Truly Privacy-Preserving Federated Analytics for Precision Medicine with Multiparty Homomorphic Encryption

David Froelicher[1,*], Juan R. Troncoso-Pastoriza[1], Jean Louis Raisaro[2,3], Michel A. Cuendet[4], Joao Sa Sousa[1], Jacques Fellay[2,5], and Jean-Pierre Hubaux[1]

[1]Laboratory for Data Security, EPFL, Lausanne, Switzerland
[2]Precision Medicine Unit, Lausanne University Hospital, Lausanne, Switzerland
[3]Data Science Group, Lausanne University Hospital, Lausanne, Switzerland
[4]Precision Oncology Center, Lausanne University Hospital, Lausanne, Switzerland
[5]School of Life Sciences, EPFL, Lausanne, Switzerland
[*]david.froelicher@epfl.ch

ABSTRACT

In biomedical research, real-world evidence, which is emerging as an indispensable complement of clinical trials, relies on access to large quantities of patient data that typically reside at separate healthcare institutions. Conventional approaches for centralizing those data are often not feasible due to privacy and security requirements. As a result, more privacy-friendly solutions based on federated analytics are emerging. They enable to simultaneously analyse medical data distributed across a group of connected institutions. However, these techniques do not inherently protect patients' privacy as they require institutions to share intermediate results that can reveal patient-level information. To address this issue, state-of-the-art solutions use additional privacy-preserving measures based on data obfuscation, which often introduce noise in the computation of the final result that can become too inaccurate for precision medicine use cases. We propose FAMHE, a modular system based on multiparty homomorphic encryption, that enables the privacy-preserving execution of federated analytics workflows yielding exact results and without leaking any intermediate information. To demonstrate the maturity of our approach, we reproduce the results of two published state-of-the-art centralized biomedical studies, and we demonstrate that FAMHE enables the efficient, privacy-preserving and decentralized execution of analyses that range from low computational complexity, such as Kaplan-Meier overall survival curves used in oncology, to high computational complexity, such as genome-wide association studies on millions of variants.

Introduction

One of the key requirements for fully realizing the potential of precision medicine is to make large amounts of medical data inter-operable and widely accessible to researchers. Today, however, medical data are scattered across various institutional strongholds, which makes accessing and aggregating such data in a centralized database extremely challenging, if not impossible. The challenges are not so much due to the technical hurdles of transporting high volumes of heterogeneous data across organizations as to the legal and regulatory barriers that make transfer of patient-level data outside a healthcare provider extremely complex and time-consuming. Moreover, stringent data-protection and privacy regulations (e.g., GDPR[1]) strongly restrict the transfer of personal data, even pseudonymized, across jurisdictions.

Federated analytics (FA) is emerging as a new paradigm that seeks to address the data governance and privacy issues related to medical data sharing[2,3]. FA enables different healthcare providers to collaboratively perform statistical analyses and develop machine learning models without exchanging the underlying datasets. Only aggregated results or model updates are transferred. As such, each healthcare provider can define its own data governance and keep control over the access to its patient-level data. FA offers unprecedented opportunities for exploiting large and diversified volumes of data spread across multiple institutions. It facilitates the development and validation of more generalizable AI algorithms that can yield more accurate and unbiased clinical recommendations and accelerate novel discoveries. This is particularly important in the context of rare-diseases and precision medicine, where the number of relevant patients in a single institution is often not sufficient to obtain unbiased results with enough statistical power.

Despite their potential, the adoption of FA-based approaches in the medical sector has been slower than expected. This is mainly due to (i) the lack of data interoperability across different healthcare providers and (ii) unresolved privacy issues related to the sharing of model updates or partial data aggregates in cleartext. Here, we focus on the second issue. Indeed, despite

patient-level data not being transferred in FA, it has been shown that the model updates (or partial aggregates) themselves can, under certain circumstances, leak sensitive personal information about the individuals used in the training sets, thus leading to re-identification, membership inference, and feature reconstruction[4,5].

Several open-source software platforms have been recently developed to provide access to FA distributed algorithms in a way that is packaged for end-users[3,6,7], e.g., DataSHIELD[6] is an open-source distributed data analysis and machine learning (ML) platform based on the open-source software R. However, none of these platforms addresses the problem of indirect data leakages that stem from the use of "vanilla" federated learning; hence their adoption poses important questions about the actual advantage in terms of eased compliance to regulations over more conventional data-analysis platforms that rely on data centralization[8–10], as partial aggregates and model updates can still be considered personal identifying data[4,11–14].

In this respect, the computer security and privacy community has proposed several advanced solutions for privacy-preserving FA based on differential privacy (diffP)[15,16], secure multiparty computation (SMC)[17,18], or homomorphic encryption (HE)[19–22] in order to protect partial aggregates (or model updates) from inference attacks during the training phase. However, the use of these techniques in FA introduces important trade-offs between privacy, utility and computational overheads that need further assessment and validation through practical use cases.

In order to ensure diffP[23] at each step of the process, some recent solutions[15,16] rely on obfuscation techniques to protect the aggregated data transferred out of healthcare centers. Yet, the use of obfuscation techniques introduces a significant trade-off between data privacy and accuracy of the end results, which is often considered unacceptable by practitioners working on highly stratified cohorts with small numbers of patients at each of the sites. The level of obfuscation needed to protect the aggregated data of each site at each training epoch often generates a final model that is too noisy to be used to derive useful conclusions in medical research. Moreover, it remains unclear how to set privacy parameters that provide acceptable mitigation of inference risks[24].

Other approaches rely on SMC[17,18] where the data are obliviously secret-shared and distributed among all participating sites or a set of computing parties, with the assumption that a majority of them are honest. Such parties then collaborate to perform a joint analysis on the data without revealing intermediate values. However, the use of SMC-based solutions includes several drawbacks that hinder their adoption in real-world medical use-cases: e.g., high network overheads and the difficulty to scale to large numbers of data providers.

Finally, a few solutions[25–27] rely on the use of HE to enable participating sites to securely centralize their data in an encrypted form to a third party (e.g., cloud provider), who can perform meaningful operations on them without learning anything about their content. Yet, standard fully homomorphic-encryption schemes, i.e. schemes enabling arbitrary computations on encrypted data, are still considered non-viable due to the high computational and storage overheads they introduce. Furthermore, these solutions introduce a single point of failure since only one entity holds the decryption key. To address this issue, distributed solutions that rely on HE[19–21,28] have been proposed. However, these solutions focus exclusively on simple computations, e.g., count and basic statistics, and are not suited for complex FA tasks.

In this paper, we present a new approach to FA based on Multiparty Homomorphic Encryption (MHE)[29] that effectively combines SMC and HE. We call our approach Federated Analytics with Multiparty Homomorphic Encryption (FAMHE). We rely on MHE to perform privacy-preserving FA by pooling the advantages of both SMC and HE and by minimizing their disadvantages. In particular, by relying on MHE and on the distributed protocols for federated analytics proposed by Froelicher et al.[22], our approach enables several sites to compute on their local patient-level data and then encrypt (*Local Computation & Encryption* in Figure 1) and homomorphically combine their local results under MHE (*Collective Aggregation* in Figure 1). The use of MHE ensures that the secret key of the underlying HE scheme never exists in full. Instead, the control over the decryption process is distributed across all participating sites, each one holding a fragment of the decryption key. This means that all participating sites have to agree to enable the decryption of any piece of data and that no single entity alone can decrypt the data. These steps can be repeated (*Iterate* in Figure 1), depending on the analytic task. At each new iteration, participating sites use the encrypted combination of the results of the previous iteration to compute on their local data without the need of decryption, e.g., gradient descent steps in the training of a regression model. The collectively encrypted and aggregated final result is eventually switched (*Collective Key Switching* in Figure 1) from an encryption under the collective public key to an encryption under the querier's public key (the blue lock in Figure 1) such that only the querier can decrypt.

Contrary to approaches that rely on obfuscation techniques to ensure privacy during the intermediate steps of a FA workflow, our approach does not generate undesired biases and achieves comparable accuracy to the one that would be obtained by conventional solutions that centralize all data in a single database. In case the final result of the analysis needs to be protected from potential inferences, our approach guarantees that obfuscation can be applied only to the decrypted final result, instead of being applied to all local model updates at each iteration during the model training process. This is possible because aggregate-level data transferred and processed across participating sites remain encrypted under the collective encryption key. Furthermore, our approach overcomes the performance limitations of both SMC and HE, as it can scale to a large number of data providers for the execution of complex real-world collaborative federated workflows, without revealing any intermediate

information. Our approach can significantly reduce the requirements for contractual agreements and the obligations of data controllers that often hinder multicentric medical studies, because data processed using MHE can be considered anonymous data under the General Data-Protection Regulation (GDPR)[11].

To demonstrate the readiness and maturity of our approach to handle real-world use cases for multicentric medical studies, as opposed to previous published attempts that focused mainly on artificial demonstrative examples[15,17,21,30], we use MHE to replicate two high-impact peer-reviewed multicentric studies that originally relied on data centralization[31,32]. In particular, we show that we can perform - in a privacy-preserving and fully decentralized way - tasks that span from analyses with low computational complexity, such as non-parametric survival analyses often used in oncology, to iterative analyses with high computational complexity, such as genome-wide association studies (GWAS) on millions of variants.

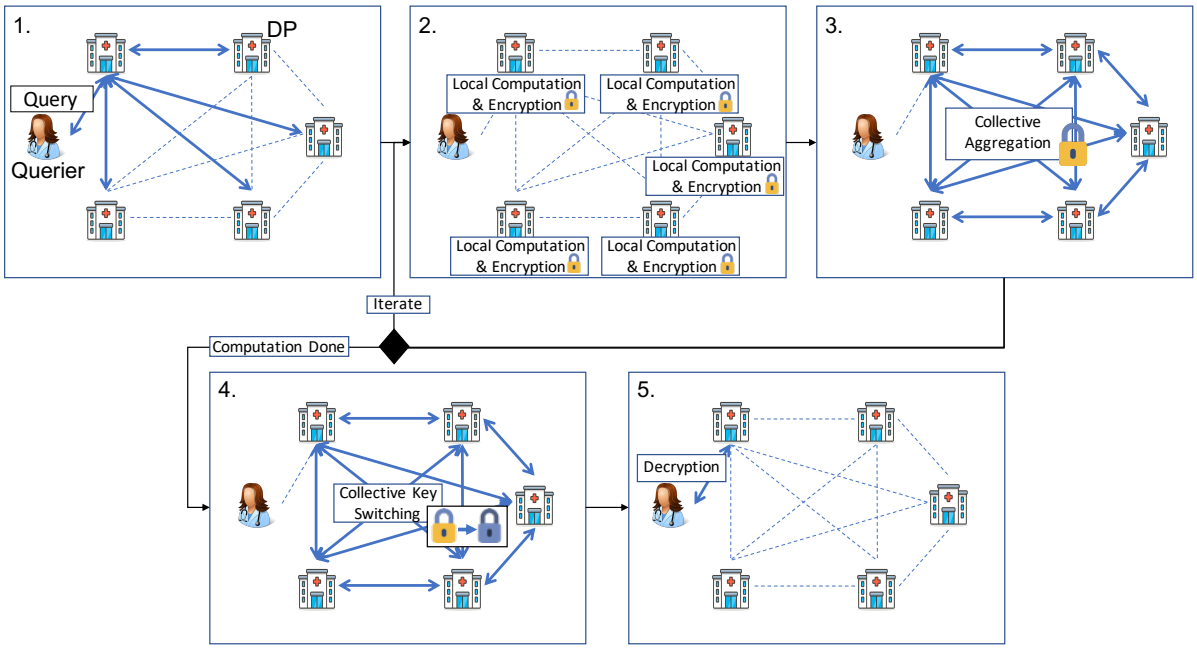


Figure 1. System Model and FAMHE Workflow. All entities are interconnected (dashed lines) and communication links at each step are shown by thick arrows. All entities (Data Providers (DPs) and Querier) are honest-but-curious and do not trust each other. In **1.** the querier sends the query (in clear) to all the DPs who (**2.**) locally compute on their cleartext data and encrypt their results with the collective public key. In **3.** the DPs' encrypted local results are aggregated. For iterative tasks, this process is repeated (**Iterate**). In **4.** the final result is then collectively switched by the DPs from the collective public key to the public key of the querier. In **5.** the querier decrypts the final result.

Results

In order to assess the computational overhead and accuracy of FAMHE in realistic conditions for large-scale multicentric research projects, we replicate two previously peer-reviewed biomedical studies by leading groups in Oncology and Genomics. The original studies relied on centralized databases, hence we artificially split the data over multiple virtual data providers (DPs) that are hosted on multiple computer nodes. We then perform FAMHE analysis with patient-level data access that is restricted to the local node, thus mimicking what individual healthcare centers would do within their secure clinical networks in a real use-case. In the first study, Samstein et al.[31] demonstrated that the tumor mutational burden (TMB) is a predictor of clinical response to Immune Checkpoint Inhibitor (ICI) treatments in patients with metastatic cancers. For this purpose, they computed Kaplan-Meier overall survival curves of 1,662 advanced-cancer patients treated with ICI and stratified by TMB values. In the second study, McLaren et al.[32] tested for the association between 4,057,178 variants and the HIV-1 viral load in an infected population of 1,857 European individuals by relying on a genome-wide association study (GWAS) that considered 12 covariates. In the original studies, the data had to be centralized in a single server, thus introducing multiple privacy risks. With FAMHE instead, the data remain stored by the data providers that directly compute on their local data. The local results are then (iteratively and) securely combined among multiple data providers by relying on FAMHE.

We implemented our solution by building on top of Lattigo[33], an open-source Go library for lattice-based cryptography. We refer to Online Methods for the detailed configuration and experimental environment settings.

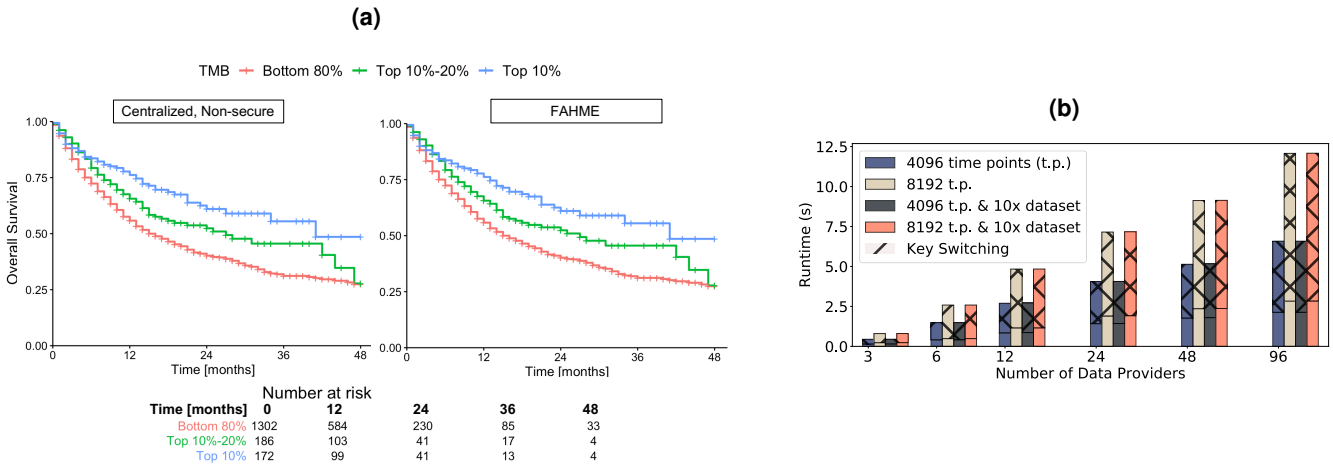


Figure 2. Secure and Distributed Reproduction of a Survival Curve Study. 2a: Survival curves generated in a centralized non-secure manner and with FAMHE on the data used by Samstein et al.[31]. With FAMHE, the original data are split among three data providers and the querier obtains exact results. The table in Figure 2a displays the number of patients at risk in time. The exact same numbers are obtained with the centralized, non-secure solution and with FAMHE. **2b**: FAMHE execution time for the computation of one (or multiple) survival curve(s) with a maximum of 8192 time points. For both the aggregation and key switching (from the collective public key to the querier's key), most of the execution time is spent in communication (up to 98%), as the operations on the encrypted data are lightweight and parallelized on multiple levels, i.e., among the data providers and among the encrypted values.

Kaplan-Meier Survival Analysis

In Figure 2a, we show an example of survival curves generated in a distributed and privacy-preserving way when the data are split over three DPs. We note that FAMHE generates the exact same survival curves as those produced by a standard centralized non-secure solution, e.g., Samstein et al.[31], as shown in Figure 2a.

In Figure 2b, we show that FAMHE efficiently achieves **exact results**, as the computation of the survival curves shown in Figure 2a is executed in less than 12 seconds, even when the data are scattered among 96 DPs. We also observe that the execution time is almost independent of the DPs' dataset size, as the same experiment performed on a 10x larger dataset (replicated 10x) takes almost the exact same time. We show that FAMHE's execution time remains below 12 seconds for up to 8192 time points. We note that the number of time points (instants at which an event can occur) is smaller than 200 in this particular study due to the rounding off of survival times to months. In summary, the FAMHE-based Kaplan-Meier estimator produces precise results and scales efficiently with the number of time points, each DPs' dataset size, and with the number of DPs. We remark that the hazard ratio, which is often computed in survival curve studies, can be directly estimated by the querier, based on the final result[34]. Otherwise, the exact computation can be implemented in the generic workflow described in Figure 1. This requires the training of proportional hazards regression models that is similar to (or can even be translated to[35]) the training of generalized linear models on which we rely to execute a GWAS, as described in the next section.

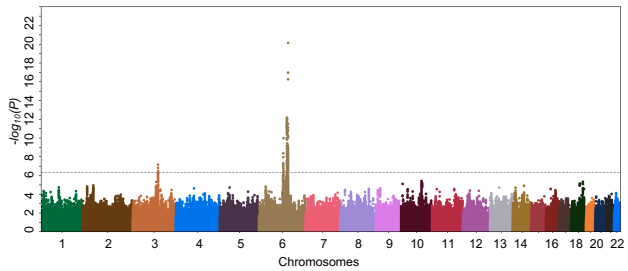
Genome-Wide Association Study (GWAS)

The execution of a GWAS requires the training of an independent regression model for each considered variant. The study performed by McLaren et al.[32] hence requires the training of more than 4 million models on a data set of 1857 samples (i.e., the subset of patients from the Swiss HIV Cohort Study that were part of the original study) with 13 features (i.e., 12 covariates and 1 variant) to test for association between the 4 million variants and the HIV-1 viral load (i.e., the label or phenotype).

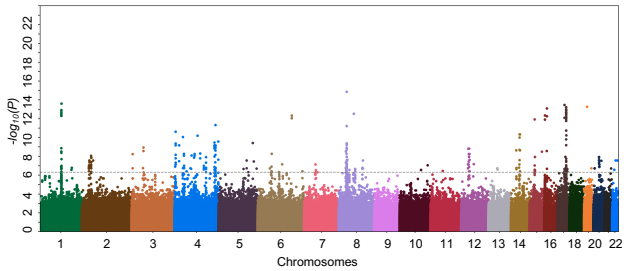
We propose two approaches, FAMHE-GWAS and FAMHE-FastGWAS, to perform this high-dimensional intensive computation in a secure and federated manner and to protect data confidentiality. In both solutions, no intermediate result is revealed at any point of the computation, and data that are exchanged between the data providers (DPs) are always collectively encrypted. In FAMHE-GWAS, we perform an exact linear regression by relying on the fact that the patients' covariates (i.e., 12 features) are common to all regression models. We compute once the covariance matrix of the covariates, before including the contribution of each variant (See Online Methods). In FAMHE-FastGWAS, we first obtain the covariate contributions by training once a linear model considering only the covariates as features. We then compute a new label (or phenotype) in which we report the covariate contributions before training in parallel 4 million uni-variate models for each of the 4 million variants (See Online Methods). In both approaches, the data providers iteratively use their local cleartext data and iteratively aggregate

their encrypted local results, following the workflow presented in Figure 1.

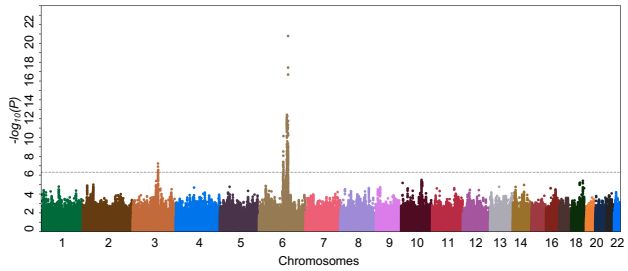
(a) *Original* Approach



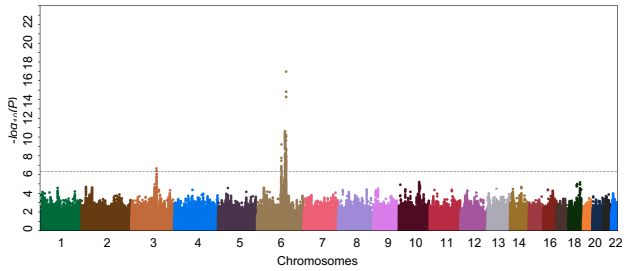
(b) *Meta-analysis* Approach



(c) FAMHE-GWAS



(d) FAMHE-FastGWAS



(e) GWAS Errors

		Indep.		Meta-ana.		FAMHE-FastGWAS		FAMHE-GWAS	
		-$\log_{10}$(P-val)	w	-$\log_{10}$(P-val)	w	-$\log_{10}$(P-val)	w	-$\log_{10}$(P-val)	w
3 DPs	all	0.369	0.04	0.448	0.04	$6.7e^{-3}$	$1.5e^{-3}$	$2.72e^{-3}$	$7.3e^{-4}$
	peaks	4.14	0.055	7.9	0.19	0.71	$6.61e^{-3}$	0.1392	$1.88e^{-7}$
6 DPs	all	0.409	0.0665	0.45	0.041	$8.3e^{-3}$	$1.61e^{-3}$	$2.78e^{-3}$	$7.4e^{-4}$
	peaks	4.86	0.12	7.95	0.195	0.82	$6.63e^{-3}$	0.1393	$2.3e^{-7}$
12 DPs	all	0.425	0.104	0.453	0.048	$9e^{-3}$	$1.63e^{-3}$	$2.79e^{-3}$	$7.7e^{-4}$
	peaks	6.619	0.126	7.99	0.197	0.848	$6.69e^{-3}$	0.1399	$3.6e^{-7}$

Figure 3. Comparison of the GWAS results obtained with different approaches with 12 DPs (when applicable). 3a *Original* is considered as the ground-truth and is obtained on a centralized cleartext dataset by relying on the PLINK[36] software. 3b *Meta-analysis* is also obtained with PLINK. 3c and 3d are the results obtained with FAMHE-GWAS and FAMHE-FastGWAS, respectively. In the original study (Figure 3a) and in our secure approach (Figures 3c and 3d), genome-wide signals of association ($log_{10}(P) < 5 \times 10^7$, dotted line) were observed on chromosomes 6 and 3. Table 3e displays the Absolute Averaged Error on the logarithm of the p-values (-$\log_{10}$(P-val)) and on the model weights (w) between *Original* and federated approaches. It also shows that one data provider performing the GWAS alone with only its local data (**Indep.**) obtains inaccurate results. For each number of data providers, we report the error averaged over all positions and the errors on the peaks identified with *Original* (see Figure 3a).

We compare FAMHE-GWAS and FAMHE-FastGWAS against (i) *Original*, the centralized non-secure approach in which the PLINK[36] software is used, (ii) *Meta-analysis*[36], a solution in which each DP locally and independently performs the GWAS and the results are then combined, and (iii) *Independent*, a solution in which a data provider uses only its part of the dataset to perform the GWAS. Note that *Meta-analysis* can also be securely executed by encrypting each DP's local result and by following the generic workflow presented in Figure 1.

In Figure 3a, we show the Manhattan plot that was obtained with *Original* (i.e., the ground-truth) and visually compare its result with *Meta-analysis*, FAMHE-FastGWAS and FAMHE-GWAS. With FAMHE we are always able to identify all

the significantly associated regions, whereas the results of the *Meta-analysis* approach are too noisy. We complement this comparison with Table 3e, in which we show the error in terms of the logarithm of the p-value (-$\log_{10}$(P-val)) and the regression weights (w) for all the considered approaches with respect to *Original*. We observe that FAMHE-FastGWAS yields an averaged error that is always smaller than 10^{-2} on the computed logarithmic p-value and thus enables the correct identification of all the peaks. The obtained error can be further reduced (by a factor of almost 3) by relying on FAMHE-GWAS.

Figure 4a shows that FAMHE is able to perform a GWAS on more than 4 million variants in less than one hour with FAMHE-FastGWAS, even when the data are spread among 12 data providers. We also note that FAMHE is highly parallelizable, i.e., doubling the number of available threads, would almost halve the execution time. We observe that FAMHE is 3-4 orders of magnitude slower than *Original*, which performs the same operation in 14 seconds when the data are stored in one single machine. Gathering all data in a single machine would require up to 30GB of communication. *Meta-analysis* takes around 5 minutes when executed on 12 DPs but yields very imprecise results. Finally, with FAMHE-GWAS, in which an exact linear regression is performed, the error is reduced (more than 3x smaller than with FAMHE-FastGWAS) but its execution time is, as shown in Figure 4a, 20x slower than FAMHE-FastGWAS. As described in Online Methods, FAMHE computes the p-values of multiple (between 512 and 8192) variants in parallel due to the *Single Instruction, Multiple Data (SIMD)* property of the cryptoscheme and is further parallelized on multiple levels, as the the data providers perform multi-threaded computations in parallel. As shown in Figure 4, FAMHE execution time increases linearly with the number of variants and better than linearly with the DP dataset size, and it decreases with the number of data providers.

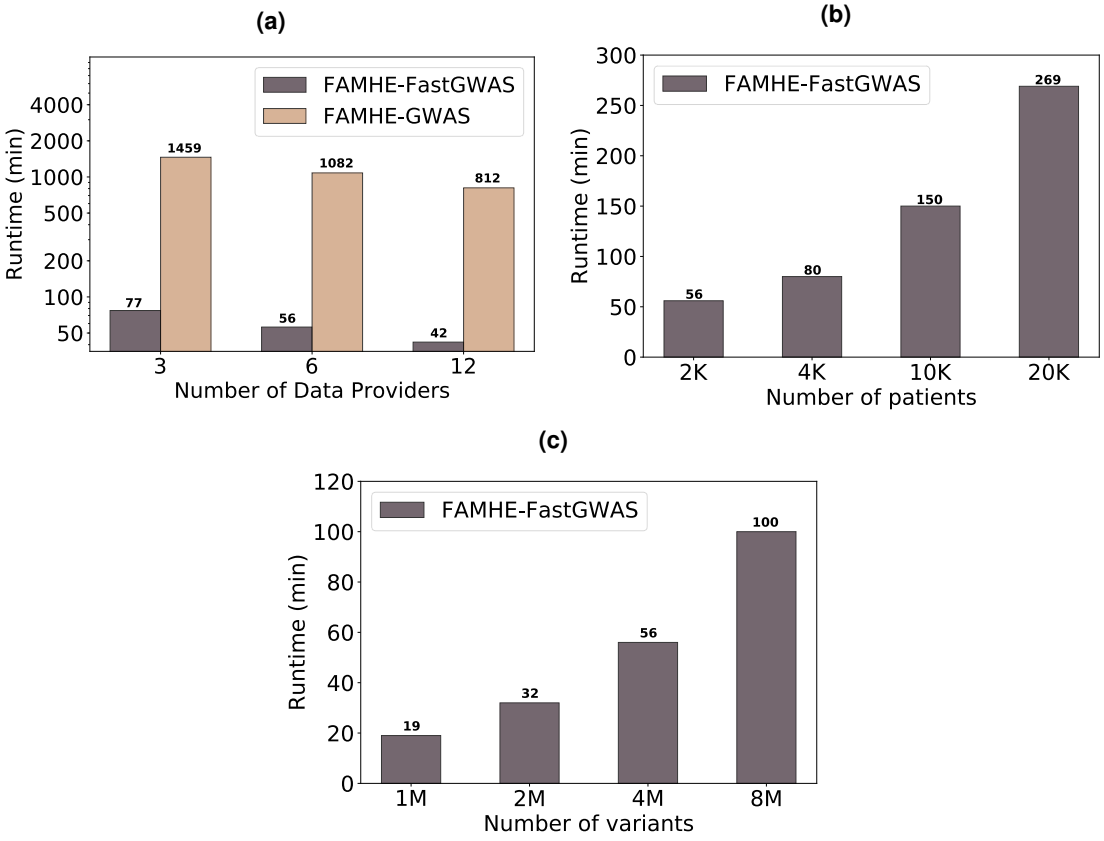


Figure 4. FAMHE Scaling. FAMHE's scaling with the number of data providers 4a, with the size of the dataset 4b and with the number of variants considered in the GWAS 4c. Unless otherwise stated, the original dataset containing 1857 samples and 4 million variants is evenly split among the data providers. By default, the number of DPs is fixed to 6. In 4b and 4c, the execution time of FAMHE-FastGWAS is shown, FAMHE-GWAS's scaling follows the same trend but is consistently 20x slower.

Discussion

FAMHE combines the best features of federated-learning solutions and of cryptographic solutions for federated analysis, while eliminating their respective disadvantages, i.e., difficulty to scale to large numbers of data providers, computational overhead, bias introduction on the final result and restrictive trust assumptions. Similar to existing federated-learning solutions, FAMHE enables a large number of data providers to keep their data locally stored under their control and to effectively collaborate in order to perform large-scale analyses. However, contrary to most federated-learning solutions[2,3,3,6,15,16], FAMHE also protects data confidentiality and does not require the addition of any noise to the results. As other proposed solutions based on advanced cryptography[17–21,37], FAMHE does not reveal intermediate values to any party. However, contrary to these solutions, FAMHE is not limited in the number of computing parties, it does not require data outsourcing and it enables the data providers to efficiently compute on their local cleartext data. FAMHE yields the same trade-off between performance and accuracy as

	Protect. Mecha.	Patient Level Data Protect.	GDPR Anonym.	Intermed. Val. Protect.	Scale w. Parties	Dishonest Majority	Preserve Utility	Comput. Flexibility	Complex Comput.	Appl. To Real Cases
AllOfUs[8] Gen. Eng.[9] UKBio.[10]	None	No	No	No	∼	No	Yes	Yes	Yes	Yes
DataSHIELD[6] Vantage6[7] SHRINE[38] Splink[3]	Aggregates	∼	No	No	Yes	No	∼	∼	Yes	∼
Bonomi et al.[15]	DiffP	Yes	No	∼	Yes	Yes	No	No	No	∼
Li et al.[16]	DiffP	Yes	No	∼	Yes	Yes	No	No	Yes	No
Cho et al.[18]	SMC	Yes	Yes	Yes	No	No	∼	No	Yes	Yes
Jagadeesh et al.[17]	SMC	Yes	Yes	Yes	No	No	∼	No	No	∼
Froelicher et al.[21] Lu et al.[37]	MHE	Yes	Yes	Yes	Yes	Yes	∼	∼	No	No
FAMHE	MHE	Yes	Yes	Yes	Yes	Yes	∼	∼	Yes	Yes

Table 1. Comparison with Existing Works. Existing solutions are compared on their protection mechanisms, on whether (i) patient levels and (ii) aggregated data are protected, (iii) the data protection satisfies the GDPR definition of anonymity, (iv) they scale with the number of data providers and computing parties, (v) a majority of computing parties can be dishonest (without deviating from the protocol), (vi) the obtained results are the same as if they were computed on a centralized cleartext dataset (i.e., utility), (vii) the system enables multiple types of computations, and (viii) the solution is tested in a real application scenario. A "∼" means that the property is partially fulfilled.

standard non-secure solutions do. As shown in Results, FAMHE can be instantiated with two different workflows for GWAS: FAMHE-GWAS performs heavier computations but yields more precise result than FAMHE-FastGWAS, which is able to perform a GWAS on 4 million variants in less than one hour. Moreover, FAMHE naturally provides exact results but can also be enhanced in order to ensure differential privacy with a minimal perturbation on the final result. Differential privacy is usually achieved by perturbing, i.e., adding noise, to the function's output such that the results obtained from two neighboring datasets, i.e., differing in only one sample, cannot be distinguished. The required noise magnitude depends on the function sensitivity. Existing federated analytics solutions based on differential privacy mechanisms usually rely on each data provider to perturb its local results before they are combined with other DPs' results. By keeping the intermediate results encrypted (and never decrypting them), FAMHE enables the perturbation to be applied on only the final result, i.e., before the *Collective Key Switching*. In this case, the DPs collectively generate the perturbating noise by generating it from composable distributions, e.g., Laplace, as explained by Kim et al.[39]. As the function's output sensitivity is the same for the DPs' local results and for the combined final result, FAMHE enables a smaller (proportionally to the number of DPs) perturbation of the final result for the same level of privacy. The choice of the privacy parameters is not trivial and is an interesting direction for future work.

The properties described above have an important consequence for the applicability and adoption of FAMHE for future real-world multicentric biomedical studies. Indeed, it was established by privacy law experts that data processed using MHE can be considered anonymous data under the General Data Protection Regulation (GDPR)[11]. Therefore, our approach significantly reduces the requirements for contractual agreements and the obligations of data controllers that often hinder multicentric medical studies. In contrast, existing solutions in which the intermediate results are often revealed to the computing entities do not comply to this definition, as intermediate results are still considered to be personal identifying data[12–14].

We summarize our comparison with existing works in Table 1 and refer to Online Methods for more details on the related work.

We have shown that complex biomedical analysis workflows can be integrated in the FAMHE generic pipeline. We have demonstrated that FAMHE is able to accurately reproduce the results of existing peer-reviewed biomedical studies when the data are split among multiple sites, while ensuring absolute data confidentiality. FAMHE is able to efficiently perform simple statistical computations, such as Kaplan-Meier survival curves, and more complex analyses such as GWAS. These use cases cover a wide-range of applications, ranging from cohort exploration to the training and evaluation of predictive regression models, and shows the generality of FAMHE. In future developments, we will extend the capabilities of FAMHE by integrating, in FAMHE's truly privacy-preserving generic workflow, more standard analysis tools and machine-learning algorithms, e.g., proportional-hazard regression models. We also believe that making FAMHE easily configurable by practitioners for many medical applications is an interesting and challenging future work. One step in this direction consists in integrating FAMHE with a user-friendly tool to facilitate the adoption of our solution by practitioners and biomedical researchers. For example, by integrating FAMHE in the MedCo[40] software, we will complement its privacy-preserving cohort-selection functionality hence obtain a user-friendly tool that streamlines the complete workflow of multicentric medical studies.

Online Methods

In this section, we describe the crypto-scheme that we rely on to build FAMHE and discuss how FAMHE differentiates itself from existing works. We describe FAMHE system and threat model, before detailing the execution of the privacy-preserving pipelines for survival curves and GWAS studies. Finally, we detail our experimental settings and how differential privacy can be ensured with FAMHE.

Cryptographic Background

The data exchanged by the data providers are always encrypted such that only the querier can decrypt the final result. For this purpose, we rely on a multiparty (or distributed) fully-homomorphic encryption scheme[29] in which the secret key is distributed among the parties, and the corresponding collective public key pk is publicly known. Thus, each party can independently compute on ciphertexts encrypted under pk (the yellow lock in Figure 1), but all parties have to collaborate to decrypt a ciphertext. Hence, encrypted data cannot be decrypted, as long as one DP is honest and refuses to participate in the decryption. This multiparty scheme also enables DPs to collectively switch the encryption key of a ciphertext from pk to another public key, i.e., the querier's key (blue lock), without decrypting. We provide a list of recurrent symbols in Table 2 (in Supplementary). Mouchet et al.[29] propose a multiparty version of the Brakerski Fan-Vercauteren (BFV) lattice-based homomorphic cryptosystem[41] and introduce interactive protocols for key management and cryptographic operations. We rely directly on this multiparty scheme for the computation of survival curves, which involves only exact integers arithmetic, and we use an adaptation to the Cheon-Kim-Kim-Song cryptosystem (CKKS)[42] (described by Froelicher et al.[22]) that enables approximate arithmetic for the GWAS operations.

In summary, these cryptoschemes enable the encryption of vectors of N values in a single ciphertext, and any operation performed on the ciphertext is performed simultaneously on all the values. We rely on this *Single Instruction, Multiple Data* property and on edge computing, i.e., the computations are pushed to the data providers, to parallelize and efficiently perform computationally-intensive tasks.

Related Work

Centralized solutions for medical data sharing[8–10] require large amounts of data to be stored in a single repository that becomes a single point of failure and that has (often) to be fully trusted.

To alleviate this trust assumption, federated learning solutions[2,3,6] have been proposed. In these solutions, the data providers keep their data locally and share only aggregates or training model updates with a central server. However, multiple research contributions[12–14] have shown that these aggregates can still reveal significant information about the data providers' data. For example, Nasirigerdeh et al. proposed SPlink[3], a federated instantiation of the Plink[36] software to perform a GWAS. With SPlink the data providers' partial covariance matrices (i.e., intermediate result) are revealed to the server that aggregates these matrices in order to perform the models training. Although the original data X is not actually transferred, it can be inferred from the covariance matrix $X^T X$ computed by the aggregating server. In FAMHE, the covariance matrix is collectively and obliviously computed by exchanging encrypted data such that the models can be trained without revealing any intermediate data.

Similarly, secure multiparty solutions[17,18] rely on secret-sharing to compute on medical data without revealing intermediate or aggregate information. Cho et al.[18] designed a three-party secret-sharing-based solution to enable GWAS execution while not revealing information on the input data. Secret-sharing-based solutions require the data providers to communicate their data to a limited number of computing nodes, i.e., outside their premises. They also require an honest majority among the

computing nodes. FAMHE efficiently scales to federated learning settings where many DPs locally keep their data and can withstand up to S-1 out of S dishonest DPs.

Distributed solutions relying on homomorphic encryption[19–21,37] to enable federated analytics in a trust model similar to FAMHE were proposed. Some of these works assume a threat model more constraining than FAMHE, as they consider an active malicious adversary, but also exclusively focus on simple computations, e.g., counts and simple statistics. We build on the multiparty homomorphic encryption-based protocols proposed by Froelicher et al.[22] to propose a generic federated workflow for medical federated analytics and to show how the demanding GWAS computation can be performed through this workflow.

Differential-privacy-based solutions[15,16], in which the intermediate values are obfuscated by a specific amount of noise, assume a paradigm different than FAMHE, as privacy is traded off with accuracy. In fact, this obfuscation decreases the data and model utility. The training of accurate models requires high-privacy budgets but the achieved privacy level remains unclear[24]. In FAMHE, similarly to standard cleartext non-secure solutions (e.g., Plink[36]), the accuracy is traded for only the performance. We show in the Results Section that FAMHE achieves a accuracy similar to standard non-secure solutions, is able to scale to a high number of data providers and yields an acceptable execution time.

System & Threat Model

FAMHE supports a network of mutually distrustful medical institutions that act as data providers (DPs) and hold subjects' records. An authorized querier (see Figure 1) can run queries without threatening the data confidentiality and subjects' privacy. The DPs and the querier are assumed to follow the protocol and to provide correct inputs. All-but-one data providers can be dishonest, i.e., they can try to infer information about other data providers by using the protocol's outputs. We assume that the DPs are available during the complete execution of a computation. However, to account for unresponsive DPs, FAMHE can use a threshold-encryption scheme, where the DPs secret-share[43] their secret keys, thus enabling a subset of the DPs to perform the cryptographic interactive protocols.

Privacy-Preserving Pipeline for Survival Curves

As presented in the Results Section, in order to showcase FAMHE accuracy and performance, we reproduced an existing peer-reviewed study by executing it in a privacy-preserving and federated workflow. We detail here this secure workflow.

Survival curves are generally estimated with the Kaplan-Meier estimator[44]

$$\hat{S}(t) = \prod_{j,\, t_j \leq T} \left(1 - \frac{d_j}{n_j}\right), \tag{1}$$

where t_j is a time when at least one event happened, d_j is the number of events at time t_j, and n_j is the number of individuals known to have survived (or at risk) just before the time point t_j. We show in Figure 2a the exact replica of the survival curve presented by Samstein et al.[31] produced by our distributed and privacy-preserving computation. In this figure, each step down is the occurrence of an event. The ticks indicate the presence of censored patients, i.e., patients who withdrew from the study. The number of censored patients at time t_j is indicated by c_j. As shown in Figure 5 (in Supplementary), to compute this curve, each data provider i locally computes, encodes and encrypts a vector of the form $n_0^{(i)}, c_0^{(i)}, d_0^{(i)}, ..., n_T^{(i)}, c_T^{(i)}, d_T^{(i)}$ containing the values $n_j^{(i)}, c_j^{(i)}, d_j^{(i)}$ corresponding to each time point t_j for $t_j = 0, ..., T$. All the DPs' vectors are then collectively aggregated. The encryption of the final result is then collectively switched from the collective public key pk to the querier's public key, who can decrypt the result with its secret key and generate the curve following Eq. (1).

Privacy-Preserving Pipeline for GWAS

We briefly describe the genome-wide association study workflow before explaining how we perform it in a federated and privacy-preserving manner.

We consider a data set of p samples, i.e., patients. Each patient is described by f features or covariates (with indexes 1 to f). We list all recurrent symbols and acronyms in Table 2 (in Supplementary). Hence, we have a covariates matrix $X \in \mathbb{R}^{(p \times f)}$. Each patient also has a phenotype or label, i.e., $y \in \mathbb{R}^{(p \times 1)}$ and v variant values, i.e., one for each variant considered in the association test. The v variant values for all p patients form another matrix $V \in \mathbb{R}^{(p \times v)}$. To perform the GWAS, for each variant i, the matrix $X' = [\mathbf{1}, X, V[:,i]] \in \mathbb{R}^{(p \times (f+2))}$, i.e., the matrix X augmented by a column of 1s (intercept) and the column of one variant i, is constructed. The vector $w \in \mathbb{R}^{(f+2)}$ is then obtained by $w = (X'^T X')^{(-1)} X'^T y$. The p-value for variant i is then obtained with $p_{val} = 2 \cdot pnorm(-|\frac{w[f+2]}{\sqrt{(MSE(y,y') \cdot (X'^T X')^{(-1)}[f+2;f+2]}}|)$ where $pnorm$ is the cumulative distribution function (CDF) of the standard normal distribution, $w[f+2]$ is the weight corresponding to the variant, $MSE(y,y')$ is the mean squared error

obtained from the prediction y' computed with w and $(X'^T X')^{(-1)}[f+2; f+2]$ corresponds to the standard error of the variant weight.

Although this computation has to be performed for each variant i, we remark that X is common to all variants. We rely on the Shermann-Morrison formula[45] and the method presented in the report on cryptographic and privacy-preserving primitives (page 52) of the WITDOM European project[46] in order to compute $(X^T X)^{(-1)}$ only once before adjusting it for each variant and thus obtain $(X'^T X')^{(-1)}$. We describe this approach, i.e., FAMHE-GWAS, in Protocol 6a. Each data provider DP_i has a subset of p_i patients. For efficiency, the DPs are organized in a tree structure and one DP is chosen as the root of the tree DP_R. We remark that this does not have any security implication as any exchanged information is collectively encrypted. In a *Collective Aggregation (CA)*, each DP encrypts ($E()$) its local result with the collective key, aggregates its children DPs encrypted results with its encrypted local result and sends the sum to its parent DP such that DP_R obtains the encrypted result aggregated among all DPs. We recall here that with the homomorphic-encryption scheme used, vectors of values can be encrypted in one ciphertext and that any operation performed on a ciphertext is simultaneously performed on all vector elements, i.e., *Single Instruction, Multiple Data (SIMD)*. We rely on this property to parallelize the operations at multiple levels: among the DPs, among the threads in each DP and among the values in the ciphertexts. For a 128-bit security level, the computations are performed simultaneously for 512 variants with FAMHE-GWAS and for 8192 with FAMHE-FastGWAS.

We rely on the Gauss-Jordan (GJ) method[47] to compute the inverse of the encrypted covariance matrix. We chose this algorithm as it requires only row operations, which can be efficiently performed with SIMD. The only operation that is not directly applicable in HE is the division that we approximated with a Chebyshev polynomial. Note that we avoid any other division in the protocol by pushing them to the last step that is executed by the querier Q after decryption. In Protocol 6a (in Supplementary), we keep $1/c$ until decryption.

In Protocol 6b (in Supplementary), we describe how we further reduce the computation overhead by obtaining the covariates' weights w' with a lightweight federated gradient descent (FGD), by reporting the obtained covariates' contributions in the phenotype y, that becomes y''; to compute the p-value, we then compute only one element of the covariance inverse matrix $(X'^T X')^{(-1)}[f+2; f+2]$, instead of the entire inverse. We follow the method described by Froelicher et al.[22] to perform the federated gradient descent, without disclosing any intermediate values.

We describe in Figure 7 (in Supplementary) how the (main) values used in both protocols are packed to optimize the communication and the amount of required operations (multiplications, rotations). We perform permutations, duplications, and rotations on cleartext data that are held by the DPs (indicated in orange in Figure 7); and we avoid, as much as possible, the operations on encrypted vectors. Note that rotations on ciphertexts are almost two orders of magnitude slower than multiplications or additions and should be avoided when possible. As ciphertexts have to be aggregated among DPs, a tradeoff has to be found between computation cost (e.g., rotations) and data packing, as a smaller packing density would require the exchange of more ciphertexts.

In both Protocols, all operations for v variants are executed in parallel, due to the ciphertext packing (SIMD). These operations are further parallelized due to multi-threading and to the distribution of the workload among the DPs. We highlight (in bold) the main steps and aggregated values in the protocol and note that DPs' local data are in cleartext, whereas all exchanged data are collectively encrypted ($E()$).

Experimental Settings

We implemented our solution by building on top of Lattigo[33], an open-source Go library for lattice-based cryptography, and Onet, an open-source Go library for building decentralized systems. The communication between data providers (DPs) is done through TCP, with secure channels (using TLS). We evaluate our prototype on an emulated realistic network, with a bandwidth of 1 Gbps and a delay of 20 ms between every two nodes. We deploy our solution on 12 Linux machines with Intel Xeon E5-2680 v3 CPUs running at 2.5GHz with 24 threads on 12 cores and 256 Gigabytes of RAM, on which we evenly distribute the DPs. We choose security parameters to always achieve a security level of 128 bits.

Differentially Private Mechanism

Differential privacy is a privacy-preserving approach for reporting results on statistical datasets, introduced by Dwork[23]. This approach guarantees that a given randomized statistic, $\mathscr{M}(DS) = R$, computed on a dataset DS, behaves similarly when computed on a neighbor dataset DS' that differs from DS in exactly one element. More formally, (ε, δ)-differential privacy[48] is defined by $\Pr[\mathscr{M}(DS) = R] \leq \exp(\varepsilon) \cdot \Pr[\mathscr{M}(DS') = R] + \delta$, where ε and δ are privacy parameters: the closer to 0 they are, the higher the privacy level is. (ε, δ)-differential privacy is often achieved by adding noise to the output of a function $f(DS)$. This noise can be drawn from the Laplace distribution with mean 0 and scale $\frac{\Delta f}{\varepsilon}$, where Δf, the sensitivity of the original real valued function f, is defined by $\Delta f = \max_{D,D'} ||f(DS) - f(DS')||_1$. Other mechanisms, e.g., relying on a Gaussian distribution, have also been proposed[23,49].

As explained before, FAMHE can enable the participants to agree on a privacy level by choosing whether to yield exact or obfuscated, i.e., differentially private results, to the querier. We also note that our solution would then enable the obfuscation of only the final result, i.e., the noise can be added before the final decryption, and all the previous steps can be executed with exact values as no intermediate value is decrypted. This is a notable improvement with respect to existing federated-learning solutions based on differential privacy[39] in which the noise has to be added by each data provider at each iteration of the training. In the solution by Kim et al.[39], each data provider perturbs its locally computed gradient such that the aggregated perturbation, obtained when the data providers aggregate (combine) their locally updated model, is ε-differentially private. This is achieved by having each data provider generating and adding a partial noise such that, when aggregated, the total noise follows the Laplace distribution. The noise magnitude is determined by the sensitivity of the computed function and this sensitivity is similar for each DP output and for the aggregated final result. This means that, as the intermediate values remain encrypted in FAMHE, a noise with the same magnitude can be added only once on the final result, thus ensuring the same level of privacy with a lower distortion of the result.

References

1. The EU General Data Protection Regulation. https://eugdpr.org/, (10.01.2021).
2. Sheller, M. J. *et al.* Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Sci. reports* **10**, 1–12 (2020).
3. Nasirigerdeh, R. *et al.* sPLINK: A Federated, Privacy-Preserving Tool as a Robust Alternative to Meta-Analysis in Genome-Wide Association Studies. *BioRxiv* (2020).
4. Zhu, L. & Han, S. Deep leakage from gradients. In *Federated Learning*, 17–31 (Springer, 2020).
5. Melis, L., Song, C., De Cristofaro, E. & Shmatikov, V. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, 691–706 (IEEE, 2019).
6. Gaye, A. *et al.* DataSHIELD: taking the analysis to the data, not the data to the analysis. *Int. journal epidemiology* **43**, 1929–1944 (2014).
7. priVAcy preserviNg federaTed leArninG infrastructurE for Secure Insight eXchange. https://distributedlearning.ai/, (11.01.2021).
8. All of Us Research Program, NIH. https://allofus.nih.gov/, 30.01.2021.
9. Genomics England. https://www.genomicsengland.co.uk/, 30.01.2021.
10. UK Biobank. https://www.ukbiobank.ac.uk/, 30.01.2021.
11. Scheibner, J. *et al.* Revolutionizing Medical Data Sharing Using Advanced Privacy Enhancing Technologies: Technical, Legal and Ethical Synthesis. *J Med Internet Res (in press). doi:10.2196/25120* (2021).
12. Wang, Z. *et al.* Beyond Inferring Class Representatives: User-Level Privacy Leakage From Federated Learning. In *IEEE INFOCOM* (2019).
13. Melis, L., Song, C., De Cristofaro, E. & Shmatikov, V. Exploiting Unintended Feature Leakage in Collaborative Learning. In *2019 IEEE Symposium on Security and Privacy (SP)* (2019).
14. Nasr, M., Shokri, R. & Houmansadr, A. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In *IEEE S&P* (2019).
15. Bonomi, L., Jiang, X. & Ohno-Machado, L. Protecting patient privacy in survival analyses. *J. Am. Med. Informatics Assoc.* **27**, 366–375 (2020).
16. Li, W. *et al.* Privacy-Preserving Federated Brain Tumour Segmentation. In *MLMI* (2019).
17. Jagadeesh, K. A., Wu, D. J., Birgmeier, J. A., Boneh, D. & Bejerano, G. Deriving genomic diagnoses without revealing patient genomes. *Science* **357**, 692–695 (2017).
18. Cho, H., Wu, D. J. & Berger, B. Secure Genome-Wide Association Analysis using Multiparty Computation. *Nat. biotechnology* **36**, 547–551 (2018).
19. Froelicher, D. *et al.* Unlynx: A Decentralized System for Privacy-Conscious Data Sharing. *PETS* (2017).
20. Raisaro, J. L. *et al.* Medco: Enabling Secure and Privacy-Preserving Exploration of Distributed Clinical and Genomic Data. *IEEE/ACM Transactions on Comput. Biol. Bioinforma.* (2018).
21. Froelicher, D., Troncoso-Pastoriza, J. R., Sousa, J. S. & Hubaux, J. Drynx: Decentralized, Secure, Verifiable System for Statistical Queries and Machine Learning on Distributed Datasets. *IEEE TIFS* DOI: 10.1109/TIFS.2020.2976612 (2020).

22. Froelicher, D. *et al.* Scalable Privacy-Preserving Distributed Learning. *PETS* (2021).

23. Dwork, C., Roth, A. *et al.* The algorithmic foundations of differential privacy. *Foundations Trends Theor. Comput. Sci.* **9**, 211–407 (2014).

24. Jayaraman, B. & Evans, D. Evaluating Differentially Private Machine Learning in Practice. In *USENIX Security* (2019).

25. Aono, Y., Hayashi, T., Trieu Phong, L. & Wang, L. Scalable and secure logistic regression via homomorphic encryption. In *ACM CODASPY* (2016).

26. Bonte, C. & Vercauteren, F. Privacy-preserving logistic regression training. *BMC medical genomics* (2018).

27. Kim, M., Song, Y., Wang, S., Xia, Y. & Jiang, X. Secure logistic regression based on homomorphic encryption: Design and evaluation. *JMIR medical informatics* (2018).

28. Raisaro, J. *et al.* SCOR: A secure international informatics infrastructure to investigate COVID-19. *J. Am. Med. Info. Assoc.* (2020).

29. Mouchet, C., Troncoso-pastoriza, J. R., Bossuat, J.-P. & Hubaux, J. P. Multiparty Homomorphic Encryption: From Theory to Practice. In *Tech. Report https://eprint.iacr.org/2020/304* (2019).

30. iDash Competition. http://www.humangenomeprivacy.org/2020/, (11.01.2021).

31. Samstein, R. M. *et al.* Tumor mutational load predicts survival after immunotherapy across multiple cancer types. *Nat. genetics* **51**, 202–206 (2019).

32. McLaren, P. J. *et al.* Polymorphisms of large effect explain the majority of the host genetic contribution to variation of HIV-1 virus load. *Proc. Natl. Acad. Sci.* **112**, 14658–14663 (2015).

33. Lattigo: A library for lattice-based homomorphic encryption in go. https://github.com/ldsec/lattigo, 10.01.2021.

34. Tierney, J. F., Stewart, L. A., Ghersi, D., Burdett, S. & Sydes, M. R. Practical methods for incorporating summary time-to-event data into meta-analysis. *Trials* **8**, 16 (2007).

35. Laird, N. & Olivier, D. Covariance analysis of censored survival data using log-linear analysis techniques. *J. Am. Stat. Assoc.* **76**, 231–240 (1981).

36. Plink Software. https://www.cog-genomics.org/plink/, 30.11.2020.

37. Lu, Y., Zhou, T., Tian, Y., Zhu, S. & Li, J. Web-Based Privacy-Preserving Multicenter Medical Data Analysis Tools Via Threshold Homomorphic Encryption: Design and Development Study. *J. medical Internet research* **22**, e22555 (2020).

38. Data Sharing Network (SHRINE). https://www.i2b2.org/work/shrine.html, (11.01.2021).

39. Kim, M., Lee, J., Ohno-Machado, L. & Jiang, X. Secure and Differentially Private Logistic Regression for Horizontally Distributed Data. *IEEE Transactions on Inf. Forensics Secur.* **15**, 695–710 (2020).

40. Medco Software. https://medco.epfl.ch/, 10.01.2021.

41. Fan, J. & Vercauteren, F. Somewhat Practical Fully Homomorphic Encryption. *IACR Cryptol. ePrint Arch.* (2012).

42. Cheon, J. H., Kim, A., Kim, M. & Song, Y. Homomorphic encryption for arithmetic of approximate numbers. In *ASIACRYPT* (2017).

43. Shamir, A. How to share a secret. *Commun. ACM* (1979).

44. Manish Kumar Goel, P. K. & Kishore, J. Understanding survival analysis: Kaplan-Meier estimate. (2010).

45. Sherman, J. & Morrison, W. J. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals Math. Stat.* **21**, 124–127 (1950).

46. WITDOM: empoWering prIvacy and securiTy in non-trusteD envirOnMents. https://cordis.europa.eu/project/id/644371/results, 30.01.2021.

47. Atkinson, K. E. *An introduction to numerical analysis* (John wiley & sons, 2008).

48. Dwork, C., McSherry, F., Nissim, K. & Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, 265–284 (Springer, 2006).

49. Ghosh, A., Roughgarden, T. & Sundararajan, M. Universally utility-maximizing privacy mechanisms. *SIAM J. on Comput.* **41**, 1673–1693 (2012).

Acknowledgements

We would like to thank Apostolos Pyrgelis for reviewing the paper. This work was partially supported by the grant #2017-201 of the Strategic Focal Area "Personalized Health and Related Technologies (PHRT)" of the ETH Domain.

Author contributions statement

J.R.T.-P, J.L.R. and J.P.H. conceived the study. D.F. and J.R.T.-P. developed the methods. D.F. and J.S.S. implemented the software and performed experiments, the results of which were validated by M.C. and J.F. All authors wrote the manuscript.

Data availability

We replicated two existing medical studies, i.e., Samstein et al.[31] and McLaren et al.[32], and refer the reader to these works to obtain the related datasets.

Additional information

The authors declare no competing financial interests.