functions *SurvDS()* and *coxphSLMADS()*. These functions are configured to reside in modified R environments located behind a firewall at each institution and process the individual-level data at each distinct repository.

dsSurvivalClient contains the functions *ds.Surv()* and *ds.coxphSLMA()*. These functions reside on the conventional R environment of the analyst. The *ds.Surv()* (assign function) calls the server-side function *SurvDS()* to assign survival objects in each site. This can then be used as the response variable in the *ds.coxphSLMA()* (aggregate) function. The *ds.coxphSLMA()* function calls and controls the corresponding server-side functions *coxphSLMADS()* and performs the regression analysis at different sites. These functions implement study-level meta-analysis (SLMA). The estimates from each site are combined and then pooled using fixed effects or random effects meta-analysis.

Computational pipeline and use case

We outline the development and code for implementing survival models (Cox regression) and meta-analysis of hazard ratios in our package (dsSurvival).

A tutorial in bookdown format is available here:

https://neelsoumya.github.io/dsSurvivalbookdown/

In the following, we demonstrate the computational steps using synthetic data. The first step is using DataSHIELD to connect to the server and loading the survival data. We assume that the reader is familiar with these details. We show the steps using synthetic data. There are 3 data sets that are held on the same server but can be considered to be on separate servers/sites. The variable *EVENT* holds the event information and variables *STARTTIME* and *ENDTIME* hold the time information. There is also age and gender information in variables named *age* and *female*, respectively. We will look at how age and gender affect survival time and then meta-analyze the hazard ratios. For details on how to setup the variables, please see the bookdown above.

Algorithm 1: Workflow for performing privacy preserving survival analysis.

```
# %Create survival object
dsSurvivalClient::ds.Surv(time='STARTTIME', time2='ENDTIME', event = 'EVENT',
 objectname='surv_object', type='counting')
# %use constructed Surv object in ds.coxph.SLMA()
coxph_model_full = dsSurvivalClient::ds.coxph.SLMA(formula = 'surv_object ~ D$age+D$female')
# %Summary of survival objects
# We can also summarize a server-side object of type survival::Surv() to provide a non-disclosive
 summary of the server-side object.
dsSurvivalClient::ds.coxphSummary(x = 'coxph_serverside')
# %Diagnostics for Cox proportional hazards models
# There are functions to test for the assumptions of Cox proportional hazards models.
dsSurvivalClient::ds.coxphSLMAassign(formula = 'surv_object ~ D$age+D$female', objectname =
 'coxph_serverside')
dsSurvivalClient::ds.cox.zphSLMA(fit = 'coxph_serverside')
```

The log-hazard ratios and their standard errors from each study can be found after running *ds.coxphSLMA()*. The hazard ratios can then be meta-analyzed using the *metafor* package [10]. Fig. 3 shows an example forest plot with meta-analysed hazard ratios. The plot shows the log hazard ratios corresponding to age in the survival model.

There are two options to generate the survival object. The analyst can generate it separately or inline (for example, by the following command: *dsSurvivalClient::ds.coxph.SLMA(formula = 'survival::Surv(time=SURVTIME,event=EVENT) ~ D$age+D$female')*). If a survival object is generated separately, it is stored on the server and can be used later in an assign function (*ds.coxphSLMAassign()*). This allows the survival model to be stored on the server and can be used later for diagnostics.

Preserving privacy and disclosure checks

Disclosure checks are an integral part of DataSHIELD and dsSurvival. dsSurvival leverages the DataSHIELD framework to ensure that multiple parties perform secure computation and only the relevant aggregated statistical details are shared. We disallow any Cox models where the number of covariate terms are greater than a fraction (default set to 20%) of the number of data points. The number of data points is the number of entries (for all patients) in the survival data. This fraction can be also be changed by the data custodian or administrator in DataSHIELD. We also deny any access to the baseline hazard function.

Diagnostics for Cox proportional hazards models

We generate diagnostics for Cox models using the function dsSurvivalClient::ds.cox.zphSLMA(). These diagnostics can allow an analyst to determine if the proportional hazards assumption in Cox proportional hazards models is satisfied. If the p-values returned by dsSurvivalClient::ds.cox.zphSLMA() are greater than 0.05 for a covariate, then the proportional hazards assumption is likely correct for that covariate.

If the proportional hazards assumptions are violated, then the analyst may wish to modify the model. Modifications may include introducing strata or using time-dependent covariates.

Discussion and Conclusion

dsSurvival is a DataSHIELD package for privacy preserving meta-analysis of survival data distributed across different sites. dsSurvival also performs federated calculation of hazard ratios. Its implementation relies exclusively on the distributed algorithm of the DataSHIELD environment. DataSHIELD facilitates important research particularly amongst institutions that are not allowed to transmit patient-level data to an outside server.

Previously building survival models in DataSHIELD involved using approximations like piecewise exponential models. This involves defining time buckets and is an additional burden on the researcher. A lack of familiarity with this approach also makes people less trusting of the results.

Previous work has looked at reducing the dimensions of a survival model and the reduced feature space model is then shared amongst multiple parties [11]. Survival analysis is also possible in DataSHIELD using dsSwissKnife [12]. However, our package offers advantages such as storing the model on the server-side, diagnostics, integration with client-side meta-analysis and future plans to add in more functionality such as survival curves.

We have released an R package for privacy preserving survival analysis in DataSHIELD. Our tool can be of great use in domains where there is a need for building survival models and there are privacy concerns about sharing data. We hope this suite of tools and tutorials will serve as a guideline on how to use survival analysis in a federated environment.

Limitations

Our approach implements study-level meta-analysis. This is a computationally faster approach but is also a limitation, especially if the units of meta-analysis are centres within a study. This may reduce the number of events per center and normality approximations implicit in two-stage meta-analysis may be violated. In the future we will implement functionality of iteratively fitting a single model across all studies. We will also develop plotting of privacy preserving survival curves and the ability to have time-dependent covariates in survival models. Our package also does not return Schoenfeld or Martingale residuals (due to privacy concerns), which are used as diagnostics for survival models. Finally in the future we will apply our package on real world data and solve any practical issues that arise.

Abbreviations

SLMA: Study Level Meta-Analysis.

Declarations

Ethics approval and consent to participate

No ethics approval and consent to participate was necessary.

Consent for publication

Not applicable.

Availability of data and materials

This study does not generate any data.

A tutorial in bookdown format with code, diagnostics, plots and synthetic data is available here:

https://neelsoumya.github.io/dsSurvivalbookdown/

All code is available from the following repositories:

https://github.com/neelsoumya/dsSurvivalClient/

https://github.com/neelsoumya/dsSurvival/

Our package can provide figures for downstream analysis. An example script for generating forest plots for hazard ratios is available here:

https://github.com/neelsoumya/dsSurvival/blob/main/forestplot_FINAL.R

Competing interests

All authors declare they have no conflicts of interest to disclose.

Funding

This work was funded by EUCAN-Connect under the European Union's Horizon 2020 research and innovation programme (grant agreement no. 824989). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. The views expressed are those of the authors and not necessarily those of the funders.

Authors' contributions

SB and GS carried out the analysis and implementation, participated in the design of the study and drafted the manuscript. TP, DA and PB gave critical comments and edited the manuscript. TB and DZ directed the study. All authors gave final approval for publication.

Acknowledgements

We acknowledge the help and support of the DataSHIELD technical team. We are especially grateful to Stephen Sharp, Elaine Smith, Stuart Wheater, Patricia Ryser-Welch and Wolfgang Viechtbauer for fruitful discussions and feedback.

References

1. Altman D, De Stavola B, Love S, Stepniewska K (1995) Review of survival analyses published in cancer journals. British journal of cancer 72: 511–518.

2. Machin D, Cheung YB, Parmar M (2006) Survival analysis: a practical approach. John Wiley & Sons.

3. Cox DR (1972) Regression models and life-tables. Journal of the Royal Statistical Society: Series B (Methodological) 34: 187–202.

4. Hartmann O, Schuetz P, Albrich WC, Anker SD, Mueller B, et al. (2012) Time-dependent cox regression: serial measurement of the cardiovascular biomarker proadrenomedullin improves survival prediction in patients with lower respiratory tract infection. International journal of cardiology 161: 166–173.

5. Blasimme A, Fadda M, Schneider M, Vayena E (2018) Data sharing for precision medicine: policy lessons and future directions. Health Affairs 37: 702–709.

6. Gaye A, Marcon Y, Isaeva J, LaFlamme P, Turner A, et al. (2014) Datashield: taking the analysis to the data, not the data to the analysis. International journal of epidemiology 43: 1929–1944.

7. DataSHIELD — DataSHIELD — Newcastle University. URL `http://www.datashield.ac.uk/`.

8. EUCAN Connect. URL https://www.eucanconnect.eu/.

9. Pastorino S, Bishop T, Crozier SR, Granström C, Kordas K, et al. (2019) Associations between maternal physical activity in early and late pregnancy and offspring birth size: remote federated individual level meta-analysis from eight cohort studies. BJOG An Int J Obstet Gynaecol 126: 459–470.

10. Viechtbauer W (2010) Conducting Meta-Analyses in R with the metafor Package. J Stat Softw 36: 1–48.

11. Yu S, Fung G, Rosales R, Krishnan S, Rao RB, et al. (2008) Privacy-preserving cox regression for survival analysis. In: Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. New York, New York, USA: ACM Press, pp. 1034–1042. doi:10.1145/1401890.1402013. URL http://dl.acm.org/citation.cfm?doid=1401890.1402013.

12. Dragan I, Sparsø T, Kuznetsov D, Slieker R, Ibberson M (2020) dsSwissKnife: An R package for federated data analysis. bioRxiv : 2020.11.17.386813.

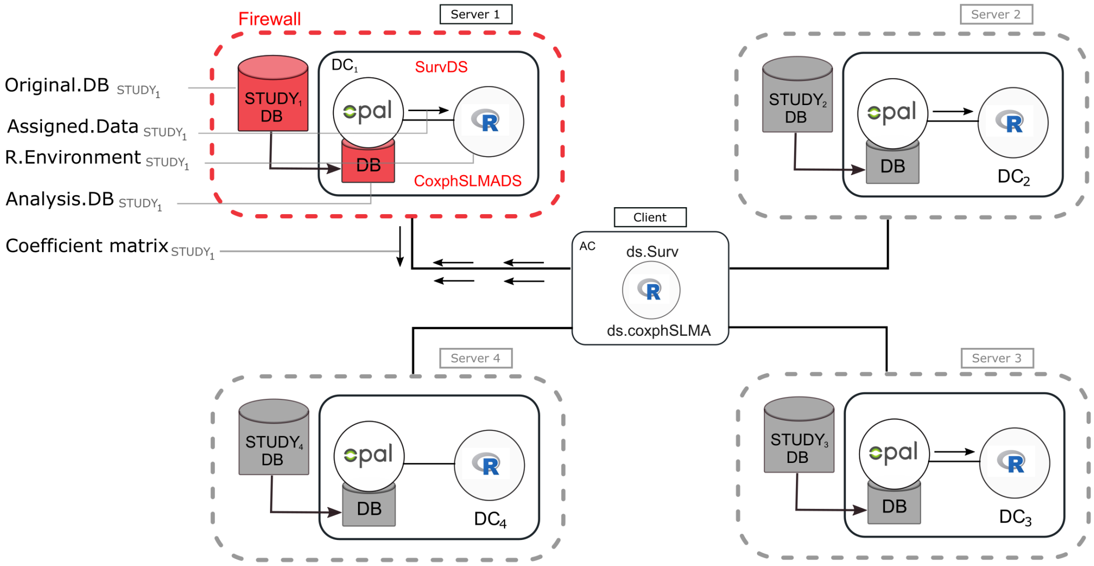

Figure 1. Client-server architecture of DataSHIELD. The diagram shows four study sites/servers (DC) each having data stored in the Original.DB. The analyst (client) sends commands from the analysis computer (AC) to each study site to request the specific data (Assigned.Data) to be analyzed. This could be all the variables or specific variables stored in Analysis.DB. R commands are also sent from the analysis computer to every study telling it to create survival objects and fit the Cox proportional hazard model. Each site responds to instructions sent by creating the survival object and fitting the model. This fitting is carried out in the R environment of each study. The coefficient matrices, standard errors, and odds ratios from each site are then pooled and meta-analyzed using fixed optimization methods, and only non-disclosive statistics are returned to the analyst.

Figures

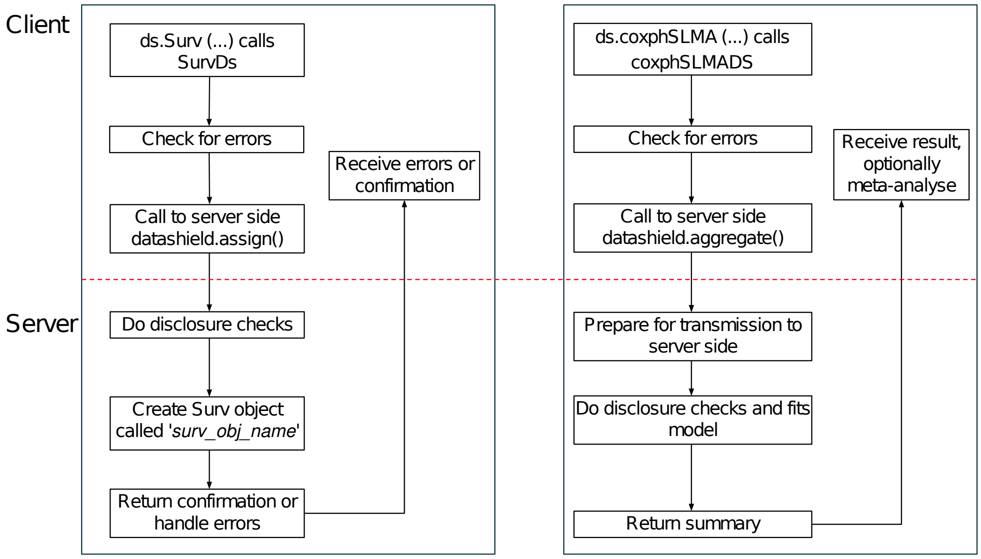
Client
ds.Surv (...) calls SurvDs
Check for errors
Call to server side datashield.assign()
Receive errors or confirmation
Server
Do disclosure checks
Create Surv object called '*surv_obj_name*'
Return confirmation or handle errors
ds.coxphSLMA (...) calls coxphSLMADS
Check for errors
Call to server side datashield.aggregate()
Receive result, optionally meta-analyse
Prepare for transmission to server side
Do disclosure checks and fits model
Return summary

Figure 2. Architecture of client and server side functions for building survival models in dsSurvival. Left panel: An assign function for creating a server-side survival object using *ds.Surv()*. Right panel: An aggregate function for a Cox proportional hazards model using *ds.coxphSLMA()*.

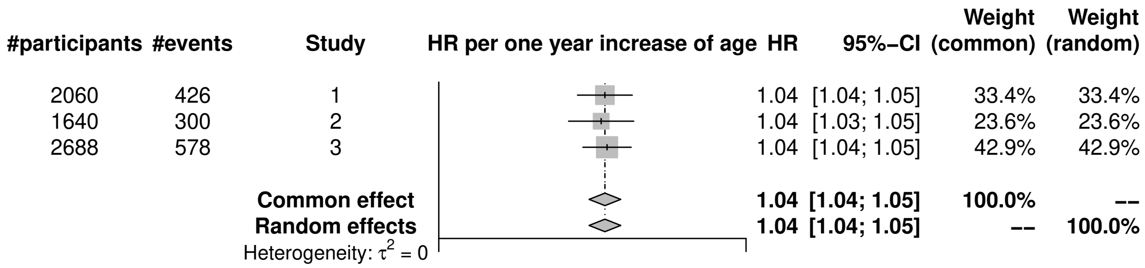

Figure 3. A plot showing the meta-analyzed hazard ratios generated from dsSurvival. A Cox proportional hazards model was fit to synthetic data. The hazard ratios correspond to age in a survival model.