nature communications

Article

https://doi.org/10.1038/s41467-025-62525-z

FedECA: federated external control arms for causal inference with time-to-event data in distributed settings

Received: 28 October 2024

Accepted: 24 July 2025

Published online: 13 August 2025

Check for updates

Jean Ogier du Terrail[1,13], Quentin Klopfenstein[1,13], Honghao Li[1,13], Imke Mayer[1], Nicolas Loiseau[1], Mohammad Hallal[1], Michael Debouver[1], Thibault Camalon[1], Thibault Fouqueray[1], Jorge Arellano Castro[1], Zahia Yanes[1], Laëtitia Dahan[2], Julien Taïeb[3], Pierre Laurent-Puig[4,5], Jean-Baptiste Bachet[6], Shulin Zhao[4], Remy Nicolle[7], Jérôme Cros[8], Daniel Gonzalez[9], Robert Carreras-Torres[10], Adelaida Garcia Velasco[10,11], Kawther Abdilleh[12], Sudheer Doss[12], Félix Balazard[1] & Mathieu Andreux[1]

External control arms can inform early clinical development of experimental drugs and provide efficacy evidence for regulatory approval. However, accessing sufficient real-world or historical clinical trials data is challenging. Indeed, regulations protecting patients' rights by strictly controlling data processing make pooling data from multiple sources in a central server often difficult. To address these limitations, we develop a method that leverages federated learning to enable inverse probability of treatment weighting for time-to-event outcomes on separate cohorts without needing to pool data. To showcase its potential, we apply it in different settings of increasing complexity, culminating with a real-world use-case in which our method is used to compare the treatment effect of two approved chemotherapy regimens using data from three separate cohorts of patients with metastatic pancreatic cancer. By sharing our code, we hope it will foster the creation of federated research networks and thus accelerate drug development.

Development of innovative drugs is a long and challenging task with increasing costs[1]. The probability of success of a new drug is low, with about 10% of drugs that enter clinical trials reaching FDA approval[2]. Phase III randomized trials, that aim at establishing clinical efficacy, fail approximately in one case out of two[3]. Improving this rate is crucial to overcome those obstacles and accelerate the access to new drugs while reducing the development costs.

Statistical methods were developed to compare the efficacy of a treatment to a control group that is built with data from external sources to the current trial (External Control Arm - ECA). ECA methods take into

[1]Owkin, Inc., New York, NY, USA. [2]Department of Digestive Oncology, Hôpital la Timone, Marseille, France. [3]GI oncology department Georges Pompidou European Hospital, Université Paris Cité, CARPEM CCC, 20 rue leblanc 75015 Paris, APHP, Paris, France. [4]Centre de Recherche des Cordeliers, Sorbonne Université, Inserm, Université Paris Cité, Paris, France. [5]Institut du Cancer Paris CARPEM, AP-HP Centre, Hôpital Européen Georges Pompidou, Paris, France. [6]Sorbonne University, Hepatogastroenterology and digestive oncology department, Pitié Salpêtrière hospital, APHP, Paris, France. [7]Université Paris Cité, Centre de Recherche sur l'Inflammation (CRI), INSERM, U1149, CNRS, ERL 8252, F-75018 Paris, France. [8]Department of Pathology, Université Paris Cité - FHU MOSAIC, Beaujon Hospital, Clichy, France. [9]Fédération Francophone de Cancérologie Digestive, Dijon, France. [10]Institut d'Investigació Biomèdica de Girona (IDIBGI), Girona, Catalonia, Spain. [11]Department of Medical Oncology, Catalan Institute of Oncology, Doctor Josep Trueta University Hospital, Girona, Catalonia, Spain. [12]Pancreatic Cancer Action Network, El Segundo, CA, USA. [13]These authors contributed equally: Jean Ogier du Terrail, Quentin Klopfenstein, Honghao Li. ✉e-mail: jean.duterrail.scientific.contact@gmail.com

account the potential bias introduced by the non-randomized nature of the control group, and enable early assessment of treatment efficacy, which can inform the transition from a single-arm phase II to a phase III clinical trial[4,5]. ECA are increasingly used in clinical applications[6], and are receiving more and more attention from regulatory agencies (FDA, EMA)[7,8]. Externally controlled trials may also substitute randomized controlled trials (RCT) in specific situations where an RCT would be deemed unfeasible or untimely. This is the case for rare diseases where patient recruitment is difficult and time-consuming[9], as well as in some oncology cases involving specific patient subgroups[6,10–12].

Statistically, the lack of randomization between the treatment group and the external control group makes a naive comparison unreliable due to confounding bias. Therefore, assuming that the treatment effect is identifiable[13], statistical or machine learning methods[14–20] are needed in this context to provide valid estimates and inferences of the treatment effect. Despite advances in statistical methods, data sharing is a major obstacle to the feasibility of ECA. Due to their sensitivity, health data are strictly regulated, e.g., by the General Data Protection Regulation (GDPR) in the EU and the Health Insurance Portability and Accountability Act (HIPAA) in the US. Even after careful pseudonymization[21], sharing health data remains a complex endeavor, notably because of data ownership and liability issues. Thereby, in practice, it is difficult to set up an ECA involving phase II data from a pharmaceutical company and potentially real-world data from different hospitals or medical institutions.

To address this data sharing challenge, various machine learning techniques have been proposed in recent years. Among them, federated learning (FL)[22], a privacy-enhancing technology (PET), makes it possible to extract knowledge and train models from multiple institutions without pooling data. It has already been used with success in similar settings to connect pharmaceutical companies[23] and hospitals[24,25] in federated research networks. Herein, we investigate the use of FL to build ECA, focusing on time-to-event outcomes such as progression-free survival (PFS) or overall survival (OS), which are predominant in oncology RCTs[26].

In this work, we present FedECA, a federated external control arm method, which is a federated version of the well-known inverse probability of treatment weighting (IPTW) statistical method[27] for time-to-event outcomes. FedECA facilitates ECAs for pharmaceutical companies by giving them access to real-world control patients from multiple institutions, while limiting patient data exposure thanks to FL. We first demonstrate the efficacy of our approach on synthetic data created using realistic data generation processes, both with in-RAM simulations and on a federated network deployed with 10 distant synthetic centers located in the cloud. We show that FedECA achieves identical conclusions to IPTW on pooled data as well as better statistical power compared to a competing method based on federated analytics (FA), while also controlling for moment differences between the two arms. Furthermore, we showcase two examples of FedECA applied to real patient data, starting with an FL simulation using real data from trials before proceeding to demonstrate the end-to-end deployment of a real federated research network between three research institutions located in different countries.

Results

FedECA, a federated ECA method

Here we describe our federated extension of the IPTW method for time-to-events outcomes: FedECA. FedECA estimates the treatment effect by comparing the experimental drug arm stored in one center with a control arm defined by external data held within different centers, as illustrated in Fig. 1. FedECA consists of three main steps, all performed via FL. It first trains a propensity score model using logistic regression to obtain weights, then fits a weighted time-to-event Cox model to correct for potential confounding bias, and finally computes an aggregated statistic which allows to test the treatment effect. See “Method overview” for more details. Simultaneously, we develop, alongside FedECA, FA methods in order to both visualize and validate the results of FedECA (see “Federated Analytics for end-to-end federated ECA analysis”) as one would in an equivalent pooled ECA analysis.

Figure 1 illustrates the advantages of our proposed method, where data can remain on the premises of the participating centers and only aggregated information is shared over multiple communication rounds. Herein, an aggregator node is responsible for orchestrating the training process, aggregating and redistributing the results to all centers, without directly seeing the raw data itself. This is in contrast to the classical ECA analysis, where data is pooled into a single center and data privacy is not an issue. In this privacy-enhanced context that we tackle, and that we describe in details in “Privacy of FedECA”, all kinds of data analyses from simple statistics such as computing global variance or histograms to more complex methods such as IPTW are markedly more difficult to apply as one can only share aggregated data.

FedECA is equivalent to a standard IPTW model trained on pooled data

In spite of such difficulties, we demonstrate both mathematically and numerically FedECA’s equivalence to the pooled IPTW model. Mathematical proof of the equivalence under proper assumptions can be found in “Inverse probability weighted WebDISCO”. In particular, an important assumption underlying this derivation is the use of Breslow’s approximation for tied times when constructing the partial likelihood of the Cox model. In addition to mathematical equivalence, we also study numerical equivalence, which could be affected by the propagation of machine precision errors during repeated computations. We demonstrate this equivalence between pooled and federated IPTW analyses on realistic simulated data with right-censored events, 10 normally distributed and correlated covariates, and a treatment allocation depending on the covariates.

We refer to “Synthetic data generating model of time-to-event outcome” for details on the data generation process. In the following numerical experiment, we compare the results obtained from a classical IPTW analysis where data are pooled into the same place with those obtained from FedECA operating in distributed settings. Here we monitor four key metrics: the hazard ratio of the treatment allocation covariate estimated from a Cox model, the partial likelihood of the Cox model, the *p*-value associated to the hazard ratio (HR), derived from a two-sided Wald test that assumes under the null hypothesis an asymptotic chi-squared distribution with one degree of freedom, and the propensity scores estimated from the logistic regression. We repeat this simulation 100 times and report the relative errors between pooled IPTW and FedECA in Fig. 2. It should be noted that the relative error we report also takes into account potential differences depending on the optimizer used to train the propensity model. For instance, in our implementation for the pooled reference, we rely on `sklearn`’s default optimizer, which is the Limited memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) optimizer, and thus differs slightly from the exact Newton-Raphson steps used in FedECA.

The results in Fig. 2 show that the relative errors between FedECA and the pooled IPTW are negligible, not exceeding 0.2%, illustrating the effectiveness of the proposed federated optimization process. Moreover, Supplementary Fig. 2 shows that the number of centers among which the data is split has no impact on FedECA’s performance. Hence, we illustrate that up to a negligible error, due to finite precision numerical errors in the optimization process, FedECA provides results that are equivalent to the classical IPTW despite not having access to all data in the same location.

FedECA and MAIC both control SMD

To assess the performance of FedECA, we compare it with another, more naïve, federated method in terms of the ability to correctly detect the

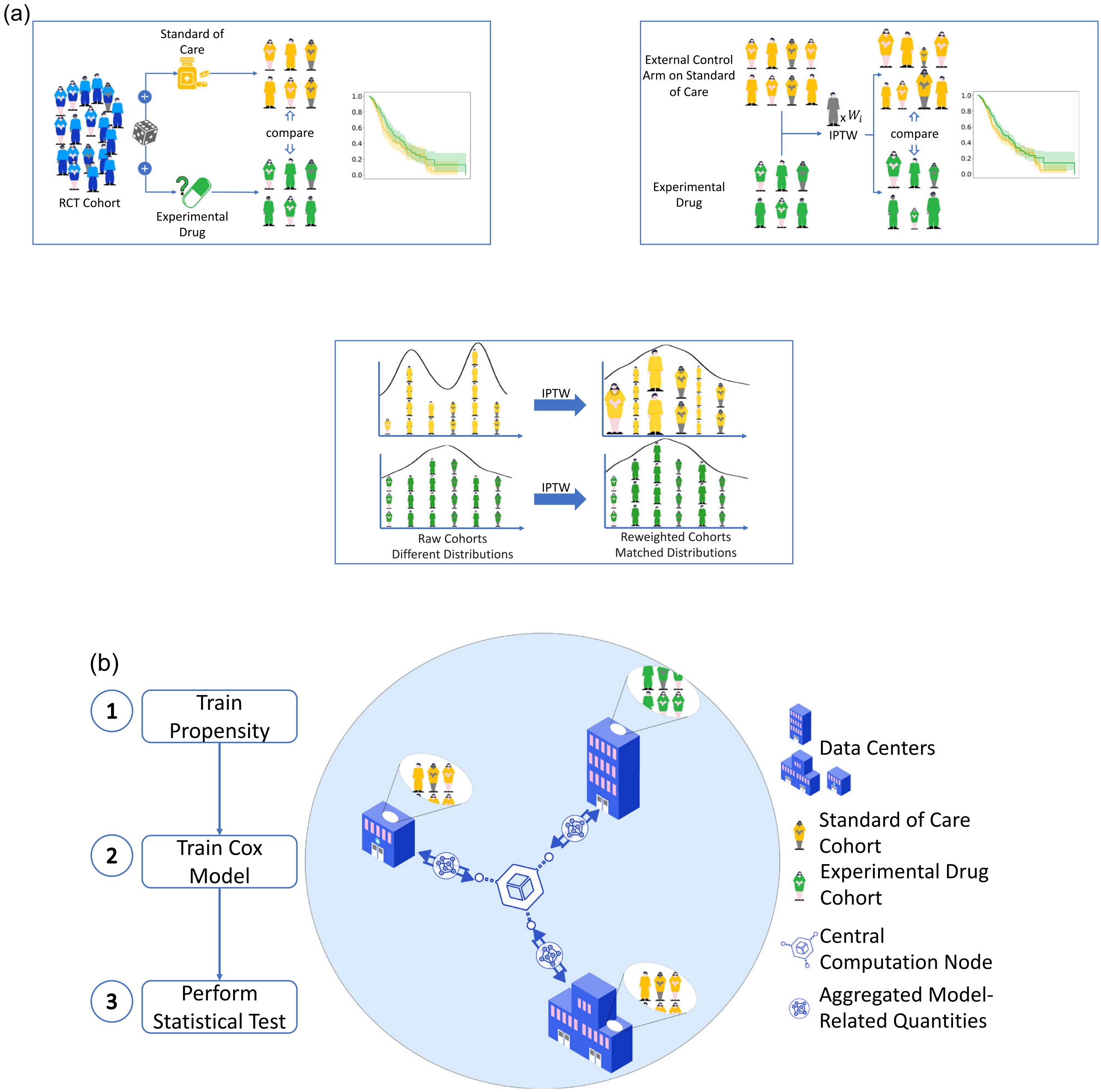

Fig. 1 | Illustration of randomized controlled trials (RCT) versus an external control arm (ECA) analysis. FedECA graphical abstract. **a** In an RCT, patients are randomly assigned to either the experimental (i.e., treatment) or the control arm. In an ECA, patients are assigned to the treatment arm, while the control arm is defined using historical data. Due to this absence of randomization and the resulting confounding, the two groups of patients cannot be compared directly. To overcome this issue, a model is used to capture the association between the treatment allocation and the confounding factors. From this model, weights are computed and are used to balance the two arms to ensure comparability. Then, the weights are incorporated into a Cox model to estimate the treatment effect. Finally, a statistical test is performed to assess the significance of the measured treatment effect. **b** In the considered setting, patient data is stored in different geographically distinct centers, and a similar analysis as in (**a**) is attempted thanks to our algorithm FedECA. A trusted third party is responsible for the orchestration of the training processes, which consists of exchanging model-related quantities across the centers. No individual patient data is shared between the centers, and only aggregated information is exchanged, which limits patient data exposure while producing equivalent results. Some of the symbols used in the figure have been bought to the Noun Project, Inc. by M.H., granting M.H. perpetual, non-exclusive, worldwide rights to such symbols.

presence of a treatment effect. We start by focusing on the reweighting step of FedECA, an important step to correct for the confounding bias, as measured by the standardized mean difference (SMD) of covariates between the two patient groups after reweighting. The SMD is a coarse, univariate measure which summarizes, for each covariate, the balance between the two groups by only looking at the first and second order moments, see "SMD estimator". However, SMD is expected by regulators[28], which ask ECA methods to control SMD below a threshold (usually 10%). In this context, the main competitor of FedECA is the matching-adjusted indirect comparison (MAIC)[29], a method that allows to reweight the individual patient data (IPD) from a treatment arm to match the mean and the standard deviation of data from an external control arm for which the IPD are not accessible. The two reweighting approaches differ mainly in two ways. First, since MAIC's reweighting procedure involves communicating statistics only once between the centers, it is essentially a FA method, whereas FedECA's reweighting uses FL. Second, while MAIC explicitly

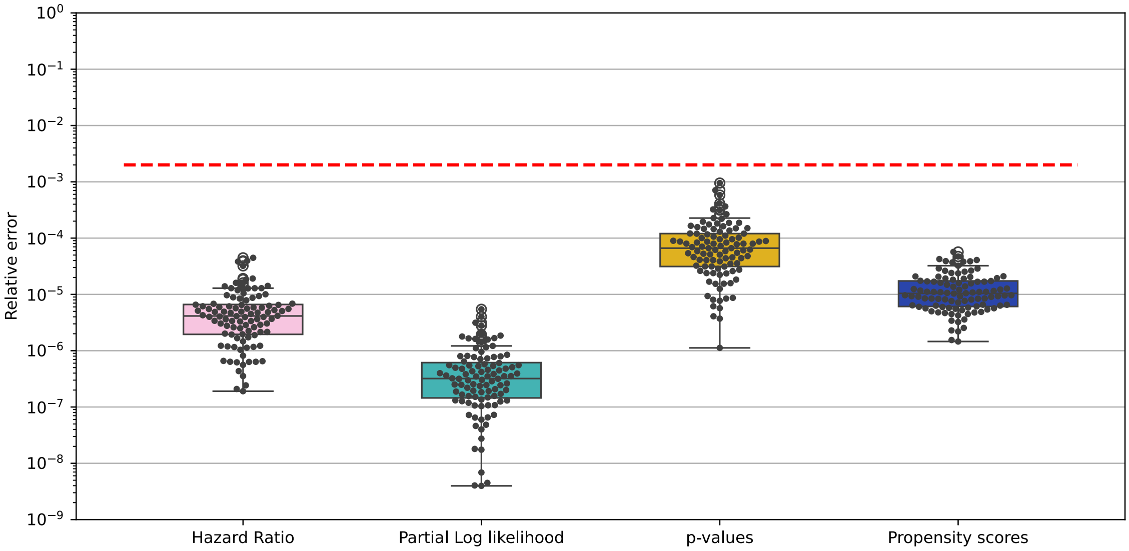

Fig. 2 | Pooled equivalence between IPTW and FedECA. Box- and swarm-plots of the relative errors between FedECA and the pooled IPTW on four different quantities: the hazard ratio of the treatment allocation covariate estimated from a Cox model, the partial likelihood of the Cox model, the *P*-value associated to the hazard ratio, and the propensity scores estimated from the logistic regression. For each quantity, relative error is defined as the absolute difference between the pooled IPTW value and the FedECA value, divided by the pooled IPTW value. Each quantity was computed from $n = 100$ repetitions of the simulation, that is computed by running FedECA and pooled IPTW on n random draws of 1000 samples with 10 covariates. Red dotted line indicates a relative error of 0.2% between FedECA and the pooled IPTW. Boxplot and swarm-plot use the seaborn Python library's default settings, that is: boxes are from the first to the third quartiles, the black line being the median, and whiskers extend to the lowest (resp. highest) data point still within 1.5 inter-quartile range of the lower (upper) quartile. No statistical test was used. Source data are provided as a Source Data file.

enforces perfect matching of low-order moments irrespective of the covariate shift, leading to zero SMD by design (see "Estimation of the treatment effect"), FedECA does multivariate balancing through the propensity scores.

To compare the two methods, we consider scenarios with different levels of covariate shift, which is a parameter that controls the intensity of the confounding factors on the treatment allocation variable, biasing the two groups (see "Synthetic data generating model of time-to-event outcome" for more details on the data generation). In this experiment, on one end of the spectrum, treatment allocation does not depend on the covariates: there is no covariate shift. Therefore, the SMDs of all covariates are small, even before reweighting. On the other end of the spectrum, treatment allocation depends more and more on the covariates values. Therefore, we expect weighting to be necessary to control SMD. The results are illustrated in Fig. 3a where we show the mean absolute SMD as a function of the covariate shift for FedECA, MAIC and the unweighted baseline. For low covariate shift (≤0.5) all three methods control the SMDs of the covariates, while for medium to high covariate shifts (>0.5), the unweighted method fails to control the SMDs, while both MAIC and FedECA control SMD. More details per-covariate are available Fig. 3b for the two extremes.

FedECA outperforms MAIC in power to detect a treatment effect

Following the comparison using SMD, we now compare the effect of the different reweighting offered by FedECA and MAIC in terms of treatment effect estimation, as measured by type I error and statistical power. After reweighting, FedECA trains a Cox model using weighted time-to-event data of both arms. The presence of a treatment effect is determined by the hazard ratio estimated from the Cox model, as well as the associated *p*-value. A *p*-value less than 0.05 is considered significant. For the type I error experiment, we generate synthetic data with no treatment effect between the two arms, while for the statistical power experiment, we generate synthetic data with a true treatment effect. Note that for MAIC, unlike for FedECA, the training of the Cox model assumes the pooling of all IPD on treatment allocation, time to event, as well as censoring. While this assumption seemingly contradicts the ECA's setup, it can be considered as an idealization of a real-world use case of ECA analysis using MAIC, as detailed in "Estimation of the treatment effect". Figure 3c shows the estimated type I error and statistical power for FedECA, MAIC and the unweighted baseline, under varying covariate shift and number of samples. One of the key factors influencing the type I error and statistical power estimation is the variance estimation method applied for each treatment effect point estimation. Here, we compared three variance estimation methods: the bootstrap estimator, the robust sandwich estimator, and the naive estimator based on the inversion of the observed Fisher information[16]. For FedECA, only the bootstrap variance estimator successfully controls the type I error at around 5%. In comparison, the robust variance estimator systematically overestimates the variance, leading to an inflated *P*-value and thus a conservative empirical type I error rate. Finally, the naive variance estimator fails to control the type I error. These results are consistent with previous work[16]. For MAIC with bootstrap variance estimation, resampling with replacement is performed only on the treatment group, since the pseudo IPD of the control arm is fixed by design. Compared with FedECA, it only controls the type I error for small covariate shifts and loses control when the covariate shift increases. The robust variance estimator shows similar variance overestimation to FedECA. For the unweighted baseline, as it cannot account for the confounding bias introduced by covariate shift, it quickly loses control of the type I error as soon as the covariate shift is no longer zero. Next, for those methods that successfully control the type I error, we compare their statistical power. FedECA with the bootstrap variance estimator shows the best performance, followed by FedECA with the robust variance estimator. Both FedECA variants outperform MAIC with the robust variance estimator as the covariate shift and number of samples change. Based on the above results, we choose the bootstrap variance estimator for all experiments on treatment effect estimation.

FedECA can be used in real-world conditions on synthetic data

We host up to 11 "servers" in the cloud (10 "centers" and 1 aggregation server) and deploy the Substra[30] software on all centers. Details of the

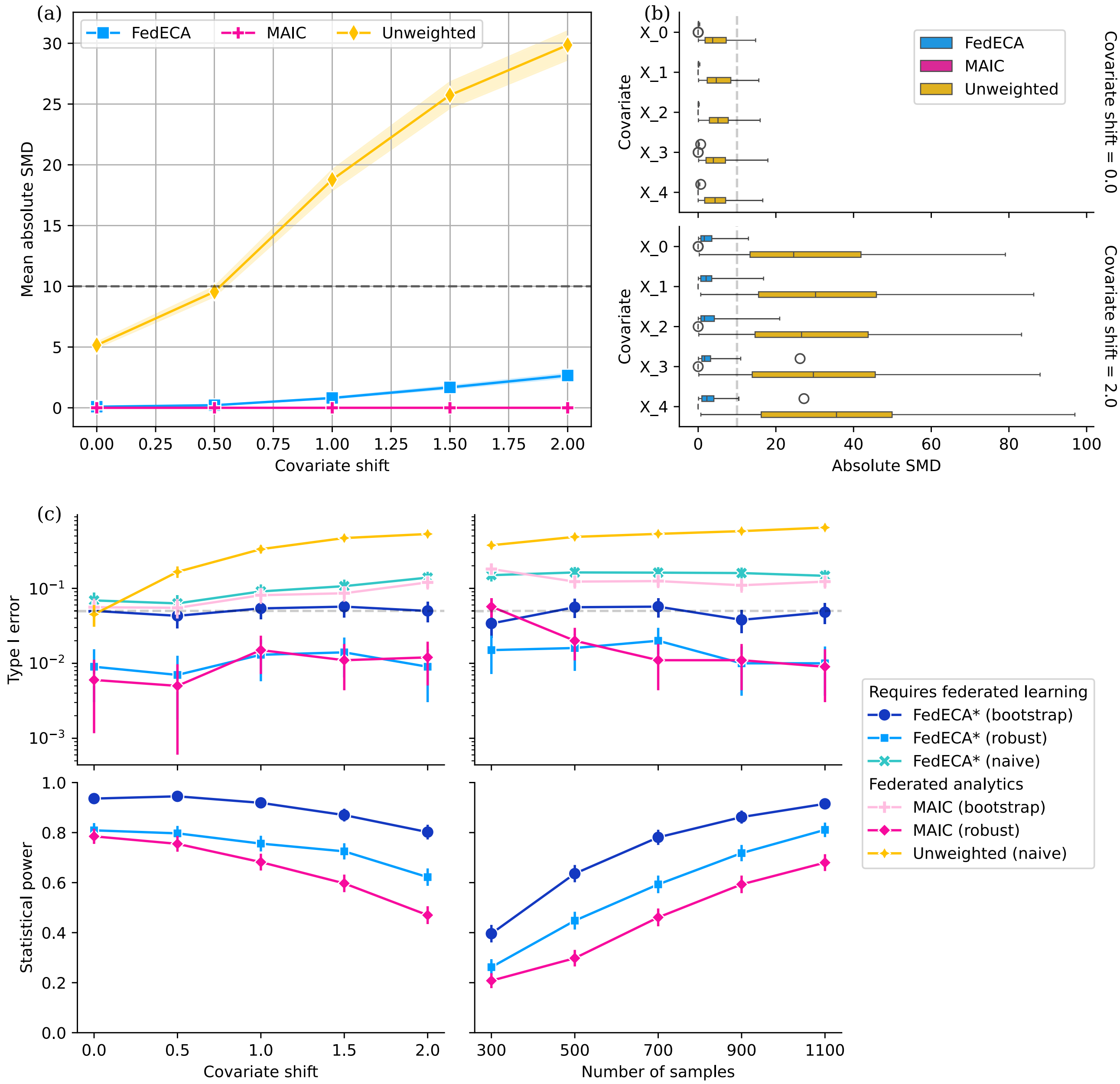

Fig. 3 | Comparison of different methods on statistical power, type I error of treatment effect estimates, as well as standardized mean difference (SMD) of covariates between the two treatment arms. a Curves representing the mean absolute SMD computed on 10 covariates as a function of the covariate shift for three different methods: FedECA, MAIC and the non-adjusted treatment effect estimation (unweighted) over $n = 100$ repetitions. Shaded area is the two-sided 95% interval around the mean assuming standard normal distributions. **b** Boxplots representing the distribution of the absolute SMD over the $n = 100$ repetitions for the first five covariates. Each estimation of SMD is based on $n = 100$ repetitions of propensity score estimation. For all simulations, we generate 10 covariates and 1000 samples. Boxplot and swarmplot uses the seaborn Python library's default settings that is: boxes are from the first to the third quartiles, the black line being the median, and whiskers extend to the lowest (resp. highest) data point still within 1.5 inter-quartile range of the lower (upper) quartile. **c** Comparison of different methods on statistical power and type I error of treatment effect estimation. Different variance estimation methods leading to different p-values are given in parentheses after each method giving point estimates of the hazard ratio. In particular, the naive variance estimation is based on the simple inversion of the observed Fisher information. For statistical power, only results of methods that consistently control the type I error around/under 0.05 (marked by gray dashed lines in top panels) are shown. Each estimation of statistical power or type I error is based on $n = 1000$ repetitions of treatment effect estimation. For bootstrap-based variance estimating methods, the number of bootstrap resampling is set to 200. For all simulations, we assume 10 covariates. The hazard ratio of the simulated treatment effect is set to 0.4 for the estimation of statistical power, and to 1.0 for the estimation of type I error. For simulations with varying covariate shifts (the two panels on the left), the number of samples is fixed at 700. For simulations with varying sample size (the two panels on the right), the covariate shift is fixed at 2.0. The asterisk on FedECA indicates that, due to the time-consuming nature of the power analysis, their more lightweight pooled-equivalent counterparts were used instead (pooled IPTW). For confidence intervals, we use the central limit theorem applied to Bernoulli variables to compute parameters of the associated normal and plot the two-sided 95% intervals as error bars. No statistical test was used. Source data are provided as a Source Data file.

Table 1 | Treatment effect estimation on radiographic progression-free survival by comparing different regimens across different trials

Experiment (Treatment vs. Control)	Method	Treatment data source	Control data source	log(HR)	HR (95% CI)	Z	p
Apa-AA-P vs. AA-P	FedECA	Trial 1	Trial 2	− 0.42	0.66 (0.55, 0.78)	− 4.72	<0.00001
	Literature[32]	Trial 1	Trial 1	− 0.36	0.70 (0.60, 0.83)	− 4.31	<0.0001
AA-P vs. P	FedECA	Trial 1	Trial 2	− 0.69	0.50 (0.42, 0.59)	− 8.04	<0.000001
	Literature[33]	Trial 2	Trial 2	− 0.63	0.53 (0.45, 0.62)	− 7.77	<0.001
AA-P vs. AA-P	FedECA	Trial 1	Trial 2	− 0.11	0.90 (0.77, 1.05)	− 1.33	0.18
Apa-AA-P vs. P	FedECA	Trial 1	Trial 2	− 0.99	0.37 (0.31, 0.45)	− 10.42	<0.000001

Trial 1 refers to NCT02257736, Trial 2 refers to NCT00887198. All *p*-values resulting from the Wald test performed on the entry of $\hat{\boldsymbol{\beta}}$ corresponding to the treatment allocation, assuming a χ^2 distribution with 1 degree of freedom, are obtained with bootstrap variance estimation. Exact *p*-values are from top to bottom (excluding results from literature) $2.3e^{-6}$, $8.9e^{-16}$, 0.184456 and $1.9e^{-25}$. Source data are provided as a Source Data file.

cloud setup are available in "Real-world experiments setup details". For each experiment, we use the first of the servers as the trusted third party performing the aggregation (the "server") and the other servers as data owners holding a different part of the data (the "centers"). Each "center" has a different set of credentials, giving it different permissions over the assets created in the federated network. Each center registers a predefined subset of the synthetic data, as if it were its own, through the Substra system. FedECA can be run on the deployed network simply by changing the type of backend used and specifying the identifiers of the datasets (hashes) registered into the Substra platform as inputs to the fit method following `scikit-learn`'s fit API[31]. An example of the FedECA Python API is given in Supplementary Fig. 1. We monitor the runtime of the full pipeline when running in-RAM and on the cloud as a function of the number of centers. We give conservative estimates by setting a large target number of rounds (20) for the training of the propensity model and the Cox model, and by computing the federated robust sandwich estimator, which adds overhead and is not necessary if using the bootstrap variance estimator.

In-RAM experiments take a few seconds with 10 centers, which is to be compared with IPTW on pooled data, which has a below-second runtime. This slowdown is mainly due to (1) the sequential processing of each client (2) the static nature of the Substra framework, which forces the execution of a higher target number of rounds than needed for convergence (see "Early stopping within a static distributed framework" for further explanation). While (2) is a fundamental limitation of Substra, (1) could be improved by using Python multiprocessing. The real-world runtime is almost constant with respect to the number of clients and is under 2 h (1 h18 min on average, with a standard deviation of around 3 min). A complete breakdown of the different runtimes across settings is shown in Supplementary Table 1. Insofar as 10 centers is already a large number in the considered cross-silo FL setup, this result suggests good scalability in terms of speed, provided that an appropriate infrastructure can be deployed across the different centers. Our result is consistent with previous Substra deployments[23].

FedECA can be used on real world use-cases: application on real prostate cancer data with simulated federated learning

We access data from two phase III trials in metastatic castration-resistant prostate cancer[32,33] from the Yale University Open Data Access (YODA) project[34,35]. In all cases, we simulate an FL setup in which data are held by two synthetic centers, one holding the treated arm and the other the remaining patients. We note that, according to our experiments Supplementary Fig. 2, the number of centers has no impact on FedECA's performance (nor does the way the data are distributed across the different centers), and we therefore expect roughly the same results if we had chosen different data splits. Full details of cohort construction can be found in "Prostate cancer cohort construction".

In the following, we present results focused on estimating the average treatment effect on radiographic progression-free survival (rPFS), the primary outcome of both trials, by comparing regimens across trials. We first show the results reported for each trial found in the associated publications. Then, we conduct simulated ECA studies by replacing the abiraterone acetate + prednisone (AA-P) arm of each trial with the same arm from the other trial. In addition, we compare the two AA-P arms to test the exchangeability of the two study populations and to validate previous ECA analyses. Finally, we compare the apalutamide + abiraterone acetate + prednisone (Apa-AA-P) arm with the prednisone (P) arm, which is not seen in the literature and demonstrates the potential usefulness of our method to provide additional evidence that is otherwise unavailable or costly to produce in terms of time and resources. In Table 1, we show the consistency of the treatment effect estimations with the published results, as well as the non-significance of the treatment effect when comparing two AA-P arms. In addition, we show a significant treatment effect of Apa-AA-P versus P, which is reasonable considering the superiority of Apa-AA-P over AA-P, and of AA-P over P.

FedECA can be used on real world use-cases: application on real metastatic pancreatic cancer data in a real deployed federated research network

We access metastatic pancreatic cancer data in three cancer centers: the Fédération Francophone de Cancérologie Digestive (FFCD), which holds data from two completed RCTs[36,37], the Institut d'Investigació Biomèdica de Girona (IDIBGI) and the Pancreatic Cancer Action Network (PanCAN), which holds data from clinical practice. For the two FFCD trials, the primary outcome is the percentage of patients alive and without radiological and/or clinical progression 6 months after the randomization, and the secondary outcomes are OS and PFS. Full details of cohort construction can be found in "Pancreatic adenocarcinoma cohort construction". Regarding the FL setup, we deploy a Substra-based federated network across the three centers. Contrary to the experiment in "FedECA can be used on real world use-cases: application on real prostate cancer data with simulated federated learning" which used a simulated FL setup, in this case, in order to perform any statistical analysis involving data from multiple centers, now one must go through the Substra federated network infrastructure and launch Substra's "compute plans", which introduces overhead and imposes constraints on what can be computed safely (see "Privacy of FedECA"). More details of this new, more stringent, FL setup can also be found in "Metastatic pancreatic adenocarcinoma data infrastructure setup".

We aim to compare the treatment efficacy of FOLFIRINOX versus gemcitabine and nab-paclitaxel on OS. Given that all three centers have patients in both treatment groups, we begin by testing a key assumption required to combine cohorts from different centers receiving the same treatment: the exchangeability between pairs of centers. For each treatment group, we compare patients from two centers and test if there is a center effect after correcting for measured confounders with IPTW (e.g., we test if patients under FOLFIRINOX

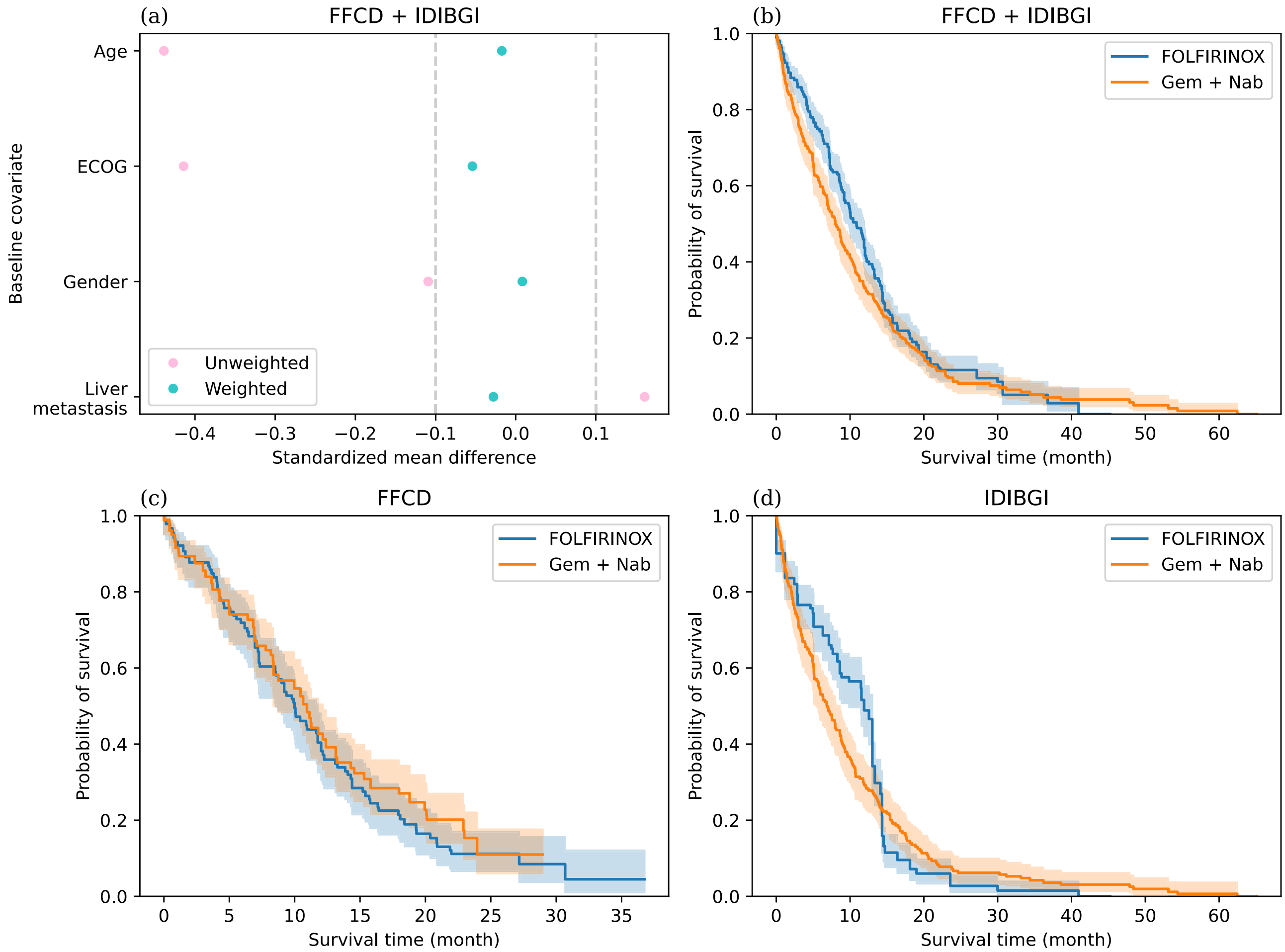

Fig. 4 | Real-world FOLFIRINOX effect estimation using FedECA versus local analyses. **a** SMD of covariates between the two arms of the combined FFCD+IDIBGI cohort, before and after weighting by FedECA's propensity model. Therefore, each dot represents the SMD over $n = 153 + 225 = 378$ samples. **b** Weighted Kaplan-Meier curves of the combined FFCD + IDIBGI cohort using FedECA's propensity model. Sample size is $n = 378$. The 95% confidence intervals displayed are obtained using the exponential Greenwood formula. **c** Weighted Kaplan-Meier curves of the FFCD cohort using a local propensity model. Sample size is $n = 225$. The 95% confidence intervals displayed are obtained using the exponential Greenwood formula. **d** Weighted Kaplan-Meier curves of the IDIBGI cohort using a local propensity model. Sample size is $n = 153$. The 95% confidence intervals displayed are obtained using the exponential Greenwood formula. Associated p-values can be found in the associated table. Source data are provided as a Source Data file.

have different outcomes between FFCD and IDIBGI). Such comparisons are done for all three pairs of centers. Unfortunately, we observe a strong center effect at the PanCAN center whose population has significantly better outcomes than the other centers in both treatment groups (Supplementary Table 9 and Supplementary Fig. 7). We suspect that there is a residual immortal time bias not addressed by the correction we performed in this cohort (see “Pancreatic adenocarcinoma cohort construction”). Consequently, we exclude the entire PanCAN cohort from the treatment effect analysis that we present in the remainder of this section, and leave the results of a federated analysis including data from all three centers in Supplementary Table 8 and Supplementary Fig. 6 for illustrative purposes.

Using the combined FFCD and IDIBGi cohort in FedECA, we compare the treatment efficacy of FOLFIRINOX versus gemcitabine and nab-paclitaxel on OS. Figure 4a demonstrates that the propensity model in FedECA, trained with FL, effectively balances the two patient groups over the group of selected covariates, reducing the SMD between the two arms to below 10% for all covariates, which was not achieved before reweighting. Table 2 shows the estimated treatment effect on overall survival of FOLFIRINOX over gemcitabine and nab-paclitaxel with an HR of 0.84 (0.68, 1.04) and an associated p-value of 0.118. While this result does not reach statistical significance, it is consistent with the literature using IPTW on pooled data (e.g., HR = 0.77 (0.70, 0.85)[38]), and supports the trend toward superiority of FOLFIRINOX over gemcitabine and nab-paclitaxel. For comparison, local analyses at each center (Table 2) result in broader confidence intervals of the estimated HR. Figure 4c, d highlight the above results by displaying propensity-weighted Kaplan-Meier curves. Note that in Fig. 4 and Supplementary Fig. 6, results on the combined cohorts are obtained through the application of FA without pooling data, see “Federated Analytics for end-to-end federated ECA analysis”, Supplementary Fig. 13 and Supplementary Fig. 14.

Discussion

In this work we have introduced FedECA, a federated extension of the IPTW method for estimating treatment effects in the context of external control arms. Our results demonstrate that FedECA replicates its pooled-equivalent counterpart IPTW up to machine precision (see Fig. 2), ensuring the same statistical properties as IPTW. Compared to the simpler FA baseline MAIC, FedECA shows superior statistical power and controls the type I error while effectively adjusting for confounding factors as shown by the standardized mean difference below 10% (see Fig. 3c). Unlike many stratified competitors[39–43], FedECA is well-suited for drug development settings, where treated and control patients are in separate locations, with pharmaceutical companies holding only treated patient data, and control patients spread across multiple

Table 2 | Estimation of treatment effect on overall survival by comparing FOLFIRINOX against gemcitabine + nab placlitaxel

Method	Data source	log(HR)	HR (95% CI)	Z	p
FedECA	FFCD, IDIBGI	− 0.17	0.84 (0.68, 1.04)	− 1.56	0.118
IPTW	FFCD	0.13	1.13 (0.80, 1.61)	0.71	0.480
IPTW	IDIBGI	− 0.12	0.89 (0.65, 1.22)	− 0.72	0.471

Results with single data source are obtained without FL. All *p*-values resulting from the Wald test performed on the entry of $\hat{\boldsymbol{\beta}}$ corresponding to the treatment allocation, assuming a χ^2 distribution with 1 degree of freedom are obtained with bootstrap variance estimation. Exact *p*-values are from top to bottom 0.118425, 0.479963 and 0.470586. Source data are provided as a Source Data file.

institutions. It also remains applicable in scenarios where both treatment arms are distributed across different institutions, as demonstrated in the metastatic pancreatic adenocarcinoma experiment. While MAIC can only estimate the average treatment effect on the control (ATC), FedECA supports the estimation of the ATE, the average treatment effect on the treated (ATT), as well as the ATC by changing the weights used in the estimation. Furthermore, FedECA also enables covariate adjustment through adjusted IPTW, providing researchers with greater flexibility in choosing between a marginal effect or a conditional one[44]. This distinction is important in the context of time-to-event outcomes due to the non-collapsibility of the hazard ratio.

FedECA is not only an algorithm but also a software solution that has been deployed and tested in real-world settings. A first experiment demonstrates that FedECA can reproduce results of individual clinical trials[32,33], while also enabling analyses that would not be feasible if each trial were analyzed separately. Although the data used in this experiment were not physically distributed in different locations, the results of those simulations are representative of what could be achieved on similar data in a real federated setting. In a second experiment, we deployed a Substra federated research network across three cancer centers in three different countries: the Fédération Française de Cancérologie Digestive (FFCD, France), the institut d'investigació biomèdica de Girona (IDIBGI, Spain) and the Pancreatic Cancer Action Network (PanCAN, USA). This study aimed to estimate the ATE of FOLFIRINOX (leucovorin and fluorouracil plus irinotecan and oxaliplatin) over gemcitabine and nab-paclitaxel[45–48]. After excluding data from one center affected by immortal time bias, our results, while not reaching statistical significance, are consistent with the findings of meta-analyses and pooled analyses from the literature. While the results of our analysis rely on a smaller sample size (n = 378) than the largest previous efforts[38,48–51], it brings additional evidence on this topic of rising medical interest[52].

One of the key challenges of real-world data is handling missing values or missing features, as encountered in our metastatic pancreatic adenocarcinoma experiment. While extensive research exists on missing data imputation in machine learning and its impact on causal inference analyses[53–58], it remains underexplored in the context of federated learning. In this study, we used a naive solution by applying MissForest[54] independently at each site. Imputation per-site is suboptimal and could possibly further increase heterogeneity and biases. Addressing these limitations requires future work on federated missing data imputation methods. Beyond missing data, another important aspect is the choice of confounding factors when building an ECA based on propensity scores[59]. FedECA remains sensitive to misspecification in the propensity score method, as its pooled version, IPTW. When building an ECA, one should carefully select the confounders to include in the propensity score method and should consider the possibility of unmeasured confounders as well as ways to perform sensitivity analysis to assess the robustness of the results[60]. Additionally, when performing an ECA analysis, it is crucial to ensure consistent data collection across centers, particularly regarding the definition of endpoints. For example, progression-free survival (PFS) is not defined in a standardized way in clinical practice, which can lead to bias and errors in the estimation of the treatment effect.

In this work, we focus on the (log) hazard ratio as the treatment effect estimand, as it is commonly used with time-to-event endpoints[32,33,38]. Other effect measures have been proposed for time-to-event outcomes, such as contrasts of restricted mean survival time (RMST)[61]. The latter has the advantage of being collapsible and offers better clinical relevance and interpretability in certain applications[62]. An IPTW-based estimator for the difference of RMST has been introduced[63] and could guide an extension of FedECA to RMST-based effect estimation in a federated setting.

Another important open research direction that we leave to future work is to study more deeply the security profile of standalone FedECA. Indeed, the federation of both the propensity score model and the Cox proportional hazards (PH) model currently requires transmitting aggregated information to the central server. In the absence of local differential privacy (DP) mechanisms, this transmitted information could, in theory, be exploited by adversarial attacks (we sketch how such attacks could be performed in "Privacy of FedECA"). For this reason, in addition to providing the pseudo-code of our FedECA algorithms, we traced in our implementation all quantities exchanged across centers when performing the different federated learning and analytics algorithms involved in FedECA allowing a full privacy audit of the implementation by experts. This information is reported in "Privacy of FedECA". We tested the application of DP to the first part of the FedECA training, which is the training of the propensity model. However, it was shown to be already detrimental to the statistical analysis (see Supplementary Figs. 3 and 11 and Supplementary Table 13). This DP experiment raises an interesting question, which is whether the use of DP in the clinical trials of tomorrow is warranted, given that it trades off the accuracy of treatment effect estimation for data privacy. We do not claim to answer this question in this work.

Another privacy-enhancing layer that could be added to FedECA would be to use secure aggregation (SA)[64] to hide individual contributions through cryptographic operations. This would demonstrably hide potentially sensitive information such as per-client risk sets and would also allow private set unions (PSU)[65] to compute the global event times, or could also be used to secure the aggregator node[64,66].

To conclude, FedECA is a federated method for real-world distributed ECA analysis. FedECA is particularly suited for the scenario where treated patient data are in a distinct center and the external control arm is split across different centers that cannot share their data. FedECA is a federated extension of IPTW that reproduces the result of a pooled analysis, yielding similar treatment effect estimation and similar statistical guarantees. Thus FedECA enables causal inference in distributed ECA settings while limiting IPD exposure. We demonstrated that FedECA is not only a simulation tool but a valid method for real-world applications, showcasing its ability in two clinically different contexts.

Implementing federated methods in real-world healthcare environments, as we did for the metastatic pancreatic cancer use case, still presents major technical and operational challenges that are absent from simulated FL. Our implementation of FedECA is based on an open-source FL software hosted by the LFAI, which had already been successfully used in the targeted high-security healthcare setting with both pharmaceutical companies and cancer centers[23,25]. Having a trusted implementation like this one, whose security has been audited and that is compatible with heterogeneous IT (Information Technology) environments, is a prerequisite for building real FL networks, as the implementation must be vetted by the different IT teams from all partner institutions. While we focus in this paper on the technical and methodological challenges, we want to emphasize that the non-algorithmic challenges associated with setting up any FL network between real institutions are still, to this day, potentially the main bottlenecks in such analyses, as we touch upon in "Related works".

We hope FedECA, alongside other real-world applications of federated learning, can shift perspectives and drive collaborations between hospitals, medical centers, and pharmaceutical companies, demonstrating that medical discoveries are achievable while minimizing patient data exposure.

Methods

Inclusion and ethics statement

We provide below the inclusion and ethics statements related to the patient data we access. Note that we only access data from past studies in a retrospective fashion.

he ethics of this retrospective study on clinical data collected during care and from past clinical trials were validated by each institution according to corresponding local regulations. We list below the corresponding statements from each of the participating cancer centers.

Regarding FFCD data, the study was conducted in accordance with the ethical principles outlined in the Declaration of Helsinki, International Council for Harmonization of Technical Requirements for Pharmaceuticals for Human Use (ICH) requirements and Good Clinical Practice guidelines; it received authorization from the French national medicines agency (ASNM), and independent ethics committee (number 214-R18 and 14-12-79 respectively for PRODIGE 35 and PRODIGE 37). The study was both registered in clinicaltrials.gov (NCT02352337 for PRODIGE 37 and NCT02827201 for PRODIGE 35) and EudraCT 2014-004449-28.

For IDIBGI data the study was approved by the Comitè d'Ètica d'Investigació amb Medicaments CEIM GIRONA the 8th of August 2023 (Acta 11/2023) under reference CEIM code 2023.165 with principal investigators ADELAIDA GARCIA VELASCO and ROBERT CARRERAS TORRES and SANOFI-AVENTIS SA as promoter.

Finally for PanCAN data the sponsor of the IRB was the Pancreatic Cancer Action Network (# KYT001) the IRB reference is 20192301, study 1265508 with main investigator Matrisian, Lynn.

For FFCD and IDIBGI, an informed consent form with non-opposition principle for the reuse of their health data for research purposes was communicated to the patient at the time of admission in each of the study centers in accordance with European regulations. For PanCAN's data, authorization was obtained to use patient data through PanCAN's Know Your Tumor program. All data from PanCAN used are fully anonymized and as a result, did not require consent.

Participants were not compensated.

Problem statement

We consider a setting where one center, e.g., a pharmaceutical company, hosts data of all treated patients and approaches several other centers to use their data as control to define a distributed ECA. Although FedECA also works in the more general case where there is no constraint on patient mixing within the participating centers.

We suppose that FDA guidelines for ECA[28] have been applied to direct data harmonization so that variables, assigned or received treatments, data formats, variable ranges, outcome definitions and inclusion criteria match across centers. However we touch on the practical challenges associated with such a requirement when studying the metastatic pancreatic adenocarcinoma use-case in "Pancreatic adenocarcinoma cohort construction". We assume that variables are not missing and relegate discussing data imputation questions associated with real-world data to "Pancreatic adenocarcinoma cohort construction" as well.

We further posit that all centers arrived at a consensus on a common list of confounding factors that influence both the exposure and the outcome of interest, we give examples of such lists in "Prostate cancer cohort construction" and "Pancreatic adenocarcinoma cohort construction".

Moreover, the studied treatment effect is the average treatment effect (ATE) evaluated using the hazard ratio (HR) with time-to-event outcomes. See the discussion regarding this design choice.

Finally we assume the deployment of a federated solution, such as Substra[30], between the centers as well as a trusted third party or aggregator. This setup is detailed for the real-world deployment use-cases in "Real-world experiments setup details".

We note that, because of the scope of this article, we do not necessarily dwell on such technicalities; in practice however, they are a crucial aspect of FL projects and should not be underestimated[24,67,68].

Federated external control arms (FedECA)

Method overview. The ECA methodology we use relies on three main steps: training a propensity score model, fitting a weighted Cox model, and testing the parameter related to the treatment. We first introduce them here in a pooled-level fashion, before explaining in detail how we adapted them to the federated setting in the next sections.

Setup and notations

Each patient is represented by covariates $\boldsymbol{X} \in \mathbb{R}^p$. It undergoes treatment $A \in \{0, 1\}$, corresponding either to the treated ($A = 1$) or control ($A = 0$) arm. We denote $\boldsymbol{x}_i$ the covariates of the i-th patient, and a_i its treatment allocation. Following treatment, the patient has an event of interest (e.g., death or disease relapse) at a random time T^*. The patient may leave the arm before the event of interest is actually observed, a phenomenon called censoring: we denote the observed time T, whose realizations are denoted t_i. We note $\delta_i = 1$ if this corresponds to a true event, resp. $\delta_i = 0$ if censorship took place. Additionally, we define the observed outcome $Y_i = (T_i, \delta_i)$. Let n denote the total number of patients, indexed by i.

Let $\mathcal{S}$ denote the finite set of all observed times, i.e. $\mathcal{S} = \{t_i\}_{i=1}^n$. At a given time s, let $\mathcal{D}_s$ denote the set of patients with an event at this time, i.e.

$$\forall s \in \mathcal{S}, \mathcal{D}_s = \{i | t_i = s, \delta_i = 1\}, \tag{1}$$

and let $\mathcal{R}_s$ denote the set of patients at risk at this time, i.e.

$$\forall s \in \mathcal{S}, \mathcal{R}_s = \{i | t_i \geq s\}. \tag{2}$$

Further, let $\mathring{\mathcal{S}}$ denote the set of times where at least one true event occurs, i.e.

$$\mathring{\mathcal{S}} = \{s \in \mathcal{S} | \mathcal{D}_s \neq \emptyset\}. \tag{3}$$

Data is distributed among K different centers, with n_k samples per center. We denote $\boldsymbol{x}_{i,k}$ the i-th covariate vector from the k-th center; accordingly, $a_{i,k}$ denotes the treatment allocation, $y_{i,k} = (t_{i,k}, \delta_{i,k})$ the observed outcome, where $t_{i,k}$ is the observed time event, and $\delta_{i,k}$ whether a true event took place. Similarly, for each time s and center k, we define the subset $\mathcal{D}_{s,k}$ and $\mathcal{R}_{s,k}$ as the respective restrictions of $\mathcal{D}_s$ and $\mathcal{R}_s$ to center k.

Propensity score model training

Due to the lack of randomization, for each sample, the probability of being assigned the treatment A might depend on the covariates $\boldsymbol{X}$. We train a propensity score model $p_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta}$ such that

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) \approx \mathbb{P}[A | \boldsymbol{X} = \boldsymbol{x}]. \tag{4}$$

We use a logistic model for $p_{\boldsymbol{\theta}}$, i.e.,

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{\theta}^T \boldsymbol{x})}. \tag{5}$$

Its negative log-likelihood is given by

$$\mathcal{J}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \{a_i \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i) + (1 - a_i)\log(1 - p_{\boldsymbol{\theta}}(\boldsymbol{x}_i))\}. \tag{6}$$

In "Federated propensity model training", we explain how this model is trained in a federated setting.

Inverse probability weighted treatment (IPTW)

For each sample i, we define an IPTW weight $w_i \in (0, +\infty)$ based on the propensity score model trained in the previous step as

$$w_i = \begin{cases} \frac{1}{\max(p_\theta(\boldsymbol{x}_i), \varepsilon)} & \text{if } a_i = 1, \\ \frac{1}{\max(1 - p_\theta(\boldsymbol{x}_i), \varepsilon)} & \text{otherwise.} \end{cases} \tag{7}$$

In order to avoid overflow errors, $\varepsilon > 0$ was set to 10^{-16} in our experiments.

We note that it might be that in the case of insufficient overlap, weights would take extreme values. In our cases it is not what we observe from looking at per-center histograms of propensity scores in Supplementary Fig. 8. In the general case future work might be needed to deal appropriately with extreme values.

We then train a weighted Cox proportional hazards (CoxPH) model with parameters $\boldsymbol{\beta} \in \mathbb{R}^q$, related to patient-specific variables $\boldsymbol{z}_i \in \mathbb{R}^q$. We stress that the variables $\boldsymbol{z}_i$ are not the same as the covariates $\boldsymbol{x}_i$. More precisely, for the vanilla IPTW method, the sole covariate used is the treatment allocation, i.e., $\boldsymbol{z}_i = a_i$. In the general case of the adjusted IPTW (adjIPTW) method, one may use additional covariates, especially if they are known confounders. We note that our federated framework can support both classical IPTW and adjIPTW, unlike in[40], although we choose to illustrate our results with IPTW for the sake of simplicity.

The CoxPH model is fitted by maximizing a data-fidelity term consisting in the partial likelihood $L(\boldsymbol{\beta})$ with Breslow's approximation for tied times[69]:

$$L(\boldsymbol{\beta}) = \prod_{i:\delta_i = 1} \left(\frac{e^{\boldsymbol{\beta}^T \boldsymbol{z}_j}}{\sum_{j:t_j \geq t_i} w_j e^{\boldsymbol{\beta}^T \boldsymbol{z}_j}} \right)^{w_i} = \prod_{s \in \mathring{\mathcal{S}}} \prod_{i \in \mathcal{D}_s} \left(\frac{e^{\boldsymbol{\beta}^T \boldsymbol{z}_i}}{\sum_{j \in \mathcal{R}_s} w_j e^{\boldsymbol{\beta}^T \boldsymbol{z}_j}} \right)^{w_i}, \tag{8}$$

where the second equation has been rewritten using the sets $\mathcal{D}_s$ and $\mathcal{R}_s$. For numerical stability, we use the negative log-likelihood $\ell(\beta) = \log L(\beta)$, which reads

$$\ell(\boldsymbol{\beta}) = -\sum_{s \in \mathring{\mathcal{S}}} \sum_{i \in \mathcal{D}_s} \left\{ w_i \boldsymbol{\beta}^T \boldsymbol{z}_i - w_i \log \left(\sum_{j \in \mathcal{R}_s} w_j e^{\boldsymbol{\beta}^T \boldsymbol{z}_j} \right) \right\}. \tag{9}$$

While $\ell(\boldsymbol{\beta})$ represents a data-fidelity term, we also add a regularization $\psi(\boldsymbol{\beta})$ with strength $\gamma > 0$, leading to the full loss

$$\mathcal{L}(\boldsymbol{\beta}) = \ell(\boldsymbol{\beta}) + \gamma \psi(\boldsymbol{\beta}). \tag{10}$$

In "Inverse probability weighted WebDISCO", we describe how we minimize the loss $\mathcal{L}$ in a federated setting, which is one of the main technical innovations of this paper.

Variance estimation and statistical testing

Once the weights are fitted, we estimate the variance matrix of $\hat{\boldsymbol{\beta}}$ using a robust variance estimator[70]. Let us denote

$$\zeta_s^0(\boldsymbol{\beta}) = \sum_{j \in \mathcal{R}_s} w_j e^{\boldsymbol{\beta}^T \boldsymbol{z}_j}, \tag{11}$$

$$\boldsymbol{\zeta}_s^1(\boldsymbol{\beta}) = \sum_{j \in \mathcal{R}_s} w_j e^{\boldsymbol{\beta}^T \boldsymbol{z}_j} \boldsymbol{z}_j, \tag{12}$$

$$\boldsymbol{\zeta}_s^2(\boldsymbol{\beta}) = \sum_{j \in \mathcal{R}_s} w_j e^{\boldsymbol{\beta}^T \boldsymbol{z}_j} \boldsymbol{z}_j \boldsymbol{z}_j^T, \tag{13}$$

and $\hat{\zeta}_s^0(\boldsymbol{\beta})$, $\hat{\boldsymbol{\zeta}}_s^1(\boldsymbol{\beta})$, $\hat{\boldsymbol{\zeta}}_s^2(\boldsymbol{\beta})$ the analogous quantities using the estimated weights $\{\hat{w}_i\}_{i=1}^n$.

Following[40,70], the robust variance estimator of the variance of $\hat{\boldsymbol{\beta}}$ takes the following form:

$$\widehat{\boldsymbol{Var}}(\hat{\boldsymbol{\beta}}) = \boldsymbol{H}^{-1} \boldsymbol{Q} (\boldsymbol{H}^{-1})^T, \tag{14}$$

where

$$\boldsymbol{H} = \sum_{s \in \mathring{\mathcal{S}}} \sum_{i \in \mathcal{D}_s} \hat{w}_i \left(\frac{\hat{\boldsymbol{\zeta}}_s^2(\hat{\boldsymbol{\beta}})}{\hat{\zeta}_s^0(\hat{\boldsymbol{\beta}})} - \frac{\hat{\boldsymbol{\zeta}}_s^1(\hat{\boldsymbol{\beta}}) \hat{\boldsymbol{\zeta}}_s^1(\hat{\boldsymbol{\beta}})^T}{\hat{\zeta}_s^0(\hat{\boldsymbol{\beta}})^2} \right), \tag{15}$$

$$\boldsymbol{Q} = \sum_{i=1}^{n} \hat{\boldsymbol{\varphi}}_i(\hat{\boldsymbol{\beta}}) \hat{\boldsymbol{\varphi}}_i(\hat{\boldsymbol{\beta}})^T, \tag{16}$$

$$\begin{aligned} \hat{\boldsymbol{\varphi}}_i(\hat{\boldsymbol{\beta}}) = {} & \delta_i \hat{w}_i \left(\boldsymbol{z}_i - \frac{\hat{\boldsymbol{\zeta}}_s^1(\hat{\boldsymbol{\beta}})}{\hat{\zeta}_s^0(\hat{\boldsymbol{\beta}})} \right) - \hat{w}_i \exp(\hat{\boldsymbol{\beta}}^T \boldsymbol{z}_i) \boldsymbol{z}_i \sum_{s' \in \mathring{\mathcal{S}}} \sum_{j \in \mathcal{D}_{s'}} \frac{\hat{w}_j \mathbb{1}_{\{s' \leq s\}}}{\hat{\zeta}_{s'}^0(\hat{\boldsymbol{\beta}})} \\ & + \hat{w}_i \exp(\hat{\boldsymbol{\beta}}^T \boldsymbol{z}_i) \sum_{s' \in \mathring{\mathcal{S}}} \sum_{j \in \mathcal{D}_{s'}} \frac{\hat{w}_j \mathbb{1}_{\{s' \leq s\}} \hat{\boldsymbol{\zeta}}_{s'}^1(\hat{\boldsymbol{\beta}})}{\hat{\zeta}_{s'}^0(\hat{\boldsymbol{\beta}})^2}, \text{ for all } i \in \mathcal{D}_s, s \in \mathring{\mathcal{S}}, \end{aligned} \tag{17}$$

with $\mathbb{1}_{\{s' \leq s\}}$ the indicator function that has the value 1 on all times s' (with events) and is 0 otherwise.

Eventually, a Wald test is performed on the entry of $\hat{\boldsymbol{\beta}}$ corresponding to the treatment allocation, assuming a χ^2 distribution with 1 degree of freedom[71].

Related works. Before diving into the details of the federation of the propensity score model and the weighted Cox model, we provide some context explaining the position of FedECA in the literature.

Methodology

In the case of binary or continuous outcomes, inverse probability of treatment weighting (IPTW) can be directly federated, and has been explored extensively[72–77]. In contrast, to the best of our knowledge, few works have explored the federation of ML model training compatible with ECA for time-to-event outcomes.

The difficulty of this federation is that the straightforward application of FL algorithms such as Federated Averaging[22] to time-to-event ML models is impossible due to the non-separability of the Cox proportional hazards (PH) loss[78,79]. Careful federation of the training of ML models capable of handling time-to-event outcomes is possible[78,79] but often requires either to use tree-based models[80,81], approximations[79,82] or can only be performed in stratified settings[39–43], which limits the applicability of such federated analyses for ECA analyses.

Indeed, existing stratified federated IPTW methods such as[40] cannot be applied to ECA as, in the realistic setting we consider, the treatment variable is constant within each center and thus comparison between the treated and untreated groups cannot be done locally from within a single center. A recent work proposed a propensity score method to estimate hazard ratios in a federated weighted Cox PH model[83]. The main difference with our work lies in the fact that they have considered propensity scores based on the combination of local propensity scores (computed in each center) and global ones, demonstrating superior performance than the global scores alone. However, as previously stated, in this paper, we consider a setting where local propensity score models cannot be trained locally to predict treatment allocation since the variable to predict is constant in each center.

In particular, we extend WebDISCO[78], which, alongside[83], is to the best of our knowledge, one of the few exact methods enabling federated learning of time-to-event models in ECA contexts. Our method can also be seen as an extension of stratified IPTW for time-to-event outcomes from[40,83] to the non-stratified case that allows application to ECA.

Our methodological contributions do not stop there as FedECA also 1. supports adjusted IPTW using any sets of covariates for the training of the Cox model, 2. proposes a federated algorithm for robust distributed estimation 3. develops an efficient bootstrap implementation in FL as well as 4. provides two federated analytics estimators (Federated SMD and Federated Kaplan-Meier) allowing to perform an end-to-end ECA study in a federated setting.

Other lines of work tackle the federated analytics setting where no learning is involved and propose to use aggregated data (AD), such as matching-adjusted indirect comparison (MAIC)[29] to perform direct comparisons in combination with the available individual patients data.

Finally, another popular research direction is to propose private representations of patient covariates[84–87] that can be pooled into a central server. These methods have the drawback of not yielding pooled-equivalent results. Furthermore, centralizing these representations increases the potential leakage risks associated with a successful, even if unlikely, attack, compared to a federated storage system.

We summarize in Supplementary Table 2 the differences between FedECA and methods from the literature.

Real-world federated learning

Besides the technical methodology differences, FedECA stands out from related works thanks to its actual implementation on a real federated learning network connecting three clinical centers described in "FedECA can be used on real world use-cases: application on real metastatic pancreatic cancer data in a real deployed federated research network". This practical application demonstrates how our method can effectively address questions about treatment efficacy in clinical practice in the case of metastatic pancreatic cancer while enhancing the privacy of individual patients' data. We believe this is FedECA's most important contribution as most of the literature on FL only studies simulated scenarios. The main reason why most FL research is theoretical is that real-world federated networks are still, to this day, highly complex to set up and operate.

This is due to several factors, notably:

- The need for trust, which is a critical factor to build collaborations between competing institutions and FL providers. In practice, trust is often established through successful prior partnerships between pairs of actors or facilitated when the principal investigator has strong credentials to show, which help them engage with new stakeholders. Open-source code is also a key enabler for trust in such collaborations.
- The need for a contractual and legal framework within which one can deploy a federated network between different legal entities respecting local jurisdictions.
- The federated learning (FL) solution must be compatible with potentially heterogeneous IT systems as some institutions may refuse to store their data in normalized cloud environments hosted in specific countries due to concerns over data ownership.
- The federated network implementation has to be vetted by each of the IT team of the partner institution to make sure there is no data leakage.
- FL collaborations also require harmonizing the data beforehand, which in practice is often mostly manual and requires the help of data engineers and doctors onsite as well as AI specialists across all participating centers and is coordinated usually through e-mails by the principal investigator.
- The usual data-science workflow is rendered much more complex by constraints on data access and sharing, which limit the kind of analyses that can be run.

The software we are using, Substra, is dockerized, has been audited for its security and is easily integrated into existing infrastructure although it usually requires DevOps (Development Operations) teams in each hospital or institution to be successfully deployed.

With this in mind, we go on to precisely describe the federation scheme we employ by specifying all quantities that are communicated between the centers and the aggregator.

Federated propensity model training. Our goal is to fit a model for the propensity score (5) based on distributed data $\{(\boldsymbol{x}_{i,k}, a_{i,k})_i\}_{k=1}^{K}$. Let $\mathcal{J}$ denote the full negative log-likelihood of the model, and $\mathcal{J}_k$ the negative log-likelihood for each center, i.e.,

$$\mathcal{J}_k(\boldsymbol{\theta}) = \sum_{i=1}^{n_k} \left\{ a_{i,k} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_{i,k}) + (1 - a_{i,k}) \log(1 - p_{\boldsymbol{\theta}}(\boldsymbol{x}_{i,k})) \right\}. \tag{18}$$

Due to the separability of each loss term in per-sample terms[88], we have

$$\mathcal{J}(\boldsymbol{\theta}) = \sum_{k=1}^{K} \mathcal{J}_k(\boldsymbol{\theta}). \tag{19}$$

Using the separability (19), it is straightforward to optimize $\mathcal{J}$ using a second-order method, since its gradient and Hessian can be computed from the sum of local quantities, see "FedECA, a federated ECA method" of[89]. We call this naïve strategy FEDNEWTONRAPHSON: its pseudocode is provided in Algorithm 1. This algorithm has a hyperparameter corresponding to the number of steps: in our numerical experiments, we noted that $E = 10$ is sufficient to obtain proper convergence.

The strategy FEDNEWTONRAPHSON requires to compute full batch gradients and Hessians, in time $O(n_k)$ on each center, and each communication with the aggregator requires the exchange of $O(p^2)$ floating numbers. In the setting of ECAs, we usually have both $n_k \leq 10^3$ and $p \leq 10^3$, making such a second-order approach tractable. We note that for larger data settings, several improvements could be considered following[89,90], which would reduce the quantities of transmitted parameters. We leave such improvements to future work.

Algorithm 1. FedNewtonRaphson

```
Initialize $\boldsymbol{\theta}_0 = 0$
for $e = 1$ to $E$ do
    Aggregator sends $\boldsymbol{\theta}_{e-1}$ to each center
    for $k = 1$ to $K$ in parallel do                    ▷ On each center
        $\boldsymbol{g}_{e,k} = \nabla_{\boldsymbol{\theta}} J_k(\theta_{e-1})$
        $\boldsymbol{H}_{e,k} = \nabla^2_{\boldsymbol{\theta}} J_k(\theta_{e-1})$
        Send $\boldsymbol{g}_{e,k}$ and $\boldsymbol{H}_{e,k}$ to the aggregator
    end for
    $\boldsymbol{g}_e = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{g}_{e,k}$                    ▷ Aggregator-side
    $\boldsymbol{H}_e = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{H}_{e,k}$
    $\boldsymbol{\theta}_e = \boldsymbol{\theta}_{e-1} - (\boldsymbol{H}_e)^{-1}\boldsymbol{g}_e$
end for
return $\boldsymbol{\theta}_E$
```

Inverse probability weighted WebDISCO. Here, we propose a method to minimize the regularized weighted CoxPH model (10) in a federated fashion. Since the non-separability of the weighted CoxPH log-likelihood $\ell(\boldsymbol{\beta})$ prevents the use of vanilla FL algorithms, we inspire ourselves from WebDISCO[78] to build a pooled-equivalent second-order method. It should be noted, however, that the method can only be applied to partial likelihood under the Breslow's approximation for tied times (8), as opposed to the Efron's approximation.

Non-separability

Compared to the logistic propensity score model, the main difficulty of federating Eq. (9) stems from the non-separability of the log-likelihood, i.e., the cross-center terms. Indeed, for any time s, the risk set $\mathcal{R}_s$ is a union of per-center terms, i.e.

$$\mathcal{R}_s = \cup_{k=1}^{K} \mathcal{R}_{s,k}. \tag{20}$$

Thus, the aggregated Eq. (9) can be rewritten as

$$\ell(\boldsymbol{\beta}) = -\sum_{k=1}^{K} \sum_{s\in\mathring{\mathcal{S}}} \sum_{i\in\mathcal{D}_{s,k}} \left\{ w_i \boldsymbol{\beta}^T \boldsymbol{z}_{i,k} - w_i \log\left(\sum_{j\in\mathcal{R}_{s,k}} w_j e^{\boldsymbol{\beta}^T \boldsymbol{z}_{j,k}} + \sum_{k'\neq k} \sum_{j\in\mathcal{R}_{s,k'}} w_j e^{\boldsymbol{\beta}^T \boldsymbol{z}_{j,k'}} \right) \right\}, \tag{21}$$

where the loss for each sample i of each center k involves terms from other samples j in other centers $k'\neq k$. The non-separability of the CoxPH loss is a well-known issue in a federated setting, and previous works have investigated reformulations to make it amenable to vanilla federated learning solvers[79]. Here, we instead adapt the WebDISCO method[78] to the weighted case in order to keep pooled-equivalent results and benefit from second-order acceleration.

Federated computation of $\nabla_{\boldsymbol{\beta}}\ell(\boldsymbol{\beta})$ and $\nabla^2_{\boldsymbol{\beta}}\ell(\boldsymbol{\beta})$

Our method consists in performing an iterative server-level Newton-Raphson descent on $\mathcal{L}$. The gradient $\nabla_{\boldsymbol{\beta}}\ell(\boldsymbol{\beta})$ and Hessian $\nabla^2_{\boldsymbol{\beta}}\ell(\boldsymbol{\beta})$ thus need to be computed in a federated fashion. These quantities can be computed in closed form as

$$\nabla_{\boldsymbol{\beta}}\ell(\boldsymbol{\beta}) = -\sum_{s\in\mathring{\mathcal{S}}} \sum_{i\in\mathcal{D}_s} \left(w_i \boldsymbol{z}_i - w_i \frac{\sum_{j\in\mathcal{R}_s} w_j e^{\boldsymbol{\beta}^T \boldsymbol{z}_j} \boldsymbol{z}_j}{\sum_{j\in\mathcal{R}_s} w_j e^{\boldsymbol{\beta}^T \boldsymbol{z}_j}} \right), \tag{22}$$

and

$$\nabla^2_{\boldsymbol{\beta}}\ell(\boldsymbol{\beta}) = \sum_{s\in\mathring{\mathcal{S}}} \sum_{i\in\mathcal{D}_s} w_i \left\{ \frac{\sum_{j\in\mathcal{R}_s} w_j e^{\boldsymbol{\beta}^T \boldsymbol{z}_j} \boldsymbol{z}_j \boldsymbol{z}_j^T}{\sum_{j\in\mathcal{R}_s} w_j e^{\boldsymbol{\beta}^T \boldsymbol{z}_j}} - \frac{\left(\sum_{j\in\mathcal{R}_s} w_j e^{\boldsymbol{\beta}^T \boldsymbol{z}_j} \boldsymbol{z}_j\right)\left(\sum_{j'\in\mathcal{R}_s} w_{j'} e^{\boldsymbol{\beta}^T \boldsymbol{z}_{j'}} \boldsymbol{z}_{j'}^T\right)}{\left(\sum_{j\in\mathcal{R}_s} w_j e^{\boldsymbol{\beta}^T \boldsymbol{z}_j}\right)^2} \right\}. \tag{23}$$

Note that the Hessian evaluated at $\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}$, $\nabla^2_{\boldsymbol{\beta}}\ell(\hat{\boldsymbol{\beta}})$, corresponds, up to a sign, to the quantity $\boldsymbol{H}$ defined in (15) for the robust variance estimator. We now define the local counterparts $\zeta^h_{s,k}(\boldsymbol{\beta})$ of the previously introduced quantities, where the sum is restricted to the risk set $\mathcal{R}_{s,k}$,

$$\zeta^0_{s,k}(\boldsymbol{\beta}) = \sum_{j\in\mathcal{R}_{s,k}} w_j e^{\boldsymbol{\beta}^T \boldsymbol{z}_j}, \tag{24}$$

$$\boldsymbol{\zeta}^1_{s,k}(\boldsymbol{\beta}) = \sum_{j\in\mathcal{R}_{s,k}} w_j e^{\boldsymbol{\beta}^T \boldsymbol{z}_j} \boldsymbol{z}_j, \tag{25}$$

$$\boldsymbol{\zeta}^2_{s,k}(\boldsymbol{\beta}) = \sum_{j\in\mathcal{R}_{s,k}} w_j e^{\boldsymbol{\beta}^T \boldsymbol{z}_j} \boldsymbol{z}_j \boldsymbol{z}_j^T. \tag{26}$$

Further, let us denote

$$W_s = \sum_{i\in\mathcal{D}_s} w_i, \tag{27}$$

$$\boldsymbol{Z}_s = \sum_{i\in\mathcal{D}_s} w_i \boldsymbol{z}_i, \tag{28}$$

and

$$W_{s,k} = \sum_{i\in\mathcal{D}_{s,k}} w_i, \tag{29}$$

$$\boldsymbol{Z}_{s,k} = \sum_{i\in\mathcal{D}_{s,k}} w_i \boldsymbol{z}_i, \tag{30}$$

where by convention, in all cases, the sum is set to 0 in case of an empty set. Eqs. (22) and (23) can be respectively rewritten as

$$\nabla_{\boldsymbol{\beta}}\ell(\boldsymbol{\beta}) = -\sum_{s\in\mathring{\mathcal{S}}} \boldsymbol{Z}_s - W_s \frac{\boldsymbol{\zeta}^1_s(\boldsymbol{\beta})}{\zeta^0_s(\boldsymbol{\beta})}, \tag{31}$$

$$\nabla^2_{\boldsymbol{\beta}}\ell(\boldsymbol{\beta}) = \sum_{s\in\mathring{\mathcal{S}}} W_s \left\{ \frac{\boldsymbol{\zeta}^2_s(\boldsymbol{\beta})}{\zeta^0_s(\boldsymbol{\beta})} - \frac{\boldsymbol{\zeta}^1_s(\boldsymbol{\beta})\boldsymbol{\zeta}^1_s(\boldsymbol{\beta})^T}{\zeta^0_s(\boldsymbol{\beta})^2} \right\}. \tag{32}$$

Using these equations, we can rewrite

$$\nabla_{\boldsymbol{\beta}}\ell(\boldsymbol{\beta}) = -\sum_{k=1}^{K} \left\{ \sum_{s\in\mathring{\mathcal{S}}} \boldsymbol{Z}_{s,k} - W_{s,k} \frac{\sum_{k'} \boldsymbol{\zeta}^1_{s,k'}(\boldsymbol{\beta})}{\sum_{k'} \zeta^0_{s,k'}(\boldsymbol{\beta})} \right\}, \tag{33}$$

$$\nabla^2_{\boldsymbol{\beta}}\ell(\boldsymbol{\beta}) = \sum_{k=1}^{K} \sum_{s\in\mathring{\mathcal{S}}} W_{s,k} \left\{ \frac{\sum_{k'} \boldsymbol{\zeta}^2_{s,k'}(\boldsymbol{\beta})}{\sum_{k'} \zeta^0_{s,k'}(\boldsymbol{\beta})} - \frac{\left(\sum_{k'} \boldsymbol{\zeta}^1_{s,k'}(\boldsymbol{\beta})\right)\left(\sum_{k'} \boldsymbol{\zeta}^1_{s,k'}(\boldsymbol{\beta})\right)^T}{\left(\sum_{k'} \zeta^0_{s,k'}(\boldsymbol{\beta})\right)^2} \right\}. \tag{34}$$

Assuming the set of all true event times $\mathring{\mathcal{S}}$ is known to all centers, we see that it is possible to reconstruct the full gradient $\nabla_{\boldsymbol{\beta}}\ell(\boldsymbol{\beta})$ and Hessian $\nabla^2_{\boldsymbol{\beta}}\ell(\boldsymbol{\beta})$ based on the 5-uplet $\{(W_{s,k}, \boldsymbol{Z}_{s,k}, \zeta^0_{s,k}(\boldsymbol{\beta}), \boldsymbol{\zeta}^1_{s,k}(\boldsymbol{\beta}), \boldsymbol{\zeta}^2_{s,k}(\boldsymbol{\beta}))\}_{s,k}$. Algorithm 2 sums up this algorithm.

Algorithm 2. FedCoxComp

```
Require: Weights $\beta$, set $\mathring{\mathcal{S}}$
Aggregator sends $\beta$ to each center
for $k = 1$ to $K$ in parallel do                          ▷ On each center
    for $s \in \mathring{\mathcal{S}}$ do
        Compute $W_{k,s}$ with (29)                        ▷ 0 if $\mathcal{D}_{s,k} = \emptyset$
        Compute $Z_{k,s}$ with (30).
    end for
    for $s \in \mathring{\mathcal{S}}$ s.t. $W_{k,s} > 0$ do                  ▷ 0 otherwise
        Compute $\zeta^0_{s,k}(\beta)$ with (24)
        Compute $\zeta^1_{s,k}(\beta)$ with (25)
        Compute $\zeta^2_{s,k}(\beta)$ with (26)
    end for
    Send back $\{(W_k, Z_k, \zeta^0_{s,k}(\beta), \zeta^1_{s,k}(\beta), \zeta^2_{s,k}(\beta))\}_{s\in\mathring{\mathcal{S}}}$
end for
Compute $\nabla_\beta \ell(\beta)$ with (33)                              ▷ On the server
Compute $\nabla^2_\beta \ell(\beta)$ with (34)
return $\nabla_\beta \ell(\beta)$, $\nabla^2_\beta \ell(\beta)$
```

Algorithm 3. non-robust FedECA

```
Require: Maximal number of steps $E$, LR schedule $(\alpha_e)_e$, regularization $\gamma$
Initialization $\boldsymbol\beta_0 = 0$
for $e = 1$ to $E$ do
    $\nabla_{\boldsymbol\beta}\ell(\boldsymbol\beta_{e-1}), \nabla^2_{\boldsymbol\beta}\ell(\boldsymbol\beta_{e-1}) = \mathrm{FedCoxComp}(\boldsymbol\beta_{e-1})$    ▷ Communication between server and centers
    $\nabla_{\boldsymbol\beta}\mathcal{L}(\boldsymbol\beta_{e-1}) = \nabla_{\boldsymbol\beta}\ell(\boldsymbol\beta_{e-1}) + \gamma\nabla_{\boldsymbol\beta}\psi(\boldsymbol\beta)$
    $\nabla^2_{\boldsymbol\beta}\mathcal{L}(\boldsymbol\beta_{e-1}) = \nabla^2_{\boldsymbol\beta}\ell(\boldsymbol\beta_{e-1}) + \gamma\nabla^2_{\boldsymbol\beta}\psi(\boldsymbol\beta)$
    $\boldsymbol\beta_e = \boldsymbol\beta_{e-1} - \alpha_e\left(\nabla^2_{\boldsymbol\beta}\mathcal{L}(\boldsymbol\beta_{e-1})\right)^{-1}\nabla_{\boldsymbol\beta}\mathcal{L}(\boldsymbol\beta_{e-1})$
    if Stopping criterion then $e = E$
    end if
end for
return $\boldsymbol\beta_E$
```

Non-robust FedECA

To optimize the full loss (10), we can now leverage the computation of the gradient and Hessian of the weighted CoxPH loss ℓ to perform a second-order Newton-Raphson descent. We follow the hyperparameters of `lifelines`[91] for this optimization. In particular, we use the same learning rate strategy, the same regularizer and the same stopping criterion. Indeed, as `lifeline`'s regularizer does not depend on data and is smooth, its gradient and Hessian can be computed on the server's side, deriving twice the following equation:

$$\mathcal{L}(\boldsymbol\beta) = \ell(\boldsymbol\beta) + \gamma\psi(\boldsymbol\beta), \tag{35}$$

with γ the strength of the regularization.

In more details for the regularizer $\psi(\boldsymbol\beta)$, we use a soft elastic-net regularization[92] with hyperparameters $\lambda > 0$ and $\alpha > 0$:

$$\psi(\boldsymbol\beta) = \lambda\left(\sum_r \phi_\alpha(\boldsymbol\beta_r)\right) + \frac{1-\lambda}{2}\parallel \boldsymbol\beta\parallel_2^2, \tag{36}$$

where ϕ_α is a smooth approximation of the absolute value that is progressively sharpened with the round e.

$$\alpha = 1.3^e, \tag{37}$$

$$\phi_\alpha(x) = \frac{1}{\alpha}(\log(1+\exp(\alpha x)) + \log(1+\exp(-\alpha x))). \tag{38}$$

We also allow for constant learning rates as in `scikit-survival`[93]. We note that implementing different learning rate strategies or regularizers should be straightforward with our implementation. Algorithm 3 summarizes the full algorithm used.

We note that in practice, due to the linearity of both models and as covariates are often low-dimensional in clinical trials, tuning hyperparameters such as learning rates and regularizations parameters is superfluous. In fact, we use the default settings without regularization in all of our experiments unless explicitly stated. We also note that regularization in linear models for treatment effect estimation must be imposed carefully to avoid the over-shrinking effect described in[94]. We still give practitioners ways to tune such hyper-parameters, as it is already the case in non-distributed softwares, in order not to loose flexibility. This allows to accomodate potential future workflows such as using deep-learning based covariates, which might be high-dimensional[95] and thus optimizing the Cox loss might, in this case, require the use of ridge regularization to keep the hessian from being ill-conditioned. Regarding federated hyper-parameter tuning in general, see "Federated hyper-parameters selection".

Statistical test. Federated robust variance estimation

The robust variance estimator can be obtained by aggregating local quantities, as we demonstrate in the following. We assume that each client has access to $\zeta^0_s(\hat{\boldsymbol\beta})$ and $\boldsymbol\zeta^{\boldsymbol 1}_{\boldsymbol s}(\hat{\boldsymbol\beta})$ for all $s \in \mathring{\mathcal{S}}$. This can be achieved by simply allowing the server to transmit the quantities $\zeta^0_{s,k}(\hat{\boldsymbol\beta})$ and $\boldsymbol\zeta^{\boldsymbol 1}_{\boldsymbol{s,k}}(\hat{\boldsymbol\beta})$ to the centers in addition to $\boldsymbol H$.

The global goal is to compute the robust estimator of the variance given by

$$\widehat{\boldsymbol{Var}}(\hat{\boldsymbol\beta}) = \boldsymbol H^{-1}\boldsymbol Q(\boldsymbol H^{-1})^T, \tag{39}$$

where $\boldsymbol H$ (15) corresponds to the Hessian $\nabla^2_{\boldsymbol\beta}\ell(\hat{\boldsymbol\beta})$ and $\boldsymbol Q$ is defined in (16). We note that through FedECA 3 each client already has access to $\boldsymbol H$.

Let us define $\boldsymbol M_k$ as

$$\boldsymbol M_k = \sum_{i=1}^{n_k}(\boldsymbol H^{-1}\hat{\boldsymbol\varphi}_i(\hat{\boldsymbol\beta}))\hat{\boldsymbol\varphi}_i(\hat{\boldsymbol\beta})^T(\boldsymbol H^{-1})^T, \tag{40}$$

where the sum is on all indices belonging to client k.

Then we have,

$$\widehat{\boldsymbol{Var}}(\hat{\boldsymbol\beta}) = \sum_{k=1}^{K}\boldsymbol M_k. \tag{41}$$

Indeed, if we let $\hat{\boldsymbol\Phi}(\hat\beta) \in \mathbb{R}^{n,p}$ be the matrix whose rows are the $\hat{\boldsymbol\varphi}_i(\hat{\boldsymbol\beta})$ for all $i \in [[1, n]]$. then we can write the variance as

$$\widehat{\boldsymbol{Var}}(\hat{\boldsymbol\beta}) = \boldsymbol H^{-1}\hat{\boldsymbol\Phi}(\hat\beta)^T\hat{\boldsymbol\Phi}(\hat\beta)(\boldsymbol H^{-1})^T, \tag{42}$$

$$\begin{aligned}\hat{\boldsymbol\Phi}(\hat\beta)^T\hat{\boldsymbol\Phi}(\hat\beta)_{i,j} &= \sum_{k=1}^{n}\left(\hat{\boldsymbol\varphi}_k(\hat{\boldsymbol\beta})\right)_i\cdot\left(\hat{\boldsymbol\varphi}_k(\hat{\boldsymbol\beta})\right)_j \\ &= \sum_{k=1}^{K}\sum_{m=1}^{n_k}\left(\hat{\boldsymbol\varphi}_m(\hat{\boldsymbol\beta})\right)_i\cdot\left(\hat{\boldsymbol\varphi}_m(\hat{\boldsymbol\beta})\right)_j,\end{aligned} \tag{43}$$

$$\widehat{\boldsymbol{Var}}(\hat{\boldsymbol\beta}) = \boldsymbol H^{-1}\hat{\boldsymbol\Phi}(\hat{\boldsymbol\beta})^T\hat{\boldsymbol\Phi}(\hat{\boldsymbol\beta})(H^{-1})^T = \sum_{k=1}^{K}\boldsymbol M_k. \tag{44}$$

Each client can compute $\hat{\boldsymbol\varphi}_{\boldsymbol i}(\hat{\boldsymbol\beta})$ with Eq. (17) for all its samples i ($\forall s, i \in \mathcal{D}_{s,k}$) as long as it has access to $\zeta^0_{s,k}(\hat{\boldsymbol\beta})$ and $\boldsymbol\zeta^{\boldsymbol 1}_{\boldsymbol{s,k}}(\hat{\boldsymbol\beta})$ for all $s \in \mathring{\mathcal{S}}$. Therefore, each client can compute the corresponding $\boldsymbol M_{\boldsymbol k}$.

This leads us to the full robust algorithm of FedECA in 5. Once the variance is estimated using the above expression, we can perform inference using, e.g., a Z-test. Note that as in `lifelines`[91] we use the Hessian of the regularized function. Therefore, to accommodate the computation of the variance, we modify non-robust FedECA as depicted in Alg. 5. Privacy-wise, this modification (a) gives each client the same knowledge as the server on the last round and (b) communicates an additional $\boldsymbol M_k$ matrix by center, which is reasonable. In addition, in the IPTW case, the matrix only the treatment allocation is used as a covariate and hence $\boldsymbol M_k$ is a scalar M_k.

Algorithm 4. RobustFedCoxComp

```
Require: Weights $\boldsymbol\beta$, set $\mathring{\mathcal{S}}$
Aggregator sends $\boldsymbol\beta$ to each center
for $k = 1$ to $K$ in parallel do                                    ▷ On each center
    for $s \in \mathring{\mathcal{S}}$ do
        Compute $W_{k,s}$ with (29)                                 ▷ 0 if $\mathcal{D}_{s,k} = \emptyset$
        Compute $\boldsymbol Z_{k,s}$ with (30).
    end for
    for $s \in \mathring{\mathcal{S}}$ s.t. $W_{k,s} > 0$ do                               ▷ 0 otherwise
        Compute $\zeta^0_{s,k}(\boldsymbol\beta)$ with (24)
        Compute $\boldsymbol\zeta^1_{s,k}(\boldsymbol\beta)$ with (25)
        Compute $\boldsymbol\zeta^2_{s,k}(\boldsymbol\beta)$ with (26)
    end for
    Send back $\{(W_k, \boldsymbol Z_k, \zeta^0_{s,k}(\boldsymbol\beta), \boldsymbol\zeta^1_{s,k}(\boldsymbol\beta), \boldsymbol\zeta^2_{s,k}(\boldsymbol\beta))\}_{s\in\mathring{\mathcal{S}}}$
end for
Compute $\nabla_{\boldsymbol\beta}\ell(\boldsymbol\beta)$ with (33)                                ▷ On the server
Compute $\nabla^2_{\boldsymbol\beta}\ell(\boldsymbol\beta)$ with (34)
return $\nabla_{\boldsymbol\beta}\ell(\boldsymbol\beta), \nabla^2_{\boldsymbol\beta}\ell(\boldsymbol\beta)$      ▷ And if it's the last round return $\forall s \in \mathring{\mathcal{S}}, \zeta^0_{s,k}(\boldsymbol\beta), \boldsymbol\zeta^1_{s,k}(\boldsymbol\beta), W_s$
```

Algorithm 5. FedECA

```
Require: Maximal number of steps $E$, LR schedule $(\alpha_e)_e$, regularization $\gamma$
Initialization $\beta_0 = 0$
for $e = 1$ to $E$ do
    $\nabla_{\boldsymbol\beta}\ell(\boldsymbol\beta_{e-1}), \nabla^2_{\boldsymbol\beta}\ell(\boldsymbol\beta_{e-1}) = \mathrm{RobustFedCoxComp}(\beta_{e-1})$    ▷ Communication between server and centers
    $\nabla_{\boldsymbol\beta}\mathcal{L}(\boldsymbol\beta_{e-1}) = \nabla_{\boldsymbol\beta}\ell(\boldsymbol\beta_{e-1}) + \gamma\nabla_{\boldsymbol\beta}\psi(\boldsymbol\beta)$
    $\nabla^2_{\boldsymbol\beta}\mathcal{L}(\boldsymbol\beta_{e-1}) = \nabla^2_{\boldsymbol\beta}\ell(\boldsymbol\beta_{e-1}) + \gamma\nabla^2_{\boldsymbol\beta}\psi(\boldsymbol\beta)$
    $\boldsymbol\beta_e = \boldsymbol\beta_{e-1} - \alpha_e \left(\nabla^2_{\boldsymbol\beta}\mathcal{L}(\boldsymbol\beta_{e-1})\right)^{-1} \nabla_{\boldsymbol\beta}\mathcal{L}(\boldsymbol\beta_{e-1})$
    if Stopping criterion then $e = E$
    end if
end for
return $\boldsymbol\beta_E$
Define $\hat{\boldsymbol\beta} = \boldsymbol\beta_E$
for $k = 1$ to $K$ in parallel do    ▷ On each center
    Send back $\boldsymbol{M}_k\boldsymbol{M}_k^T$ where $\boldsymbol{M}_k = \sum_{s\in\mathring{\mathcal{S}}}\sum_{i\in\mathcal{D}_{s,k}} \boldsymbol{H}^{-1}\hat{\boldsymbol\varphi}_i(\hat{\boldsymbol\beta})$.
end for
Compute $\widehat{\boldsymbol{Var}}(\hat{\boldsymbol\beta}) = \sum_{k=1}^K \boldsymbol{M}_k\boldsymbol{M}_k^T$    ▷ On the server
return $\widehat{\boldsymbol{Var}}(\hat{\boldsymbol\beta})$
```

Federated bootstrap estimation

When implementing bootstrap in federated setups, the most straightforward implementation is to bootstrap samples per-center. However this creates some edge cases if centers have small sample sizes (e.g., $\exists\, k|n_k = 1$) and risks underestimating the variance compared to the pooled case. In order to circumvent this issue we label all samples from 1 to n and label clients' samples irrespective of their order. This requires only sharing the number of samples by clients and nothing else. In practice we label the centers from 1 to K randomly and then assign the number from 1 to n_1 to the first client's samples and so forth. As each client knows its indices we can then ask the server to sample indices from 1 to n as we would in the pooled case and then send the current set of indices chosen for each bootstrap to all clients that would then use it to bootstrap its own cohort. This way, there is no bias in sampling, even for arbitrarily small clients. We refer to this alternative as global bootstrap.

We therefore implement both, but use in our experiments this global bootstrap where we sample with replacement the global distributed cohort as if it were pooled.

Distributed computation with Substra introduces an overhead per atomic task executed locally on each partner's machine mainly due to docker image building. This overhead is not negligible and can become a bottleneck when performing bootstrapping, which naïvely necessitate to execute $O(n_{rounds}{*}n_{bootstraps})$ tasks per-client. To alleviate this issue we implement a more efficient bootstrapping strategy where we only have to run $O(n_{rounds})$ tasks. This is achieved by employing hooks so that each task, instead of executing its normal code, bootstraps itself and then executes all its bootstraped versions producing a list of bootstraped results per task. This requires to also modify the aggregation steps to be able to aggregate each bootstrap run independently and then redispatch all bootstraped aggregations to each client. Details of this non-trivial implementation trick can be found in the bootstraper.py script.

Note that we could also add another layer of parallelization inside each task by using Python multi-processing as each bootstrap run is independent of each other. However, the impact of this optimization would be negligible with respect to the overhead introduced by the distributed constraints. With this optimization, a Substra experiment with 200 bootstraps lasts less than an hour instead of ≈ 200 hours naïvely without this parallelization layer, which would have made the bootstrap variance estimation impractical, irrespective of the size of the federated network. Note that other kinds of parallelization schemes could also be undertaken, such as running multiple training jobs (so-called "Compute Plans" in Substra) in parallel as was done in MELLODDY[23]. However, this option necessitates scaling servers' computational resources (CPUs, RAM) linearly with the number of training jobs in parallel, which is impractical.

Privacy of FedECA. We consider that time-to-event and censorship are safe to share, this is a strong assumption but is often used in clinical trials as KM curves are released[96]. More generally, every federated computational graph involved in FedECA and created by our implementation of the above algorithms as well the ones underpinning the FA methods of "Federated analytics for end-to-end federated ECA analysis" can be audited easily thanks to Supplementary Fig. 10 and 12–14. Each individual variable name in those graphs can be understood thanks to the associated tables, Supplementary Tables 11, 12, 10, 14 and 15. Details on how this tracing step is performed are available in 4.8.3.

Regarding the security of the covariates, we place ourselves in the "honest-but-curious" threat model, described in more detail in Substra's documentation[97].

The only covariate used when doing IPTW is the treatment allocation, which is known throughout centers. Therefore, the only quantities tied to the covariates that are communicated are (1) the gradients of the propensity model, and (2) the scalar product of covariates and propensity model weights that are exposed through the propensity scores, averaged on risk sets and on distinct event times. Regarding the first point, we propose an implementation of a differentially private version of the propensity model training that we describe in the next paragraph. Regarding the second point, we assume that the dimension p of the covariate vector is such that $p > > 1$ and therefore that leaking scalar products is an acceptable risk in this context; This is a strong assumption. In the general case it could theoretically allow for attacks such as membership attacks[98]. Making the pipeline end-to-end differential private (DP) is an open problem. One could in principle rely again on DP to either add noise to the scalar products themselves or to the propensity scores when training the Cox PH model. However, this would affect the result even more than when applying DP only to the propensity model training, which already has a strong effect see Supplementary Fig. 3. Another research avenue would be to increase the average/minimum size of the per-client risk sets by discretizing the times and applying random quantization mechanisms (RQM)[99]. We note that in this second case another downside would be that, in addition to destabilizing the training of the Cox model, it would artificially create more ties in the data, which would in return affect the quality of the Breslow estimator.

Because it exposes sums of additional covariates, studying the privacy of the federation of adjusted IPTW requires a specific treatment that we leave to future work.

Differential privacy of the propensity model

We list here some properties of DP that are relevant to our implementation and refer the reader to the work of[100] or[101] for a more complete exposition of the topic:

DP provides slack parameters (ϵ, δ), which allow to strike a trade-off between model accuracy and privacy of individual contributions.

A process $\mathcal{M}$ is (ϵ, δ)-DP if and only if $\forall D, D\prime$ adjacent (differing by one element), we have:

$$p(\mathcal{M}(D) \in S) \leq p(\mathcal{M}(D') \in S) \cdot \exp(\epsilon) + \delta \tag{45}$$

Perfect privacy guarantees are only obtained by taking $(\epsilon, \delta) = (0, 0)$ which makes the process $\mathcal{M}$ provably indistinguishable from the addition or removal of one individual. In practice in real-world deployments it seems ϵ between 0.1 and 50 are used depending on the application[101] with different values of δ. DP benefits from nice composability properties[100] and can thus be applied easily to ML training methods that are iterative by nature and can therefore be applied to FL as well[102].

We use the Opacus library[103], which implements the privacy accountant method of[102] to train the propensity model within FedECA with differential privacy (DP) with various (ϵ, δ) couples.

Our implementation is available in the script torch_dp_fed_avg_algo.py and uses Rényi differential privacy (RDP)[104], which gives tighter bounds alongside with Poisson sampling.

Federated analytics for end-to-end federated ECA analysis

Regulators ask for SMD and Kaplan-Meier survival curves[105] in addition to the hazard ratio (HR), the associated confidence intervals (CI) and *P*-value in order to validate the reweighting of the propensity model and to be able to describe the patient population's time-to-events distribution within the two groups.

Therefore, we also implement two federated analytics methods that we call Fed-Kaplan and Fed-SMD in order to compute such quantities globally on distributed data without compromising data.

It is to be noted that Fed-Kaplan can be directly derived from FedECA as it requires the same quantities, namely the risk sets and number of occurrence of events. However, Fed-SMD requires the communication of additional second order terms which increases the attack surface of FedECA.

Federated Kaplan-Meier estimator. We follow FedECA implementation to compute per-center and communicate the unique times of events $\mathcal{S}_k$, the (weighted) risk set $\mathcal{R}_{s,k}$ and the (weighted) number of deaths occurring at these times $\mathcal{D}_{s,k}$ for each arm.

This enables to compute in the server $\mathcal{R}_s$ and $\mathcal{D}_s$ which then allows to compute the Kaplan-Meier estimator at each time t of a predefined grid for each arm as well as the Greenwood and exponential Greenwood confidence intervals[106].

For completeness, we remind the reader of these well-known formulas that we rewrite using our notations:

$$
\begin{aligned}
\hat{S}(t) &= \prod_{s\in\mathring{\mathcal{S}}|s\le t}\left(1-\frac{\sum_{j\in\mathcal{D}_s} w_j}{\sum_{k\in\mathcal{R}_s} w_k}\right),\\
\mathrm{Var}(\hat{S}) &= \hat{S}(t)^2 \prod_{s\in\mathring{\mathcal{S}}|s\le t}\frac{\sum_{j\in\mathcal{D}_s} w_j}{\left(\sum_{k\in\mathcal{R}_s} w_k\right)\times\left(\sum_{k\in\mathcal{R}_s} w_k-\sum_{j\in\mathcal{D}_s} w_j\right)},\\
Z(t) &= \log(\log\hat{S}(t)),\\
\mathrm{Var}[\hat{Z}(t)] &= \frac{1}{(\log\hat{S}(t))^2}\prod_{s\in\mathring{\mathcal{S}}|s\le t}\frac{\sum_{j\in\mathcal{D}_s} w_j}{\left(\sum_{k\in\mathcal{R}_s} w_k\right)\times\left(\sum_{k\in\mathcal{R}_s} w_k-\sum_{j\in\mathcal{D}_s} w_j\right)}.
\end{aligned}
$$

With $\hat{S}(t)$ the Kaplan-Meier estimator of the survival function and $\mathrm{Var}(\hat{S})$ and $\mathrm{Var}[\hat{Z}(t)]$ respectively the Greenwood and exponential Greenwood estimators of the variance of the Kaplan-Meier estimator at time t. In practice exponential Greenwood shall be used[106] and this is what we display in the results.

SMD estimator. Computing SMD in a federated setting

We compute the standardized mean difference (SMD) for each covariate before and after weighting as defined by:

$$SMD = \frac{\bar{x}_1-\bar{x}_2}{\sqrt{\frac{s_1^2+s_2^2}{2}}}. \tag{46}$$

Where $\bar{x}_1$ and $\bar{x}_2$ are the means of the covariate in the two arms, and s_1^2 and s_2^2 are the variances of the covariate in the two arms. As explained in[107,108] we use the variance of the groups before weighting as a normalizer. We compute this quantity in a federated fashion efficiently in two aggregation rounds by developing the variance following[109]:

$$\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2 = \frac{1}{n-1}\left(\sum_{i=1}^{n}x_i^2-n\bar{x}^2\right). \tag{47}$$

Effectively each center transmits uncentered moments of order 1 and 2 and the server uses them to derive the centered moments of order 1 and 2.

Datasets and cohorts construction

Synthetic data generating model of time-to-event outcome. To illustrate the performance of our proposed FL implementation, we rely on simulations with synthetic data. We simulate covariates and related time-to-event outcomes respecting the proportional hazards (PH) assumption, with the baseline hazard function derived from a Weibull distribution. For simplicity, we assume a constant treatment effect across the population. The data generation process consists of several consecutive steps that we describe below, assuming our target is a dataset with p covariates and n samples.

First, a design matrix $\boldsymbol{X}=[\boldsymbol{X}^{(1)},\dots,\boldsymbol{X}^{(p)}]\in\mathbb{R}^{n\times p}\sim\mathcal{N}(0,\boldsymbol{\Sigma})$ is drawn from a multivariate normal distribution to obtain (baseline) observations for n individuals described by p covariates. The covariance matrix $\boldsymbol{\Sigma}$ is taken to be a Toeplitz matrix such that the covariances between pairs ($\boldsymbol{X}^{(i)}$, $\boldsymbol{X}^{(j)}$) of covariates decay geometrically. In other words, for a fixed $\rho > 0$, we have $\mathrm{cov}(\boldsymbol{X}^{(i)}, \boldsymbol{X}^{(j)}) = \rho^{|i-j|}$. Such a covariance matrix implies a locally and hierarchically grouped structure underlying the covariates, which we choose to mimic the potentially complex structure of real-world data. To reflect the varying correlations of the covariates with the outcome of interest, the coefficients $\boldsymbol{\beta}_i$ of the linear combination used to build the hazard ratio are drawn from a standard normal distribution.

$$
\begin{aligned}
\boldsymbol{\Sigma} &= \text{Toeplitz}\,(1,\rho,\rho^2,\cdots,\rho^{p-1}),\\
\boldsymbol{X} &\in \mathbb{R}^{n\times p}\sim\mathcal{N}(0,\boldsymbol{\Sigma}),\\
\boldsymbol{\beta} &\in \mathbb{R}^{p}\sim\mathcal{N}(0,1).
\end{aligned} \tag{48}
$$

In the context of clinical trials with external control arms, which implies non-randomized treatment allocation, we simulate the treatment allocation in such a way that it depends on the covariates. More precisely, we introduce the treatment allocation variable A that follows a Bernoulli distribution, where the probability of being treated (the propensity score) q depends on a linear combination of the covariates, connected by a logit link function g. The coefficients $\boldsymbol{\alpha}_i$ of the linear combination are drawn from a uniform distribution, where the range $k \ge 0$ is symmetric around 0 and is normalized by the number of covariates. The degree of influence of the covariates on A can be regulated by adjusting the value of k. The greater the value of k, the stronger the influence, and therefore the lower the degree of overlap between the distributions of propensity scores of the treated and (external) control groups. Conversely, $k = 0$ removes the dependence, leading to a randomized treatment allocation.

$$
\begin{aligned}
\boldsymbol{\alpha} &\in \mathbb{R}^{p}\sim p^{-1/2}U(-k,k),\\
q_i &= g^{-1}(\boldsymbol{\alpha}^T\boldsymbol{X}_i) = \left(1+e^{-\boldsymbol{\alpha}^T\boldsymbol{X}_i}\right)^{-1},\\
a_i|\boldsymbol{X}_i &\sim \text{Bern}\,(q_i).
\end{aligned} \tag{49}
$$

Once drawn, the treatment allocation variable A_i is composed with the constant treatment effect, defined here as the hazard ratio μ, to obtain the final hazard ratio h_i for each individual. The time-to-event T_i^* of each sample is then drawn from a Weibull distribution with shape v and the scale depending on h_i and v. Meanwhile, for all samples we assume a constant dropout (or censoring) rate d across time, resulting in a censoring time that follows an exponential distribution.

$$
\begin{aligned}
h_i(a_i) &= \mu^{a_i}\exp(\boldsymbol{\beta}^T\boldsymbol{X}_i),\\
T_i^* &\sim \mathcal{W}(h_i(a_i)^{-\frac{1}{\nu}},\nu),\\
C_i &\sim \mathcal{E}(d)
\end{aligned} \tag{50}
$$

Finally, the event indication variable δ_i can be derived from T_i^* and C_i: $\delta_i = \mathbb{1}_{T_i^*\le C_i}$. And the observed outcome Y_i for the ith individual is defined as the couple $Y_i = (T_i = \min(T_i^*, C_i), \delta_i)$, i.e., it corresponds to the observed time and the information on whether an event is observed.

Prostate cancer cohort construction. We access data of two phase III randomized clinical trials from the Yale University Open Data Access

(YODA) project[34,35] of patients with metastatic castration-resistant prostate cancer. The first trial, NCT02257736, has apalutamide, abiraterone acetate and prednisone (Apa-AA-P) for the treatment arm, and placebo, abiraterone acetate and prednisone (AA-P) for the control arm. The primary outcome is radiographic progression-free survival (rPFS). The second trial, NCT00887198, has AA-P for the treatment arm, and placebo and prednisone (P) for the control arm. The primary outcomes are overall survival (OS) and rPFS. We thus artificially distribute the data of the first trial to one "client" (simulated server) and the data of the second arm to the second client replicating natural splits such as in[82] to simulate a federated learning setup.

While all patients are randomized in each trial, it is still necessary to correct for potential confounding when comparing arms from different trials. First, the inclusion/exclusion criteria of both trials were aligned, patients in NCT02257736 with present visceral metastases at randomization were excluded to match the exclusion criterion of NCT00887198. Then a group of variables of patient's baseline characteristics were chosen for propensity-weighting based on literature review as well as on their availability in both trials. The chosen covariates are age, body-mass index (BMI), eastern cooperative oncology group (ECOG), brief pain inventory (BPI) score and bone-metastasis-only. We present baseline characteristics for each trial in Supplementary Tables 3, 4.

We then filter these patients to remove non-informative patients and patients with missing survival information. The final full cohort consists of $n = 1927$ patients ($n = 839$ for NCT02257736 and $n = 1088$ for NCT00887198) in three treatment arms (Apa-AA-P, AA-P and P). We infer the missing covariates on a per-center basis using MissForest[54]. The flow diagram of the cohort construction is present in Supplementary Fig. 5, including different ECA experiments conducted in this study.

We note that, since we submitted our research plan proposal to YODA (provided in Supplementary Fig. 9) in order to access the data, we departed from the original plan in the following ways:

- IPTW is studied instead of G-computation
- we do not study conformal prediction
- time-to-event endpoints are studied instead of change in SLD or change in PSA

Pancreatic adenocarcinoma cohort construction

Cohort construction. We access data from three different sources: the Fédération Francophone de Cancérologie Digestive (FFCD), the Institut d'Investigació Biomèdica de Girona (IDIBGI), and the Pancreatic Cancer Action Network (PanCAN). The FFCD data consists of a subset of two clinical trials: PRODIGE 35[36] and PRODIGE 37[37] that respectively compare the first line efficacy of, for PRODIGE 35, 6 months of FOLFIRINOX (arm A), 4 months of FOLFIRINOX followed by leucovorin plus fluorouracil maintenance treatment for controlled patients (arm B), and a sequential treatment alternating gemcitabine and fluorouracil, leucovorin, and irinotecan every 2 months (arm C) and for PRODIGE 37: alternately receive gemcitabine + nab-paclitaxel for 2 months then FOLFIRI.3 for 2 months (arm A), or gemcitabine + nab-paclitaxel alone until progression (arm B). We use both the FOLFIRINOX arm A from PRODIGE 35 ($n = 92$) and the gemcitabine + nab paclitaxel arm B from PRODIGE 37 with ($n = 61$). The inclusion criteria of this new subset is thus metastatic pancreatic adenocarcinoma patients with a performance status eastern cooperative oncology group (ECOG) of either 0, 1 or 2. We select patients with the same inclusion criteria treated with FOLFIRINOX or gemcitabine + nab-paclitaxel from clinical practice data from IDIBGI and PanCAN. In IDIBGI we find $n = 33$ patients treated with FOLFIRINOX and $n = 192$ with gemcitabine + nab-paclitaxel. In PanCAN we find $n = 91$ patients treated with FOLFIRINOX and $n = 86$ with gemcitabine + nab-paclitaxel patients that meet the criteria, totaling $n = 177$ patients out of 181 originally available, excluding ECOG 3 and 4. Among the 177 patients, 2 are censored at the time the study starts. Therefore, their data is not informative for the Cox model fitting but might still be useful for the estimation of the propensity model. In addition, we identify in the PanCAN cohort the presence of an immortal time bias due to biopsy collection, i.e., a patient will enter the PanCAN database only if they have had at least one biopsy before their last known follow-up. Consequently, patients who died before having any biopsy will not be included in the database, and patients in the database will be alive at least until their first biopsy. The time interval between the start of treatment and the first biopsy is therefore an immortal time for all patients in the PanCAN cohort. Such immortal time bias will lead to inflated survival rates[110] and is crucial to the present study. To correct for this bias, we thus retrieve for all PanCAN patients the date of their first biopsy and adjust their entry date into the study by taking the latest date between the first biopsy and the start of first-line treatment for metastatic pancreatic cancer.

For each patient we access the following covariates: age at diagnosis, ECOG performance status, biological gender determined by self-report and whether or not patients have liver metastasis following the literature[48] and restrictions due to data availability for covariates in each center. We present baseline characteristics for each of the centers in Supplementary Tables 5–7 respectively for FFCD, IDIBGI and PanCAN. We then filter these patients to remove non-informative patients, i.e., patients with missing treatment or survival information. The final full distributed cohort consists of $n = 555$ patients ($n = 153$ for FFCD, $n = 225$ for IDIBGI, and $n = 177$ for PanCAN). We infer the missing covariates on a per-center basis using MissForest[54] considering ECOG as a numerical variable because it is ordered, and apply minimum-maximum normalization to numerical variables using [0, 100] for age values and [0.0, 2.0] for ECOG loosely following[48].

Practical considerations associated with setting-up real-world federated learning collaborations. While clinical trials data is well-standardized, real-world data from centers from multiple continents are not and need to be harmonized for the federation to be considered. Clinical practice data has to be extracted by partners from different local sources stored in different databases and accessed by different internal toolings leading to a variety of extracted formats. We ask the centers to align on a common data dictionary created from FFCD data, which acts as the reference center as RCT data is already well-curated. We share this dictionary as a Google Sheet to all partners specifying expected variables, units formats and possible values. Resulting data extracts have missing values, and some data entries contain errors. Thus, while some parts can be automated, the whole process from data extraction to data harmonization involves a non-trivial amount of back-and-forth between data engineers from partner centers, medical doctors, data stewarts and data scientists in order to perform thorough quality checks of the input data. It is interesting to note that, while we did not use large language models (LLMs)[111] in this work, they could certainly be useful to streamline parts of this process[112], However, end-to-end automation seems out of reach with current technology[113].

Real-world experiments setup details

All experiments in this article are simulated in-RAM with the exception of two experiments: the first one which uses synthetic data and splits it into 10 cloud nodes and the pancreatic adenocarcinoma use-case.

We refer to those two experiments as real-world in order to distinguish the complexity of their deployment from in-RAM simulation cases.

In both cases, we use the Substra platform[30] to deploy the federated learning network over secure cloud-based infrastructures.

Substra is distributed with Helm charts for each component. The charts package all the files required for a deployment in a Kubernetes cluster. Provisioning of the clusters and Substra deployment are performed using a private Terraform module (known as infrastructure as-code). We detail below the two different deployments.

Synthetic data infrastructure setup. For this experiment, the clusters are hosted on Google Kubernetes engine (GKE) but Substra's deployment is cloud-agnostic. Provisioning of the GKE cluster and Substra deployment are performed using a private Terraform module (known as infrastructure as-code). For this experiment, we used 11 Kubernetes clusters:

- 1 cluster is hosting the Substra orchestrator - single source of truth within the federation - as well as a Substra Backend and Frontend, which makes it capable of receiving and performing aggregation tasks. Substra's documentation refers to this cluster as "AggregationNode".
- 10 clusters are hosting a Substra Backend (and Frontend) only and perform compute tasks on local data. Substra's documentation refers to each of these clusters as "TrainDataNode".

Clusters are physically in Belgium according to Google ("zone europe-west1" https://cloud.google.com/compute/docs/regions-zones?hl=en). GKE version used is `1.27.2-gke.1200` and the machines used are the "n1-standard-16" https://cloud.google.com/compute/docs/general-purpose-machines?hl=en#n1_machine_types. Regarding the communication protocol between centers, the organizations communicate with the orchestrator via gRPC and over http(s) one to another. Since the experiment is simulated in an internal environment using synthetic data we chose not to enforce mutual transport layer security (mTLS). More information can be found in Substra's documentation: https://docs.substra.org/en/latest/documentation/components.html.

Metastatic pancreatic adenocarcinoma data infrastructure setup. Similarly, we deploy within Owkin Inc.'s own Federated Research Network cloud infrastructure four nodes: a node for each of the three participating centers and an additional server node, the "AggregationNode", with responsibilities described above. Each node here is independent: the nodes are deployed in different regions with different cloud providers. PanCan data are located in the US, while Idibgi and FFCD data are hosted in Europe. The precise version information of the Substra versions used for this deployment is `0.51.0+dev` for `substra-frontend`, `0.47.0+d5dfbdb6` for `substra-backend` and `0.42.0+e6b1bddb` for the `orchestrator` repository.

Partner centers uploaded their data to the corresponding nodes.

Estimation of the treatment effect

We compare FedECA to several competitors, which, with the exception of pooled IPTW, are all adapted to the ECA setup, where the IPD of a local cohort are accessible and only aggregated statistics are accessible for an external cohort. Given the time-to-event nature of the outcome, we choose to estimate the hazard ratio under the proportional hazards assumption as a measure of the treatment effect. For all competitors, data is used to fit a Cox model as implemented in the `lifelines` library[91] to obtain the estimation.

Unweighted Cox regression

We implement a naïve Cox model regressing the observed outcome Y on the treatment allocation variable A, without using the weights of the samples. This corresponds to an unadjusted comparison between the treated and untreated groups, which would be valid in a randomized setting but not in an external control arm case. In the ECA setup, this estimator corresponds to the WebDISCO method, and we use the implementation provided by the authors of this method.

MAIC

Although not a method originally proposed for ECA, the MAIC method can be adapted to perform ECA analysis under proper assumptions: First, for the reweighting step, we make use of the implementation available in the `indcomp` package (https://github.com/AidanCooper/indcomp). Specifically, the IPD of the local cohort are reweighted so that a specified group of covariates matches the external cohort in terms of means and variances, creating, by design, reweighted data with zero SMD relative to the external cohort for each covariate. Then, in the absence of IPD of the external cohort, in order to train the Cox model to estimate treatment effect, we further assume that the (aggregated) risk set of the external cohort is also accessible. In this case, methods based on the digitization of the Kaplan-Meier curve[114] can be used to construct pseudo IPD as an approximation to the IPD of the external cohort. The pseudo IPD are then assigned a uniform weight of one and combined with the reweighted local cohort to estimate the treatment effect. In our simulation experiments, for reasons of simplicity and without loss of validity, we use the real IPD of the external cohort as an idealization of the pseudo IPD, which sets the upper bound of MAIC's performance in the ECA setup.

Pooled IPTW

The general concept and strategy of IPTW has been described before (see "Method overview"). In the implementation, the core estimation process is divided into two key steps. First, the propensity scores are estimated using unpenalized logistic regression or, alternatively, they can be provided externally to the estimator. These scores are then used to compute inverse probability weights tailored to the effect estimand. For the average treatment effect (ATE), weights are based on the inverse of propensity scores for both treated and control groups. For the average treatment effect on the treated (ATT), the weights involve a combination of treatment indicators (for the treated individuals) and inverse propensity scores (for the control individuals). Second, the treatment effect estimation is performed by fitting a weighted Cox proportional hazards model, where the inverse probability weights are incorporated in the regression model of the observed outcome Y on the treatment allocation A.

Competing paradigms

We already motivated the choice of IPTW as the best weighting method for small sample sizes. However other methods than weighting and matching could be considered for federation as well such as G-computation[17,18] or doubly debiased machine learning[19,20] as their performance should be comparable[20]. We leave their federation to future work.

Experiments details

Early stopping within a static distributed framework. As Substra is static and requires to fix the number of federated rounds a priori, we implement early-stopping for the stopping criterion on the Hessian norm by running up to MAX_{iter} rounds (20 in practice) and backtrack to find the first round where convergence was achieved.

Federated hyper-parameters selection. We note that, due to distribution and privacy constraints, standard practices such as cross-validation might become difficult to setup. In practice, in the case there is one sample per patient, which is what we study here, what we recommend is to treat the distributed datasets as a single distributed dataset, as in the federated global bootstrap, and proceed to splitting accordingly. We further recommend per-center stratification for fairness considerations and to avoid pathological cases. We note that, naïvely, any stratification on a private variable would require exposing and sharing this variable at least to the server, which limits the kinds of cross-validation that can be applied. We refer to the works of[115–117] for going beyond those recommandations.

Software and reproducibility. Following the recent trend of switching from R to Python for implementing statistical software[93,118,119], we chose Python as the base language for our implementation. This choice is also motivated by the fact that most FL research implementation code is written in Python. We follow reference survival analysis packages implementation design choices, such as `lifelines`[91] and `scikit-survival`[93]. We use the Substra software[30], which is an open-source software that has been audited and validated by security teams of both hospitals and pharmaceutical companies across different FL projects. Substra has demonstrated its ability to be deployed in real-world conditions for biomedical research purposes in the MELLODDY project[23,23], as well as in the HealthChain project on breast cancer treatment response prediction[25].

FedECA is available as a Python package on Github: https://github.com/owkin/fedeca for non-commercial use.

The code in this repository follows best practices such as continuous integration (CI), thorough code testing (coverage of code at 82% on commit `a6ec22c`), deployed documentation using Github pages as well as the use of pre-commit hooks to help manage the repository's evolution.

The availability of the code not only ensures the reproducibility of the results presented in this article as well as the possibility to audit its implementation, but also opens the possibility for other research teams to perform real-world federated ECA.

Indeed, a user can launch FedECA running the exact same code either in-RAM for simulations, or on a real deployed substra network in real conditions by modifying the backend type, as shown in Supplementary Fig. 1.

The FedECA repository contains a quickstart as well as detailed documentation and comments, which should allow easy replication.

All quantitative figures in this article with synthetic data can be reproduced by following instructions in `experiments/README.md`. The associated yaml configurations provide all hyper-parameters that were used.

For experiments on 10 centers replication involves deploying a substra network, which require some development operations (DevOps) capabilities. However, details in "Synthetic data infrastructure setup" should be sufficient to reproduce the results. The associated experiment script is defined in real_world_runtimes.yaml.

For experiments on YODA data, we install the fedeca package within the YODA platform, split the data in such a way that the control arm and the treatment arm are in two separate groups, and run fedeca with bootstrap variance estimation. Scripts used to preprocess the data and run the experiments are available in the yoda folder in the `fedeca` repository.

For experiments on metastatic pancreatic adenocarcinoma data, we use the fedeca package unaltered on commit `a6ec22c` after having registered the data in the Substra platform. Obfuscated versions of the scripts that ran on the deployed platform and that were used to generate the related figures in the article are available in the pdac folder in the `fedeca` repository. By obfuscated, we mean that dataset hashes or URLs in this script were converted to random strings so they cannot be mapped to any of the original data or servers.

The nature of this last experiment is such that replications require data access which might be restricted, see "Datasets and cohorts construction". However once access to data is obtained and federated network is deployed all experiments should be easily reproduced thanks to the above scripts.

Regarding the automatic tracing of all quantities communicated by FedECA's federated algorithms such as the one reported in Supplementary Fig. 10 the logging code originally developed by[120] and relying on remote methods decorators can be found in the jean/logging branch. Executing `plot_graphs_and_tables.py` automatically generates all corresponding graphs and tables in the article.

Further questions can be addressed to the corresponding author J.O.d.T. through the creation of github issues or via direct e-mail.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

All data generated in this study can be re-generated using the scripts provided in https://github.com/owkin/fedeca. Data are available from the Yale University Open Data Access (YODA) Project under restricted access to qualified researchers. Access can be obtained by following instructions on the YODA Project website https://yoda.yale.edu/request/. The entire data access process takes approximately 3 months. Access is provided for 1 year but can be renewed for additional years. The FFCD, IDIBGI data are available under restricted access to qualified researchers, access can be obtained by contacting directly the main investigators in each center: J.-B. B. for FFCD and R. C. for IDIBGI. Expected timeframe for access is a few months. Data can be available for any negotiated durations. Data from PanCAN are available under restricted access to qualified researchers, access can be obtained by submitting a proposal for review at www.pancan.org/spark. Data will be provided to qualified and approved researchers within one month of request. The duration of data access varies based on contractual agreement, but generally is two or three years. The clinical data used in this study are subject to access restrictions due to both regulatory requirements related to healthcare data sharing as well as contractual requirements related to intellectual property (IP) considerations. Source data are provided with this paper.

Code availability

The integrality of the code is publicly released and openly available for research purposes under a research only license at the following URL: https://github.com/owkin/fedeca.

References

1. DiMasi, J. A., Grabowski, H. G. & Hansen, R. W. Innovation in the pharmaceutical industry: new estimates of r&d costs. *J. Health Econ.* **47**, 20–33 (2016).
2. Hay, M., Thomas, D. W., Craighead, J. L., Economides, C. & Rosenthal, J. Clinical development success rates for investigational drugs. *Nat. Biotechnol.* **32**, 40–51 (2014).
3. Dowden, H. & Munro, J. Trends in clinical success rates and therapeutic focus. *Nat. Rev. Drug Discov.* **18**, 495–496 (2019).
4. Ventz, S. et al. Design and evaluation of an external control arm using prior clinical trials and real-world data. *Clin. Cancer Res.* **25**, 4993–5001 (2019).
5. Yin, X. et al. Historic clinical trial external control arm provides an actionable gen-1 efficacy estimate before a randomized trial. *JCO Clin. Cancer Inform.* **7**, e2200103 (2023).
6. Wang, X. et al. Current perspectives for external control arms in oncology clinical trials: Analysis of EMA approvals 2016-2021. *J. Cancer Policy* **35**, 100403 (2023).
7. Center for Drug Evaluation, Center for Biologics Evaluation Research, and Oncology Center of Excellence Research. Considerations for the design and conduct of externally controlled trials for drug and biological products. https://www.fda.gov/media/164960/download (2023).
8. European Medicines Agency. Reflection paper on establishing efficacy based on single-arm trials submitted as pivotal evidence in a marketing authorisation. https://www.ema.europa.eu/en/documents/scientific-guideline/reflection-paper-establishing-efficacy-based-single-arm-trials-submitted-pivotal-evidence-marketing_en.pdf (2023).
9. Khachatryan, A., Read, S. H. & Madison, T. External control arms for rare diseases: building a body of supporting evidence. *J. Pharmacokinet. Pharmacodyn.* **50**, 501–506 (2023).
10. Mishra-Kalyani, P. S. et al. External control arms in oncology: current use and future directions. *Ann. Oncol.* **33**, 376–383 (2022).
11. Lambert, J. et al. Enriching single-arm clinical trials with external controls: possibilities and pitfalls.*Blood Adv.* **7**, 5680–5690 (2022).

12. Przepiorka, D. et al. FDA approval: blinatumomab. *Clin. Cancer Res.* **21**, 4035–4039 (2015).
13. Robins, J. M. Data, design, and background knowledge in etiologic inference. *Epidemiology* **12**, 313–320 (2001).
14. Austin, P. C. An introduction to propensity score methods for reducing the effects of confounding in observational studies. *Multivar. Behav. Res.* **46**, 399–424 (2011).
15. Lunceford, J. K. & Davidian, M. Stratification and weighting via the propensity score in estimation of causal treatment effects: a comparative study. *Stat. Med.* **23**, 2937–2960 (2004).
16. Austin, P. C. Variance estimation when using inverse probability of treatment weighting (IPTW) with survival analysis. *Stat. Med.* **35**, 5642–5655 (2016).
17. Robins, J. A new approach to causal inference in mortality studies with a sustained exposure period-application to control of the healthy worker survivor effect. *Math. Model.* **7**, 1393–1512 (1986).
18. Chatton, A. et al. G-computation, propensity score-based methods, and targeted maximum likelihood estimator for causal inference with different covariates sets: a comparative simulation study. *Sci. Rep.* **10**, 9219 (2020).
19. Chernozhukov, V. et al. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal* https://doi.org/10.1111/ectj.12097 (2018).
20. Loiseau, N. et al. External control arm analysis: an evaluation of propensity score approaches, g-computation, and doubly debiased machine learning. *BMC Med. Res. Methodol.* **22**, 1–13 (2022).
21. Ohmann, C. et al. Sharing and reuse of individual participant data from clinical trials: principles and recommendations. *BMJ open* **7**, e018647 (2017).
22. McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, 1273–1282. (PMLR, 2017).
23. Oldenhof, M. et al. Industry-scale orchestrated federated learning for drug discovery. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. (AAAI Press, 2023).
24. Pati, S. et al. Federated learning enables big data for rare cancer boundary detection. *Nat. Commun.* **13**, 7346 (2022).
25. Ogier du Terrail, Jean et al. Federated learning for predicting histological response to neoadjuvant chemotherapy in triple-negative breast cancer. *Nat. Med.* **29**, 135–146 (2023).
26. Le-Rademacher, J. & Wang, X. Time-to-event data: an overview and analysis considerations. *J. Thorac. Oncol.* **16**, 1067–1074 (2021).
27. Rosenbaum, P. R. & Rubin, D. B. The central role of the propensity score in observational studies for causal effects. *Biometrika* **70**, 41–55 (1983).
28. US Food and Drug Administration, et al. Considerations for the design and conduct of externally controlled trials for drug and biological products. https://www.fda.gov/regulatory-information/search-fda-guidance-documents/considerations-design-and-conduct-externally-controlled-trials-drug-and-biological-products (2023).
29. Signorovitch, J. E. et al. Matching-adjusted indirect comparisons: a new tool for timely comparative effectiveness research. *Value Health* **15**, 940–947 (2012).
30. Galtier, M. N. & Marini, C. Substra: a framework for privacy-preserving, traceable and collaborative machine learning. *arXiv*. https://doi.org/10.48550/arXiv.1910.11567 (2019).
31. Pedregosa, F. et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
32. Saad, F. et al. Apalutamide plus abiraterone acetate and prednisone versus placebo plus abiraterone and prednisone in metastatic, castration-resistant prostate cancer (ACIS): a randomised, placebo-controlled, double-blind, multinational, phase 3 study. *Lancet Oncol.* **22**, 1541–1559 (2021).
33. Ryan, C. J. et al. Abiraterone in metastatic prostate cancer without previous chemotherapy. *N. Engl. J. Med.* **368**, 138–148 (2013).
34. Krumholz, H. M. & Waldstreicher, J. The yale open data access (yoda) project–a mechanism for data sharing. *N. Engl. J. Med.* **375**, 403–405 (2016).
35. Ross, J. S. et al. Overview and experience of the Yoda project with clinical trial data sharing after 5 years. *Sci. Data* **5**, 1–14 (2018).
36. Dahan, L. et al. Randomized phase ii trial evaluating two sequential treatments in first line of metastatic pancreatic cancer: results of the panoptimox-prodige 35 trial. *J. Clin. Oncol.* **39**, 3242–3250 (2021).
37. Rinaldi, Y. et al. Gemcitabine plus nab-paclitaxel until progression or alternating with FOLFIRI. 3, as first-line treatment for patients with metastatic pancreatic adenocarcinoma: The Federation Francophone de cancérologie digestive-prodige 37 randomised phase II study (Firgemax). *Eur. J. Cancer* **136**, 25–34 (2020).
38. Chan, KelvinK. W. et al. Real-world outcomes of folfirinox vs gemcitabine and nab-paclitaxel in advanced pancreatic cancer: a population-based propensity score-weighted analysis. *Cancer Med.* **9**, 160–169 (2020).
39. Buchanan, A. L. et al. Worth the weight: using inverse probability weighted Cox models in AIDS research. *AIDS Res. Hum. Retroviruses* **30**, 1170–1177 (2014).
40. Shu, D., Yoshida, K., Fireman, B. H. & Toh, S. Inverse probability weighted Cox model in multi-site studies without sharing individual-level data. *Stat. methods Med. Res.* **29**, 1668–1681 (2020).
41. Luo, C. et al. ODACH: a one-shot distributed algorithm for the Cox model with heterogeneous multi-center data. *Sci. Rep.* **12**, 6627 (2022).
42. Li, D., Lu, W., Shu, D., Toh, S. & Wang, R. Distributed Cox proportional hazards regression using summary-level information.*Biostatistics* **24**, 776–794 (2022).
43. Park, J. A., Kim, T. H., Kim, J. & Park, Y. R. WICOX: Weight-based integrated Cox model for time-to-event data in distributed databases without data-sharing. *IEEE J. Biomed. Health Inform.* **27**, 526–537 (2022).
44. Daniel, R., Zhang, J. & Farewell, D. Making apples from oranges: Comparing noncollapsible effect estimators and their standard errors after adjustment for different covariate sets. *Biom. J.* **63**, 528–557 (2021).
45. Pusceddu, S. et al. Comparative effectiveness of gemcitabine plus nab-paclitaxel and folfirinox in the first-line setting of metastatic pancreatic cancer: a systematic review and meta-analysis. *Cancers* **11**, 484 (2019).
46. Chiorean, ElenaGabriela, Cheung, W. Y., Giordano, G., Kim, G. & Al-Batran, Salah-Eddin Real-world comparative effectiveness of nab-paclitaxel plus gemcitabine versus folfirinox in advanced pancreatic cancer: a systematic review. *Ther. Adv. Med. Oncol.* **11**, 1758835919850367 (2019).
47. Williet, N. et al. Folfirinox versus gemcitabine/nab-paclitaxel as first-line therapy in patients with metastatic pancreatic cancer: a comparative propensity score study. *Ther. Adv. Gastroenterol.* **12**, 1756284819878660 (2019).
48. Klein-Brill, A., Amar-Farkash, S., Lawrence, G., Collisson, E. A. & Aran, D. Comparison of folfirinox vs gemcitabine plus nab-paclitaxel as first-line chemotherapy for metastatic pancreatic ductal adenocarcinoma. *JAMA Netw. open* **5**, e2216199–e2216199 (2022).
49. Hegewisch-Becker, S. et al. Tpk-group (tumour registry pancreatic cancer). Results from the prospective German TPK clinical cohort

study: treatment algorithms and survival of 1,174 patients with locally advanced, inoperable, or metastatic pancreatic ductal adenocarcinoma. *Int. J. Cancer* **144**, 981–990 (2019).

50. Riedl, J. M. et al. Gemcitabine/nab-paclitaxel versus folfirinox for palliative first-line treatment of advanced pancreatic cancer: a propensity score analysis. *Eur. J. Cancer* **151**, 3–13 (2021).

51. Chun, JungWon et al. Comparison between folfirinox and gemcitabine plus nab-paclitaxel including sequential treatment for metastatic pancreatic cancer: a propensity score matching approach. *BMC Cancer* **21**, 537 (2021).

52. The Lancet Gastroenterology Hepatology. Cause for concern: the rising incidence of early-onset pancreatic cancer (2023).

53. Pantanowitz, A. & Marwala, T. Missing data imputation through the use of the random forest algorithm. In *Advances in computational intelligence*, pages 53–62. (Springer, 2009).

54. Stekhoven, D. J. & Bühlmann, P. Missforest-non-parametric missing value imputation for mixed-type data. *Bioinformatics* **28**, 112–118 (2012).

55. Cerda, P., Varoquaux, Gaël & Kégl, Balázs Similarity encoding for learning with dirty categorical variables. *Mach. Learn.* **107**, 1477–1494 (2018).

56. Le Morvan, M., Josse, J., Moreau, T., Scornet, E. & Varoquaux, Gaël Neumiss networks: differentiable programming for supervised learning with missing values. *Adv. Neural Inf. Process. Syst.* **33**, 5980–5990 (2020).

57. Mayer, I. et al. Doubly robust treatment effect estimation with missing attributes. *Ann. Appl. Stat.* **14**, 1409–1431 (2020).

58. Le Morvan, M., Josse, J., Scornet, E. & Varoquaux, Gaël What's a good imputation to predict with missing values? *Adv. Neural Inf. Process. Syst.* **34**, 11530–11540 (2021).

59. Yao, L. et al. A survey on causal inference. *ACM Trans. Knowl. Discov. Data* **15**, 1–46 (2021).

60. Imbens, G. W. Sensitivity to exogeneity assumptions in program evaluation. *Am. Econ. Rev.* **93**, 126–132 (2003).

61. Zhao, L. et al. On the restricted mean survival time curve in survival analysis. *Biometrics* **72**, 215–221 (2016).

62. Pak, K. et al. Interpretability of cancer clinical trial results using restricted mean survival time as an alternative to the hazard ratio. *JAMA Oncol.* **3**, 1692–1696 (2017).

63. Conner, S. C. et al. Adjusted restricted mean survival times in observational studies. *Stat. Med.* **38**, 3832–3860 (2019).

64. Bonawitz, K. et al. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 1175–1191, (2017).

65. Bloom, B. H. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM.* **13**, 422–426 (1970).

66. Marchand, T., Muzellec, B., Béguier, C., Ogier du Terrail, J. and Andreux, M., Securefedyj: a safe feature gaussianization protocol for federated learning. In Koyejo, S. et al. editors, *Advances in Neural Information Processing Systems*, **35**, 36585–36598. (Curran Associates, Inc., 2022).

67. Kairouz, P. et al. Advances and open problems in federated learning. *Found. Trends® Mach. Learn.* **14**, 1–210 (2021).

68. Bujotzek, Markus Ralf, et al. "Real-world federated learning in radiology: hurdles to overcome and benefits to gain." Journal of the American Medical Informatics Association 32.1, 193–205 (2025).

69. Breslow, N. E. Analysis of survival data under the proportional hazards model. *Int. Stat. Rev. / Rev. Int. de. Stat.* **43**, 45–57 (1975).

70. Binder, D. A. Fitting Cox's proportional hazards models from survey data. *Biometrika* **79**, 139–147 (1992).

71. Klein, J. P. et al. *Survival analysis: techniques for censored and truncated data*, **1230**, (Springer, 2003).

72. Toh, S. et al. Combining distributed regression and propensity scores: a doubly privacy-protecting analytic method for multicenter research. *Clinical Epidemiology*, 1773–1786, (2018).

73. Xiong, R. et al. Federated causal inference in heterogeneous observational data. Statistics in Medicine 42. **24**, 4418–4439 (2023).

74. Han, L. et al. Federated adaptive causal estimation (face) of target treatment effects. Journal of the American Statistical Association, 1–14 (2025).

75. Han, L. Shen, Z. & Zubizarreta, J. Multiply robust federated estimation of targeted average treatment effects. *NeurIPS*. **36**, 70453–70482 (2023).

76. Tarumi, S., Suzuki, M., Yoshida, H., Miyauchi, S. & Kurazume, R. Personalized federated learning for institutional prediction model using electronic health records: A covariate adjustment approach. In *2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 1–4. (IEEE, 2023).

77. Almodóvar, A., Parras, J. & Zazo, S. Propensity weighted federated learning for treatment effect estimation in distributed imbalanced environments. *Comput. Biol. Med.* **178**, 108779 (2024).

78. Chia-Lun, L. et al. WebDISCO: a web service for distributed Cox model learning without patient-level data sharing. *J. Am. Med. Inform. Assoc.* **22**, 1212–1219 (2015).

79. Andreux, M., Manoel, A., Menuet, R., Saillard, C. and Simpson, C. Federated survival analysis with discrete-time Cox models. *arXiv* https://doi.org/10.48550/arXiv.2006.08997 (2020).

80. Wang, X. et al. SurvMaximin: robust federated approach to transporting survival risk prediction models. *J. Biomed. Inform.* **134**, 104176 (2022).

81. Alberto, A. & Matteucci, M. Federated survival forests. *International Joint Conference on Neural Networks (IJCNN)*. (IEEE, 2023).

82. Terrail, JeanOgierdu et al. Flamby: datasets and benchmarks for cross-silo federated learning in realistic healthcare settings. *Adv. Neural Inf. Process. Syst.* **35**, 5315–5334 (2022).

83. Huang, C., Wei, K., Wang, C., Yu, Y. & Qin, G. Covariate balance-related propensity score weighting in estimating overall hazard ratio with distributed survival data. *BMC Med. Res. Methodol.* **23**, 233 (2023).

84. Rassen, J. A., Avorn, J. & Schneeweiss, S. Multivariate-adjusted pharmacoepidemiologic analyses of confidential information pooled from multiple health care utilization databases. *Pharmacoepidemiol. drug Saf.* **19**, 848–857 (2010).

85. Lee, J. et al. Privacy-preserving patient similarity learning in a federated environment: development and analysis. *JMIR Med. Inform.* **6**, e7744 (2018).

86. Yuji, K. et al. Collaborative causal inference on distributed data. *Expert. Syst. Appl.* **244**, 123024 (2024).

87. Imakura, A., Tsunoda, R., Kagawa, R., Yamagata, K. & Sakurai, T. DC-COX: Data collaboration Cox proportional hazards model for privacy-preserving survival analysis on multiple parties. *J. Biomed. Inform.* **137**, 104264 (2023).

88. Yang, T. et al. Applied federated learning: improving Google keyboard query suggestions. *arXiv* https://doi.org/10.48550/arXiv.1812.02903 (2018).

89. Islamov, R., Qian, X. & Richtárik, P. Distributed second-order methods with fast rates and compressed communication. In *International conference on machine learning*, 4617–4628 (PMLR, 2021).

90. Li, T. et al. Feddane: A federated Newton-type method. In *2019, 53rd Asilomar Conference on Signals, Systems, and Computers*, 1227–1231 (IEEE, 2019).

91. Davidson-Pilon, C. lifelines: survival analysis in python. *J. Open Source Softw.* **4**, 1317 (2019).

92. Zou, H. & Hastie, T. Regularization and variable selection via the elastic net. *J. R. Stat. Soc.: Ser. B*. **67**, 301–320 (2005).
93. Pölsterl, S. scikit-survival: A library for time-to-event analysis built on top of scikit-learn. *J. Mach. Learn. Res*. **21**, 8747–8752 (2020).
94. Hahn, P.R., Carvalho, C.M., Puelz, D. & He, J. Regularization and confounding in linear regression for treatment effect estimation. *Bayesian Anal*. **13**, 163–182 (2018).
95. Courtiol, P. et al. Deep learning-based classification of mesothelioma improves the prediction of patient outcome. *Nat. Med*. **25**, 1519–1525 (2019).
96. Liu, N., Zhou, Y. & Lee, J. J. IPDfromKM: reconstruct individual patient data from published Kaplan-Meier survival curves. *BMC Med. Res. Methodol*. **21**, 111 (2021).
97. Substra Team. Our privacy strategy. In Substra documentation, version 1.0.0 (Owkin, 2024). Available at: https://docs.substra.org/en/stable/additional/privacy-strategy.html (accessed Aug. 10, 2025) (2024).
98. Shokri, R., Stronati, M., Song, C. & Shmatikov, V. Membership inference attacks against machine learning models. In the *2017 IEEE symposium on security and privacy (SP)*, 3–18. (IEEE, 2017).
99. Youn, Y., Hu, Z., Ziani, J. & Abernethy, J. Randomized quantization is all you need for differential privacy in federated learning. preprint at *arXiv* https://doi.org/10.48550/arXiv.2306.11913 (2023).
100. Dwork, C. et al. The algorithmic foundations of differential privacy. *Found. Trends® Theor. Comput. Sci*. **9**, 211–407 (2014).
101. Damien Desfontaines. A list of real-world uses of differential privacy. 10 Ted is writing things (personal blog). https://desfontain.es/privacy/real-world-differential-privacy.html (2021).
102. Abadi, M. et al. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318 (2016).
103. Yousefpour, A. et al. Opacus: User-friendly differential privacy library in pytorch. *arXiv* preprint arXiv:2109.12298, (2021).
104. Mironov, I. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pages 263–275. (IEEE, 2017).
105. Kaplan, E. L. & Meier, P. Nonparametric estimation from incomplete observations. *J. Am. Stat. Assoc*. **53**, 457–481 (1958).
106. Sawyer, S. The greenwood and exponential greenwood confidence intervals in survival analysis. *Applied survival analysis: regression modeling of time to event data*, pages 1–14, (2003).
107. Greifer, N. Covariate balance tables and plots: a guide to the 'cobalt` package. https://ngreifer.github.io/cobalt/articles/cobalt.html#variance-in-standardized-mean-differences-and-correlations (2023).
108. Austin, P. C. A critical appraisal of propensity-score matching in the medical literature between 1996 and 2003. *Stat. Med*. **27**, 2037–2049 (2008).
109. Pébay, P., Terriberry, T. B., Kolla, H. & Bennett, J. Numerically stable, scalable formulas for parallel and online computation of higher-order multivariate central moments with arbitrary weights. *Comput. Stat*. **31**, 1305–1325 (2016).
110. Suissa, S. Immortal time bias in pharmacoepidemiology. *Am. J. Epidemiol*. **167**, 492–499 (2007).
111. Vaswani, A. Attention is all you need. *Advances in Neural Information Processing Systems*, (2017).
112. Kather, JakobNikolas, Ferber, D., Wiest, I. C., Gilbert, S. & Truhn, D. Large language models could make natural language again the universal interface of healthcare. *Nat. Med*. **30**, 2708–2710 (2024).
113. Singhal, K. et al. Large language models encode clinical knowledge. *Nature* **620**, 172–180 (2023).
114. Guyot, P., Ades, A. E., Ouwens, MarioJ. N. M. & Welton, N. J. Enhanced secondary analysis of survival data: reconstructing the data from published Kaplan-Meier survival curves. *BMC Med. Res. Methodol*. **12**, 2 (2012).
115. Dai, Z., Low, BryanKianHsiang & Jaillet, P. Federated bayesian optimization via thompson sampling. *Adv. Neural Inf. Process. Syst*. **33**, 9687–9699 (2020).
116. Khodak, M. et al. Federated hyperparameter tuning: challenges, baselines, and connections to weight-sharing. *Adv. Neural Inf. Process. Syst*. **34**, 19184–19197 (2021).
117. Wang, Z., Kuang, W., Zhang, C., Ding, B. & Li, Y. FedHPO-bench: A benchmark suite for federated hyperparameter optimization. In Krause, A. et al. editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, 35908–35948. PMLR, 23–29 Jul (2023).
118. Bertrand, Q., Klopfenstein, Q., Bannier, P. A., Gidel, G. & Massias, M. Beyond l1: faster and better sparse models with skglm. In *NeurIPS* (2022).
119. Muzellec, B., Teleńczuk, M., Cabeli, V. & Andreux, M. PyDESeq2: a Python package for bulk RNA-seq differential expression analysis. *Bioinformatics* **39**, btad547 (2023).
120. Muzellec, B., Marteau-Ferey, U. & Marchand, T. Fedpydeseq2: a federated framework for bulk RNA-seq differential expression analysis. *bioRxiv*, pages 2024–12, (2024).

Acknowledgments

This study, carried out under YODA Project 2023-5198, used data obtained from the Yale University Open Data Access Project, which has an agreement with JANSSEN RESEARCH & DEVELOPMENT, L.L.C. The interpretation and reporting of research using this data are solely the responsibility of the authors and does not necessarily represent the official views of the Yale University Open Data Access Project or JANSSEN RESEARCH & DEVELOPMENT, L.L.C. The data coming from the Pancreatic Cancer Action Network was collected through the SPARK health data platform and derives from the "PanCAN Know Your Tumor Program". We especially thank the patients that shared their data with this program. Part of this work corresponding to work-package 4 of the RHU AI-TRIOMPH and carried out by Owkin France was supported by Agence Nationale de la Recherche as part of the France 2030 plan with reference ANR-23-RHUS-0012 (H.L. and F.B). We thank Cameron Davidson Pilon for developing the wonderful python package that is `lifelines` and that was a source of inspiration for this work. We thank Arthur Pignet for discussions on efficient bootstrap implementation with substra as well as for his code allowing efficient distributed computation of moments. We thank Constance Beguier for her help in deriving the distributed computation of the Kaplan-Meier estimator. We thank Ulysse Marteau-Ferey for his help applying and modifying the Substra tracing code that he developed for another concurrent research work. We thank Andy Karabajakian for having helped answer questions on chemotherapies combinations for metastatic pancreatic cancers. We thank Maylis Largeteau and Parjeet Kaur for their help in writing the FedECA license. Finally, we thank Jean-Philippe Vert and Nathan Noiry for their insightful comments and suggestions.

Author contributions

M.A. and F.B. conceived the idea of investigating federated methods for external control arms and supervised the paper writing and both contributed equally. J. O.d.T., Q.K. and H.L. wrote the paper and led the research. Q.K. implemented the data generation process as well as the pooled baselines with the help of H.L. and J. O.d.T designed, implemented and launched all federated learning related code. M.A. and J. O.d.T wrote the differential privacy federated code together. H.L. and I.M. helped writing the paper and performing experiments. Notably H.L. performed the YODA experiments with the help of I.M. and J.O.d.T. N. L. reviewed the statistical methodology and helped writing the paper. M. H. provided support for creating the figures and feedbacks on the writing of the paper. M. D. on one side and T. C. and T. F. on the other were in charge of respectively administrating the federated learning network infrastructure and deploying Substra. L. D., J. T., P. L.-P., J.-B. B., S. Z., R. N. and J. C. were responsible for the data from the PRODIGE35 and PRODIGE37 trials held at FFCD. D. G. reviewed the paper. In addition to providing data from listed clinical trials, J. T. discussed the study design and reviewed the manuscript R. C. T. and A. G. V. curated the data from IDIBGI, K. A. and S. D. curated the data from PanCAN under the guidance of J. A. C. and Z. Y. in charge of data-harmonization and specifications.

Competing interests

The authors declare the existence of a financial competing interest. Some authors are or were employed by Owkin, Inc. during their time on the project (J. O.d.T, Q.K., M.A., H.L, I.M., N.L., M.H., M.D., T.C., T.F., F.B., J.A.C., Z.Y.). P. L.-P. has received honoraria for consulting and/or advisory board for AMGEN, Pierre Fabre, Biocartis, Servier and BMS. J.B. Bachet has received personal fees from Amgen, Bayer, Bristol Myers Squibb, GlaxoSmithKline, Merck Serono, Merck Sharp & Dohme, Pierre Fabre, Sanofi, Servier, and non-financial support from Amgen, Merck Serono, and Roche, outside the submitted work. J. T. has received honoraria as a speaker and/or in an advisory role from AMGEN, Astelllas, Astra Zeneca, Boehringer, BMS, Merck KGaA, MSD, Novartis, ONO pharmaceuticals, Pierre Fabre, Roche Genentech, Sanofi, Servier and Takeda. A. G. V. has received honoraria as a speaker and/or in an advisory role from Astra Zeneca, Merck Serono, MSD, Novartis, Roche Genentech, Sanofi, and Servier. R. N. has received honoraria as a consultant from Cure51. Part of this work corresponding to work-package 4 of the RHU AI-TRIOMPH and carried out by Owkin France was supported by Agence Nationale de la Recherche as part of the France 2030 plan with reference ANR-23-RHUS-0012 (H.L and F.B.). The remaining authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-025-62525-z.

Correspondence and requests for materials should be addressed to Jean Ogier du Terrail.

Peer review information *Nature Communications* thanks Ruoxuan Xiong, and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

Reprints and permissions information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.