Privacy-aware Artificial Intelligence in Systems Medicine

Kumulative Dissertation

eingereicht an der
Fakultät für Mathematik, Informatik und Naturwissenschaften
der Universität Hamburg

zur
Erlangung des akademischen Grades

Doktor der Naturwissenschaften (Dr. rer. nat.)

vorgelegt beim Promotionsausschuss Informatik im Jahr 2023 von

M.Sc.
Julian Oskar Matschinske

geboren in Fürth

Tag der Disputation:	14.02.2024
Mitglieder der Prüfungskommission:	
Vorsitzender:	Prof. Dr. Matthias Rarey
Stv. Vorsitzender:	Prof. Dr. Stefan Kurtz
1. Gutachter:	Prof. Dr. Jan Baumbach
2. Gutachter:	Prof. Dr. Richard Röttger
3. Gutachter:	Prof. Dr. Gunnar W. Klau

Abstract

Bioinformatics is grappling with an explosion of data, creating both opportunities and challenges for scientific discovery and healthcare. This thesis stands at the crossroads of systems medicine and privacy-aware artificial intelligence (AI), offering contributions that aim to harness the potential of this data-rich landscape. Central to the thesis are the web tools CoVex, sPLINK, FeatureCloud, and AIMe, each designed to address unique challenges, publicly and freely available to the research community.

Within the ambit of systems medicine, CoVex emerges as a tool in the realm of infectious diseases and drug repurposing. Deploying network exploration and ranking algorithms like centrality measures, CoVex identifies intricate disease pathways and potential drug targets. Its purpose mainly lies in drug repurposing achieved by its capability to explore integrated virus-protein, protein-protein and protein-drug interaction networks to identify alternative applications for existing drugs, thereby accelerating the medical response to urgent challenges like the COVID-19 pandemic.

Privacy-aware AI is the second major pillar of the thesis, with a focus on federated learning (FL) as an enabling technology. The tools sPLINK and FeatureCloud are introduced to demonstrate this approach. sPLINK, specialized for genome-wide association studies (GWAS), preserves data privacy without compromising analytical robustness. FeatureCloud expands upon this by serving as a versatile, FL platform, thereby facilitating large-scale analyses across multiple institutions while adhering to stringent data privacy norms. It employs and integrates state-of-the-art privacy-enhancing techniques (PETs), such as differential privacy (DP) and secure multiparty computation (SMPC), to protect sensitive patient data. Evaluation of FeatureCloud shows that the results are sufficiently close or even identical to centrally performed analyses, thereby demonstrating the efficacy and applicability of FL in a cross-silo context.

The thesis also brings forth the AIMe registry, aiming to create a foundation for transparency, reproducibility, and reliability in biomedical AI. By setting standards and ensuring correct and complete reporting, AIMe acts as a central hub for vetting and disseminating AI tools, increasing validation and reproducibility of results reported in biomedical research.

As we traverse an era defined by rapid data proliferation and stringent data protection laws, this thesis demonstrates that specialized tools and versatile platforms are valuable additions to the research landscape. CoVex, sPLINK, AIMe and FeatureCloud each have unique specializations, yet they all contribute to more efficient research in systems medicine: making integrated data quickly accessible to researchers, allowing large-scale analyses across distributed datasets, and ensuring valid and reproducible reporting of results.

Kurzfassung

Die Bioinformatik sieht sich einer Explosion von Daten gegenübergestellt, die neue Möglichkeiten aber auch Herausforderungen für wissenschaftliche Erkenntnisse und die Gesundheitsversorgung mit sich bringt. Diese Dissertation positioniert sich im Schnittbereich zwischen Systemmedizin und datenschutzbewusster künstlicher Intelligenz (KI) und erbringt Beiträge, die das Potential dieser datenreichen Umgebung erschließen sollen. Im Mittelpunkt stehen die Web-Tools CoVex, sPLINK, FeatureCloud und AIMe, die entwickelt wurden, um damit verbundene Herausforderungen zu bewältigen und die der Forschungsgemeinschaft öffentlich und frei zur Verfügung stehen.

Im Rahmen der Systemmedizin tritt CoVex als Werkzeug im Bereich der Infektionskrankheiten und der Medikamentenumwidmung in Erscheinung. Durch den Einsatz von Netzwerk-Erkundungs- und Ranking-Algorithmen, wie Zentralitätsmaßen, identifiziert CoVex komplexe Krankheitspfade und potenzielle neue Zielproteine für Medikamente. Der Hauptzweck liegt in der Medikamentenumwidmung, die mittels integrierter Virus-Protein-, Protein-Protein- und Protein-Medikament-Interaktionsnetzwerke alternative Anwendungen für bestehende Medikamente erkundet. Damit soll die Reaktion auf dringende Herausforderungen wie die COVID-19-Pandemie beschleunigt werden.

Datenschutzbewusste KI ist der zweite Hauptteil der Dissertation, wobei das föderierte Lernen (FL) als wesentliche Technologie im Fokus steht. Die Tools sPLINK und FeatureCloud werden vorgestellt, um diesen Ansatz zu demonstrieren. sPLINK, spezialisiert auf genomweite Assoziationsstudien (engl. GWAS), gewährleistet die Datensicherheit ohne Kompromisse bei der analytischen Robustheit einzugehen. FeatureCloud erweitert dieses Konzept, indem es als vielseitige Plattform für FL dient und so großangelegte Analysen über mehrere Institutionen hinweg ermöglicht, während es strengen Datenschutznormen gerecht wird. Es verwendet und integriert moderne Techniken zur Verbesserung der Privatsphäre, wie Differential-Privacy (DP) und Secure-Multiparty-Computation (SMPC), um sensible Patientendaten zu schützen. Die Evaluation von FeatureCloud zeigt, dass die Ergebnisse ausreichend nah oder sogar identisch mit zentral durchgeführten Analysen sind, wodurch die Wirksamkeit und Anwendbarkeit von FL im übergreifenden Kontext bestätigt wird.

Die Dissertation stellt auch das AIMe-Register vor, welches eine Grundlage für Transparenz, Reproduzierbarkeit und Zuverlässigkeit für biomedizinische KI schaffen soll. Durch das Definieren von Standards und die Sicherstellung korrekter und vollständiger Berichte fungiert AIMe als zentrale Datenbank zur Überprüfung und Veröffentlichung von KI-Tools, wodurch die Validierung und Reproduzierbarkeit von Ergebnissen in der biomedizinischen Forschung erhöht wird.

In einer Ära, die durch schnelle und fortschreitende Datenerzeugung und strenge Datenschutzgesetze geprägt ist, zeigt diese Dissertation, dass spezialisierte Tools und vielseitige Plattformen sinnvolle Ergänzungen in der Forschungslandschaft darstellen. CoVex, sPLINK, AIMe und FeatureCloud haben jeweils ihren eigenen Schwerpunkt, tragen jedoch alle dazu bei, die Forschung in der Systemmedizin effizienter zu gestalten: Sie machen integrierte Daten schnell für Forscherinnen und Forscher zugänglich, ermöglichen großangelegte Analysen über verteilte Datensätze hinweg und gewährleisten eine valide und reproduzierbare Dokumentation dieser Ergebnisse und ihrer Entstehung.

Danksagung

Zuallererst möchte ich meinem Doktorvater, Prof. Jan Baumbach, meinen tiefen Dank aussprechen. Er stand mir stets mit Rat und Tat zur Seite und hat mir mit seinem Lehrstuhl die idealen Bedingungen für meine Forschung geschaffen.

Ein herzlicher Dank gilt auch meinen Kolleginnen und Kollegen, die ich als stets hilfsbereit und freundlich in Erinnerung behalten werde. Sie waren maßgeblich für den inspirierenden und kreativen Austausch verantwortlich, der meine Arbeit bereichert hat. Insbesondere möchte ich Julian Späth hervorheben, mit dem ich eng zusammengearbeitet habe, und der mit seiner Hilfsbereitschaft und Zuverlässigkeit eine große Hilfe war.

Des Weiteren danke ich den Sekretärinnen, Science Managern und technischen Mitarbeitern, die immer zur Stelle waren, wenn Unterstützung benötigt wurde, sowie den Mitgliedern der Prüfungskommission und den Gutachtern für ihre Zeit und Bereitschaft.

Schließlich gilt mein tiefster Dank meinen Eltern. Ihre unermüdliche Unterstützung hat es mir ermöglicht, mich mit voller Energie meiner Forschung zu widmen. Ohne ihre Hilfe und Unterstützung wäre mein Weg weitaus herausfordernder gewesen.

Contents

List of Figures

List of Acronyms and Abbreviations

AI	Artificial intelligence
ATC	Anatomical therapeutic chemical
CICD	Continuous integration and continuous delivery
DEG	Differentially expressed gene
DL	Deep learning
DP	Differential privacy
EMA	European Medicines Agency
FDA	Food and Drug Administration
FL	Federated learning
GDPR	General Data Protection Regulation
GWAS	Genome-wide association study
GUI	Graphical user interface
HE	Homomorphic encryption
HPID	Host-Pathogen Interaction Database
HTTPS	Hypertext transfer protocol secure
IID	Independent and identically distributed data
ILPD	Indian Liver Patient Dataset
LR	Linear regression
ML	Machine learning
MSE	Mean squared error
PET	Privacy-enhancing technique
PPI	Protein-protein interaction
RF	Random forest
SDK	Software development kit
SMPC	Secure multi-party computation
SVM	Support vector machine

1 Introduction

As in many areas of computer science, bioinformatics is a strongly data-driven discipline at the intersection of biomedicine and data science. The accumulated data keeps growing in terms of quantity, quality and coverage of captured biomedical processes, opening up new opportunities and challenges alike [123].

Until today, medicine mostly divides diseases by the organs affected (e.g., heart diseases), symptoms (e.g., hypertension) or even the person who discovered the disease (e.g., Alzheimer's). This usually reflects a lack of deeper understanding of the disease since it only conveys a superficial view [120] (see Figure 1.1). Large-scale data analysis, especially through machine learning, bridges this gap by sifting through vast amounts of data, identifying patterns, and drawing correlations that may not be evident to traditional methods. It thereby supports the systems medicine approach by enabling the identification of intricate disease mechanisms and complex interactions in the human body.

In this thesis, methods from the area of machine learning are investigated and applied, and made available to other researchers via collaborative tools, putting a special focus on federated learning, a technique that allows for training machine learning models on large amounts of distributed data while maintaining data privacy. These methods do not start at the symptoms in a 'top down' approach, but at biomedical data of various forms, capturing a piece of the complex processes happening in the human organism with little assumptions about the disease mechanism, following the data-driven philosophy of systems medicine.

1.1 Topics

This thesis revolves around three interconnected themes - systems medicine, artificial intelligence, and data privacy. Each of these areas offers unique perspectives and challenges that merge at the intersection of modern scientific research and technological advancements in bioinformatics. The main goal of the thesis is to make useful tools available to researchers that enable them to use technologies and methods in these areas.

The following sections briefly define the most important topics and put them into context.

1.1.1 Systems medicine

Systems medicine provides novel perspectives on disease mechanisms by looking at the whole human organism. It examines complex interactions and mechanisms in the human body within its environment, providing a deeper understanding of the root causes of a disease [119, 15].

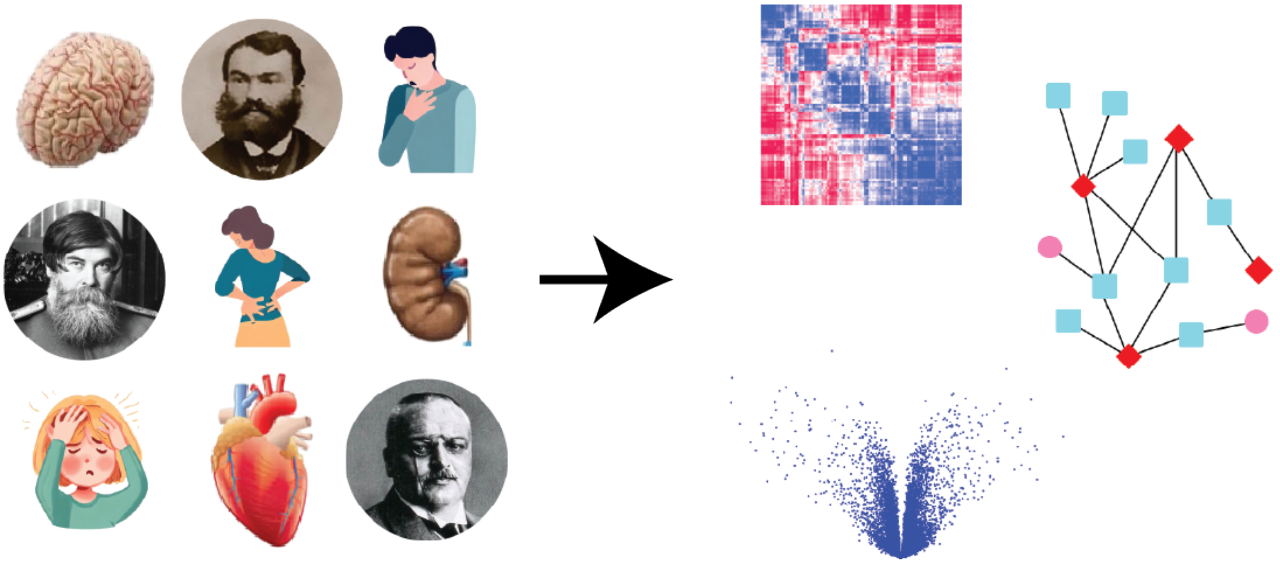


Figure 1.1: Diseases currently are often named after organs (e.g., brain, heart, liver), symptoms (e.g., headache, backache, whooping cough) or doctors who first investigated them (e.g., Bekhterev's, Parkinson's, Alzheimer's). Systems medicine looks at the underlying disease mechanisms involving omics data (e.g., genes, proteins, metabolites interactions) instead of symptoms or affected body parts to provide holistic explanations and treatments.

In the last decades, acquisition of data has become cheaper and their variety has grown hugely [150]. The available data is often difficult to interpret and of enormous size, which poses a challenge to understand it and make use of it. At the same time, it can be assumed with a high degree of plausibility that the acquired data potentially provides valuable insights into disease mechanisms and other processes in a living organism [207]. The task of making sense of this data, putting together these pieces with the help of machine learning and other methods, and obtaining a better understanding of these processes is a challenging endeavor, but can provide insights that go beyond the superficial and symptomatic level.

Systems medicine contributes to many areas in bioinformatics and medicine, including drug repurposing and novel treatments, but also by revealing knowledge that supports other areas of medicine. The heterogeneous nature of the data can be a valuable source, but also makes it difficult to work with in some cases [207].

1.1.2 Drug repurposing

Drugs are usually developed as a treatment for a particular disease and then admitted for that disease by the respective authorities, e.g., the Food and Drug Administration (FDA) in the US [191] or European Medicines Agency (EMA) in the EU [73]. Finding new drugs is a lengthy process and having a newly discovered drug admitted can take even longer [48, 102].

From an administrative point of view, these drugs are then approved treatments for a particular disease. But from a biomedical point of view, these drugs are merely substances having a certain impact on the processes in the human organism. It is therefore very much possible that a drug originally approved for disease A can be effective for disease B as well (or even more so), even if the disease appears very different on the symptomatic level (see Section 1.4.1).

Finding and applying such drugs, originally developed for another disease or purpose, is hence referred to as drug repurposing. Both systems medicine, and machine learning can be of great help to identify potential drug candidates as a first step in the process of drug repurposing [151, 192]. See Section 2.2 more information about the connection of drug repurposing to systems medicine.

1.1.3 Artificial intelligence

As introduced before, biomedicine today generates all sorts of data with of different size, shape, quality and biomedical significance [137]. In many cases, we do not understand (yet) whether and, if so, how it relates to phenotypical phenomena we are interested in, like diseases [165]. The sheer quantity of data renders efforts to manually examine the data futile in many cases [112].

Artificial intelligence and machine learning are techniques which help analyze these large quantities of data and which have seen impressive advances in the last decade [107]. It also has proven highly effective in the biomedical field due to its versatility and adaptability, allowing it to be applied to many types of data that can be accumulated in biomedicine [65]. This includes omics data, medical images, electronic health records and many more. Generally, the performance of a machine learning method, i.e. quality of its predictions, improves when the amount of training data grows, provided it is not redundant and of high quality.

There is often confusion between the terms *artificial intelligence (AI)* and *machine learning (ML)* and they are used interchangeably in many cases. For now, we define AI as a superset of ML. A more detailed definition can be found in Section 3.1.

1.1.4 Data privacy

Privacy is a big concern when it comes to storing and processing medical data, particularly patient data [96]. While the information contained in it can be of high interest to researchers, it can also be abused by health insurance companies or employers and lead to discrimination of patients with allegedly higher risks of illness.

While it is often hard to tell what information is contained in biomedical data, this very uncertainty must cause us to treat it as carefully as possible. From the perspective of research, there is a big interest in maintaining public trust in how data is treated, i.e., how it is processed and stored, to keep getting access to it [76].

Privacy awareness relates to the enduring effort to maintain patients' privacy wherever possible. A stronger version would be privacy preservation, which establishes the promise to not reveal anything that could possibly identify a patient's identity. This promise is hard to maintain in a provable sense and even more so in a practical sense. There are such techniques, however, which are introduced in Chapter 3. However, they usually lead to worse results through added noise or complicate the technical implementation. Balancing prediction quality and privacy is therefore a non-trivial task and needs to be considered at each step when dealing with sensitive patient data.

1.2 Motivation

The global research community garners significant benefits from tools and software contributed by fellow researchers, which provide novel perspectives and insights into varied scientific fields [155]. Particularly, web tools stand out due to their easily accessible nature and versatility of usage in diverse contexts. These tools serve as a bridge connecting and amalgamating the advancements made across different areas, especially within the biomedical domain, thus fostering interdisciplinary collaborations. By offering researchers the capability to swiftly access information, these tools enable a seamless flow of knowledge and data beyond geographical and disciplinary boundaries.

In many instances, such collaboration is perceived as a dynamic extension of traditional scientific publications. This is largely due to the fact that these web tools offer the advantage of an interactive graphical user interface and visualization capabilities that enrich the overall research experience. In tandem, these tools are often built upon complex algorithms operating on the backend or server side, thereby bringing a level of sophistication to the process. Additionally, programming libraries and software development kits (SDKs) fall into this bracket, as they can be utilized by other researchers to enrich their work and establish a robust foundation for future exploration [16].

The field of systems medicine emphasizes the need to integrate as many potential data sources as possible [41]. Consequently, collaborative online tools and freely available software systems align perfectly with this approach, facilitating the reflection of this principle in active research. These resources pave the way for enhanced integration and knowledge sharing across different domains, thus fostering a culture of collective growth within the scientific community.

The outbreak of the Coronavirus pandemic in 2020 has highlighted, more than ever, the importance of rapid response measures in tackling highly transmissible diseases [148, 91]. Until effective therapeutics or vaccines are discovered, measures like lockdowns remain the only immediate solution to reduce the reproduction rate of the virus. However, these measures significantly affect social life and the economy [182]. Therefore, the concept of drug repurposing emerges as a promising solution, with its potential to shorten approval times and speed up the journey from laboratory to clinic [157].

In circumstances where time is of paramount importance, such as during a pandemic, the availability of useful, timely data is vital to understand a disease better and potentially discover a cure. However, some types of data are highly sensitive and are not freely shareable within the research community. This is particularly true for patient data, which is stringently protected by legislation in most regions globally [197]. The General Data Protection Regulation (GDPR) imposed by the European Union is a prime example of such legislation that sets strict data protection requirements, potentially hampering data sharing without comprehensive legal preparation.

In an attempt to reconcile the conflicting objectives of data protection and accelerating research progress, the concept of federated learning has been introduced [108]. This novel approach ensures data protection by leaving the data at the point of collection, such as a hospital, while sharing only generalized information about a disease mechanism with other researchers. This innovative strategy strikes a balance between safeguarding patient privacy and fostering scientific discovery, without sacrificing either [164].

However, federated learning presents significant challenges compared to conventional machine learning methods. Instead of executing a machine learning algorithm on a single computer with a single dataset, federated learning requires a network of interconnected computers. Given these additional complexities, a general framework that aims to streamline the development, dissemination, and execution of federated algorithms could be a game-changer [101].

Driven by these considerations, there is an ever-growing demand for public and user-friendly tools, such as those discussed in this thesis. These tools tap into the immense potential of machine learning and systems medicine to improve our approach towards scientific research and discovery.

1.3 Contents of the thesis

Two related areas are investigated in this thesis: systems medicine, with a focus on data integration, and privacy in medicine related to machine learning and artificial intelligence, using federated learning. It aims to outline the interconnection of the topics systems medicine, artificial intelligence and privacy awareness and demonstrates solutions in the form of freely available online tools that can be used by the research community to help harvest their potential.

To reconcile privacy-awareness and research interests, a federated learning platform called FeatureCloud has been developed and is introduced in the second part.

1.3.1 Structure

In the first chapter, the main topics are briefly introduced and the motivation for collaborative tools in their domains is outlined. At the end of the chapter, related work can be found which provides the scientific context of the thesis.

The topics are described further in the two main chapters about systems medicine and privacy-aware AI, providing more details about the respective fields.

Chapter 2 is dedicated to systems medicine, with a special focus on drug repurposing. It provides an overview of network-based approaches to first find indirect drug targets to subsequently identify potential drugs, involving the required data and the employed network algorithms. At the end of the chapter, the main result, CoVex is shown, an interactive interactome explorer for drug repurposing with a focus on Covid-19.

Chapter 3 first defines the terms used for machine learning in the remainder of the chapter. Then, several examples of ML models are introduced and their respective strengths and weaknesses are discussed. The main focus of the chapter lies on federated learning and privacy preserving techniques, which are presented and related to each other. The chapter concludes with the results related to these topics, mainly the universal FeatureCloud platform, sPLINK for genome-wise association studies and the AIMe registry for thorough reporting on ML methods.

Chapter 4 discusses the results, again divided into a section about systems medicine and privacy-preserving AI. It highlights the contributions and also mentions the shortcomings of the developed tools and platforms.

Chapter 5 provides potential extensions and grounds for further research in the area of network-based drug repurposing and federated learning techniques.

1.3.2 Software and tools

Various tools and software systems were implemented in the context of this thesis, which implement systems medicine approaches and federated machine learning and make them available to the research community in the manner described. The following paragraphs briefly introduce 4 of them and highlight their purpose.

CoVex, the coronavirus explorer is an interactive resource and exploration tool, able of showing the early available data about virus-host interactions [1]. It integrates various data sources and allows for a systems medicinal approach to drug repurposing by suggesting drug targets further down the disease pathway. Ultimately, it outputs a list of drugs, ranked by graph algorithms according to their position in the protein-protein-drug interaction network. The pandemic caused by SARS-CoV-2 (also known as 'Coronavirus') is the primary use case for this kind of systems medicine-based drug repurposing, highlighting the necessity of collaborative approaches and quickly available tools in such a situation, but other viruses could be explored like this as well.

sPLINK is a federated tool for genome-wide association studies (GWAS) [2]. It combines the advantages of meta studies, whose results become worse with imbalances data, and centralized GWAS, where data sharing is necessary and data privacy cannot be maintained. It served as a proof-of-concept for federated machine learning and implement one useful algorithm for a federated setting.

AIMe, a registry for artificial intelligence in biomedical research is tackling the problem of incomplete, misleading or incorrect reporting on machine learning methods in scientific publications [3].

[1] Website: https://exbio.wzw.tum.de/covex/

[2] Website: https://exbio.wzw.tum.de/splink/

[3] Website: https://aime-registry.org/

FeatureCloud is a platform which provides the necessary infrastructure for federated machine learning [4]. It hosts federated applications ('apps') on a public registry from where they can be used by researchers to conduct a collaborative machine learning project. It also offers help during development of such apps to third-party developers, who can contribute their own AI algorithms to a public registry ('AI Store').

Figure 1.2 shows how the standalone tools CoVex and sPLINK, and the FeatureCloud platform relate to each other.

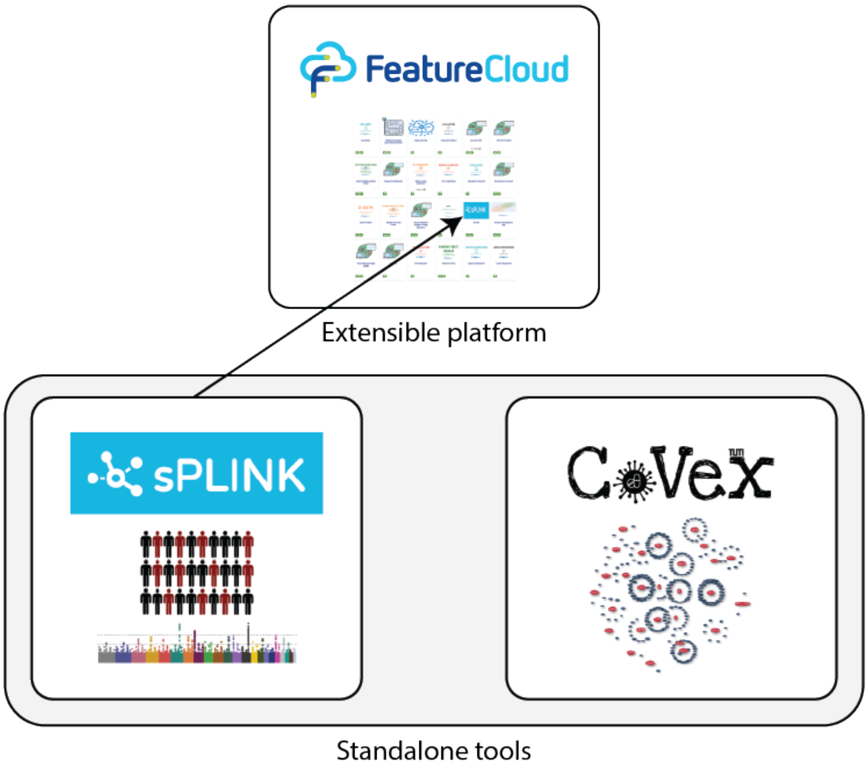


Figure 1.2: CoVex and sPLINK are standalone tools with their own website and interface. FeatureCloud is a platform which is extensible by third-party developers through apps, into which the sPLINK implementation has been integrated.

1.4 Related work

This section provides an overview of the related work in the context of systems medicine, with a special focus on drug repurposing, and privacy-aware artificial intelligence.

1.4.1 Systems medicine

The concept of drug repurposing has gained significant attention in recent years due to its potential to accelerate the drug development process by reducing associated costs and timelines [157]. This approach involves identifying novel therapeutic indications for existing drugs, thereby cutting down the time-intensive and costly process of de novo drug development. Multiple computational and experimental approaches have been deployed to achieve this, including drug-target interaction prediction, network-based methodologies, and a range of data mining techniques, all leveraging massive biomedical and pharmacological data repositories [50].

Notable examples of successful drug repurposing include sildenafil and bupropion. Sildenafil was originally developed to treat angina but later found a robust market as a treatment for erectile dysfunction [67, 63]. Bupropion, initially marketed as an antidepressant, has since been repurposed as a smoking cessation aid [93], demonstrating efficacy in multiple clinical trials [98]. The success of these and other cases has

[4] Website: https://featurecloud.ai/

spurred further investigation into drug repurposing, highlighting its viability as a strategic approach in drug discovery and development.

Systems medicine, which involves the integration of computational modeling with clinical and biological data to provide comprehensive insights into disease pathways and drug actions, has been increasingly applied to drug repurposing [88]. In the context of drug repurposing, a variety of network-based methods have been employed [12], such as drug-target networks, protein-protein interaction networks, and gene-disease networks, to unveil potential new indications for existing drugs [36]. For instance, a pioneering network-based approach called PRINCE (Prioritization and Complex Elucidation), was developed to predict novel drug-disease associations. This approach integrates drug-target interaction data with a human protein-protein interactome, providing insights into drug-disease connections [195].

Beyond network-based approaches, machine learning techniques have increasingly been incorporated in drug repurposing efforts within systems medicine, showcasing the potential of artificial intelligence (AI) in this field. For instance, deep learning algorithms have been used to predict drug-disease associations based on gene expression data [5], a technique that offers superior predictive performance compared to conventional methods [216]. Ensemble learning, another advanced AI technique, has also been utilized to predict drug repurposing candidates for multiple diseases using diverse data sources, including drug-target interactions and disease similarity networks [210].

In summary, the application of systems medicine to drug repurposing, through leveraging advanced computational techniques, shows significant potential in revolutionizing drug development, promising a shorter and more cost-effective pipeline. These advancements not only have implications for the discovery of novel therapeutic applications for existing drugs, but also herald a new era of precision medicine, where treatments are tailored to individual patient profiles based on complex genetic, environmental and lifestyle data.

1.4.2 Privacy-aware AI

Federated learning (FL) [108] has emerged as a robust technique that significantly addresses privacy and security concerns, enabling collaborative machine learning without sharing raw data. It operates on the principle of decentralized training and model aggregation, thereby allowing multiple parties to jointly train a global model while retaining their local data. This creates an environment that fosters collaboration while ensuring stringent data privacy [127].

In an endeavor to make federated learning accessible to a wider user group, numerous frameworks have been developed [208]. These range from technical solutions such as programming libraries, which necessitate a certain level of programming skills, to more user-friendly solutions that cater to users with varying degrees of technical proficiency. The gamut of these frameworks can broadly be categorized into two types: backend-focused frameworks and all-in-one frameworks.

Backend-focused frameworks primarily cater to developers by providing a suite of tools and methods to implement federated and privacy-aware algorithms [170]. They offer an advanced platform for the development and integration of these sophisticated algorithms into different applications. However, these platforms often require users to have substantial programming experience or even a strong background in software development [22]. This requirement poses a significant barrier to clinical experts and researchers who might lack such programming skills but could immensely benefit from applying federated learning to their research work.

On the other hand, all-in-one frameworks are specifically designed to make privacy-aware analyses accessible to users without in-depth programming skills. These platforms provide a graphical user interface (GUI), simplifying the process of implementing federated learning and thus widening its user base [153, 181]. The emphasis is on user-friendliness, which democratizes federated learning, extending its reach beyond the technical realm.

A major focus area within these frameworks is the inclusion of privacy-enhancing techniques (PETs) [179]. PETs encompass a range of strategies and tools developed to protect sensitive information within the data used for federated learning. These include methods like differential privacy, homomorphic encryption, and secure multi-party computation, all designed to add another layer of privacy protection during the data analysis process.

However, while these all-in-one frameworks have managed to increase accessibility and incorporate PETs, they come with their own set of limitations. For instance, many of these frameworks suffer from a lack of extensibility or have a narrow focus on specific algorithms such as deep learning, or are limited to specific applications such as neuroimaging or genomics [164]. These constraints restrict their applicability and versatility, preventing their use in a wider array of use cases and research fields.

In conclusion, while federated learning, with its privacy-enhancing measures, has shown immense promise as a technique for collaborative machine learning, there is a pertinent need for the development of more versatile, extensible, and user-friendly frameworks [178]. These new frameworks should cater to a wide range of users and applications, facilitating privacy-aware analysis for everyone from developers to clinicians and researchers, across various domains and use cases [117].

2
Systems medicine

Systems medicine is an emerging interdisciplinary field that aims to understand the complex interactions between biological systems and improve healthcare by leveraging computational and experimental approaches. This comprehensive approach integrates different levels of information, such as genes, proteins, metabolites, and clinical data, to achieve a holistic understanding of human health and diseases [88, 105].

A central premise of systems medicine is that biological systems exhibit emergent properties that can only be understood by studying the interactions among their components. These interactions, referred to as the 'interactome', are essential to understanding how complex biological functions emerge from simpler molecular components [10]. Systems medicine employs mathematical modeling, bioinformatics, and high-throughput experimental approaches to unravel the complexity of the interactome and identify key molecular players and pathways involved in disease processes [100].

One of the main applications of systems medicine lies in the area of precision medicine. By incorporating multi-omics data (e.g., genomics, transcriptomics, proteomics, and metabolomics) and patient-specific clinical information, systems medicine can help identify individual-specific disease signatures and predict optimal treatment strategies [37]. This approach, also known as personalized medicine, aims to tailor treatments to individual patients based on their unique genetic and molecular profiles, thus improving therapeutic outcomes and reducing adverse side effects [43].

Systems medicine is also instrumental in drug discovery and repurposing efforts. By generating comprehensive maps of disease-associated molecular pathways and investigating the effects of drug candidates on these pathways, researchers can identify novel therapeutic targets and assess the efficacy and safety of existing drugs in treating various conditions [209]. This approach not only expedites the drug development process but also reduces the associated costs by uncovering new uses for approved drugs [7].

Another application of systems medicine lies in the identification and validation of biomarkers for early disease detection and prognosis. By integrating multi-omics data and clinical information (e.g., in the form of electronic health records), researchers can identify specific molecular signatures that correlate with disease states or treatment responses. These biomarkers can then be used for early diagnosis, monitoring disease progression, and evaluating therapeutic outcomes [95].

Despite its potential, systems medicine faces several challenges [8], including the integration and interpretation of vast amounts of heterogeneous data [2], developing accurate and predictive models, and translating findings into clinically actionable insights [92]. Ongoing efforts to address these challenges include the development of advanced computational methods, the establishment of interdisciplinary collaborations, and the promotion of data sharing and standardization initiatives [9].

2.1 Infectious diseases

Infectious diseases, caused by pathogens such as bacteria, viruses, parasites, or fungi, that are transmitted from one organism to another, continue to pose significant threats to global health [135]. The complexity of host-pathogen interactions and the rapid emergence of drug-resistant strains present major challenges in the prevention, diagnosis, and treatment of infectious diseases [19]. Systems medicine, with its integrative and holistic approach, offers a promising avenue to tackle these challenges by unraveling the intricate interplay between pathogens and their hosts [24].

Given the complexity and heterogeneity of infectious diseases, a fundamental understanding of their unique aspects is crucial for their effective management. One of the key aspects of infectious diseases is the dynamic and complex interplay between the host and the pathogen. Systems medicine can be employed to study these interactions on multiple levels, such as genetic, transcriptomic, proteomic, and metabolomic. By integrating data from these different levels, researchers can develop a comprehensive understanding of the molecular mechanisms involved in infection processes, immune responses, and host susceptibility to various pathogens [21].

Viral infections In contrast to bacterial, fungal, and parasitic infections, viral infections are peculiar in their very nature of pathogenesis. While all infectious diseases involve the introduction of a foreign organism or agent into the host system, the distinctive element of viral infections lies in their specific mode of replication and their intimate interaction with the host's cellular machinery [173]. Viruses, essentially nucleic acid encased in a protein shell, lack the cellular structure required for self-sustaining reproduction [4, 106]. They, therefore, rely on host cells to replicate, using the host cell's molecular machinery to synthesize their components and assemble new virus particles. This distinction has far-reaching implications for both the progression of disease and its treatment strategies [45].

The rapid mutation rates of many viruses, particularly RNA viruses, lead to the emergence of drug-resistant strains, rendering certain antiviral treatments ineffective over time [173]. Vaccination remains one of the most effective strategies against viral diseases, priming the immune system to recognize and combat the virus upon subsequent exposures [154].

However, vaccine development is a lengthy and complex process, requiring careful evaluation of safety, efficacy, and long-term protection. Another challenge is that viruses can undergo antigenic drift or shift, altering their surface proteins and potentially escaping immune detection – a phenomenon observed notably in influenza viruses [31].

Therefore, the strategies to combat viruses involve a combination of antiviral drugs, vaccines, and therapeutic antibodies, with ongoing research focusing on innovative methods like gene editing and interfering RNA technologies to hinder viral replication [38].

Network-based approaches, a cornerstone of systems medicine, can be used to model host-pathogen interactions, allowing researchers to identify essential molecular players, signaling pathways, and potential therapeutic targets. This is particularly relevant for viral infections. The immunological response triggered by a viral infection involves complex signaling networks and feedback loops that spread through multiple organ systems. For instance, virus-host protein interaction networks can provide insights into the mechanisms by which viruses hijack host cellular machinery, while immune response networks can help elucidate the complex interplay between the immune system and invading pathogens [77].

2.2 Drug repurposing

Drug repurposing, also known as drug repositioning, is the process of identifying new therapeutic uses for existing drugs. This approach can significantly reduce the time and cost associated with traditional drug discovery and development, as approved drugs have already undergone extensive safety and pharmacokinetic evaluations [7]. One of the compelling arguments for drug repurposing, especially in the context of viral infections, lies in the rapid mutation rates observed in many viruses [206]. Traditional drug development is a lengthy process, and by the time a new drug is developed, tested, and approved, the target virus might have already mutated, potentially rendering the drug less effective [42]. Drug repurposing can address this challenge by swiftly transitioning an already approved drug to combat a newly emerged or mutated viral strain, offering a more adaptive response to the dynamic landscape of infectious diseases [157]. In the context of infectious diseases, drug repurposing holds great potential for rapidly addressing emerging threats and combating drug-resistant pathogens [132].

Systems medicine, with its data-driven and integrative approach, can play a crucial role in facilitating drug repurposing efforts for infectious diseases [163]. Systems medicine can help identify drug repurposing candidates by integrating various data types, such as pathogen-protein interactions, drug-target interactions, and protein-protein interaction networks (see Figure 2.1). By analyzing these data through computational approaches, researchers can uncover potential connections between approved drugs and infectious disease pathways, as well as predict off-target effects that might be therapeutically beneficial [209].

Such pathways can be interpreted as a series of chemical reactions occurring within a cell that lead to certain cellular functions or outcomes. They can be impacted by various factors, including genes, proteins, pathogens and external substances like drugs [4]. Infectious disease pathways refer to the sequences of molecular events that get triggered during an infection, which might involve the pathogen's life cycle, host immune responses, and changes in cellular physiology. Drug repurposing is inherently linked to these pathways because a drug developed for one purpose might inadvertently influence a pathway in a manner that proves beneficial against an infectious disease [7]. Understanding these pathways, therefore, allows scientists to predict how existing drugs might be repurposed to modulate these processes, potentially halting the progression of the disease or mitigating its effects [103, 162].

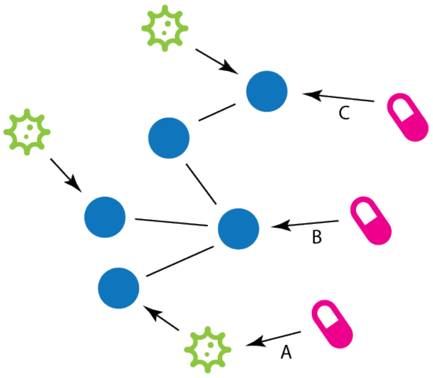


Figure 2.1: The simplified interactome network consists of viral proteins (green), human proteins (blue) and drugs (pink). One can distinguish between drug targets that are directly targeted by a virus (A), indirect targets (B) and pathogen protein targets (C).

2.2.1 Datasets

Datasets are foundational pillars in the domain of systems medicine and drug repurposing. Their extensive information provides a nuanced view of viral-host interactions, especially when considering the ever-evolving nature of viral strains and their intricate modes of interaction with host cellular systems. Such datasets not only collate data but also streamline the intricate analysis required to discern patterns, identify potential drug targets, and predict therapeutic efficacy [12].

Protein-protein interaction (PPI) networks are pivotal in drug repurposing strategies. These networks, through their comprehensive mapping of molecular interactions, offer more than just biological data. They provide a roadmap to the complex landscape of intracellular processes, facilitating the identification of proteins and pathways pivotal for cell functionality. The STRING database, for instance, is a comprehensive repository of PPIs, which has been utilized for research on diseases ranging from malaria to certain types of cancers [187]. Additionally, BioGRID, another vast PPI resource, has facilitated understanding of genetic and protein interactions in various organisms, revealing potential chokepoints for therapeutic interventions [35].

Pathogen-host interaction datasets provide information on the interactions between pathogens and the host organisms they infect. These datasets capture the molecular strategies employed by pathogens to invade, thrive, and reproduce, often at the host's expense. The Host-Pathogen Interaction Database (HPIDB), for example, has become an invaluable tool for researchers studying infectious diseases, offering insights into the molecular tussles between hosts and their invaders [139]. Such resources have proven crucial in the study of viruses like HIV, highlighting how they exploit host cells for reproduction, and suggesting intervention points for drug development [70].

Protein-drug networks are multifaceted repositories that shed light on the intricate relationships between drugs and their target proteins. The significance of such networks extends to drug repurposing efforts, where understanding these relationships can lead to unforeseen therapeutic applications. The DrugBank database stands as a testament to the potential of such datasets. It not only provides details on drug-protein interactions but has also enabled researchers to identify novel uses for old drugs, such as the repurposing of thalidomide from a sedative to a potent immunomodulatory agent for conditions like multiple myeloma [202]. Moreover, the exploration of these networks has unveiled secondary or off-target effects of drugs. Sometimes, these unintended interactions can be harnessed for therapeutic benefit in contexts other than their original intent, broadening the horizon of drug repurposing [209, 89].

In summary, the vastness and depth of these datasets play an essential role in modern drug discovery, especially in the realm of drug repurposing. Through meticulous analysis and integration, they offer a perspective through which researchers can envision novel therapeutic strategies, making them invaluable in the fight against infectious diseases in systems medicine.

2.3 Network exploration and ranking

Network ranking is a powerful computational approach that utilizes the topological properties of biological networks to identify promising drug repurposing candidates [114]. By leveraging the inherent complexity of systems medicine, this method facilitates the discovery of novel therapeutic targets and the prediction of drug-disease relationships. In this section, the principles of network ranking, its applications in drug repurposing, and how it complements traditional drug discovery strategies within the context of systems medicine are discussed.

Network ranking is based on the idea that the importance of a node (e.g., a protein or drug) within a biological network is determined by its topological properties, ergo its position in the network (e.g., using centrality measures [11]). By quantifying these properties, network ranking algorithms assign a score to each node, reflecting its significance in the context of the network. Highly ranked nodes are considered critical for maintaining the network's stability and function and are, therefore, more likely to be associated with diseases or drug action [75].

Network ranking can be applied to various types of networks, including protein-protein interaction (PPI) networks, drug-target networks, and disease-gene networks. The following sections provide an overview of prominent network-based ranking algorithms.

2.3.1 Centrality measures

In graph theory, centrality measures provide a quantifiable way to identify the most 'important' nodes within a network (see Figure 2.2). Several centrality measures exist, including degree centrality, closeness centrality, and betweenness centrality, each illuminating different aspects of the nodes' importance [142].

Degree centrality, defined as the number of connections a node has to other nodes, can be used to identify highly connected proteins (hubs) or drugs in the network (see Figures 2.2a, 2.2d). High-degree centrality may indicate essential proteins in a biological system or drugs with multiple targets [12].

The degree of a node is the number of edges that it has. For an unweighted undirected graph, it is a straightforward count of the edges. In a directed graph, one can differentiate between in-degree and out-degree. The degree centrality for a node v, $C_D(v)$, in a graph with N nodes is given by:

$$C_D(v) = \frac{\deg(v)}{N-1} \tag{2.1}$$

where $\deg(v)$ is the degree of v, and $N-1$ is the maximum possible degree.

Closeness centrality measures the inverse of the average shortest path length from a node to all other nodes in the network, capturing the 'reach' of a node (see Figures 2.2b, 2.2e). Proteins or drugs with high closeness centrality may have a wider influence on the network due to their 'proximity' to other nodes [99].

The closeness centrality for a node v, $C_C(v)$, in a connected graph is given by:

$$C_C(v) = \frac{N-1}{\sum_{u \neq v} d(v,u)} \tag{2.2}$$

where $d(v,u)$ is the shortest-path distance from v to u, and the sum is over all nodes u not equal to v.

Betweenness centrality emphasizes nodes that serve as bridges between different parts of the network (see Figures 2.2c, 2.2f). Proteins or drugs with high betweenness centrality may affect communication or interaction pathways within the network and thus could play a critical role in biological systems [58].

The betweenness centrality for a node v, $C_B(v)$, is given by:

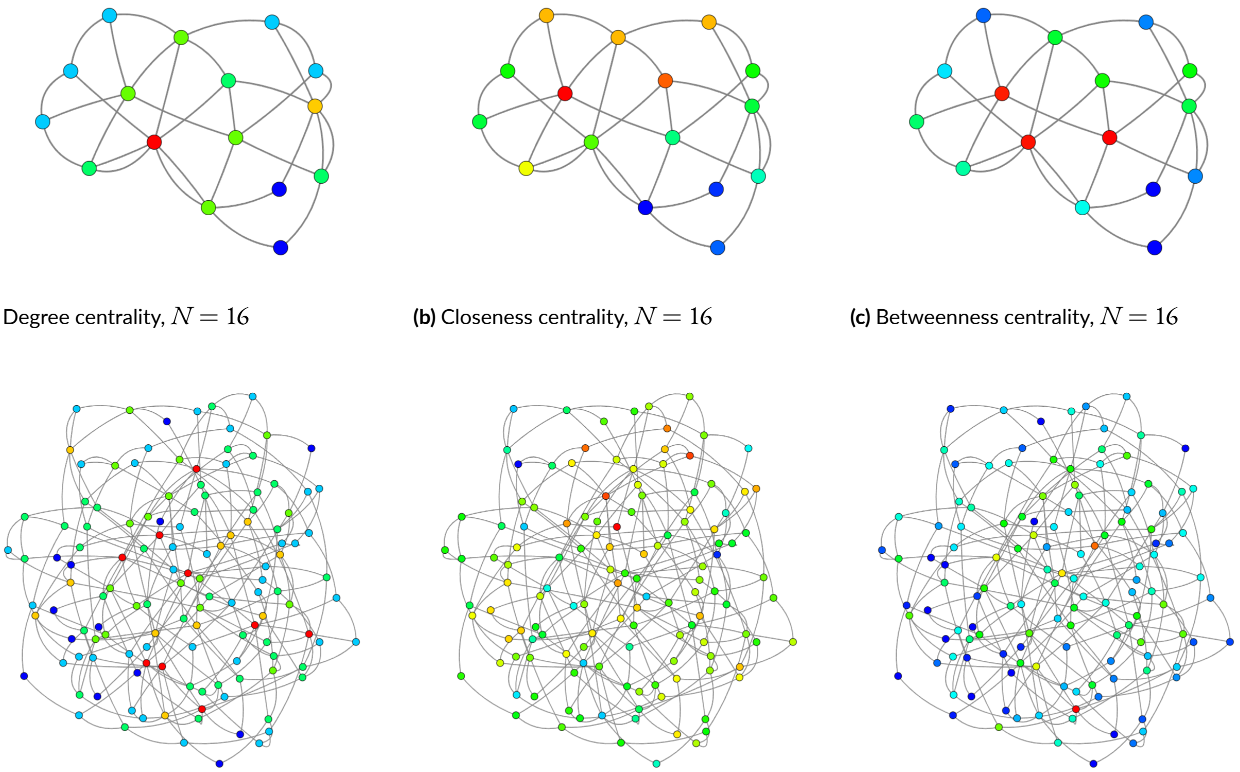
$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{2.3}$$

where σ_{st} is the total number of shortest paths from node s to node t and $\sigma_{st}(v)$ is the number of those paths that pass through v.

2.3.2 Steiner tree

A Steiner Tree is a graph that connects a given subset of vertices (called terminal vertices) with the minimum total edge weight possible [49]. Originating from the realm of combinatorial optimization, the use of Steiner Trees presents a promising avenue in the realm of systems medicine, particularly for drug ranking in drug repurposing based on PPI and protein-drug networks.

A Steiner Tree approach identifies the minimal set of additional nodes (e.g., proteins), called Steiner vertices, and associated interactions that connect the terminal nodes [94].



(a) Degree centrality, $N = 16$

(b) Closeness centrality, $N = 16$

(c) Betweenness centrality, $N = 16$

(d) Degree centrality, $N = 128$

(e) Closeness centrality, $N = 128$

(f) Betweenness centrality, $N = 128$

Figure 2.2: Degree centrality, closeness centrality and betweenness centrality assign different scores to each node. The scores are linearly normalized and encoded as color ascendingly as blue (lowest), green, yellow, orange, red (highest). Degree centrality assigns high scores independently of the position in the network, whereas closeness and betweenness centrality take the position within the network into account. Source code available at `https://github.com/jm9e/centrality-networks`.

The rationale for using a Steiner Tree in drug repurposing lies in the network's inherent ability to capture indirect interactions and pathways between proteins and drugs [40]. Drugs targeting proteins in the resulting Steiner Tree may have a higher likelihood of influencing the disease process, even if they do not directly target the disease proteins. Therefore, these drugs may be repurposed for treating the disease.

To rank drugs for repurposing, one can score the drugs based on their association with the proteins in the Steiner Tree. Drugs associated with multiple proteins or highly connected proteins in the tree would receive higher scores. This way, the drugs that could potentially exert a broader influence on the disease-associated network would be ranked higher.

Moreover, the approach could be refined by considering edge weights representing interaction strengths, or by incorporating additional network measures, such as centrality, to prioritize highly influential proteins and their associated drugs.

The Steiner Tree problem is an NP-hard problem and thus not efficiently solvable [104]. However, heuristic approaches exist [72], such as the Multi-Steiner Tree approach[1], a customized version of the Kou et al. [3, 110] approach, to obtain an efficient approximation (see Section 1).

Multi-Steiner Tree initially utilizes the Kou et al. procedure to calculate the premier Steiner Tree, denoted as T. Concurrently, a depth-first search is executed to find all bridges within the graph. A bridge is understood as an edge, the removal of which would result in the disconnection of the graph. Let L signify the collection of edges present in T, C be the cost related to T, and τ be a user-specified tolerance detailing the extent to which the costs of the forthcoming trees may surpass C. Additionally, let k signify the count of the already discovered trees (initialized to 1) and U represent the set of returned nodes (initialized to the nodes contained in T). The steps in 1 are iterated until $k = K$ or L is empty. Subsequently, the subgraph elicited by U is returned.

2.3.3 TrustRank

TrustRank[1], an algorithm originally designed to combat web spam [78], can be effectively leveraged for drug repurposing using PPI and protein-drug networks. TrustRank operates on the principle of 'guilt by association' - it differentiates reputable (or 'trustworthy') web pages from spam by iteratively distributing trust scores from a manually selected set of reputable seed pages to their linked pages [28] (see Section 2).

In the context of drug repurposing, TrustRank can be applied to a network where nodes represent drugs or proteins and edges represent known drug-protein interactions, following a network topology approach as before [50]. The trust scores in this network can be interpreted as confidence in the potential of a drug to be repurposed for a particular disease.

Proteins known to be associated with a disease can be chosen as seeds, and these seeds are initially assigned high trust scores. The algorithm then propagates these scores throughout the network along interaction edges, distributing a protein's score among its associated drugs [39] and, if we consider drug-drug interaction data, potentially further among other drugs.

In this iterative process, drugs associated (directly or indirectly) with many disease proteins or with highly trusted disease proteins tend to receive high trust scores, hence they are ranked highly for repurposing. As a result, TrustRank also allows for the consideration of indirect drug-disease associations that may be overlooked by approaches considering only direct interactions [71].

[1] Implementations for TrustRank and Multi-Steiner Trees can be found at https://github.com/jm9e/network-algorithms

Algorithm 1 Multi-Steiner Tree

```
Require: $G, K, \tau$
$T \leftarrow$ computeSteinerTreeKou($G$)
$bridges \leftarrow$ findBridges($G$)
$L \leftarrow$ edges($T$)
$C \leftarrow$ cost($T$)
$k \leftarrow 1$
$U \leftarrow$ nodes($T$)
while $k \neq K$ AND $L$ is not empty do
    $e \leftarrow L$.pop()
    if $e$ in $bridges$ then
        continue
    end if
    $G$.removeEdge($e$)
    $T' \leftarrow$ computeSteinerTreeKou($G$)
    if cost($T'$) $\leq C \times 100 + \tau$ then
        $U$.add(nodes($T'$))
        $k \leftarrow k + 1$
    end if
    $L \leftarrow L$.intersection(edges($T'$))
    $G$.addEdge($e$)
end while
return $G$.subgraph($U$)
```

Algorithm 2 TrustRank

```
Require: $graph, seedSet, d, maxIterations$
$N \leftarrow$ number of nodes in graph
$trustScores \leftarrow$ array of size $N$
for $i$ in $0$ to $N - 1$ do
    if $i$ in $seedSet$ then
        $trustScores[i] \leftarrow 1/$size of $seedSet$
    else
        $trustScores[i] \leftarrow 0$
    end if
end for
for $iteration$ in $0$ to $maxIterations - 1$ do
    $newTrustScores \leftarrow$ array of size $N$
    for $i$ in $0$ to $N - 1$ do
        $newTrustScores[i] \leftarrow (1 - d) \times trustScores[i]$
        for $j$ in $0$ to $N - 1$ do
            if $graph[i][j] > 0$ then
                $newTrustScores[i] \leftarrow newTrustScores[i] + d \times trustScores[j]/$outdegree($j$)
            end if
        end for
    end for
    $trustScores \leftarrow newTrustScores$
end for
return $trustScores$
```

2.4 COVID-19 pandemic

Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) is a novel coronavirus that has wreaked havoc across the globe, leading to the unprecedented COVID-19 pandemic. First identified in Wuhan, China, in December 2019, this virulent pathogen has swiftly spread, posing immense challenges to healthcare systems, economies, and societies at large [213]. As an enveloped, positive-sense, single-stranded RNA virus, it falls under the Coronaviridae family. The virus exhibits a propensity for the human respiratory system, presenting symptoms that vary widely in severity, from mild manifestations to life-threatening conditions, especially in vulnerable demographics [90].

Given the urgency precipitated by the rapid global spread of the virus and the mounting death toll, the scientific community was pressed to explore all potential avenues to identify viable therapeutic interventions. In this race against time, drug repurposing emerged as an especially promising strategy. This approach, which involves identifying new therapeutic uses for already approved or investigational drugs, offers a significant advantage in terms of time. Traditionally, developing a new drug from scratch demands years, if not decades, of rigorous research and testing. However, repurposed drugs have already cleared several safety and efficacy benchmarks, which accelerates their deployment in response to emergent health crises [157].

The systems medicine paradigm offers a robust framework to facilitate drug repurposing, especially against complex diseases like COVID-19. Central to this approach are PPI networks, pathogen-host interactions, and protein-drug networks, as introduced before. These integrative network-based methodologies offer insights into the intricate molecular interplay underlying diseases, paving the way for the identification of potential therapeutic targets [12].

Focusing on SARS-CoV-2, researchers have meticulously mapped virus-host protein-protein interactions (PPIs) specific to the infection. By scrutinizing this PPI network, pivotal host proteins and cellular pathways co-opted by the virus have been spotlighted [70]. Harnessing this knowledge, scientists can pinpoint existing drugs with the potential to modulate these host factors or pathways. The objective is to either target the proteins directly or to impede the broader cellular mechanisms that the virus exploits [213].

Furthermore, the data-driven nature of systems medicine aids in the identification of potential drug candidates that might have been overlooked using conventional methods. For instance, by analyzing the network interactions, researchers can unveil drugs that, although not directly connected to the primary disease proteins, can influence the broader disease process by targeting the peripheral proteins in the network [129].

In the context of the COVID-19 pandemic, where swift action was paramount, the efficiency and precision offered by systems medicine and drug repurposing stood out as invaluable assets in the quest for therapeutic solutions.

2.5 Results

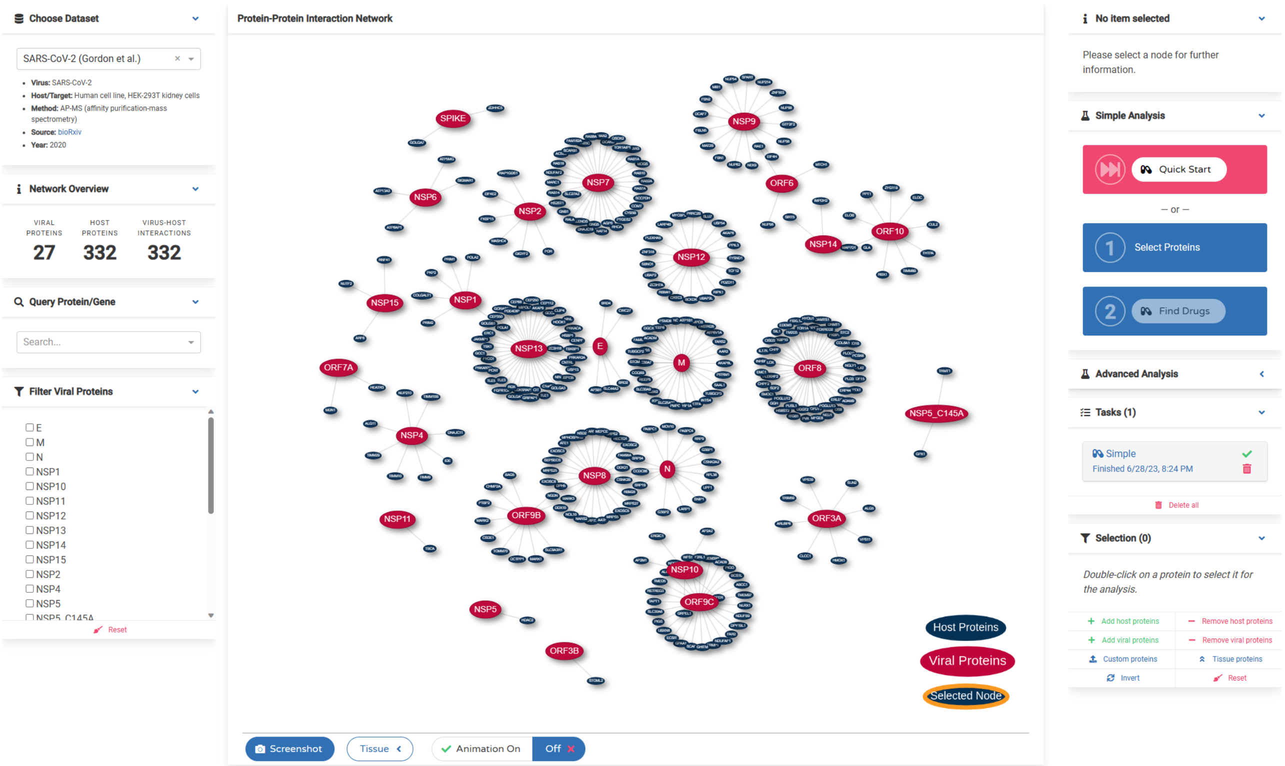
To help combat the Corona pandemic, the methods described in the previous sections have been integrated into the developed web tool CoVex (Corona Virus Explorer) [171] and made available to researchers worldwide, as one of the first tools of this kind made available in this situation.

CoVex is a data-driven web-based platform designed to explore the virus-host interaction landscape of SARS-CoV-2, the causative agent of the COVID-19 pandemic. It was released at an early stage of the SARS-CoV-2 pandemic to provide quick and easy access to the data available at the time, integrated into a publicly available online tool[2]. CoVex integrates multiple layers of information, including virus-host protein-protein interactions, human protein-protein interactions, and gene expression data, to provide a comprehensive network of SARS-CoV-2 interactions within the human host.

[2] Website: https://exbio.wzw.tum.de/covex/

One of the main features of CoVex is its ability to display and analyze virus-host interaction networks. Users can investigate SARS-CoV-2-host protein interactions, identify potential functional modules within the network, and predict the impact of these interactions on cellular processes. CoVex can also provide insights into the molecular mechanisms of viral pathogenesis, facilitating the identification of potential therapeutic targets and the development of effective antiviral strategies.

Furthermore, CoVex allows users to explore gene expression data from various tissues and cell types, helping researchers understand how the virus affects different cellular environments. By integrating this information with protein-protein interaction data, CoVex enables the identification of host factors that may contribute to the susceptibility and severity of COVID-19 infections.



Choose Dataset

SARS-CoV-2 (Gordon et al.)

- **Virus:** SARS-CoV-2
- **Host/Target:** Human cell line, HEK-293T kidney cells
- **Method:** AP-MS (affinity purification-mass spectrometry)
- **Source:** bioRxiv
- **Year:** 2020

Network Overview

VIRAL PROTEINS	HOST PROTEINS	VIRUS-HOST INTERACTIONS
27	332	332

Query Protein/Gene

Search...

Filter Viral Proteins

- □ E
- □ M
- □ N
- □ NSP1
- □ NSP10
- □ NSP11
- □ NSP12
- □ NSP13
- □ NSP14
- □ NSP15
- □ NSP2
- □ NSP4
- □ NSP5
- □ NSP5_C145A

Protein-Protein Interaction Network

No item selected

Please select a node for further information.

Simple Analysis

Tasks (1)

Simple
Finished 6/28/23, 8:24 PM

Selection (0)

Double-click on a protein to select it for the analysis.

Figure 2.3: The CoVex interface provides users with a virus-protein interaction network at the beginning, only showing viral proteins and directly affected human proteins. On the left, a summary of the selected dataset is shown and users can search for specific viral or human proteins. To the right, users can select parts of the network and start their analysis.

2.5.1 Integrated datasets

For CoVex, a network of virus-host interactions was obtained by integrating data from various sources. This includes SARS-CoV-2 AP-MS data, reported by Gordon et al. [70] which contains 332 high-confidence virus-host interactions for 27 SARS-CoV-2 proteins. Additionally, interactions from SARS-CoV-1 were included from the databases VirHostNet [74] and Pfefferle et al. [152] which contain 24 and 113 interactions, respectively.

PPIs were sourced from the integrated interactions database [109] and subsequently filtered based on their experimental validation. Through this process, an interactome was generated, comprising 17,666 proteins that are connected via 329,215 interactions.

To achieve a wide range of drug-target associations, multiple databases were compiled, including ChEMBL [130], DrugBank [202], DrugCentral [193], Target Therapeutic Database [198], Guide To Pharmacology [6], PharmGKB [13], and BindingDB [64]. From this pool, only drugs that demonstrated a binding affinity value (EC50, IC50, Kd, and Ki) $< 10\mu M$ were considered for further analysis [188, 211]. Additionally, the

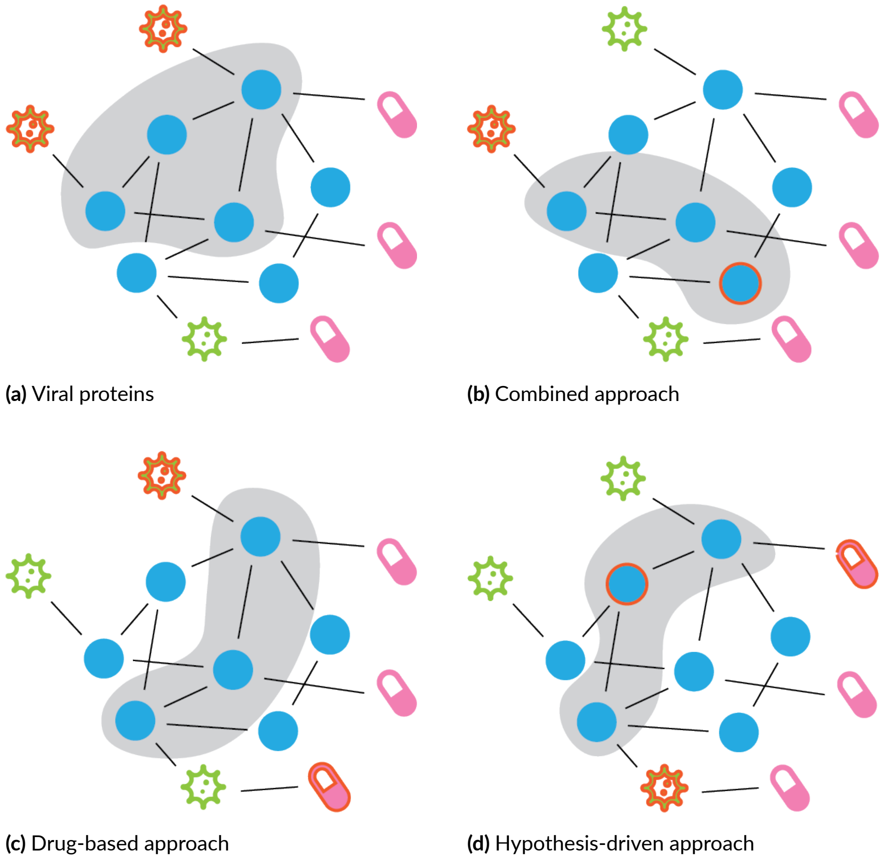


Figure 2.4: Different application scenarios in CoVex can be achieved by selecting different seed proteins (i.e., viral proteins and host proteins, highlighted with orange). Depending on the selected seed nodes, algorithms such as the Multi-Steiner Tree identify pathways (indicated in gray).

included drugs needed to be mappable to DrugBank IDs and target host proteins to be deemed relevant to the network.

In addition to that, drugs that were currently undergoing clinical trials for the treatment of COVID-19 were identified. The relevant drugs, mappable to DrugBank IDs, were sourced from platforms such as ClinicalTrials.gov, the EU Clinical Trials Register [34], and the International Clinical Trials Registry Platform. The network was further enriched with a total of 6861 drugs through that process, out of which 67 were in clinical trials. These drugs were linked to 52,860 drug-target associations that were also integrated into the network.

Per-tissue median gene expression levels from the GTEx data portal [118] adds another layer of information to the network, allowing for tissue-specific filtering and visualization of gene expression values in the CoVex tool.

2.5.2 Application scenarios

There are different ways in which CoVex can be used to identify promising drug candidates (see Figure 2.4). These approaches differ in terms of the selection of the seeds and the underlying assumptions about the disease.

While CoVex has been implemented mainly for the SARS-CoV-2 virus, integrating the relevant datasets, it can also be used for other diseases by providing a custom selection of seeds using the ’custom proteins’ feature.

The following subsections provide an overview of the main application scenarios of CoVex.

Viral proteins

In this application scenario, the user begins by selecting a set of proteins related to a virus of interest (see Figure 2.4a). Leveraging the PPI network, they identify the biological mechanism or pathway exploited by the virus for infection.

For instance, one could consider certain proteins that form the outer structure of a virus and hence facilitate its entry into host cells, which is seen as a virus-host interaction in the network. By selecting the these viral proteins or their direct interactors, an algorithm such as a Multi-Steiner Tree is used to unravel the biological pathway involved.

Analysis of the resulting network can lead to the identification of new potential drug targets. Further, the user employs network measures like closeness centrality to pinpoint drugs that could influence this pathway. They may discover relevant drugs that target a key protein within the pathway, including both approved and experimental therapeutics.

To deepen the understanding of the interplay between the key protein and other proteins known to participate in viral entry, the user can leverage additional features available in the computational tool at hand. This may lead to the identification of additional proteins that connect the key protein with other relevant proteins involved in the viral infection process.

These proteins are likely functionally related through a specific system or pathway, which could be targeted by the aforementioned drugs. In conclusion, through this process, the tool enables to identify proteins that play a significant role in viral host cell entry and can be targeted by a range of therapeutics. Importantly, this process can also lead to the identification of additional protein targets that are functionally related to critical proteins involved in the infection process, even if those critical proteins were not part of the initial protein set used for the analysis.

Combination of Viral Proteins and Selected Host Proteins

In this application scenario, the user begins with both viral proteins and a list of proteins of interest, again to identify a pathway or biological mechanisms that could potentially be targeted by drugs (see Figure 2.4b). Specifically, this could involve investigating viral proteins known to suppress host immunity and the corresponding host immune response pathways.

As an example, the user selects certain viral proteins that are believed to be involved in suppressing innate immune response and promoting apoptosis. Concurrently, they assemble a list of proteins of interest based on differentially expressed genes (DEGs) from a relevant study. These DEGs could be related to the immune response mounted by host cells when infected by a virus, with emphasis on genes associated with the host pathway relating to infection by another viral pathogen.

Once they have selected the viral proteins and DEGs, these are again used as seeds for an algorithm like the Multi-Steiner Tree to extract a potential immune-related mechanism. The resulting subnetwork should ideally reveal that the viral proteins are closely related to the DEGs in the PPI network.

Upon analyzing this network using measures like closeness centrality, the user can obtain a high rank for certain drugs that have immunomodulatory effects and see which are currently under clinical trials. Administration of these drugs could potentially mitigate immune-mediated lung injury and reduce functional deterioration caused by an overactive host inflammatory response, which is particularly crucial in later stages of a disease.

Other drugs that target this subnetwork may also be identified for further examination in downstream analyses. The tool also provides the flexibility for users to supply a custom list of proteins as seeds to seek out drugs that could target a proposed mechanism of interest.

Drug-based approach

In this scenario, starting with a set of drugs of interest, a top-down approach can be used to extract potential host mechanisms and additional drugs that target the proteins involved in these mechanisms (see Figure 2.4c). For example, one could identify a list of drugs currently under clinical trials for a specific disease and categorize them based on their anatomical therapeutic chemical (ATC) classification.

The focus could be on drugs from a specific class, such as immunostimulants, and their target proteins as starting seeds. In parallel, the interactors of immune-related viral proteins could be selected as end-point seeds. By applying the Multi-Steiner Tree algorithm, one can identify pathways of interacting proteins that link the selected drugs (and their target proteins) with the chosen viral proteins.

Among these connector proteins, genes associated with a specific system, such as cytokine signaling in the immune system, could be found. Notably, one might discover a specific gene that is highly expressed in certain body tissues and can be inhibited by an investigational drug, which is a potential therapeutic for inflammatory and autoimmune processes.

Hypothesis-driven approach

In this scenario, the user starts from a hypothesis-driven mixed selection of viral and host proteins, as well as drugs. The aim is to utilize PPIs to uncover a full mechanism or pathway and to propose additional drug candidates.

As an example, one could consider a recent hypothesis suggesting that a virus interferes with the formation of hemoglobin in erythrocytes, leading to symptoms of hypoxia. This hypothesis would also serve to explain why certain drugs are effective, as they may prevent viral proteins from competing for crucial components in hemoglobin.

Based on this hypothesis, one could investigate the pathways connecting these viral proteins with the effective drugs (see Figure 2.4d). To do this, one selects known host proteins that bind to these components as seeds. These could be proteins that interact with the viral proteins and those that bind to other relevant components.

Employing a complex algorithm for drug target discovery, followed by a network measure like closeness centrality for drug discovery, one can identify other potential therapeutic drugs, in addition to those already under clinical trials. Importantly, one might find drugs that are approved for related conditions, fitting the investigated hypothesis.

Also, some drugs are widely used therapeutically for related imbalances, thus providing more potential therapeutic options. However, it is essential to note that the scientific evidence supporting the initial hypothesis might be limited, and such a hypothesis is used merely to illustrate the potential of a tool like CoVex for network medicine investigation and hypothesis testing capabilities.

2.5.3 Summary

In light of the challenges COVID-19 has posed to health systems, economies, and societies worldwide, efforts to find effective treatment options have remained at the forefront of scientific endeavors. The extended duration required to develop vaccines has magnified the importance of swiftly identifying therapeutic agents to manage and control the symptoms of COVID-19. Particularly, drug repurposing stands out as a viable strategy, given its potential to expedite clinical trial processes.

CoVex has emerged as an advancement in this effort, mainly due to its quick availability. It is a user-friendly, interactive web platform that synergizes data from both SARS-CoV-1 and SARS-CoV-2 virus-host interactions with the broader human interactome and multiple drug-target databases, fitting into the Systems Medicine paradigm. The platform's design facilitates an efficient exploration of the virus-host-drug interactome, enabling users to pinpoint potential drug targets and candidates for repurposing swiftly. Such streamlined access to integrated data bridges the gap between computational data mining and practical drug discovery, addressing the often times unstructured process in identifying drug repurposing candidates.

While CoVex stands as a potent tool in research, it's imperative to approach its outputs with a discerning perspective. It offers suggestions for potential drug candidates but does not ascertain their antiviral efficacy. The intricate dynamics of viral-host interactions mean that drugs targeting specific proteins might not guarantee an antiviral outcome. These candidates require rigorous validation through subsequent investigations, genetic or chemical assays, and clinical trials (see Section 4.1). Furthermore, the database upon which CoVex draws, while extensive, has its limitations. CoVex focuses on identifying novel drug targets within the human interactome, eschewing drugs that directly target viral proteins.

Recognizing the intrinsic challenges of navigating the complex algorithms and parameters, CoVex has incorporated features like task queuing for parallel executions. Plans for the platform's evolution include the development of guidelines aiding users in method selection suited to their research queries, continuous updates on virus-host interactions, and integration of ongoing clinical trial data.

3
Privacy-aware AI

In systems medicine, as well as in adjacent disciplines such as bioinformatics and computational biology, the analysis of large, complex, and often high-dimensional datasets is fundamental to extracting meaningful knowledge [87]. Such datasets may include genomic sequences, proteomic profiles, medical images, and electronic health records, among others. Traditional algorithmic and statistical methods have been used to uncover patterns or make predictions, but they often necessitate a profound understanding of the biological, clinical, or physiological mechanisms at play [82]. Artificial intelligence methodologies offer a distinct advantage in this regard [189, 97]. Unlike traditional methods, they are capable of autonomously learning from the data, effectively ”teaching themselves” to identify subtle relationships or patterns within the data without requiring an in-depth understanding of the underlying processes [160, 46]. This capability not only accelerates the speed of data analysis but also enables the discovery of novel insights that might be overlooked by conventional approaches.

3.1 Definitions

Like for any statistical or algorithmic approach, certain terms are used in the context of machine learning, which are briefly introduced in this section [20, 82].

Since there is often a confusion between the terms artificial intelligence (AI) and machine learning (ML), in the remainder of this thesis, the terms are defined as follows [85, 169]:

Artificial intelligence is a broader term and encompasses ML as well as other techniques such as algorithms and systems embedding ML. This term is arguably much more vague and purposefully creates a link to human intelligence which these techniques try to mimic [169]. Consequently, this term can often be found in texts addressing an audience not necessarily familiar with the inner workings of the field.

Machine learning relates to methods and algorithms that are capable of finding rules or patterns that allow for predicting a certain label [68]. They usually employ a pre-chosen *model* which is optimized by an *optimizer*. A model consists of adjustable *parameters* which are modified (*optimized*) during the training process and a fixed structure [167]. The choice of the model significantly influences the performance of its resulting trained version and is far from trivial [20]. Section 3.2 sheds more light on different model types and their characteristics.

While the term artificial intelligence is arguably more popular, this term will be avoided in the remainder of this chapter, to serve a precise language [85].

A piece of data to which ML should be applied is called a *dataset*, which usually follows a certain structure. It contains *samples*, each consisting of *features* and potentially a *label*. A dataset D of n samples and m features can therefore be perceived as a set of vectors as in Equation 3.1, where y_i are the labels and $x_{i,j}$ are the feature values.

$$D = \{(y_1, x_{1,1}, ..., x_{1,m}), ..., (y_n, x_{n,1}, ..., x_{n,m})\} \tag{3.1}$$

When referring to all features $(x_{i,1}, ..., x_{i,m})$ of a sample i, the notation X_i is used.

Samples could represent patients for instance, where features could be gene expression values and the label be either 1 or 0, specifying whether the respective patient has a particular disease (e.g., cancer) or not.

A prediction model M with its parameters p can be perceived as a function which is presented with an instance X and returns an y value: $M_p(X) = y$. In order to obtain the parameters p for a model M, the optimizer O is applied to a dataset, which can be formalized as $O_M(D) = p$.

To compare the performance of different models, various metrics exist. *Accuracy*, representing the percentage of correctly classified samples, and *mean squared error* (MSE) are commonly used, the latter defined as $\frac{1}{n}\sum_{i=1}^{n}(M(x_i) - y_i)^2$.

3.2 Models and Optimizers

There are many different types of ML methods, ranging from conventional, statistical methods, to more modern, computationally intensive approaches such as deep learning. There are also meta-level techniques that aim to find the best model or model configuration for a specific task [84]. These models vary in performance, size, and complexity [20, 68]. The first task when applying ML to a dataset is therefore to choose the appropriate model [156].

Statistical models like linear regression (see Section 3.2.1) and logistic regression are among the simplest but still effective ML techniques [82]. These models are well-suited for problems where the relationship between the features and the labels is approximately linear. Their main advantage is interpretability, which makes them favored in fields like healthcare and social sciences.

Tree-based models such as decision trees (see Section 3.2.2) and their ensembles like Random Forests and Gradient Boosting Machines are non-linear models used for both classification and regression tasks [27]. They partition the feature space into smaller regions to make predictions, which gives them great flexibility in modeling complex relationships.

Support vector machines (SVM) are popular for tasks that require a boundary to separate different classes [44]. They can be used for both linear and non-linear classifications by employing different kernels. SVMs are especially effective when the number of features is large relative to the number of samples.

Neural networks (see Section 3.2.3) are deep learning methods employing networks with many layers to learn from the data [68]. These models are particularly good at learning from unstructured data like images, text, and sound but require substantial computational resources.

AutoML and meta-learning techniques aim to automate the process of selecting the best model and its hyperparameters [84]. These methods can be particularly useful for those who are not experts in machine learning but still need to solve complex problems.

Optimizers serve as the engine that powers the training phase of machine learning models. Their primary function is to search the parameter space to find the optimal set of parameters that minimizes the loss function [68]. This effectively tunes the model to make more accurate predictions on unseen data [20]. While some models like linear regression offer analytical solutions for optimization [82], others — particularly more complex models like neural networks — pose significant computational challenges in finding the optimal solution [115]. As models grow in power and flexibility, capturing increasingly complex relationships in the data, the optimization problem correspondingly increases in difficulty [186]. The choice of an optimizer, particularly for neural networks, is not a peripheral decision but a pivotal one that substantially influences both the model's performance and the computational resources required for training [167].

3.2.1 Linear regression

One of the simplest ML methods, often used in statistics and predictive modeling, is linear regression [134]. This method serves to illustrate the foundational concepts introduced in Section 3.1. Linear regression can be formalized as shown in Equation 3.2, where β represents the model parameters, and ε is a normally-distributed error term with mean zero.

$$M_\beta(X) = X\beta + \varepsilon \tag{3.2}$$

The task of an optimizer in the context of linear regression is to find the set of parameters β that minimizes the error term or, equivalently, a *loss function*. In many cases, this loss function L is the MSE, expressed as

$$L(\beta) = \frac{1}{n}\|X\beta - y\|^2,$$

and the optimal parameters can be analytically computed via

$$\beta = (X^T X)^{-1} X^T y$$

in a single iteration [20].

Linear regression models are considered basic continuous models that are relatively straightforward to optimize, particularly when the number of features is low. However, they make several assumptions, including the independent and linear relationship between each feature variable and the label [82]. This assumption implies that each feature contributes independently to the output, and thus, the model may not capture more complex, interdependent relationships between features and labels.

3.2.2 Decision trees

Decision trees serve as another elementary example of machine learning algorithms and contrast with linear regression methods in various ways [158]. While linear regression algorithms are continuous and linear by nature, decision trees are inherently discrete and capable of capturing nonlinear relationships. The difference between continuous and discrete methods and the implications for federated learning are shown in Section 3.3.1.

The core concept of a decision tree is to generate a tree-like model of decisions formulated as rules. Given a sample, the algorithm traverses the tree from the root, applying rules at each node, until reaching a leaf node that contains the predicted label [26]. Each decision rule at a node evaluates to either `true` or `false`, guiding the algorithm down one of two possible branches.

The optimization process of decision trees differs substantially from that of linear regression. While linear regression models can be optimized analytically to find the global minimum of a loss function [134], decision trees require heuristic techniques aimed at minimizing a measure known as *impurity* [138]. Two common measures include the Gini index G and entropy E, defined as in Equations 3.3 and 3.4 respectively, where p_i is the probability or relative frequency of a sample having the label c_i.

$$G = 1 - \sum_{i \in C} p_i^2 \qquad (3.3)$$

$$E = \sum_{i \in C} p_i \log p_i \qquad (3.4)$$

Heuristics are necessary because the search space for the "best" decision tree is often vast and computationally intractable. The tree is constructed by iteratively partitioning the dataset using rules that minimize the resulting impurity. This involves testing various feature thresholds and selecting the one that most effectively reduces impurity. The algorithm proceeds recursively, applying the same logic to the subsequent partitions of the data.

Finding an optimal decision tree is computationally challenging—an NP-hard problem [113]. This makes it particularly difficult for large sets of features. In contrast, linear regression methods can often handle larger feature sets more efficiently due to their analytical solvability.

Decision trees are also prone to overfitting, especially when the tree is deep and captures noise in the training data [133]. Overfitting leads to poor generalization to new, unseen data. Countermeasures include tree pruning techniques and setting a minimum number of samples required for leaf nodes.

3.2.3 Neural networks

Neural networks have gained significant prominence for their capacity to model intricate and non-linear relationships in data [68]. They are designed to approximate the structural and functional aspects of biological neural systems, comprising an interconnected web of neurons and edges (or synapses).

Though they are a powerful tool, capturing the limelight of machine learning in recent years [176], they are not a panacea for all data-driven problems [55]. The complexity and capabilities of neural networks have made them the subject of extensive theoretical and empirical investigation. This section offers a concise overview of the core concepts behind neural networks.

Architecture in neural networks generally involves multiple *layers* of *neurons*, which are interconnected by edges assigned specific *weights*. The input values propagate through these layers, with each layer transforming the values by multiplying them with the associated weights.

Unlike simpler models like linear regression or decision trees, where the primary task of the optimizer is to adjust a relatively small number of parameters or rules, the optimizer's objective in neural networks is more computationally demanding. It needs to find optimal weights across potentially thousands or millions of connections in the network architecture.

Optimization of a neural network is commonly achieved through a technique called *back-propagation* [168]. Back-propagation calculates the gradient of the loss function concerning each weight by traversing the network from the output layer back to the input layer. The weights are then updated in the opposite direction of the gradient, a process often referred to as gradient descent [167]. This is a markedly different optimization landscape than simpler models like linear regression, which can find the optimal parameters analytically, or decision trees, which rely on heuristic methods.

The iterative application of back-propagation allows the neural network to 'learn' how to perform its task more effectively over time. This methodology offers both the benefit and the challenge of being able to capture complex relationships in data but requires significant computational resources to optimize effectively [18].

3.2.4 Ensembles and other techniques

Ensemble techniques provide a robust approach to improve the predictive power of individual machine learning models by combining their predictions to create a more accurate overall prediction. The integration of multiple models can significantly enhance performance, offering improved stability and a decrease in both bias and variance [47]. This section will briefly introduce the most common ensemble techniques.

Voting techniques operate by soliciting predictions from each model in the ensemble. The final prediction is determined by taking the majority vote for classification tasks, or by averaging the predicted values in regression tasks [214]. Voting offers a straightforward way to improve model performance, but it assumes that each model in the ensemble has an equal say, which might not be optimal if the models have different levels of reliability.

Bagging (bootstrap aggregating) trains multiple models independently, each on a different random subset of the training data, obtained by resampling with replacement [25]. The ensemble prediction is then an average of the predictions from individual models. This technique is particularly effective for models that have high variance, such as decision trees. Random forests, which combine bagging with decision trees, are a classic application of this approach [27].

Boosting consists of training multiple models sequentially, each focusing on correcting the errors of its predecessor [59]. The final prediction is a weighted sum of the predictions from all models, where the weights reflect each model's performance. Boosting is highly effective for reducing bias and can improve accuracy, but it is more susceptible to overfitting if the individual models are complex and not sufficiently regularized [175].

Stacking uses a meta-model to combine the predictions of multiple base models. Each base model is trained independently, and their predictions serve as input features for the meta-model, which makes the final prediction [14]. Stacking extends the idea of voting by allowing the meta-model to learn how to optimally combine predictions from different base models based on their individual performance [204].

The choice of ensemble technique to apply is highly problem-specific and should consider the characteristics of the individual models, such as their complexity, tendency to overfit or underfit, and their computational requirements [166].

3.2.5 Comparison

Choosing the right machine learning model for a particular application is far from trivial. While automation and AutoML tools are designed to alleviate this complexity, they are not a panacea. They often perform a broad search across multiple algorithms and hyperparameters, which can be both time-consuming and computationally expensive due to the sheer number of combinations [84, 57].

Method	Complexity	Overfitting Risk	Interpretability	High Dimensionality Handling
Linear Regression	Low	Low	High	Moderate
Decision Tree	Medium	High	High	Poor
Random Forest	High	Moderate	Moderate	Moderate
Neural Network	Very High	High	Low	Excellent

Table 3.1: The table shows a multifaceted comparison of popular machine learning methods displaying complexity, overfitting, interpretability and handling of high dimensional data.

Table 3.1 elucidates how various algorithms perform across different dimensions, providing a more holistic view that can guide the selection process.

Computational Complexity is an essential factor to consider, especially in real-time applications or when computational resources are scarce. A model with 'Low' computational complexity like Linear Regression is easy to train and quick to deploy. This can be crucial in applications such as real-time analytics where latency matters. On the other hand, models with 'High' or 'Very High' complexity, such as Random Forests, often provide higher accuracy at the cost of longer training times and higher resource consumption. Random Forests, which consist of an ensemble of decision trees, are particularly computationally demanding but often yield higher performance [27, 18].

Overfitting Risk describes the vulnerability of a model to fit the noise in the training data instead of the underlying trend. Models with 'High' overfitting risk, like Decision Trees, require meticulous tuning and regularization to generalize well to new data. This makes them less ideal for scenarios where the model has to adapt quickly to new data. 'Low' risk models like Linear Regression are generally more robust to overfitting, especially when the amount of data is large compared to the number of features [82]. Random Forests, which ensemble multiple Decision Trees, inherently mitigate some risk of overfitting by averaging predictions but still require parameter tuning to balance bias-variance tradeoff [27, 82].

Interpretability has emerged as a significant concern, especially in sensitive domains like healthcare and criminal justice, where model decisions can have profound impacts on human lives. Models with 'High' interpretability, like Linear Regression and Decision Trees, allow for easier scrutiny and are more transparent in how they arrive at a prediction. This can be vital for ethical considerations and for gaining stakeholder trust. On the other hand, 'Low' interpretability models like Neural Networks act more like "black boxes," making it hard to understand their decision-making process. This poses challenges in scenarios requiring accountability [32].

High Dimensionality Handling is a critical factor in modern datasets that often contain hundreds or even thousands of features. Models that are rated 'Excellent' for high dimensionality, such as Neural Networks, are well-suited for complex tasks like image or speech recognition, where the feature space is inherently high-dimensional. 'Poor' performers like Decision Trees can struggle with high dimensionality due to the "curse of dimensionality," which can lead to overfitting and increased computational costs. The ability to handle high-dimensional data efficiently is often crucial for applications in bioinformatics, natural language processing, and computer vision [17].

The complexity of model selection is further amplified in specialized applications such as biomedical data analysis. Here, not only must one consider the computational and statistical aspects of the models, but also understand the intricate biological processes that generate the data [160].

In conclusion, while each model type has its own set of merits and demerits, no single algorithm is a "one-size-fits-all" solution. The task at hand—whether it's real-time analytics, ethical considerations in healthcare, or the management of high-dimensional data—necessitates a unique blend of characteristics. Thus, it is not just about maximizing performance metrics, but also about aligning the model's strengths and weaknesses with the project's specific requirements and constraints [144].

3.3 Federated learning

Federated learning (FL) presents an innovative approach to training machine learning models that aligns with emerging data privacy regulations and computational decentralization [127]. Unlike traditional machine learning where data is pooled into a centralized location for model training, FL allows the model to learn from decentralized data residing on multiple devices or servers. The ultimate goal is to develop a global model that performs as if it were trained on a centrally-located, aggregate dataset, without the need for data to ever leave its original location.

The Federated Paradigm

The foundational principle of FL is to keep data localized, thus preserving privacy and potentially even enhancing security. In this setup, each local device or node performs model training on its own subset of the data and then communicates the model updates (e.g., gradients, parameters) to a central server. The central server aggregates these updates to construct a global model. Importantly, no raw data ever leaves the local device, ensuring a level of data privacy and security that traditional centralized models cannot offer.

Methodological Adaptations

While FL has been extensively researched in the context of neural networks [127, 122, 143], extending this paradigm to other machine learning methods remains an open area of research. This thesis aims to expand FL to incorporate the methodologies discussed in Section 3.2.

Each method comes with its own set of challenges and opportunities when adapted to a federated context. For instance, methods like Linear Regression, which have lower computational complexity, might seem straightforward to federate but can encounter issues related to data distribution and privacy leakage. On the other hand, complex methods like Random Forests present challenges in aggregating trees from various devices but offer advantages in mitigating overfitting and improving model performance.

Federated linear regression

In case of linear regression (see Section 3.2.1), a federated version can be created by splitting the optimizer into sub-parts ($X_i^T X_i$ and $X_i^T y_i$) that can be calculated independently, as shown in Figure 3.5 for two participants a and b.

$$\beta = (X^T X)^{-1} X^T y = (X_a^T X_a + X_b^T X_b)^{-1} (X_a^T y_a + X_b^T y_b) \tag{3.5}$$

The aggregator then simply sums up the parts $X_i^T X_i$ and $X_i^T y_i$ to obtain $X^T X$ and $X^T y$ and thereby β, the global model. In this case, the global model entirely matches the classical method and nothing is lost in the course of federating the method (assuming no precision loss occurs due to floating point rounding errors).

Federated average

Federated average is a common technique that can be applied to a range of continuous models (see Section 3.3.1).

For neural networks (see Section 3.2.3), federated average can be applied as follows: Each participant trains one or several epochs and sends the resulting (local) model to a central aggregator. Once the aggregator has received all local models and merges them into a global model by averaging their weights. This requires all participants to use the same neural network architecture that they have agreed upon before. In most cases, the central aggregator sends an initial, randomly initialized model to all participants to avoid an initial divergence.

This method performs well for independent and identically distributed data (IID), where there is no class imbalance across different participants. However, performance can degrade substantially in non-IID conditions. Several approaches, such as data stratification and advanced aggregation algorithms, are being explored to address this issue [212]. One of the central challenges in FL is to adapt these averaging techniques to handle the data distribution skews inherently present in decentralized setups [117, 101].

3.3.1 Continuous and discrete models

The methods previously introduced can be categorized into two distinct types of models: continuous and discrete [196, 82]. This classification is linked to the predictive behavior exhibited by these models. In a continuous model, a minor alteration in the input leads to a proportionally small fluctuation in the prediction. Conversely, in a discrete model, even an insignificant change in the input can yield a drastically different outcome.

This distinction can be formally articulated, as demonstrated in Equations 3.6 and 3.7. Here, M_c and M_d denote a continuous and a discrete model, respectively, and $\mathbb{D}$ signifies the input space for these models.

$$\forall x \in \mathbb{D}.\ \lim_{h \to 0+} M_c(x-h) - M_c(x) = 0 \tag{3.6}$$

$$\exists x \in \mathbb{D}, e > 0 \in \mathbb{R}.\ \lim_{h \to 0+} M_d(x-h) = e \tag{3.7}$$

Equation 3.6 stipulates that in continuous models, the discrepancy in predictions asymptotically approaches zero as the input difference nears zero. In layman’s terms, a minuscule modification to the input results in an equivalently small change in the output.

Equation 3.7, conversely, indicates that in discrete models, the prediction can leap by a quantity e even when the input changes only slightly. Simply put, in a discrete model, a minor input adjustment may lead to a significant, abrupt output transformation.

Continuous and discrete models evidently capture different kinds of relationships between the target label and input data, and a parallel distinction can be applied to variables.

Example For instance, when considering disease variables like ’mild,’ ’elevated,’ and ’severe,’ numerical values such as ’1,’ ’2,’ and ’3’ could be assigned to reflect the severity hierarchy. These are termed *ranked* variables and possess a natural ordinal structure. However, in cases where no such ordering exists, for instance with values like ’subtype A’ and ’subtype B,’ a numerical representation might induce misleading or incorrect implications. Numeric values can insinuate relationships between categories that extend beyond mere ordering. For instance, labeling a severe condition as ’3’ does not mean it is three times as severe as a mild condition labeled ’1.’

Discrete models are generally more compatible with categorical variables, especially when these variables cannot be sensibly rendered in continuous models due to the absence of a natural ordinality. As such, discrete models are often preferred for such scenarios.

Another alternative is the ’one-hot’ encoding method [111], which expands a single categorical variable into multiple variables, each representing a unique category, with precisely one set to ’1’ and all others set to ’0.’

In the context of FL, discrete models often pose challenges that are not easily surmountable [101]. Models like decision trees and random forests are intrinsically more complex to train in a federated manner compared to continuous models, such as linear regression and neural networks. The latter can be averaged across all participating clients to construct a global model, which is not feasible for the former due to their nonlinear and non-continuous nature.

To elucidate, consider the median as a discrete analogue to the average. While federated averages can be computed by weighting local averages by sample size, federated medians do not have a straightforward computation method, as shown by Equations 3.8 and 3.9, where C represents the set of clients and n_c is the sample size for each client c.

$$\mathrm{Avg}_F = \sum_{c \in C} (\mathrm{Avg}_c \cdot n_c) \Big/ |C| \tag{3.8}$$

$$\mathrm{Med}_F \neq \sum_{c \in C} (\mathrm{Med}_c \cdot n_c) \Big/ |C| \tag{3.9}$$

To address these challenges, specialized methods like Federated Boosting [116] have been proposed. However, such techniques entail more complex mechanisms and higher communication overhead than Federated Averaging methods for continuous models.

Ensemble techniques stand as an exception, offering a natural federation method that involves combining local classifiers into a global ensemble. For instance, clients A and B might each train 100 decision trees, and the global ensemble would comprise 200 trees, equally sourced from each client. While differing from a true global ensemble, these federated ensembles still deliver comparable performance [83].

3.4 Privacy-enhancing techniques

While FL already offers a foundational layer of privacy by eliminating the need for raw data transfer between server and client devices, this alone is not a full-fledged solution for data privacy. The trained models and their associated parameters, which are exchanged during the training process, can potentially reveal sensitive information about the underlying data. To further enhance privacy measures and offer a comprehensive approach to safeguarding data, various supplemental techniques are often employed, so-called privacy-enhancing techniques (PETs). This section explores some of the most commonly used techniques, providing an in-depth look at their advantages, limitations, and ideal use-cases.

3.4.1 Differential privacy

Differential privacy (DP) is a statistical technique aimed at providing means to access the usefulness of data while maintaining the privacy of individual data points [53]. It has found wide application in census data, medical research, and more. The core principle is to add a layer of randomized noise to the output of an algorithm or query, thereby preventing the identification of any single entry in the dataset.

The formal mathematical definition of DP can be seen in Equation 3.10. Here, $\mathcal{A}(\cdot)$ symbolizes an algorithm that operates on a dataset. D_1 and D_2 are datasets differing by exactly one element but are otherwise identical to the original dataset D. The term $\operatorname{im}\mathcal{A}$ signifies the range or image of $\mathcal{A}$, encompassing all possible algorithmic outcomes.

$$\forall S \subseteq \operatorname{im}\mathcal{A}(D).\ \Pr[\mathcal{A}(D_1) \in S] \leq e^{\varepsilon} \cdot \Pr[\mathcal{A}(D_2) \in S] \tag{3.10}$$

In simpler terms, the result obtained from $\mathcal{A}(D_1)$ should be statistically indistinguishable from that obtained from $\mathcal{A}(D_2)$, safeguarding whether a particular data point was included in D or not. The parameter ε is particularly important; it quantifies the privacy loss in the data. Lower values of ε offer stronger privacy but at the cost of reduced data utility [52].

3.4.2 Homomorphic encryption

Homomorphic encryption (HE) is a cryptographic approach that allows computations to be performed on encrypted data [194]. This characteristic is invaluable for privacy-centric applications, especially for tasks that require outsourcing computational workload to third-party cloud services. Unlike conventional encryption methods that require decryption before any computational task, HE allows for operations to be performed directly on encrypted data, thus preserving privacy throughout the computation process.

Equation 3.11 captures this attribute succinctly:

$$\mathcal{A}(d) = \operatorname{dec}(\mathcal{A}(\operatorname{enc}(d))) \tag{3.11}$$

It's worth noting that HE is computationally intensive and may not be suitable for real-time or resource-constrained applications [62].

3.4.3 Secure multi-party computation

Secure multi-party computation (SMPC) enables multiple parties to collaboratively compute a function over their respective inputs, all the while maintaining the privacy of each individual input [66]. This is particularly useful in scenarios such as collaborative research between institutions, where data sharing may be legally or ethically constrained.

In SMPC, cryptographic protocols facilitate the computations so that each party only gets the final output and learns nothing about other parties' specific inputs. One well-known technique within SMPC is additive secret sharing, a form of secret sharing where a secret is divided into multiple shares. Each party gets a share, and the secret can only be reconstructed when all the shares are combined [177].

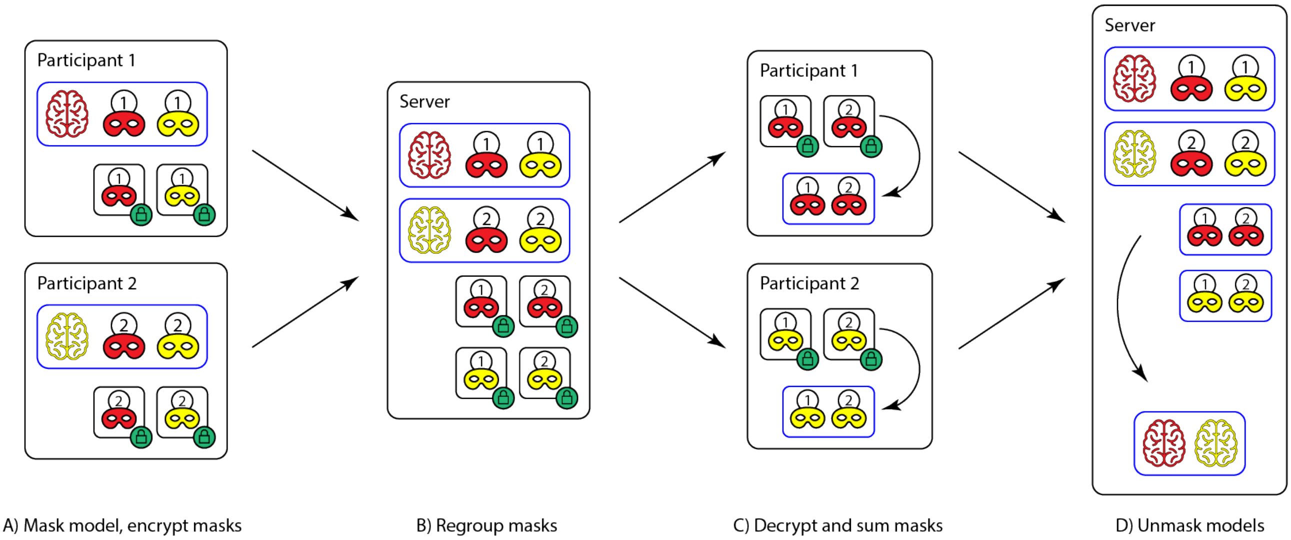


Figure 3.1: Secret sharing across multiple parties involves multiple steps if the participants communicate via a central server. The server must not see the individual data pieces since that would eliminate the masking of the individual data pieces. Participants first mask their data (A) and produce encrypted two random values (using two participants' public keys) and two masked values. They then send the encrypted random values and the masked model to the server (B). The server then sends the encrypted random values to the participants who can decrypt them (C). Finally, the participants send the decrypted and aggregated random values to the server who then can obtain the unmasked global model (D). Source: [126]

3.4.4 Combinations

FL can be synergistically combined with various PETs introduced earlier to provide multi-layered privacy guarantees. The following subsections elaborate on the viable combinations and their specific advantages and limitations.

Combining FL with DP

DP can be integrated into FL by introducing noise into the local model updates, effectively obscuring the individual contributions from each participant's dataset. This is an extension of the same privacy-preserving principle used in non-federated settings, where a single participant's data should not be identifiable through the model's parameters [51].

However, challenges arise when applying DP in an iterative learning setting. Although a single round of FL might preserve privacy to an acceptable extent, privacy leakage can accumulate across multiple rounds of model updates. This cumulative effect undermines the intended privacy protections over time. Despite this limitation, it is important to note that such behavior does not violate the formal definition of DP presented in Equation 3.10. In an iterative setting, the image im $\mathcal{A}$ should be considered as the set of all partial results across iterations. DP could still be applied effectively by dynamically adjusting the privacy parameter, ε, to mitigate the increased privacy risk in multi-round scenarios [1].

Combining FL with SMPC

SMPC offers an ideal complement to FL. Given that FL inherently involves multiple parties working collaboratively, these parties can naturally assume the roles necessary for SMPC protocols, such as additive secret sharing [66]. In this setup, SMPC would ensure that each participating node only learns the final model parameters and not the individual updates or data from other participants.

Hybrid Approaches: FL with DP and SMPC

A more advanced privacy-preserving approach would be to combine both DP and SMPC within an FL framework. In this hybrid model, SMPC can be used to securely aggregate the local model updates, while DP adds an extra layer of noise to the aggregated updates. This would make it extremely challenging for an adversary to reverse-engineer the original data from the shared model parameters [128].

By adopting such hybrid approaches, it is possible to leverage the strengths of multiple privacy-preserving techniques to address the weaknesses or limitations inherent in each individual method. This results in a more robust and secure FL environment.

3.5 Robustness and reproducibility

Robustness and reproducibility are pivotal concerns, not only in the realm of FL but also in ML at large. Addressing these issues substantiates the integrity of the methodological pipeline and fosters trust among researchers and end-users.

Robustness in ML denotes the resilience of a model when faced with input data that deviates from the initial training set. A robust model should not only manage anomalies such as noise and outliers but also accommodate scenarios where the training data is not fully representative. This is especially crucial in healthcare applications, where inaccuracies can have far-reaching implications [69, 30].

Reproducibility, on the other hand, ensures that subsequent researchers can replicate the original study's results using the same data and procedures. This not only maintains transparency and accountability but also facilitates peer verification and further research. Variability in hardware and software configurations can further complicate reproducibility [54].

Both robustness and reproducibility should be rigorously assessed when publishing new methods. This entails supplying all necessary resources for reproduction and evaluating the model's robustness across varied datasets.

In the context of biomedical data generated over the last two decades, an alarming trend has emerged: many AI models lack reproducibility due to poor adherence to best practices in ML. This noncompliance has culminated in opaque and dubious decision-making processes, eroding trust in AI systems [201].

3.5.1 Technical considerations

Traditional ML methods typically run on a single machine and manage a singular dataset. However, federated methods operate across multiple machines, encompass various datasets, and engage multiple researchers, thereby complicating their implementation, deployment, and execution.

Implementation of the core architecture of FL methods incorporates communication steps, enabling local models to share parameters with a central aggregator. These steps are integral to developing the global model, either in one go or iteratively. The sharing process involves the serialization and deserialization of parameters across the network, usually the internet. Proper serialization techniques are essential to ensure that data integrity is maintained during transmission [127].

Deployment, while more commonly used in web development, in an FL context involves ensuring uniformity in the software versions run by all participants. Any updates or changes must be synchronously propagated across all nodes to maintain consistency. This is critical as inconsistent software versions can lead to model divergence [128].

Execution, finally, is more complex, due to the distributed nature of FL, requiring advanced orchestration and synchronization techniques. The aim is to keep all components running in harmony, thereby fulfilling the overarching objectives of the federated model. This involves aligning multiple clocks, coordinating network schedules, and handling node failures efficiently [205].

3.6 Federated learning platform

As outlined in Section 3.5.1, FL requires more technical considerations than classical machine learning due to its complexity. Many of these added complexities could be covered by an integrated platform, significantly reducing the overhead caused by them. Such a platform ideally makes it almost as easy to deal with federated algorithms as classical single-computer applications or scripts.

To achieve that, a thorough list of requirements is necessary, to ensure that the developed system is practically usable and abides by legal and practical restrictions. The requirements set out here can be distinguished into privacy and security-related requirements, technical requirements, and usability-related requirements. All three are crucial for a system that aims to be used in practice.

3.6.1 Requirements

The platform aims to be used for both academic research and in clinical practice. All requirements in this section are therefore aligned with this scenario. Its base technology should be FL, allowing multiple hospitals other research institutions to perform collaborative studies. They were collected within the consortium of the FeatureCloud EU Horizon 2020 project, whose partners have expertise in bioinformatics, law and ethics, data privacy, software development, and explainable AI.

To ensure that the developed system meets all required properties, a list of requirements is provided here to test the implemented system against.

Technical requirements

Technical requirements relate to properties of the system that are necessary from a technical point of view. They play a vital role in the design of an FL platform tailored for hospitals, ensuring essential system properties from a technical perspective, such as scalability and stability.

Req. 1. *The system must run on the major operating systems Linux, Windows and MacOS.*

The platform must be versatile and compatible, capable of running seamlessly on major operating systems including Linux, Windows, and MacOS (Requirement 1). This compatibility empowers hospitals to integrate the platform into their existing infrastructure effortlessly, facilitating widespread adoption and utilization.

Req. 2. *The system must support parallel execution of multiple studies.*

To support efficient collaboration and accelerate research advancements, the system must provide support for parallel execution of multiple studies (Requirement 2). This capability allows hospitals to concurrently work on multiple research projects, maximizing productivity and enabling the rapid dissemination of scientific knowledge within the healthcare community.

Req. 3. *The system must not enforce the usage of a specific programming language.*

Promoting flexibility and adaptability, the system should not impose the usage of a specific programming language (Requirement 3). Hospitals can leverage their preferred programming languages and tools, fostering productivity and reducing barriers to adoption. This flexibility enables hospitals to capitalize on their existing expertise and infrastructure, enhancing efficiency and enabling seamless integration.

Req. 4. *The system must not require opening ports for execution.*

In order to prioritize security and simplify deployment within hospital networks, the system must not require the opening of ports for execution (Requirement 4). By avoiding the need for open ports, the platform enhances security measures and facilitates integration within the hospital's existing network infrastructure.

Req. 5. *The system must be extensible and allow for 3rd-party applications.*

The platform must be designed with extensibility in mind, allowing for the integration of third-party applications (Requirement 5). This extensibility promotes collaboration with external partners, facilitating the integration of cutting-edge technologies and expanding the platform's capabilities to meet evolving healthcare needs.

By addressing these technical requirements, the FL platform empowers hospitals to engage in collaborative research while ensuring compatibility, scalability, security, and flexibility. The platform serves as a robust foundation for facilitating advancements in healthcare by harnessing the collective intelligence of the medical community in a secure and efficient manner.

Privacy and security-related requirements

Since FeatureCloud is an EU Horizon 2020 project, its privacy-related requirements are mostly derived from current legislation in the EU (i.e., the GDPR). Biomedical data, such as omics data, can identify a person. Therefore, it must be considered to be personal data in the sense of the GDPR and cannot be shared with other parties without explicit consent.

To ensure compliance with privacy and security regulations, several key requirements have been identified for the development of a FL platform tailored for hospitals.

Req. 6. *Primary data must not leave the original storage location.*

Requirement 6 mandates that primary data must remain within the original storage location. By adhering to this requirement, the platform ensures that sensitive data does not leave the hospital's designated storage infrastructure, thereby reducing the risk of unauthorized access and maintaining compliance with privacy regulations.

Req. 7. *All data leaving the data location must retain the anonymity of its participants.*

Requirement 7 emphasizes the importance of preserving participant anonymity when data is shared outside the storage location. By implementing robust anonymization techniques, the platform safeguards the identities of individuals within the shared data, promoting secure and privacy-preserving data collaboration among hospitals.

Req. 8. *The system must ensure that 3rd-party applications follow all security and privacy requirements.*

The platform must enforce strict security and privacy requirements for third-party applications integrated into the system (Requirement 8). This requirement guarantees that any external applications meet the necessary standards for data protection and privacy, mitigating potential risks associated with unauthorized access or misuse of sensitive information.

By addressing these privacy-related requirements, the FL platform for hospitals enables secure and privacy-preserving collaboration, ensuring compliance with the GDPR and fostering a trusted environment for data sharing and research advancements in the medical field.

Usability-related requirements

Usability-related requirements aim to make the system usable by its future users at the respective locations. They are of paramount importance in the development of a FL platform tailored for hospitals, as they aim to ensure the system's usability for its future users at their respective locations.

Req. 9. *The system must not require programming skills to be used but provide a graphical user interface.*

To enhance accessibility and ease of use, the platform must provide a graphical user interface (GUI) that does not require programming skills (Requirement 9). By offering a user-friendly GUI, hospitals can effectively utilize the platform's capabilities without the need for specialized programming knowledge. This empowers healthcare professionals to focus on their research and analysis tasks, enabling efficient utilization of the platform's resources and functionalities.

Req. 10. *Running and maintenance of the system must be possible without regular help of technicians.*

In addition to user-friendliness, the system should be designed to allow running and maintenance without regular assistance from technicians (Requirement 10). Hospitals should have the capability to manage and maintain the platform independently, minimizing reliance on technical support. This self-sufficiency ensures that hospitals can efficiently operate the platform, perform necessary updates, and address routine maintenance tasks without disruption.

By addressing these usability-related requirements, the FL platform for hospitals aims to optimize user experience, facilitating the seamless adoption and utilization of the platform across various medical institutions. An intuitive GUI and the ability to independently manage the system's operation and maintenance empower healthcare professionals to leverage the platform's capabilities effectively, promoting collaborative research.

3.7 Results

This section contains results related to privacy-preserving AI, which make use of the methods described in this chapter.

3.7.1 sPLINK and HyFed

The development of sPLINK [141] marks a significant advancement in the realm of privacy-aware genomic analytics. Through the utilization of the HyFed framework [140], sPLINK offers a robust, privacy-preserving platform for GWAS. It serves as an example how federated approaches can be used for a specific problem: large-scale distributed GWAS.

Architecture and privacy preservation

sPLINK's architecture is based on the HyFed framework which incorporates four main components: WebApp, client, compensator, and server (see Figure 3.2). This architecture contributes to privacy preservation (client, server, and compensator) and usability (client and WebApp). The server plays a central role, coordinating the training process and ensuring component synchronization. It is also responsible for calculating the global model by aggregating noisy local models from clients and then subtracting the aggregated compensator noise to reveal the genuine global model. These steps are crucial for maintaining the confidentiality of the original data parameters across cohorts.

The compensator, a lightweight component running on a different machine than the server, maintains the utility of the global model by aggregating noise values from each client and sending this aggregated noise to the server. This feature is integral to sPLINK's capacity for preserving the privacy of the cohorts' data while maintaining the utility of the global model.

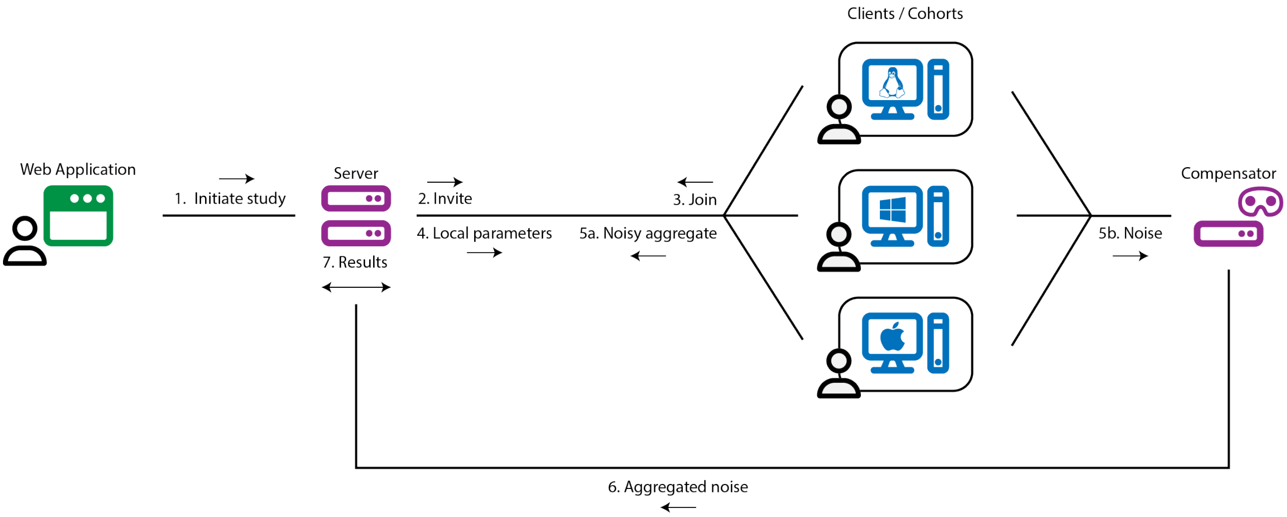


Figure 3.2: A study in sPLINK involves multiple steps. First, the coordinator initiates a study using the web interface (1). Then, other parties are invited by sending tokens to them (2). Using these tokens, they join the study (3). After that, local study parameters are sent to the clients (4). During the computation, clients send noisy parameters to the server and the noise to the compensator (5). The compensator aggregates the noise and sends it to the server (6). Combining the aggregated noisy parameters and the noise aggregate, the server obtains the results and sends them to the clients and displays them on the web interface (7).

Analytical robustness and accuracy

The comparative analysis confirms that sPLINK maintains high levels of analytical accuracy, exhibiting statistically insignificant divergence from the results generated by traditional PLINK. Its robustness against data heterogeneity across cohorts presents a significant improvement over existing meta-analysis tools.

Computational performance and resource utilization

By adopting HyFed's client component, sPLINK benefits from multi-threading and data chunking capabilities. This allows for efficient utilization of computational resources and scalability to handle large datasets, particularly when working with millions of SNPs.

Usability and security considerations

sPLINK's user experience benefits from HyFed's WebApp component, which provides functionalities like account sign-up, project initialization, and real-time tracking of project progress. From a security standpoint, the HyFed architecture ensures authentication protocols for all participants and components involved, enhancing the overall security of the federated GWAS process.

Network efficiency and scalability

The network assessments indicate that bandwidth consumption remains within acceptable bounds, and secure channels (HTTPS) are employed for all communications between client-server, client-compensator, and compensator-server, thus ensuring both efficiency and security.

Summary

sPLINK demonstrates significant potential to set new standards in the realm of secure and collaborative genomic research. Its robust handling of issues like cross-study heterogeneity positions it as a formidable alternative to current meta-analysis techniques. In conclusion, sPLINK emerged as a useful tool that achieves both privacy preservation and analytical accuracy, and balances it with computational efficiency in GWAS.

3.7.2 FeatureCloud

sPLINK shows that AI in biomedical contexts is possible while still preserving privacy for patients. However, it is very limited in its application domain since it is restricted to GWAS. FL has much potential beyond that and the challenged that have to be solved are similar for other applications: inviting other parties to a federated workflow, orchestration and execution of the workflow, and obtaining the results.

FeatureCloud [126] was therefore developed to streamline development of federated methods and take away these efforts (see Figure 3.3). It provides a general solution for the steps involved and enables developers to concentrate on the method they develop rather than setting up the infrastructure around it.

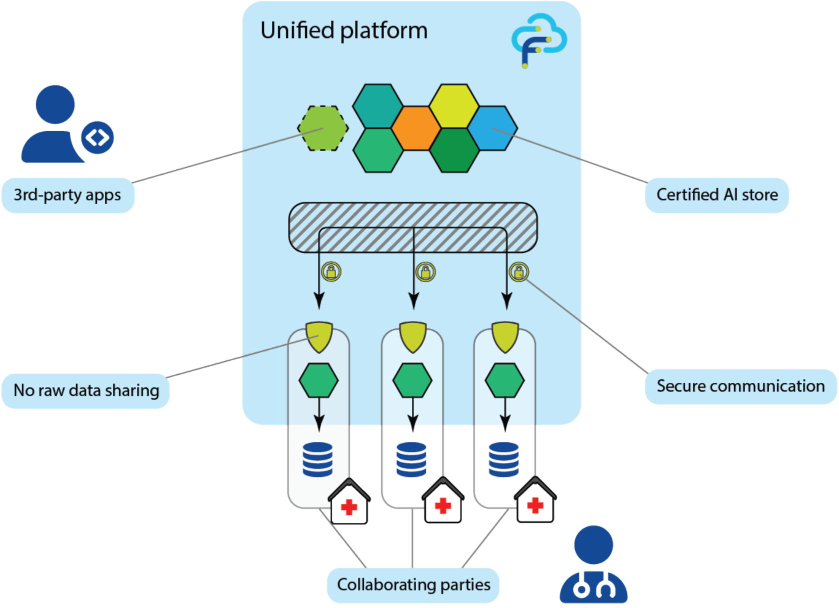


Figure 3.3: Overview of the FeatureCloud Platform: Within a federated study, healthcare facilities retain all raw or primary data on-site. FeatureCloud orchestrates the secure deployment, execution, and intercommunication of vetted AI algorithms available from its dedicated AI Store, catering to both developers and end-users. Source: [126]

Universal platform

FeatureCloud is a universal, algorithm-agnostic platform, i.e. it allows for execution of algorithms of different kinds. This is achieved by offering a generic API which can be used by third-party implementations, handling communication between different participants. These implementations need to be packaged as Docker [131] images to allow their execution independently from the operating system and isolate them for security reasons. Docker remains the only dependency of FeatureCloud. Given that it is available for a wide range of operating systems, this enables FeatureCloud to run on most computer systems.

When a federated workflow is executed, all participating locations need to run the so-called FeatureCloud controller inside a Docker container. This program instructs the Docker engine to download, start and stop the containers running the respective apps required for a workflow.

A workflow can consist of multiple apps which are executed sequentially in a federated fashion. Data is passed between them by putting the output data into a mounted volume, which is then detached from the app container and attached to the next container in the workflow. This way, all data still remains local and only the individual apps in a workflow communicate with each other. The first app receives its input (i.e., the local data) from the user at each site, and the final results can be found in the output volume of the last app.

To use FeatureCloud, users are provided with a GUI, implemented as a web application, thereby running in a standard web browser. For a federated study, one of the participating hospitals needs to assume the role of the 'coordinator', who is in charge of setting up the workflow and inviting other participants to the study. This is done by generating a token and sending it to the invited participants. The invited participants can then join the study using the token. They also have to have the FeatureCloud controller running on their computers. Once all participants have joined, the coordinator starts the study and the FeatureCloud system takes over the execution of the study, until all apps have run through. The results of the federated workflow are automatically shared with all participants, who can download them, again using the web GUI.

AI Store

A key part of the platform is the AI Store (see Figure 3.4), which contains all apps developed by external contributors that can be used within the federated workflows. Technically, these apps are Docker images which contain the app implementations. Using Docker allows the developers to choose their favorite programming technologies, since their implementation remains entirely isolated within the Docker containers. All they need to do is implement the generic FeatureCloud API, which is used to connect the app containers with the controller during a running workflow. The apps can be enriched with meta information, such as a title, description, the type of the app, the employed PETs, a link to the source code and whether the app has been certified.

Certification is another crucial aspect, which is done by the FeatureCloud maintainers and ensures the app actually abides by the requirements, particularly that it does not leak any primary data. Due to the genericity of the apps, it is not feasible to automatically determine whether it complies with all requirements. Therefore, manual vetting of apps is necessary before they can be safely used.

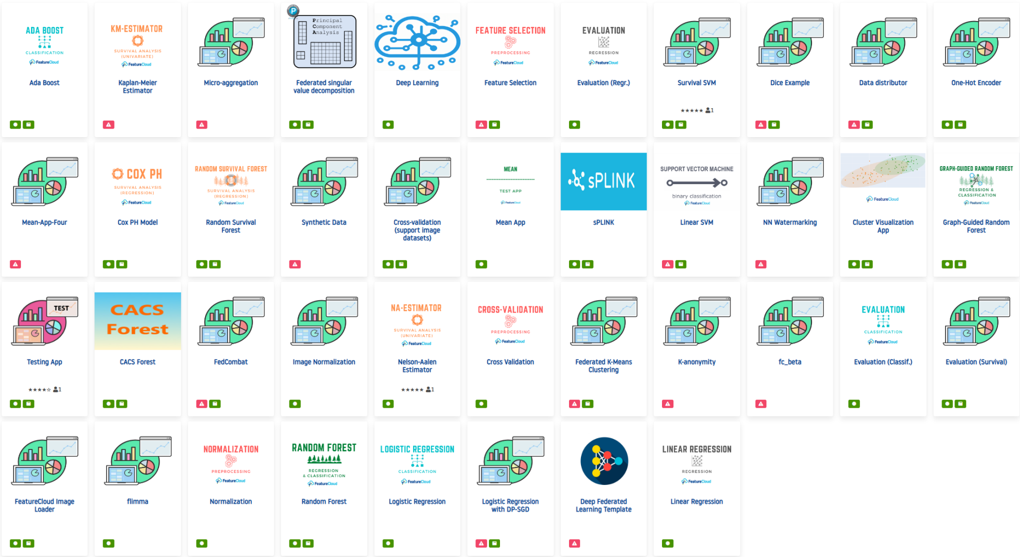


Figure 3.4: ML apps in the FeatureCloud AI Store range from statistical methods (e.g., integrated in an apper version of sPLINK) to neural networks, but also involve preprocessing (e.g., normalization) and evaluation apps.

Privacy-preserving technologies

FL already increases privacy but is not sufficient in some cases, as outlined in Section 3.4. For this reason FeatureCloud supports PETs that can either be used by developers directly, or implemented inside an app.

A basic implementation of SMPC is integrated into the FeatureCloud system and can be enabled via the API. This requires the shared data to be numeric (i.e. fixed-point numbers or integers).

DP must be integrated into the individual applications themselves, as DP cannot be generally 'switched on' without extending the algorithm itself, as outlined in Section 3.4.1.

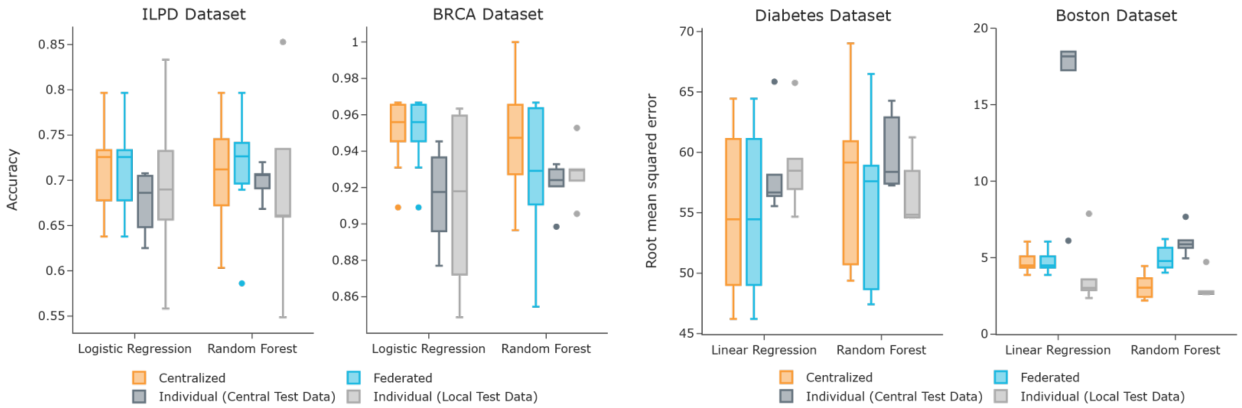


Figure 3.5: The box plots depict 10-fold cross-validation scores for assorted classification and regression algorithms, with the exception of the deep learning model, which used a test set for evaluation. Centralized outcomes are marked in orange, federated outcomes in blue, and individual local outcomes at each participant are in gray shades. Each model's performance was assessed using both the full test set (dark gray) and individual local portions of the test set (light gray). Federated logistic and linear regression models exhibit performance metrics nearly identical to their centralized counterparts, while federated Random Forest and deep learning models demonstrate comparable effectiveness. Source: [126]

Evaluation

In an effort to assess the practical utility of FeatureCloud in the realm of privacy-aware artificial intelligence, several workflows were implemented on the platform. These workflows employed different data sets and were designed to perform both classification and regression tasks. Notably, each workflow adhered to a structured pipeline that included 10-fold cross-validation, standardization, model training, and a final evaluation phase, with the exception of deep learning (DL) tasks that utilized a 20% test set to speed up the training process.

Datasets For classification tasks, the workflows processed the Indian Liver Patient Dataset (ILPD, [161]) and the Cancer Genome Atlas Breast Invasive Carcinoma dataset [184]. Regression workflows utilized the Diabetes dataset [56] and the Boston House Prices dataset [79]. A large dataset from the Survey of Health, Aging, and Retirement in Europe [23] was chosen for DL regression. Following data pre-processing, the latter dataset had 42,894 samples available for model training and evaluation.

Performance analysis The study found that the FeatureCloud-based federated models exhibited performance metrics nearly identical to those generated by centralized scikit-learn [147] models. This was observed for both logistic and linear regression models (see Figure 3.5). The Random Forest (RF) models also showed similar, if not slightly better, performance in the federated learning environment due to the inherent randomness in bootstrapping.

Further comparisons were made between federated models and those trained by individual participants. It was found that models trained by individual participants often underperformed compared to federated models, particularly in local evaluations. This underscores the efficacy of FeatureCloud in generating models that are better generalized.

Scalability and network traffic In assessing FeatureCloud's scalability, varying numbers of participants were considered, ranging from 2 to 8, a range typical for cross-silo studies. Experiments were performed under both regular and throttled internet conditions to gauge the impact of network constraints. Results indicate that the platform can efficiently handle an increasing number of participants with minimal impact on runtime (see Figure 3.6). This suggests that FeatureCloud is not only scalable but also robust enough to be used in diverse settings, including tightly-regulated medical research environments.

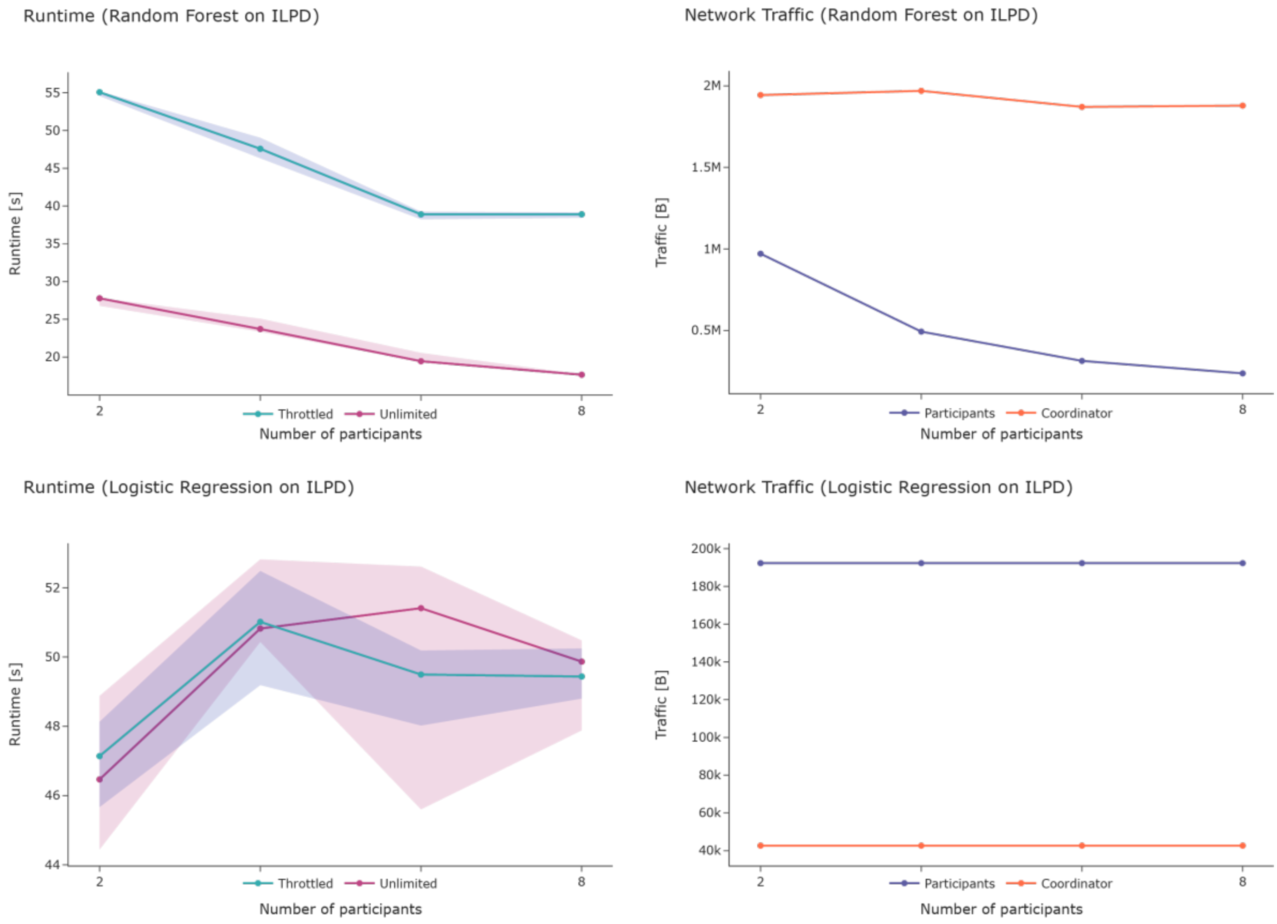


Figure 3.6: The left-hand plots display runtime metrics under both unlimited and bandwidth-restricted (throttled) network conditions, while the right-hand plots provide insights into network traffic for both the coordinator and individual participants. Median values, derived from 10 independent runs, are represented by the lines. Shaded regions around the lines indicate the 25th and 75th percentile range, capturing the variance across the runs. Source: [126]

Compliance with requirements

The technical requirements laid out in Section 3.6.1 have been fulfilled by choosing Docker as a virtualization technique, thereby not requiring a particular OS or programming language (Requirements 1, 3). Parallel execution is possible using multiple Docker containers running in parallel, orchestrated by the FeatureCloud controller (Requirement 2). Fulfillment of Requirement 4, not requiring to open any ports, is ensured by making the controller connect to an outside relay server. The integrated AI Store allows 3rd party developers to publish additional methods (Requirements 5).

Privacy and security-related requirements were also maintained. By employing FL, no patient data leaves the hospitals (Requirements 6, 7). A complete isolation of the running app containers, preventing them from accessing the internet, further contributes to that. All apps published in the AI Store undergo a manual certification process, which ensures Requirement 8.

Usability-related Requirements and 10 were fulfilled as well. Running apps is possible without requiring technical or programming skills via a graphical user interface (9). The same applies to the general maintenance of the system: after setting it up at a hospital, it can be used without continuous maintenance, since updates of apps or the controller are achieved by automatically pulling new Docker images containing required updates (10).

Summary

In summary, FeatureCloud offers a universal platform for FL workflows, simplifying the development and execution process. It supports various algorithms through Docker images, and the FeatureCloud controller manages the orchestration of app containers. The AI Store hosts apps developed by external contributors and ensures their compliance with privacy requirements. By incorporating privacy-preserving technologies, FeatureCloud enhances data privacy and security. The platform enables developers to concentrate on algorithm development, promoting the advancement of FL while preserving patient privacy in biomedical and other domains. All requirements set out in Section 3.6.1 were fulfilled. See Section 4.2 for possible shortcomings and Section 5.2 for potential future extensions.

3.7.3 AIMe registry

The results shown so far indicate that ML is applicable and useful in biomedical settings. A persisting issue that comes with increasing use of this technology is validity and reproducibility of published results, though. To address the lack of adherence to best practices in ML and the reporting of AI methods and results, several guidelines have been proposed in biomedical and clinical research [121].

However, these guidelines do not provide practical means to identify biomedical AIs that do not adhere to recommended best practices. To fill this gap, the AIMe registry [125] for AI in biomedical research was developed as a community-driven platform that allows authors to generate accessible and citable reports of their AI systems[1]. The AIMe registry follows a generic minimal information standard that can be applied to any biomedical AI system. It aims to increase transparency and reproducibility in biomedical AI research and facilitate the adoption of AI systems in clinical settings and is updated regularly based on feedback from the scientific community.

To use the registry, authors fill in a multi-page form that is divided into the sections Metadata, Purpose, Data, Method, and Reproducibility. The form is interactive, i.e. it contains conditional questions or response options based on the user input. For example, when an author specifies that thay have used 'other' test metrics for their method, they are requested to elaborate which ones. The reason behind this is to obtain as precise answers as possible.

[1] https://aime-registry.org

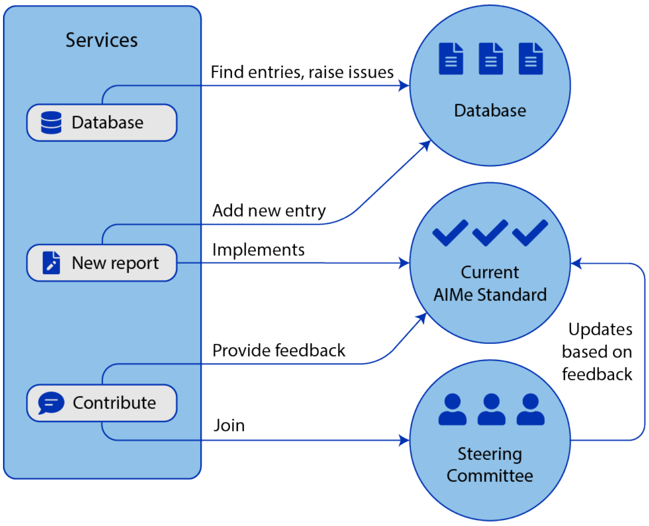


Figure 3.7: The AIMe registry user flow starts encompasses users finding reports, adding new entries, raising issues, and potentially joining the steering committee. New entries are added to the database, where they can be found by other users. Users can raise issues if they have concerns about a report. The steering committee creates new versions of the specification every 1-2 years. Source: [125]

A key feature of AIMe are the validation and reproducibility scores. They provide an estimate on how thorough and complete the validation and reproducibility measures are which the author provided. To achieve that, each answer option to a question can have a validation or reproducibility value, which can be activated by selecting the respective answer. For example, if an author asserts that all means (including dependencies) to easily re-run their AI have been provided, they increase their reproducibility score (see Question R.1.1).

> ”Do you provide all means (including dependencies) to easily re-run your AI?”
> – Question R.1.1 of the AIMe.2021 specification

The main part of the registry is the searchable database. It supports search by keyword, tags or searches within the different parts of the reports. From the list of results, users can directly generate a BibTeX citation code.

AIMe thereby addresses the reporting deficit in biomedical AI research by providing a practical means for authors to report on their AI systems and generate accessible and citable reports and also incentivizes authors to adhere to best practices and complete documentation through the report scores.

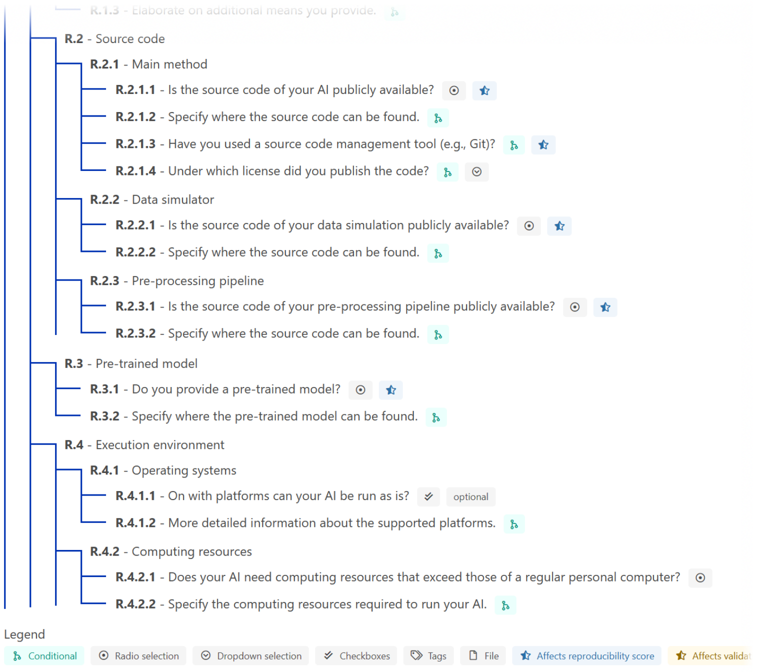
Transparency and governance

If any claims in a report are deemed incorrect, users can raise an issue directly on the registry platform. Authors have a two-week window to respond to correction requests. Failure to do so will result in the issue being published and attached to the report, further augmenting the platform’s commitment to transparency.

Governance of AIMe is managed by a steering committee, open to all researchers. The committee is responsible for ongoing updates to the registry’s specification. The executive board, a subset of the steering committee, oversees the issue resolution process to ensure fairness and transparency.

3.7.4 Summary

The field of privacy-preserving AI has given rise to a variety of specialized tools designed for unique applications. One such tool is sPLINK, which focuses on Genome-Wide Association Studies (GWAS). On the other hand, the AIMe registry serves as a comprehensive resource for providing detailed information about a broad array of machine learning methodologies.

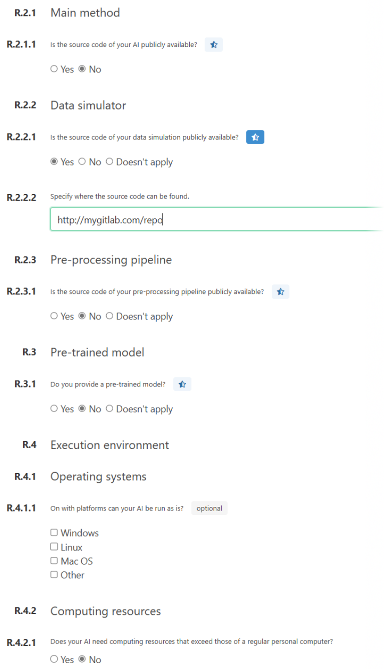


R.1.3 - Elaborate on additional means you provide.

- **R.2** - Source code
  - **R.2.1** - Main method
    - **R.2.1.1** - Is the source code of your AI publicly available?
    - **R.2.1.2** - Specify where the source code can be found.
    - **R.2.1.3** - Have you used a source code management tool (e.g., Git)?
    - **R.2.1.4** - Under which license did you publish the code?
  - **R.2.2** - Data simulator
    - **R.2.2.1** - Is the source code of your data simulation publicly available?
    - **R.2.2.2** - Specify where the source code can be found.
  - **R.2.3** - Pre-processing pipeline
    - **R.2.3.1** - Is the source code of your pre-processing pipeline publicly available?
    - **R.2.3.2** - Specify where the source code can be found.
- **R.3** - Pre-trained model
  - **R.3.1** - Do you provide a pre-trained model?
  - **R.3.2** - Specify where the pre-trained model can be found.
- **R.4** - Execution environment
  - **R.4.1** - Operating systems
    - **R.4.1.1** - On with platforms can your AI be run as is? optional
    - **R.4.1.2** - More detailed information about the supported platforms.
  - **R.4.2** - Computing resources
    - **R.4.2.1** - Does your AI need computing resources that exceed those of a regular personal computer?
    - **R.4.2.2** - Specify the computing resources required to run your AI.

Legend

(a) The AIMe question tree displays the whole specification as a tree. Icons the the right show the type of the question entry (e.g., optional, dropdown) and whether they affect a score.



R.2.1 Main method

R.2.1.1 Is the source code of your AI publicly available?

○ Yes ● No

R.2.2 Data simulator

R.2.2.1 Is the source code of your data simulation publicly available?

● Yes ○ No ○ Doesn't apply

R.2.2.2 Specify where the source code can be found.

http://mygitlab.com/repo

R.2.3 Pre-processing pipeline

R.2.3.1 Is the source code of your pre-processing pipeline publicly available?

○ Yes ● No ○ Doesn't apply

R.3 Pre-trained model

R.3.1 Do you provide a pre-trained model?

○ Yes ● No ○ Doesn't apply

R.4 Execution environment

R.4.1 Operating systems

R.4.1.1 On with platforms can your AI be run as is? optional

□ Windows
□ Linux
□ Mac OS
□ Other

R.4.2 Computing resources

R.4.2.1 Does your AI need computing resources that exceed those of a regular personal computer?

○ Yes ● No

(b) The AIMe specification is automatically rendered as a web form for users.

Figure 3.8: The AIMe specification contains a series of questions diveded into multiple sections (e.g., method, reproducibility, validation).

For those seeking a more universal solution within the realm of biomedicine, FeatureCloud emerges as a holistic platform that incorporates multiple facets of AI techniques while maintaining stringent privacy standards.

A more in-depth discussion on the significance and implications of these contributions to privacy-preserving AI is covered in Section 4.2. Future prospects and possible enhancements to these existing systems will be explored in Section 5.2.

4
Discussion

This section discusses the results presented in this thesis in the area of systems medicine and privacy-aware AI, following the previous structure.

4.1 Systems medicine

Systems medicine has been introduced as a paradigm shift in contemporary biomedical research, where it has emerged as a promising and integrative discipline. It leverages the power of computational tools and mathematical models to decipher the complexity of biological systems, enabling a more nuanced understanding of the intricate interplay between genes, proteins, and cellular networks [86, 203, 119].

4.1.1 Contributions of CoVex

CoVex, the coronavirus explorer, was presented as an innovative and comprehensive systems medicine approach for the analysis and interpretation of SARS-CoV-2 data. The principal aim of CoVex is to facilitate the exploration and understanding of the complex molecular interactions underlying the virus-host relationship and to accelerate the development of targeted therapeutic strategies against COVID-19, in particular repurposed drugs.

This research has showcased the potential of CoVex as a versatile tool to investigate the multifaceted aspects of coronavirus infection. By integrating proteomics and interactomics, CoVex provides a holistic and systems-level perspective of the viral disease modules. This approach potentially allows for the identification of novel drug targets, prediction of drug repurposing candidates, and elucidation of host-pathogen interaction dynamics.

4.1.2 Challenges and limitations

Although the utility of systems medicine has been underscored by the rapid dissemination of large-scale, high-dimensional data sets [61, 145, 174], there are inherent challenges.

CoVex serves as an example of how systems medicine can be employed to analyze and integrate this wealth of information, potentially leading to actionable insights and discoveries that may not be possible through traditional reductionist approaches, since they lie in the interplay between interacting elements.

While some studies like Gordon et al. [70] have used similar approaches, their focus has been more narrow, not considering the wider network context.

However, the quality of the curated network in CoVex is directly tied to the quality of the data available in the referenced sources. The tool does not distinguish between drug-target interactions based on differing sources or experimental rigor. This way the network can be used by the integrated algorithms introduced in Section 2.3 but lacks potentially important information.

It should also be stressed again that CoVex does not list drugs directly targeting viral proteins, focusing instead on unveiling indirect drug targets within the human interactome.

Furthermore, even though CoVex serves as a useful tool in SARS-CoV-1 and -2 research, findings from such platforms demand careful interpretation. They merely propose potential drug candidates, whose antiviral properties are not assured and require further exploration. Proposed drugs target virus-associated proteins, but their actual effect necessitates confirmation via additional investigations. In some cases, a drug that inhibits a cofactor could even bolster the virus. Therefore, after establishing a target, potential drug candidates must undergo thorough review and testing by clinical experts, adhering to established procedures and clinical trials.

Although CoVex offers valuable insights into the interactome, it has not led to the discovery of a successful drug against COVID-19 and must therefore be regarded as just a one piece in the search for novel SARS-CoV-2 targets. This underscores the challenges inherent in translating insights from complex network analyses into viable, effective therapeutics.

4.1.3 Future implications and extensions

Despite its limitations, CoVex has the potential for broader applications beyond COVID-19. It could be adapted to investigate other infectious diseases, providing valuable insights into their molecular mechanisms (see Section 5.1.1). This adaptability makes CoVex a promising approach in the emerging landscape of systems medicine tools. Yet, findings from platforms like CoVex need careful interpretation; proposed drug candidates must undergo thorough review and clinical trials.

4.2 Privacy-aware AI

The rapid advancements in the field of privacy-aware AI have given rise to a variety of tools, each with its unique strengths and distinct purposes. Two such tools presented in this thesis, sPLINK and FeatureCloud, exemplify the two ends of the spectrum: standalone tools designed for specific purposes and infrastructural platforms supporting a variety of applications.

4.2.1 Implementation and dissemination

sPLINK serves as an exemplary model of a standalone tool specifically designed for GWAS. This user-friendly, hybrid federated tool enhances the privacy of cohort data without compromising the accuracy of the test results. It is consistent with existing tools like PLINK in terms of data formats and results, thereby providing a seamless experience for the users. The tool exhibits robustness against the heterogeneity of phenotype distributions and confounding factors across datasets. With practical runtime and acceptable network usage, sPLINK demonstrates its value as a tool that performs collaborative GWAS in a privacy-aware manner. Its specificity and fine-tuned design cater directly to the needs of GWAS [180], making it a robust and efficient standalone tool for this purpose.

On the other end of the spectrum, FeatureCloud presents itself as a comprehensive and extensible platform, serving as an infrastructure for the development and application of privacy-preserving FL workflows. The power of FeatureCloud lies in its high generalization capability, allowing for the application of a myriad of machine learning workflows across a wide variety of data types. FeatureCloud also offers pre-built solutions for common use-cases and application templates for developers, further showcasing its versatility. However, the platform's openness and flexibility present challenges, including the need for standard data formats and preprocessable initial data. As a platform, FeatureCloud is designed to support a broad range of applications, thereby providing a foundation for developing diverse privacy-aware AI tools.

The contrast between sPLINK and FeatureCloud represents the diversity in design and application in privacy-aware AI tools. Standalone tools like sPLINK, with their specific focus and design, can provide targeted and efficient solutions for particular use-cases. In contrast, infrastructure platforms like FeatureCloud provide the necessary foundation and support to develop a wide range of applications, offering flexibility and versatility.

The coexistence and cooperation of standalone tools and infrastructural platforms can significantly boost the progress in the privacy-aware AI field. While standalone tools offer expertise and efficient solutions for specific tasks, platforms like FeatureCloud can facilitate their implementation and extend their reach. By acknowledging the strengths of both types of tools, the AI community can work towards more holistic, robust, and privacy-aware solutions.

As the computational demands of biomedical research continue to increase, the scalability of tools like sPLINKand FeatureCloud becomes a critical concern [117, 22]. Although the current versions perform admirably in benchmark tests, their ability to handle ever-expanding data sets is a question that merits further investigation.

4.2.2 Reproducibility and transparancy

With AI methods comes the necessity for clear, transparent descriptions of the AI methods employed [149]. A lack of transparency and standardization in the reporting of AI methodologies can hinder the reproducibility, comparison, and evaluation of biomedical AI's results, thereby obstructing the progress of AI in research and practice.

In the context of biomedical AI, reproducibility and transparency of AI methods are paramount. With AI becoming more prevalent in biology and medicine, basic information about data, methods, and AI implementation is often found lacking in relevant publications. This gap constitutes a significant hurdle for developing new AI methods and for applying AI in research and practice [172, 200, 185].

To bridge this gap, the community-driven AIMe registry has been developed for the rapid proliferation of AI in biology and medicine, providing a community-driven registry. It ensures the quality, reliability, and reproducibility of biomedical AI systems by allowing authors to register their AI tools and enabling researchers and practitioners to find existing AI systems relevant for their work. This not only increases the accessibility of information about these systems but also aids researchers and practitioners in finding AI systems relevant to their application scenarios. Through this, the quality, reliability, and reproducibility of biomedical AIs can be significantly improved.

4.2.3 Regulation and public perception

As privacy-aware AI tools like FeatureCloud gain traction, they will invariably come into contact with existing and evolving data protection laws, such as GDPR in the European Union [33]. Navigating this complex regulatory landscape will be a vital aspect of future development in the field.

Another significant challenge facing the adoption of privacy-aware AI tools is public perception. Public understanding and acceptance are key to the broader implementation of these technologies. Hence, future efforts may need to focus on not just technical development but also public engagement and education [199].

4.3 Summary

The discussion of both systems medicine and privacy-aware AI demonstrates the nuanced complexities and promise inherent in these emerging disciplines. While standalone tools like CoVex and sPLINK showcase the focused, specialized utility of domain-specific approaches, platforms such as FeatureCloud offer versatility, serving as scaffolds for a multitude of applications.

One of the compelling insights from this research is that the coexistence of these diverse tools can form a synergistic landscape, allowing each to contribute its own strengths to the overarching aim of advancing science and medicine. For instance, standalone tools can become modules within broader infrastructural platforms, potentially offering pre-validated solutions that are integrated into more complex systems.

However, the gaps in reproducibility, data quality, and regulatory compliance present hurdles that can potentially slow down progress in both fields. The creation of a registry like AIMe is a step in the right direction, but further work is needed to improve standardization, testing, and awareness in the research community.

As we move toward a future where approaches like those implemented in these tools and platforms increasingly influence biomedical research and practice, their ethical and societal implications must be carefully considered. This presents not only a technical challenge but also necessitates a multi-disciplinary approach involving policy makers, clinicians, and the public.

While this thesis contributes to the understanding and development of tools and methodologies in systems medicine and privacy-aware AI, it also raises several questions that merit further investigation, like how we can better ensure the reliability of network data in systems medicine tools such as CoVex or how privacy-aware AI tools can scale to meet the increasing data demands without sacrificing performance or privacy, particularly when combined with a federated database.

5 Conclusion and outlook

This section provides an outlook on possible extensions to the tools introduced in this thesis and potential further research.

5.1 Systems medicine

In reviewing the landscape of computational biology and drug discovery, it becomes apparent that platforms fostering collaborative research and providing quick access to data like CoVex can play an important role in the exploration and understanding of complex biological systems. CoVex, a data-driven web-based platform, made a contribution to our knowledge of the SARS-CoV-2 virus-host interaction landscape during the COVID-19 pandemic. It integrated multiple layers of information, including virus-host protein-protein interactions, human protein-protein interactions, and gene expression data, offering an in-depth view of the interactions within the human host.

The fact that no drug against SARS-CoV-2 could be found during the Covid-19 pandemic does not devalue the potential of such technology. The network medicine approach, which forms the foundation of CoVex, can find vast applicability in other disease contexts [129], such as oncology. Cancer is a complex disease that involves various genetic and epigenetic alterations [146], leading to disruptions in normal cellular processes and pathways. Understanding these intricate interactions requires a systems-level view, which is where network medicine comes in. The data-driven approach adopted by CoVex could provide meaningful insights, potentially leading to innovative therapeutic strategies.

5.1.1 Extensions

An example of how the approaches underlying CoVex can be extended to other disease contexts is the Cancer Driver Drug Interaction Explorer (CADDIE) [81]. CADDIE is a web application designed to systematically discover drug repurposing candidates in oncology, whose implementation is based on CoVex. It integrates a vast array of information, from human gene-gene and drug-gene interactions to cancer driver genes and their respective mutation frequencies. The tool not only identifies potential drug targets but also aids in the selection of therapeutic options based on network medicine algorithms, akin to the network medicine approach in CoVex. This illustrates the adaptability and utility of CoVex's foundational principles in tackling complex diseases like cancer, thus underlining the broad applicability of such technology in systems medicine.

Another innovative extension is Drugst.One[1] [80]. It aims to serve as a customizable plug-and-play solution for biomedical web-application developers, further generalizing the network medicine approach. Drugst.One integrates various databases to provide a feature-rich network explorer that can help identify drug targets and assess drug repurposing potential. Like CoVex, it also offers a multi-omics and network-based approach for understanding complex biological systems, but goes a step further by making it easily integrable into various web applications.

These tools—CADDIE and Drugst.One—provide glimpses into the future of systems medicine, signaling the onset of a more universal or generalized framework that could seamlessly integrate various layers of interaction data. As these platforms evolve, we can anticipate a streamlined process for investigating interactomes across a broad spectrum of diseases, thereby accelerating the path from discovery to drug development, and ultimately, benefiting patients across the globe.

5.1.2 Conclusion

The success of CoVex, as measured not by immediate therapeutic discovery but by its conceptual contributions, reception by the research community and its future potential, show the enduring value of a network medicine approach in the evolving field of computational biology and drug discovery. We are only just beginning to tap into the potential of these approaches, and the future, it seems, holds promising avenues for exploration and discovery in this field.

5.2 Privacy-aware AI

Reflecting on the recent developments in privacy-aware AI, encouraging advancements represented by systems such as FeatureCloud and sPLINK can be seen. These tools and platforms have demonstrated that infrastructural challenges, particularly in the domain of secure and privacy-preserving data analysis, can be effectively addressed through thoughtful design and robust technologies.

FeatureCloud, with its generalized architecture, offers a practical solution to these challenges. It serves as a proof-of-concept that privacy-aware AI systems are not just theoretically possible, but can be implemented successfully. The system's application in various fields, including bioinformatics and health informatics, and the current number of available apps underscore its versatility and effectiveness.

sPLINK, both as a standalone tool and subsequently as FeatureCloud app demonstrates the practical applicability of such platforms in the context of GWAS. By integrating SMPC methodologies, sPLINK has proven the feasibility of conducting complex genetic analyses without compromising the privacy of individual-level data. This marks an advancement in the genomics field, where the balance between data utility and privacy has always been a challenging issue.

5.2.1 Extensions

Looking forward, there are potential extensions to this general framework. The integration of more advanced PETs into the FeatureCloud API, such as HE, could further enhance the privacy-preserving capabilities of the system, allowing for even more complex computations to be performed securely [136].

Apart from that, the AIMe registry providing extensive means to report on how results can be reproduced, how validity of the results is ensured, as well as privacy-related mechanisms put in place. AIMe and FeatureCloud therefore could be integrated by requiring each app to provide an AIMe report, providing all these details.

A second direction is the development of a federated database system. This could automate the selection of participants with suitable data, streamlining the setup process for various analyses. Not only would this

[1] https://pypi.org/project/drugstone/

make it easier to initiate a workflow, it could also eliminate the need for time-consuming manual steps such as curating and formatting data, thereby improving the efficiency of research and analysis workflows [159].

This currently poses a problem for the seamless integration of federated learning platforms into existing systems. Various data formats exist and cannot be mapped easily onto other formats required by a particular system. This could be solved by establishing standardized data formats. Such efforts are ongoing but have not been broadly adopted yet for analyses [129].

In conclusion, FeatureCloud and sPLINK exemplify the promise of privacy-aware AI, particularly FL. With future developments and enhancements, platforms such as FeatureCloud have the potential to impact various fields, from genomics to health informatics and beyond, all while prioritizing privacy and data security.

5.2.2 Conclusion

The intersection of standalone tools such as sPLINK, the universal FeatureCloud platform and the AIMe registry presents a broad picture of the landscape of privacy-aware AI. Registries like AIMe ensure that the development and application of these tools are transparent, reproducible, and reliable.

The coexistence of standalone tools, infrastructural platforms, and transparency-focused registries like AIMe can propel significant advancements in the fields of systems medicine and privacy-aware AI. They not only offer robust and efficient solutions for diverse applications but also ensure the transparency and reproducibility necessary for the sustainable growth of AI in biology and systems medicine.

5.3 Future of medicine

As we look toward the horizon of biomedical research, the future of medicine is increasingly exciting, teeming with opportunities awaiting to be unearthed. This thesis has explored network algorithms in systems medicine for drug repurposing, a universal platform for machine learning, and the ethical imperative of privacy-aware AI. While each of these fields alone holds promise, their convergence may very well yield big advancements.

5.3.1 A new paradigm

The advent of large-scale data analysis and machine learning has laid the groundwork for a paradigm shift in medicine, transcending organ-specific or symptom-based understanding of diseases. With platforms like the ones introduced in this thesis, we are not merely peering into a kaleidoscope of biological processes, but we are trying to actively untangle the webs of complexity that underlie diseases as intricate as cancer or as emergent as COVID-19 used to be. As these platforms and their underlying approaches become more nuanced, integrating genomic, proteomic, and metabolomic data, a holistic view of human health will evolve. This paves the way for more targeted treatments, effective prevention strategies, and even cures for diseases that have long perplexed humanity.

ML is not just a tool in this new paradigm; it is the compass by which we navigate this complex landscape. As this thesis has shown, federated learning platforms like FeatureCloudcan provide the means to bridge the gap between data availability and privacy, allowing researchers worldwide to collaborate on a scale never before imagined. The marriage of ML with federated systems provides a model for how future research can be conducted—efficiently, securely, and inclusively.

While technological advancements herald a new era, the ethical considerations of privacy and data security become more critical than ever. The GDPR and similar legislation worldwide act as the ethical backbone, ensuring that innovation doesn't compromise individual privacy. Tools like FeatureCloudand sPLINK help pave the way for a future where data privacy and scientific advancement are not mutually exclusive but harmoniously coexistent.

5.3.2 In closing

The medical community stands on the brink of a big step forward—one that melds data, computation, and ethics into a unified, powerful force for human health. This thesis has explored some elements of this impending transformation. Yet, we have barely scratched the surface, given the advances we have seen in the past. Systems medicine, ML, and privacy-aware AI will continue to enrich our understanding of health and disease. And as we move forward, the question is not if these technologies will redefine medicine, but how quickly can we harness their full potential.

Summary

This thesis explored the burgeoning fields of artificial intelligence (AI) within systems medicine, laying a foundational understanding for its two main pillars: systems medicine and privacy-aware AI. While the intersection of these domains may seem initially distant, the present investigation demonstrated that their conjunction has considerable potential for modern medical research.

Regarding systems medicine, the thesis first detailed the utility of integrated Protein-Protein Interaction (PPI) networks. The CoVex tool stands out as a robust example, originally conceptualized for investigating the intricacies of SARS-CoV-2 pathways. Beyond its original intent, CoVex serves a conceptual basis (e.g., demonstrated by CADDIE), allowing researchers across the globe to delve into various disease pathways, making it a valuable exploratory resource. Drug repurposing was also underscored, gaining attention especially in rapid response scenarios like pandemics, due to its quick availability. Through the explanation of pertinent algorithms like centrality measures, TrustRank, and Multi-Steiner Trees, the thesis elucidated how such tools can be leveraged for potential therapeutic interventions. The specific application on integrated PPI networks merged with viral proteins and extant drugs demonstrates an intriguing blueprint for drug repurposing strategies, which can further contribute to pharmacological research in the future.

However, despite the promising strides in systems medicine, the remaining challenges were discussed. The robustness of any computational tool hinges largely on the quality of the data it processes. For CoVex, while it offers a novel approach in understanding the virus-host dynamics and advancing drug repurposing, its outcomes necessitate judicious interpretation. Researchers are required to ensure that findings from such tools undergo rigorous clinical validations, given the high stakes associated with medical interventions. Furthermore, while the tool has the potential to be extended to other diseases, the interpretative caution remains paramount.

Subsequently, the realm of privacy-aware AI was explored, underscoring its pivotal role in the contemporary data landscape. With the proliferation of massive datasets in genomics and other omics data, and the imperative to maintain data privacy due to current legislation such as the General Data Protection Regulation (GDPR) in the European Union, tools like sPLINK present an innovative stride. By fostering multi-institutional Genome-Wide Association Studies via federated learning (FL), it exemplifies the potential to conduct intricate analyses without compromising on data security. Such pioneering endeavors in FL were further expanded by the FeatureCloud platform. By democratizing the FL concept across diverse algorithms, and enriching it with additional privacy-enhancing techniques (PETs), it circumvents many developmental challenges, thus accelerating research endeavors. Moreover, with the increasing influx of machine learning (ML) methodologies, ensuring their veracity and reproducibility becomes paramount. The AIMe registry fills this void, advocating for rigorous standards and reporting in the ML domain, thereby fortifying the research ecosystem's integrity.

Delving into ML within privacy-aware AI, the thesis transitioned from the foundational concepts to more complex methodologies. Beginning with the rudiments of models and optimizers, the thesis then delved into prevalent ML paradigms, ranging from linear regression and decision trees and random forests to the more complex methods provided by neural networks. Complementing these, the thesis also explained the emergent PETs in more detail, namely differential privacy and secure multi-party computation. Each technique bolstered understanding of robust data security in collaborative environments, notably in sensitive areas like medical research.

Alongside the technical advancements, the ethical and regulatory dimensions that shape the field were emphasized as well. With tools like sPLINK, offering dedicated solutions, and platforms like FeatureCloud, providing broader adaptability, the biomedical research community finds itself equipped with a diverse toolkit. Yet, the path ahead is multifaceted. The challenge isn't solely technical; it also encompasses ensuring explainability, navigating intricate data protection laws, and fostering positive public perceptions. The involvement of community-driven initiatives, exemplified by the AIMe registry, underscores the need for collective responsibility in guiding the next phases of biomedical AI.

In conclusion, this dissertation shows the potential of privacy-aware AI in systems medicine while ensuring robust privacy measures. As the medical landscape evolves, insights and tools like the ones emanating from this research can play an important role, contributing to modern medical research.

Zusammenfassung

Diese Dissertation untersucht die aufstrebenden Bereiche der künstlichen Intelligenz (KI) innerhalb der Systemmedizin und vertieft das Verständnis für ihre zwei Hauptpfeiler Systemmedizin und datenschutzbewusste KI. Auch wenn die Bereiche zunächst entfernt erscheinen, zeigt die Arbeit, dass sie in Kombination großes Potenzial für die moderne medizinische Forschung darstellen.

Bezüglich der Systemmedizin beschreibt die Arbeit zuerst die Vorteile von integrierten Protein-Protein-Interaktionsnetzwerken (PPI). Das CoVex-Tool sticht als Beispiel hervor, das ursprünglich zur Untersuchung von SARS-CoV-2-Krankheitspfaden konzipiert ist. Jenseits seiner ursprünglichen Zielsetzung dient CoVex als konzeptionelle Grundlage (z.B. demonstriert durch CADDIE), die es Forschern weltweit ermöglicht, verschiedene Krankheitswege zu erkunden, und macht es somit zu einer wertvollen Ressource für die medizinische Forschung. Insbesondere der Einsatz zur Arzneimittel-Umwidmungen wird hervorgehoben, vor allem in Szenarien wie Pandemien, wo seine schnelle Verfügbarkeit eine wichtige Rolle spielen kann. Durch die Vorstellung relevanter Algorithmen wie Zentralitätsmaßen oder dem TrustRank- und Multi-Steiner-Tree-Algorithmus, zeigt die Arbeit, wie derartige Tools potenziell für therapeutische Interventionen eingesetzt werden können. Deren gezielte Anwendung auf integrierte PPI-Netzwerke, integriert mit viralen Proteinen und bereits bekannten Medikamenten, lässt sie als vielversprechende Strategie zur Medikamentenumwidmung erscheinen.

Trotz dieser vielversprechenden Fortschritte in der Systemmedizin, werden auch die damit verbundenen Herausforderungen diskutiert. Die Robustheit eines jeden datenverabeitenden Tools hängt stark von der Qualität der zu Grunde liegenden Daten ab. Bei CoVex, das zwar einen neuartigen Ansatz darstellt, um die Dynamik zwischen Virus und Wirt zu verstehen, und dadurch etwaige Medikamentenumwidmungen zu ermöglichen, müssen die Ergebnisse mit Vorsicht interpretiert werden. Forscherinnen und Forscher müssen dabei sicherstellen, dass die Ergebnisse einer strengen klinischen Validierung unterzogen werden, angesichts der hohen Risiken, die mit medizinischen Eingriffen verbunden sind. Obwohl das Tools das Potenzial besitzt, auf weitere Krankheiten ausgedehnt zu werden, bleibt die medizinische Deutung der Ergebnisse also von größter Wichtigkeit.

Darauffolgend wird der Bereich der datenschutzbewussten KI untersucht, wobei ihre zentrale Rolle in der gegenwärtigen Datenlandschaft betont wird. Mit der Verbreitung von Datensätzen enormer Größe in der Genomik und anderen Omicsdaten und der Notwendigkeit, den Datenschutz aufgrund der aktuellen Gesetzgebung, wie der Datenschutz-Grundverordnung (DSGVO) der Europäischen Union, zu wahren, stellt das sPLINK-Tool einen wertvollen Beitrag dar. Indem es institutionenübergreifende genomweite Assoziationsstudien mittels föderiertem Lernen ermöglicht, wird demonstriert, dass komplexe Analysen durchgeführt werden können, ohne die Datensicherheit zu beeinträchtigen. Solche durch föderiertes Lernen ermöglichte Ansätze werden dann durch die FeatureCloud-Plattform weiter ausgebaut. Indem sie das Konzept des föderierten Lernens generalisiert und es mit zusätzlichen Techniken zur Erhöhung der Privatsphäre anreichert, erleichtert sie einige Probleme bei der Entwicklung solcher Algorithmen und beschleunigt damit die Forschung in diesem Bereich. Zudem wird mit dem steigenden Aufkommen von Methoden des maschinellen Lernens (ML) die Sicherstellung ihrer Korrektheit und Reproduzierbarkeit immer wichtiger. Das AIMe-Register nimmt sich dieses Problems an und regt strenge Standards und vollständiges Reporting im ML-Bereich an, wodurch die Integrität des Forschungsökosystems gestärkt wird.

Im Zuge der Vorstellung von datenschutzbewusster KI beginnt die Arbeit mit den Grundkonzepten und führt weiter zu komplexeren Methoden. Nachdem die Grundlagen von Modellen und Optimierern erläutert wurden, widmet sich die Arbeit dann gängigen ML-Methoden, angefangen bei linearer Regression und Entscheidungsbäumen bishin zu komplexeren Methoden, wie neuronalen Netzen. Die Arbeit erklärt auch die aufgekommenen Techniken zur Verbesserung der Privatsphäre, insbesondere Differential-Privacy und Secure-Multi-Party-Computation. Neben föderiertem Lernen zeigen diese Techniken, wie Datensicherheit in kollaborativen Umgebungen, insbesondere in sensiblen Bereichen wie der medizinischen Forschung, gewährleistet werden können.

Neben den technischen Fortschritten werden auch die ethischen und regulatorischen Dimensionen diskutiert, die in diesem Feld anzutreffen sind. Mit Tools wie sPLINK, die gezielte Lösungen für ein bestimmtes Problem (hier GWAS) bieten, und Plattformen wie FeatureCloud, die universell einsetzbar sind, ist die biomedizinische Forschungsgemeinschaft mit einer großen Bandbreite ausgestattet. Die Herausforderungen sind jedoch nicht nur technischer Natur; auch die medizinische Erklärbarkeit der generierten Modelle, die Einhaltung der Datenschutzgesetze und eine positive öffentliche Wahrnehmung sind wichtige Elemente. Gemeinschaftliche Initiativen, in dieser Arbeit verkörpert durch das AIMe-Register, können hierzu einen wichtigen Beitrag leisten.

Zusammenfassend zeigt diese Dissertation das Potenzial von KI in der Systemmedizin auf, bei gleichzeitiger Gewährleistung des Datenschutzes. Während sich die medizinische Forschungslandschaft diesbezüglich weiterentwickelt, können die in dieser Arbeit vorgestellen Ansätze eine wichtige Rolle spielen.

References

[1] Martin Abadi et al. “Deep learning with differential privacy”. In: *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 2016, pp. 308–318.

[2] Animesh Acharjee et al. “Integration of multi-omics data for prediction of phenotypic traits using random forest”. In: *BMC bioinformatics* 17.5 (2016), pp. 363–373.

[3] Ajit Agrawal, Philip Klein, and Ramamoorthi Ravi. “When trees collide: An approximation algorithm for the generalized Steiner problem on networks”. In: *Proceedings of the twenty-third annual ACM symposium on Theory of computing*. 1991, pp. 134–144.

[4] B. Alberts, A. Johnson, J. Lewis, et al. *Molecular Biology of the Cell*. 4th. New York: Garland Science, 2002.

[5] Alexander Aliper et al. “Deep learning applications for predicting pharmacological properties of drugs and drug repurposing using transcriptomic data”. In: *Molecular pharmaceutics* 13.7 (2016), pp. 2524–2530.

[6] Jane F Armstrong et al. “The IUPHAR/BPS Guide to PHARMACOLOGY in 2020: extending immunopharmacology content and introducing the IUPHAR/MMV Guide to MALARIA PHARMACOLOGY”. In: *Nucleic acids research* 48.D1 (2020), pp. D1006–D1021.

[7] Ted T Ashburn and Karl B Thor. “Drug repositioning: identifying and developing new uses for existing drugs”. In: *Nature reviews Drug discovery* 3.8 (2004), pp. 673–683.

[8] Charles Auffray, Dominique Charron, and Leroy Hood. *Predictive, preventive, personalized and participatory medicine: back to the future*. 2010.

[9] Varsha Dave Badal et al. “Challenges in the construction of knowledge bases for human microbiome-disease associations”. In: *Microbiome* 7.1 (2019), pp. 1–15.

[10] Albert-Laszlo Barabasi and Zoltan N Oltvai. “Network biology: understanding the cell’s functional organization”. In: *Nature reviews genetics* 5.2 (2004), pp. 101–113.

[11] Albert-László Barabási. “The network takeover”. In: *Nature Physics* 8.1 (2012), pp. 14–16.

[12] Albert-László Barabási, Natali Gulbahce, and Joseph Loscalzo. “Network medicine: a network-based approach to human disease”. In: *Nature reviews genetics* 12.1 (2011), pp. 56–68.

[13] Julia M Barbarino et al. “PharmGKB: a worldwide resource for pharmacogenomic information”. In: *Wiley Interdisciplinary Reviews: Systems Biology and Medicine* 10.4 (2018), e1417.

[14] Eric Bauer and Ron Kohavi. “An empirical comparison of voting classification algorithms: Bagging, boosting, and variants”. In: *Machine learning* 36.1 (1999), pp. 105–139.

[15] Jan Baumbach and Harald H.H.W. Schmidt. “The End of Medicine as We Know It: Introduction to the New Journal, Systems Medicine”. In: *Systems Medicine* 1.1 (2018), pp. 1–2. DOI: 10.1089/sysm.2017.28999.jba. eprint: https://doi.org/10.1089/sysm.2017.28999.jba. URL: https://doi.org/10.1089/sysm.2017.28999.jba.

[16] Brett K Beaulieu-Jones and Casey S Greene. “Reproducibility of computational workflows is automated using continuous analysis”. In: *Nature biotechnology* 35.4 (2017), pp. 342–346.

[17] Richard Bellman and Robert Kalaba. “Dynamic programming and statistical communication theory”. In: *Proceedings of the National Academy of Sciences* 43.8 (1957), pp. 749–751.

[18] Yoshua Bengio. “Practical recommendations for gradient-based training of deep architectures”. In: *Neural Networks: Tricks of the Trade: Second Edition*. Springer, 2012, pp. 437–478.

[19] Kirandeep Bhullar et al. “Antibiotic resistance is prevalent in an isolated cave microbiome”. In: *PloS one* 7.4 (2012), e34953.

[20] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[21] J-P Boissel et al. “Bridging systems medicine and patient needs”. In: *CPT: pharmacometrics & systems pharmacology* 4.3 (2015), pp. 135–145.

[22] Keith Bonawitz et al. “Towards federated learning at scale: System design”. In: *Proceedings of machine learning and systems* 1 (2019), pp. 374–388.

[23] Axel Börsch-Supan et al. “Data Resource Profile: the Survey of Health, Ageing and Retirement in Europe (SHARE)”. In: *International Journal of Epidemiology* 42.4 (2013), pp. 992–1001. ISSN: 1464-3685, 0300-5771. DOI: 10.1093/ije/dyt088. URL: https://doi.org/10.1093/ije/dyt088.

[24] Jean Bousquet et al. “Systems medicine and integrated care to combat chronic noncommunicable diseases”. In: *Genome medicine* 3 (2011), pp. 1–12.

[25] Leo Breiman. “Bagging predictors”. In: *Machine learning* 24 (1996), pp. 123–140.

[26] Leo Breiman. *Classification and regression trees*. Routledge, 2017.

[27] Leo Breiman. “Random forests”. In: *Machine learning* 45.1 (2001), pp. 5–32.

[28] Sergey Brin and Lawrence Page. “The anatomy of a large-scale hypertextual web search engine”. In: *Computer networks and ISDN systems* 30.1-7 (1998), pp. 107–117.

[29] Han Cao et al. “dsMTL: a computational framework for privacy-preserving, distributed multi-task machine learning”. In: *Bioinformatics* 38.21 (Sept. 2022), pp. 4919–4926. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btac616. eprint: https://academic.oup.com/bioinformatics/article-pdf/38/21/4919/46697912/btac616.pdf. URL: https://doi.org/10.1093/bioinformatics/btac616.

[30] Nicholas Carlini and David Wagner. “Towards evaluating the robustness of neural networks”. In: *2017 ieee symposium on security and privacy (sp)*. Ieee. 2017, pp. 39–57.

[31] Fabrice Carrat and Antoine Flahault. “Influenza vaccine: the challenge of antigenic drift”. In: *Vaccine* 25.39-40 (2007), pp. 6852–6862.

[32] Rich Caruana et al. “Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission”. In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 2015, pp. 1721–1730.

[33] Fred H Cate, Peter Cullen, and Viktor Mayer-Schonberger. “Data protection principles for the 21st century”. In: (2013).

[34] Fred Charatan. “US launches new clinical trials database”. In: *BMJ: British Medical Journal* 320.7236 (2000), p. 668.

[35] Andrew Chatr-Aryamontri et al. “The BioGRID interaction database: 2017 update”. In: *Nucleic acids research* 45.D1 (2017), pp. D369–D379.

[36] Bin Chen et al. “Reversal of cancer gene expression correlates with drug efficacy and reveals therapeutic targets”. In: *Nature communications* 8.1 (2017), p. 16022.

[37] Rui Chen et al. “Personal omics profiling reveals dynamic molecular and medical phenotypes”. In: *Cell* 148.6 (2012), pp. 1293–1307.

[38] Shuliang Chen, Xiao Yu, and Deyin Guo. “CRISPR-Cas targeting of host genes as an antiviral strategy”. In: *Viruses* 10.1 (2018), p. 40.

[39] Feixiong Cheng et al. “A network-based drug repositioning infrastructure for precision cancer medicine through targeting significantly mutated genes in the human cancer genomes”. In: *Journal of the American Medical Informatics Association* 23.4 (2016), pp. 681–691.

[40] Feixiong Cheng et al. “Prediction of drug-target interactions and drug repositioning via network-based inference”. In: *PLoS computational biology* 8.5 (2012), e1002503.

[41] Gilles Clermont et al. “In silico design of clinical trials: a method coming of age”. In: *Critical care medicine* 32.10 (2004), pp. 2061–2070.

[42] Myron S Cohen et al. "The spread, treatment, and prevention of HIV-1: evolution of a global pandemic". In: *The Journal of clinical investigation* 118.4 (2008), pp. 1244–1254.

[43] Francis S Collins and Harold Varmus. "A new initiative on precision medicine". In: *New England journal of medicine* 372.9 (2015), pp. 793–795.

[44] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine learning* 20 (1995), pp. 273–297.

[45] Mark R Denison et al. "Coronaviruses: an RNA proofreading machine regulates replication fidelity and diversity". In: *RNA biology* 8.2 (2011), pp. 270–279.

[46] Vasant Dhar. "Data science and prediction". In: *Communications of the ACM* 56.12 (2013), pp. 64–73.

[47] Thomas G Dietterich. "Ensemble methods in machine learning". In: *International workshop on multiple classifier systems*. Springer. 2000, pp. 1–15.

[48] Joseph A DiMasi, Henry G Grabowski, and Ronald W Hansen. "Innovation in the pharmaceutical industry: new estimates of R&D costs". In: *Journal of health economics* 47 (2016), pp. 20–33.

[49] Stuart E Dreyfus and Robert A Wagner. "The Steiner problem in graphs". In: *Networks* 1.3 (1971), pp. 195–207.

[50] Joel T Dudley, Tarangini Deshpande, and Atul J Butte. "Exploiting drug–disease relationships for computational drug repositioning". In: *Briefings in bioinformatics* 12.4 (2011), pp. 303–311.

[51] Cynthia Dwork. "The promise of differential privacy: a tutorial on algorithmic techniques". In: *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, D (Oct. 2011)*. Citeseer. 2011, pp. 1–2.

[52] Cynthia Dwork, Aaron Roth, et al. "The algorithmic foundations of differential privacy". In: *Foundations and Trends® in Theoretical Computer Science* 9.3–4 (2014), pp. 211–407.

[53] Cynthia Dwork et al. "Calibrating noise to sensitivity in private data analysis". In: *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*. Springer. 2006, pp. 265–284.

[54] Cynthia Dwork et al. "The reusable holdout: Preserving validity in adaptive data analysis". In: *Science* 349.6248 (2015), pp. 636–638.

[55] Bradley Efron. "Prediction, Estimation, and Attribution". In: *Journal of the American Statistical Association* 115.530 (2020), pp. 636–655. DOI: 10.1080/01621459.2020.1762613. eprint: https://doi.org/10.1080/01621459.2020.1762613. URL: https://doi.org/10.1080/01621459.2020.1762613.

[56] Bradley Efron et al. "Least angle regression". In: *The Annals of Statistics* 32.2 (2004), pp. 407–499. DOI: 10.1214/009053604000000067. URL: https://doi.org/10.1214/009053604000000067.

[57] Matthias Feurer et al. "Efficient and robust automated machine learning". In: *Advances in neural information processing systems* 28 (2015).

[58] Linton C Freeman. "A set of measures of centrality based on betweenness". In: *Sociometry* (1977), pp. 35–41.

[59] Yoav Freund and Robert E Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting". In: *Journal of computer and system sciences* 55.1 (1997), pp. 119–139.

[60] Gihanna Galindez et al. "Lessons from the COVID-19 pandemic for advancing computational drug repurposing strategies". In: *Nature Computational Science* 1.1 (2021), pp. 33–41. ISSN: 2662-8457. DOI: 10.1038/s43588-020-00007-6. URL: https://doi.org/10.1038/s43588-020-00007-6.

[61] Yiyue Ge et al. "A data-driven drug repositioning framework discovered a potential therapeutic agent targeting COVID-19". In: *BioRxiv* (2020), pp. 2020–03.

[62] C Gentry. "A fully homomorphic encryption scheme-Stanford University, Ph. D. thesis, 2009". In: (2009).

[63] Hossein A Ghofrani, Ian H Osterloh, and Friedrich Grimminger. "Sildenafil: from angina to erectile dysfunction to pulmonary hypertension and beyond". In: *Nature reviews Drug discovery* 5.8 (2006), pp. 689–702.

[64] Michael K Gilson et al. “BindingDB in 2015: a public database for medicinal chemistry, computational chemistry and systems pharmacology”. In: *Nucleic acids research* 44.D1 (2016), pp. D1045–D1053.

[65] Jeremy Goecks et al. “How machine learning will transform biomedicine”. In: *Cell* 181.1 (2020), pp. 92–101.

[66] Oded Goldreich. “Secure multi-party computation”. In: *Manuscript. Preliminary version* 78.110 (1998).

[67] Irwin Goldstein et al. “Oral sildenafil in the treatment of erectile dysfunction”. In: *The Journal of Urology* 167.2 (2002), pp. 1197–1203.

[68] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[69] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. “Explaining and harnessing adversarial examples”. In: *arXiv preprint arXiv:1412.6572* (2014).

[70] David E Gordon et al. “A SARS-CoV-2 protein interaction map reveals targets for drug repurposing”. In: *Nature* 583.7816 (2020), pp. 459–468.

[71] Assaf Gottlieb et al. “INDI: a computational framework for inferring drug interactions and their associated recommendations”. In: *Molecular systems biology* 8.1 (2012), p. 592.

[72] Clemens Gröpl et al. “Approximation Algorithms for the Steiner Tree Problem in Graphs”. In: *Steiner Trees in Industry*. Ed. by Xiu Zhen Cheng and Ding-Zhu Du. Boston, MA: Springer US, 2001, pp. 235–279. ISBN: 978-1-4613-0255-1. DOI: 10.1007/978-1-4613-0255-1_7. URL: https://doi.org/10.1007/978-1-4613-0255-1_7.

[73] *Guideline on the scientific application and the practical arrangements necessary to implement Commission Regulation (EC) No 507/2006 on the conditional marketing authorisation for medicinal products for human use falling within the scope of Regulation (EC) No 726/2004*. Tech. rep. Amsterdam, The Netherlands: European Medicines Agency, 2016. URL: https://www.ema.europa.eu/en/documents/scientific-guideline/guideline-scientific-application-practical-arrangements-necessary-implement-commission-regulation-ec/2006-conditional-marketing-authorisation-medicinal-products-human-use-falling_en.pdf.

[74] Thibaut Guirimand, Stéphane Delmotte, and Vincent Navratil. “VirHostNet 2.0: surfing on the web of virus/host molecular interactions data”. In: *Nucleic acids research* 43.D1 (2015), pp. D583–D587.

[75] Emre Guney et al. “Network-based in silico drug efficacy screening”. In: *Nature communications* 7.1 (2016), p. 10331.

[76] Gamze Gürsoy. “Genome Privacy and Trust”. In: *Annual Review of Biomedical Data Science* 5.1 (2022). PMID: 35508070, pp. 163–181. DOI: 10.1146/annurev-biodatasci-122120-021311. eprint: https://doi.org/10.1146/annurev-biodatasci-122120-021311. URL: https://doi.org/10.1146/annurev-biodatasci-122120-021311.

[77] Sigrun M Gustafsdottir et al. “Multiplex cytological profiling assay to measure diverse cellular states”. In: *PloS one* 8.12 (2013), e80999.

[78] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. “Combating web spam with trustrank”. In: *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. 2004, pp. 576–587.

[79] David Harrison and Daniel L Rubinfeld. “Hedonic housing prices and the demand for clean air”. In: *Journal of Environmental Economics and Management* 5.1 (1978), pp. 81–102. ISSN: 0095-0696. DOI: https://doi.org/10.1016/0095-0696(78)90006-2. URL: https://www.sciencedirect.com/science/article/pii/0095069678900062.

[80] Michael Hartung et al. “A plug-and-play solution to bring network exploration and drug repurposing to biomedical web platforms”. In: (2022).

[81] Michael Hartung et al. “Cancer driver drug interaction explorer”. In: *Nucleic Acids Research* 50.W1 (May 2022), W138–W144. ISSN: 0305-1048. DOI: 10.1093/nar/gkac384. eprint: https://

academic.oup.com/nar/article-pdf/50/W1/W138/44379561/gkac384.pdf. URL: https://doi.org/10.1093/nar/gkac384.

[82] Trevor Hastie et al. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer, 2009.

[83] Anne-Christin Hauschild et al. "Federated Random Forests can improve local performance of predictive models for various healthcare applications". In: *Bioinformatics* 38.8 (Feb. 2022), pp. 2278–2286. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btac065. eprint: https://academic.oup.com/bioinformatics/article-pdf/38/8/2278/49009424/btac065.pdf. URL: https://doi.org/10.1093/bioinformatics/btac065.

[84] Xin He, Kaiyong Zhao, and Xiaowen Chu. "AutoML: A survey of the state-of-the-art". In: *Knowledge-Based Systems* 212 (2021), p. 106622.

[85] J Matthew Helm et al. "Machine Learning and Artificial Intelligence: Definitions, Applications, and Future Directions". en. In: *Curr Rev Musculoskelet Med* 13.1 (Feb. 2020), pp. 69–76.

[86] Leroy Hood, Rudi Balling, and Charles Auffray. "Revolutionizing medicine in the 21st century through systems approaches". In: *Biotechnology journal* 7.8 (2012), pp. 992–1001.

[87] Leroy Hood and Mauricio Flores. "A personal view on systems medicine and the emergence of proactive P4 medicine: predictive, preventive, personalized and participatory". In: *New biotechnology* 29.6 (2012), pp. 613–624.

[88] Leroy Hood and Stephen H Friend. "Predictive, personalized, preventive, participatory (P4) cancer medicine". In: *Nature reviews Clinical oncology* 8.3 (2011), pp. 184–187.

[89] Andrew L Hopkins. "Network pharmacology: the next paradigm in drug discovery". In: *Nature chemical biology* 4.11 (2008), pp. 682–690.

[90] Chaolin Huang et al. "Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China". In: *The lancet* 395.10223 (2020), pp. 497–506.

[91] Franziska Hufsky et al. "Computational strategies to combat COVID-19: useful tools to accelerate SARS-CoV-2 and coronavirus research". In: *Briefings in Bioinformatics* 22.2 (Nov. 2020), pp. 642–663. ISSN: 1477-4054. DOI: 10.1093/bib/bbaa232. eprint: https://academic.oup.com/bib/article-pdf/22/2/642/36654876/bbaa232.pdf. URL: https://doi.org/10.1093/bib/bbaa232.

[92] Tim Hulsen et al. "From big data to precision medicine". In: *Frontiers in medicine* 6 (2019), p. 34.

[93] Richard D Hurt et al. "A comparison of sustained-release bupropion and placebo for smoking cessation". In: *New England Journal of Medicine* 337.17 (1997), pp. 1195–1202.

[94] Frank K Hwang and Dana S Richards. "Steiner tree problems". In: *Networks* 22.1 (1992), pp. 55–89.

[95] Clifford R Jack et al. "Hypothetical model of dynamic biomarkers of the Alzheimer's pathological cascade". In: *The Lancet Neurology* 9.1 (2010), pp. 119–128.

[96] Bart Jacobs and Jean Popma. "Medical research, big data and the need for privacy by design". In: *Big Data & Society* 6.1 (2019), p. 2053951718824352.

[97] Michael I Jordan and Tom M Mitchell. "Machine learning: Trends, perspectives, and prospects". In: *Science* 349.6245 (2015), pp. 255–260.

[98] Douglas E Jorenby et al. "A controlled trial of sustained-release bupropion, a nicotine patch, or both for smoking cessation". In: *New England Journal of Medicine* 340.9 (1999), pp. 685–691.

[99] Maliackal Poulo Joy et al. "High-betweenness proteins in the yeast protein interaction network". In: *Journal of Biomedicine and Biotechnology* 2005.2 (2005), p. 96.

[100] Michael J Joyner and Bente K Pedersen. "Ten questions about systems biology". In: *The Journal of Physiology* 589.5 (2011), pp. 1017–1030.

[101] Peter Kairouz et al. "Advances and open problems in federated learning". In: *Foundations and Trends® in Machine Learning* 14.1–2 (2021), pp. 1–210.

[102] Kenneth I Kaitin and Joseph A DiMasi. “Pharmaceutical innovation in the 21st century: new drug approvals in the first decade, 2000–2009”. In: *Clinical pharmacology & therapeutics* 89.2 (2011), pp. 183–188.

[103] Minoru Kanehisa et al. “KEGG: new perspectives on genomes, pathways, diseases and drugs”. In: *Nucleic acids research* 45.D1 (2017), pp. D353–D361.

[104] Richard M Karp. *Reducibility among combinatorial problems*. Springer, 2010.

[105] Marc W Kirschner. “The meaning of systems biology”. In: *Cell* 121.4 (2005), pp. 503–504.

[106] D Knipe et al. *Fields Virology, Volumes 1 and 2*. Lippincott Williams & Wilkins: Philadelphia, PA, USA, 2013.

[107] Anton Kocheturov, Panos M Pardalos, and Athanasia Karakitsiou. “Massive datasets and machine learning for computational biomedicine: trends and challenges”. In: *Annals of Operations Research* 276.1 (2019), pp. 5–34.

[108] Jakub Konečný et al. “Federated Learning: Strategies for Improving Communication Efficiency”. In: *NIPS Workshop on Private Multi-Party Machine Learning*. 2016. URL: https://arxiv.org/abs/1610.05492.

[109] Max Kotlyar et al. “IID 2018 update: context-specific physical protein–protein interactions in human, model organisms and domesticated species”. In: *Nucleic acids research* 47.D1 (2019), pp. D581–D589.

[110] Lawrence Kou, George Markowsky, and Leonard Berman. “A fast algorithm for Steiner trees”. In: *Acta informatica* 15 (1981), pp. 141–145.

[111] Max Kuhn, Kjell Johnson, et al. *Applied predictive modeling*. Vol. 26. Springer, 2013.

[112] Pedro Larranaga et al. “Machine learning in bioinformatics”. In: *Briefings in bioinformatics* 7.1 (2006), pp. 86–112.

[113] Hyafil Laurent and Ronald L Rivest. “Constructing optimal binary decision trees is NP-complete”. In: *Information processing letters* 5.1 (1976), pp. 15–17.

[114] Duc-Hau Le. “Network-based ranking methods for prediction of novel disease associated microRNAs”. In: *Computational Biology and Chemistry* 58 (2015), pp. 139–148.

[115] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), pp. 436–444.

[116] Qinbin Li, Zeyi Wen, and Bingsheng He. “Practical federated gradient boosting decision trees”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 04. 2020, pp. 4642–4649.

[117] Tian Li et al. “Federated optimization in heterogeneous networks”. In: *Proceedings of Machine learning and systems* 2 (2020), pp. 429–450.

[118] John Lonsdale et al. “The genotype-tissue expression (GTEx) project”. In: *Nature genetics* 45.6 (2013), pp. 580–585.

[119] Joseph Loscalzo and Albert-Laszlo Barabasi. “Systems biology and the future of medicine”. In: *Wiley Interdisciplinary Reviews: Systems Biology and Medicine* 3.6 (2011), pp. 619–627.

[120] Joseph Loscalzo, Isaac Kohane, and Albert-Laszlo Barabasi. “Human disease classification in the postgenomic era: a complex systems approach to human pathobiology”. In: *Molecular systems biology* 3.1 (2007), p. 124.

[121] Wei Luo et al. “Guidelines for developing and reporting machine learning predictive models in biomedical research: a multidisciplinary view”. In: *Journal of medical Internet research* 18.12 (2016), e323.

[122] Priyanka Mary Mammen. “Federated learning: opportunities and challenges”. In: *arXiv preprint arXiv:2101.05428* (2021).

[123] Vivien Marx. “The big challenges of big data”. In: *Nature* 498.7453 (2013), pp. 255–260.

[124] Julian Matschinske et al. “Individuating Possibly Repurposable Drugs and Drug Targets for COVID-19 Treatment Through Hypothesis-Driven Systems Medicine Using CoVex”. In: *Assay and Drug Development Technologies* 18.8 (2020), pp. 348–355. ISSN: 1540-658X. DOI: 10.1089/adt.2020.1010. URL: https://doi.org/10.1089/adt.2020.1010.

[125] Julian Matschinske et al. “The AIMe registry for artificial intelligence in biomedical research”. In: *Nature Methods* 18.10 (2021), pp. 1128–1131. ISSN: 1548-7105. DOI: 10.1038/s41592-021-01241-0. URL: https://doi.org/10.1038/s41592-021-01241-0.

[126] Julian Matschinske et al. “The FeatureCloud Platform for Federated Learning in Biomedicine: Unified Approach”. English. In: *Journal of Medical Internet Research* 25 (July 2023). ISSN: 1438-8871. DOI: 10.2196/42621.

[127] Brendan McMahan et al. “Communication-efficient learning of deep networks from decentralized data”. In: *Artificial intelligence and statistics*. PMLR. 2017, pp. 1273–1282.

[128] H Brendan McMahan et al. “Federated learning of deep networks using model averaging”. In: *arXiv preprint arXiv:1602.05629* 2 (2016), p. 2.

[129] Jörg Menche et al. “Uncovering disease-disease relationships through the incomplete interactome”. In: *Science* 347.6224 (2015), p. 1257601.

[130] David Mendez et al. “ChEMBL: towards direct deposition of bioassay data”. In: *Nucleic acids research* 47.D1 (2019), pp. D930–D940.

[131] Dirk Merkel. “Docker: lightweight linux containers for consistent development and deployment”. In: *Linux journal* 2014.239 (2014), p. 2.

[132] Andrea Miró-Canturri, Rafael Ayerbe-Algaba, and Younes Smani. “Drug repurposing for the treatment of bacterial and fungal infections”. In: *Frontiers in microbiology* 10 (2019), p. 41.

[133] Tom M Mitchell. *Machine learning*. 1997.

[134] Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. *Introduction to linear regression analysis*. John Wiley & Sons, 2021.

[135] David M Morens, Gregory K Folkers, and Anthony S Fauci. “The challenge of emerging and reemerging infectious diseases”. In: *Nature* 430.6996 (2004), pp. 242–249.

[136] Viraaji Mothukuri et al. “A survey on security and privacy of federated learning”. In: *Future Generation Computer Systems* 115 (2021), pp. 619–640.

[137] Travis B Murdoch and Allan S Detsky. “The inevitable application of big data to health care”. In: *Jama* 309.13 (2013), pp. 1351–1352.

[138] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[139] Bindu Nanduri, Cathy R Gresham, and Fiona M McCarthy. “HPIDB: A Database for Host-Pathogen Interactions”. In: *Plant and Animal Genome XXVII Conference (January 12-16, 2019)*. PAG. 2019.

[140] Reza Nasirigerdeh et al. *HyFed: A Hybrid Federated Framework for Privacy-preserving Machine Learning*. 2021. DOI: 10.48550/ARXIV.2105.10545. URL: https://arxiv.org/abs/2105.10545.

[141] Reza Nasirigerdeh et al. “sPLINK: a hybrid federated tool as a robust alternative to meta-analysis in genome-wide association studies”. In: *Genome Biology* 23.1 (2022), p. 32. ISSN: 1474-760X. DOI: 10.1186/s13059-021-02562-1. URL: https://doi.org/10.1186/s13059-021-02562-1.

[142] Mark Newman. *Networks*. Oxford university press, 2018.

[143] Dinh C Nguyen et al. “Federated learning for smart healthcare: A survey”. In: *ACM Computing Surveys (CSUR)* 55.3 (2022), pp. 1–37.

[144] Ziad Obermeyer et al. “Dissecting racial bias in an algorithm used to manage the health of populations”. In: *Science* 366.6464 (2019), pp. 447–453.

[145] Xiaoqin Pan et al. “Deep learning for drug repurposing: Methods, databases, and applications”. In: *Wiley interdisciplinary reviews: Computational molecular science* 12.4 (2022), e1597.

[146] “Pathway and network analysis of cancer genomes”. In: *Nature methods* 12.7 (2015), pp. 615–621.

[147] Fabian Pedregosa et al. “Scikit-learn: Machine learning in Python”. In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

[148] Noah C Peeri et al. “The SARS, MERS and novel coronavirus (COVID-19) epidemics, the newest and biggest global health threats: what lessons have we learned?” In: *International journal of epidemiology* 49.3 (2020), pp. 717–726.

[149] Roger D Peng. “Reproducible research in computational science”. In: *Science* 334.6060 (2011), pp. 1226–1227.

[150] Yasset Perez-Riverol et al. “Quantifying the impact of public omics data”. In: *Nature Communications* 10.1 (Aug. 2019), p. 3512. ISSN: 2041-1723. DOI: 10.1038/s41467-019-11461-w. URL: https://doi.org/10.1038/s41467-019-11461-w.

[151] Azam Peyvandipour et al. “A novel computational approach for drug repurposing using systems biology”. In: *Bioinformatics* 34.16 (Mar. 2018), pp. 2817–2825. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bty133. eprint: https://academic.oup.com/bioinformatics/article-pdf/34/16/2817/25441943/bty133.pdf. URL: https://doi.org/10.1093/bioinformatics/bty133.

[152] Susanne Pfefferle et al. “The SARS-coronavirus-host interactome: identification of cyclophilins as target for pan-coronavirus inhibitors”. In: *PLoS pathogens* 7.10 (2011), e1002331.

[153] Sergey M Plis et al. “COINSTAC: a privacy enabled model and prototype for leveraging and processing decentralized brain imaging data”. In: *Frontiers in neuroscience* 10 (2016), p. 365.

[154] Stanley Plotkin. “History of vaccination”. In: *Proceedings of the National Academy of Sciences* 111.34 (2014), pp. 12283–12287.

[155] Jason Priem and Kaitlin Light Costello. “How and why scholars cite on Twitter”. In: *Proceedings of the American Society for Information Science and Technology* 47.1 (2010), pp. 1–4.

[156] Foster Provost and Tom Fawcett. *Data Science for Business: What you need to know about data mining and data-analytic thinking.* ” O’Reilly Media, Inc.”, 2013.

[157] Sudeep Pushpakom et al. “Drug repurposing: progress, challenges and recommendations”. In: *Nature reviews Drug discovery* 18.1 (2019), pp. 41–58.

[158] J. Ross Quinlan. “Induction of decision trees”. In: *Machine learning* 1 (1986), pp. 81–106.

[159] Anichur Rahman et al. “Federated learning-based AI approaches in smart healthcare: concepts, taxonomies, challenges and open issues”. In: *Cluster computing* 26.4 (2023), pp. 2271–2311.

[160] Alvin Rajkomar, Jeffrey Dean, and Isaac Kohane. “Machine learning in medicine”. In: *New England Journal of Medicine* 380.14 (2019), pp. 1347–1358.

[161] Bendi Ramana and N. Venkateswarlu. *ILPD (Indian Liver Patient Dataset).* UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5D02C. 2012.

[162] Mathias Rask-Andersen et al. “Advances in kinase targeting: current clinical use and clinical trials”. In: *Trends in pharmacological sciences* 35.11 (2014), pp. 604–620.

[163] A Srinivas Reddy and Shuxing Zhang. “Polypharmacology: drug discovery for the future”. In: *Expert review of clinical pharmacology* 6.1 (2013), pp. 41–47.

[164] Nicola Rieke et al. “The future of digital health with federated learning”. In: *NPJ digital medicine* 3.1 (2020), p. 119.

[165] Dan Robinson et al. “Integrative clinical genomics of advanced prostate cancer”. In: *Cell* 161.5 (2015), pp. 1215–1228.

[166] Lior Rokach. “Ensemble-based classifiers”. In: *Artificial intelligence review* 33 (2010), pp. 1–39.

[167] Sebastian Ruder. “An overview of gradient descent optimization algorithms”. In: *arXiv preprint arXiv:1609.04747* (2016).

[168] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), pp. 533–536.

[169] Stuart Russell. *Artificial Intelligence: A Modern Approach, eBook, Global Edition*. Pearson Education, Limited, 2016.

[170] Theo Ryffel et al. “A generic framework for privacy preserving deep learning”. In: *arXiv preprint arXiv:1811.04017* (2018).

[171] Sepideh Sadegh et al. “Exploring the SARS-CoV-2 virus-host-drug interactome for drug repurposing”. In: *Nature Communications* 11.1 (July 2020), p. 3518. ISSN: 2041-1723. DOI: 10.1038/s41467-020-17189-2. URL: https://doi.org/10.1038/s41467-020-17189-2.

[172] Geir Kjetil Sandve et al. “Ten simple rules for reproducible computational research”. In: *PLoS computational biology* 9.10 (2013), e1003285.

[173] Rafael Sanjuán et al. “Viral mutation rates”. In: *Journal of virology* 84.19 (2010), pp. 9733–9748.

[174] Lucía Prieto Santamaría et al. "Integrating heterogeneous data to facilitate COVID-19 drug repurposing". In: *Drug Discovery Today* 27.2 (2022), pp. 558–566.

[175] Robert E Schapire. "The strength of weak learnability". In: *Machine learning* 5 (1990), pp. 197–227.

[176] Jürgen Schmidhuber. "Deep learning in neural networks: An overview". In: *Neural networks* 61 (2015), pp. 85–117.

[177] Adi Shamir. "How to share a secret". In: *Communications of the ACM* 22.11 (1979), pp. 612–613.

[178] Micah J Sheller et al. "Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data". In: *Scientific reports* 10.1 (2020), pp. 1–12.

[179] Reza Shokri and Vitaly Shmatikov. "Privacy-preserving deep learning". In: *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. 2015, pp. 1310–1321.

[180] Suyash S Shringarpure and Carlos D Bustamante. "Privacy risks from genomic data-sharing beacons". In: *The American Journal of Human Genetics* 97.5 (2015), pp. 631–646.

[181] Santiago Silva et al. "Fed-biomed: A general open-source frontend framework for federated learning in healthcare". In: *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*. Springer, 2020, pp. 201–210.

[182] Shweta Singh et al. "Impact of COVID-19 and lockdown on mental health of children and adolescents: A narrative review with recommendations". In: *Psychiatry research* 293 (2020), p. 113429.

[183] Julian Späth et al. "Privacy-aware multi-institutional time-to-event studies". In: *PLOS Digital Health* 1.9 (Sept. 2022), pp. 1–16. DOI: 10.1371/journal.pdig.0000101. URL: https://doi.org/10.1371/journal.pdig.0000101.

[184] W. Nick Street, W. H. Wolberg, and O. L. Mangasarian. "Nuclear feature extraction for breast tumor diagnosis". In: *Biomedical Image Processing and Biomedical Visualization*. Ed. by Raj S. Acharya and Dmitry B. Goldgof. Vol. 1905. International Society for Optics and Photonics. SPIE, 1993, pp. 861–870. DOI: 10.1117/12.148698. URL: https://doi.org/10.1117/12.148698.

[185] Aaron Stupple, David Singerman, and Leo Anthony Celi. "The reproducibility crisis in the age of digital medicine". In: *NPJ digital medicine* 2.1 (2019), p. 2.

[186] Ilya Sutskever et al. "On the importance of initialization and momentum in deep learning". In: *International conference on machine learning*. PMLR. 2013, pp. 1139–1147.

[187] Damian Szklarczyk et al. "STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets". In: *Nucleic acids research* 47.D1 (2019), pp. D607–D613.

[188] Alan Talevi. "Multi-target pharmacology: possibilities and limitations of the "skeleton key approach" from a medicinal chemist perspective". In: *Frontiers in pharmacology* 6 (2015), p. 205.

[189] Eric J Topol. "High-performance medicine: the convergence of human and artificial intelligence". In: *Nature medicine* 25.1 (2019), pp. 44–56.

[190] Reihaneh Torkzadehmahani et al. "Privacy-Preserving Artificial Intelligence Techniques in Biomedicine". EN. In: *Methods of Information in Medicine* 61.S 01 (2022), e12–e27. ISSN: 0026-1270. DOI: 10.1055/s-0041-1740630. URL: http://www.thieme-connect.com/products/ejournals/abstract/10.1055/s-0041-1740630.

[191] U.S. Food and Drug Administration. *The Drug Development Process*. Online. 2018. URL: https://www.fda.gov/patients/learn-about-drug-and-device-approvals/drug-development-process.

[192] Fabio Urbina, Ana C Puhl, and Sean Ekins. "Recent advances in drug repurposing using machine learning". In: *Current Opinion in Chemical Biology* 65 (2021), pp. 74–84.

[193] Oleg Ursu et al. "DrugCentral 2018: an update". In: *Nucleic acids research* 47.D1 (2019), pp. D963–D970.

[194] Marten Van Dijk et al. "Fully homomorphic encryption over the integers". In: *Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryp-*

tographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings 29. Springer. 2010, pp. 24–43.

[195] Oron Vanunu et al. "Associating genes and protein complexes with disease via network propagation". In: *PLoS computational biology* 6.1 (2010), e1000641.

[196] Vladimir N Vapnik. "An overview of statistical learning theory". In: *IEEE transactions on neural networks* 10.5 (1999), pp. 988–999.

[197] Effy Vayena, Alessandro Blasimme, and I Glenn Cohen. "Machine learning in medicine: addressing ethical challenges". In: *PLoS medicine* 15.11 (2018), e1002689.

[198] Yunxia Wang et al. "Therapeutic target database 2020: enriched resource for facilitating research and early development of targeted therapeutics". In: *Nucleic acids research* 48.D1 (2020), pp. D1031–D1041.

[199] Jess Whittlestone et al. "Ethical and societal implications of algorithms, data, and artificial intelligence: a roadmap for research". In: *London: Nuffield Foundation* (2019).

[200] Mark D Wilkinson et al. "The FAIR Guiding Principles for scientific data management and stewardship". In: *Scientific data* 3.1 (2016), pp. 1–9.

[201] Greg Wilson et al. "Best practices for scientific computing". In: *PLoS biology* 12.1 (2014), e1001745.

[202] David S Wishart et al. "DrugBank 5.0: a major update to the DrugBank database for 2018". In: *Nucleic acids research* 46.D1 (2018), pp. D1074–D1082.

[203] Olaf Wolkenhauer et al. "The road from systems biology to systems medicine". In: *Pediatric research* 73.2 (2013), pp. 502–507.

[204] David H Wolpert. "Stacked generalization". In: *Neural networks* 5.2 (1992), pp. 241–259.

[205] Cong Xie, Sanmi Koyejo, and Indranil Gupta. "Asynchronous federated optimization". In: *arXiv preprint arXiv:1903.03934* (2019).

[206] Xuan Xu et al. "Facilitating antiviral drug discovery using genetic and evolutionary knowledge". In: *Viruses* 13.11 (2021), p. 2117.

[207] Ryo Yamada et al. "Interpretation of omics data analyses". In: *Journal of Human Genetics* 66.1 (Jan. 2021), pp. 93–102. ISSN: 1435-232X. DOI: 10.1038/s10038-020-0763-5. URL: https://doi.org/10.1038/s10038-020-0763-5.

[208] Qiang Yang et al. "Federated machine learning: Concept and applications". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 10.2 (2019), pp. 1–19.

[209] Muhammed A Yıldırım et al. "Drug—target network". In: *Nature biotechnology* 25.10 (2007), pp. 1119–1126.

[210] Xiangxiang Zeng et al. "deepDR: a network-based deep learning approach to in silico drug repositioning". In: *Bioinformatics* 35.24 (2019), pp. 5191–5198.

[211] S Zhang, H Zhao, and R John. "Development of a quantitative relationship between inhibition percentage and both incubation time and inhibitor concentration for inhibition biosensors—theoretical and practical considerations". In: *Biosensors and Bioelectronics* 16.9-12 (2001), pp. 1119–1126.

[212] Yue Zhao et al. "Federated learning with non-iid data". In: *arXiv preprint arXiv:1806.00582* (2018).

[213] Yadi Zhou et al. "Network-based drug repurposing for novel coronavirus 2019-nCoV/SARS-CoV-2". In: *Cell discovery* 6.1 (2020), p. 14.

[214] Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. CRC press, 2012.

[215] Olga Zolotareva et al. "Flimma: a federated and privacy-aware tool for differential gene expression analysis". In: *Genome Biology* 22.1 (2021), p. 338. ISSN: 1474-760X. DOI: 10.1186/s13059-021-02553-2. URL: https://doi.org/10.1186/s13059-021-02553-2.

[216] Nansu Zong et al. "Deep mining heterogeneous networks of biomedical linked data to predict novel drug–target associations". In: *Bioinformatics* 33.15 (2017), pp. 2337–2344.

Publications

Included in this thesis

The 4 publications listed here are part of this thesis and can be found in the same order following this part of the appendix.

A CoVex

The paper entitled *Exploring the SARS-CoV-2 virus-host-drug interactome for drug repurposing* and published in *Nature Communications* in 2020 contains the details of CoVex, presented in Chapter 2, Section 2.5.

Contribution My responsibility for this work was managing the development process of the web tool, designing both the software and system architecture and setting up the deployment process using continuous integration and continuous delivery (CICD). I also contributed significantly to both the backend and frontend code.

Reference Sepideh Sadegh et al. "Exploring the SARS-CoV-2 virus-host-drug interactome for drug repurposing". In: *Nature Communications* 11.1 (July 2020), p. 3518. ISSN: 2041-1723. DOI: 10.1038/s41467-020-17189-2. URL: https://doi.org/10.1038/s41467-020-17189-2

Full list of authors Sepideh Sadegh*, Julian Matschinske*, David B Blumenthal, Gihanna Galindez, Tim Kacprowski, Markus List, Reza Nasirigerdeh, Mhaned Oubounyt, Andreas Pichlmair, Tim Daniel Rose, Marisol Salgado-Albarrán, Julian Späth, Alexey Stukalov, Nina K Wenke, Kevin Yuan, Josch K Pauling, Jan Baumbach

*contributed equally

B sPLINK

The paper entitled *sPLINK: A hybrid federated tool as a robust alternative to meta-analysis in genome-wide association studies* and published in *Genome Biology* in 2022 contains the details of sPLINK, presented in Chapter 3, Section 3.7.1.

Contribution My contribution for this work was designing the base architecture and contribute significantly to both backend and frontend development, as well as evaluating and testing the tool. I also contributed to writing the paper.

Reference Reza Nasirigerdeh et al. "sPLINK: a hybrid federated tool as a robust alternative to meta-analysis in genome-wide association studies". In: *Genome Biology* 23.1 (2022), p. 32. ISSN: 1474-760X. DOI: 10.1186/s13059-021-02562-1. URL: https://doi.org/10.1186/s13059-021-02562-1

Full list of authors Reza Nasirigerdeh, Reihaneh Torkzadehmahani, Julian Matschinske, Tobias Frisch, Markus List, Julian Späth, Stefan Weiss, Uwe Völker, Esa Pitkänen, Dominik Heider, Nina Kerstin Wenke, Georgios Kaissis, Daniel Rueckert, Tim Kacprowski, Jan Baumbach

C FeatureCloud

The paper entitled *The FeatureCloud platform for federated learning in biomedicine: Unified approach* and published in the *Journal of Medical Internet Research* in 2023 contains the details of FeatureCloud, presented in Chapter 3, Section 3.7.2.

Contribution My role in this work during the development process was leading the development team, choosing the appropriate technologies, and significantly contribute to the code. I also assumed the role of the *product owner*, in agile development terms, talking to the involved stakeholders, managing the tickets for the developers and moderating the developer meetings. I also wrote the paper, together with Julian Späth, the other first author, and evaluated the platform regarding runtime and network traffic.

Reference Julian Matschinske et al. "The FeatureCloud Platform for Federated Learning in Biomedicine: Unified Approach". English. In: *Journal of Medical Internet Research* 25 (July 2023). issn: 1438-8871. doi: 10.2196/42621

Full list of authors Julian Matschinske*, Julian Späth*, Mohammad Bakhtiari, Niklas Probul, Mohammad Mahdi Kazemi Majdabadi, Reza Nasirigerdeh, Reihaneh Torkzadehmahani, Hanne Hartebrodt, Balazs-Attila Orban, Sándor-József Fejér, Olga Zolotareva, Supratim Das, Linda Baumbach, Josch K Pauling, Olivera Tomašević, Béla Bihari, Marcus Bloice, Nina C Donner, Walid Fdhila, Tobias Frisch, Anne-Christin Hausschild, Dominik Heider, Andreas Holzinger, Walter Hötzendorfer, Jan Hospes, Tim Kacprowski, Markus Kastelitz, Markus List, Rudolf Mayer, Mónika Moga, Heimo Müller, Anastasia Pustozerova, Richard Röttger, Christina A Saak, Anna Saranti, Harald HHW Schmidt, Christof Tschohl, Nina K Wenke, Jan Baumbach

*contributed equally

D AIMe

The paper entitled *The AIMe registry for artificial intelligence in biomedical research* and published in *Nature Methods* in 2021 contains the details of AIMe, presented in Chapter 3, Section 3.7.3.

Contribution My responsibility in this work was developing the whole system, consisting of a frontend webapp and a backend, testing it, and deploying it. I also contributed to writing the paper.

Reference Julian Matschinske et al. "The AIMe registry for artificial intelligence in biomedical research". In: *Nature Methods* 18.10 (2021), pp. 1128–1131. issn: 1548-7105. doi: 10.1038/s41592-021-01241-0. url: https://doi.org/10.1038/s41592-021-01241-0

Full list of authors Julian Matschinske, Nicolas Alcaraz, Arriel Benis, Martin Golebiewski, Dominik G Grimm, Lukas Heumos, Tim Kacprowski, Olga Lazareva, Markus List, Zakaria Louadi, Josch K Pauling, Nico Pfeifer, Richard Röttger, Veit Schwämmle, Gregor Sturm, Alberto Traverso, Kristel Van Steen, Martiela Vaz de Freitas, Gerda Cristal Villalba Silva, Leonard Wee, Nina K Wenke, Massimiliano Zanin, Olga Zolotareva, Jan Baumbach, David B Blumenthal

Additional publications

Systems medicine

Reference Gihanna Galindez et al. “Lessons from the COVID-19 pandemic for advancing computational drug repurposing strategies”. In: *Nature Computational Science* 1.1 (2021), pp. 33–41. issn: 2662-8457. doi: 10.1038/s43588-020-00007-6. url: https://doi.org/10.1038/s43588-020-00007-6

Reference Franziska Hufsky et al. “Computational strategies to combat COVID-19: useful tools to accelerate SARS-CoV-2 and coronavirus research”. In: *Briefings in Bioinformatics* 22.2 (Nov. 2020), pp. 642–663. issn: 1477-4054. doi: 10.1093/bib/bbaa232. eprint: https://academic.oup.com/bib/article-pdf/22/2/642/36654876/bbaa232.pdf. url: https://doi.org/10.1093/bib/bbaa232

Reference Julian Matschinske et al. “Individuating Possibly Repurposable Drugs and Drug Targets for COVID-19 Treatment Through Hypothesis-Driven Systems Medicine Using CoVex”. In: *Assay and Drug Development Technologies* 18.8 (2020), pp. 348–355. issn: 1540-658X. doi: 10.1089/adt.2020.1010. url: https://doi.org/10.1089/adt.2020.1010

Privacy-aware AI

Reference Anne-Christin Hauschild et al. “Federated Random Forests can improve local performance of predictive models for various healthcare applications”. In: *Bioinformatics* 38.8 (Feb. 2022), pp. 2278–2286. issn: 1367-4803. doi: 10.1093/bioinformatics/btac065. eprint: https://academic.oup.com/bioinformatics/article-pdf/38/8/2278/49009424/btac065.pdf. url: https://doi.org/10.1093/bioinformatics/btac065

Reference Olga Zolotareva et al. “Flimma: a federated and privacy-aware tool for differential gene expression analysis”. In: *Genome Biology* 22.1 (2021), p. 338. issn: 1474-760X. doi: 10.1186/s13059-021-02553-2. url: https://doi.org/10.1186/s13059-021-02553-2

Reference Julian Späth et al. “Privacy-aware multi-institutional time-to-event studies”. In: *PLOS Digital Health* 1.9 (Sept. 2022), pp. 1–16. doi: 10.1371/journal.pdig.0000101. url: https://doi.org/10.1371/journal.pdig.0000101

Reference Reihaneh Torkzadehmahani et al. “Privacy-Preserving Artificial Intelligence Techniques in Biomedicine”. EN. in: *Methods of Information in Medicine* 61.S 01 (2022), e12–e27. issn: 0026-1270. doi: 10.1055/s-0041-1740630. url: http://www.thieme-connect.com/products/ejournals/abstract/10.1055/s-0041-1740630

Reference Reza Nasirigerdeh et al. *HyFed: A Hybrid Federated Framework for Privacy-preserving Machine Learning*. 2021. doi: 10.48550/ARXIV.2105.10545. url: https://arxiv.org/abs/2105.10545

Reference Han Cao et al. “dsMTL: a computational framework for privacy-preserving, distributed multi-task machine learning”. In: *Bioinformatics* 38.21 (Sept. 2022), pp. 4919–4926. issn: 1367-4803. doi: 10.1093/bioinformatics/btac616. eprint: https://academic.oup.com/bioinformatics/article-pdf/38/21/4919/46697912/btac616.pdf. url: https://doi.org/10.1093/bioinformatics/btac616

A
CoVex

Sepideh Sadegh et al. “Exploring the SARS-CoV-2 virus-host-drug interactome for drug repurposing”. In: *Nature Communications* 11.1 (July 2020), p. 3518. ISSN: 2041-1723. DOI: 10.1038/s41467-020-17189-2. URL: https://doi.org/10.1038/s41467-020-17189-2

nature COMMUNICATIONS

ARTICLE

https://doi.org/10.1038/s41467-020-17189-2 OPEN

Check for updates

Exploring the SARS-CoV-2 virus-host-drug interactome for drug repurposing

Sepideh Sadegh[1,6], Julian Matschinske[1,6], David B. Blumenthal[1], Gihanna Galindez[1], Tim Kacprowski[1], Markus List[1], Reza Nasirigerdeh[1], Mhaned Oubounyt[1], Andreas Pichlmair[2], Tim Daniel Rose[3], Marisol Salgado-Albarrán[1,4], Julian Späth[1], Alexey Stukalov[2], Nina K. Wenke[1], Kevin Yuan[1], Josch K. Pauling[3] & Jan Baumbach[1,5]✉

Coronavirus Disease-2019 (COVID-19) is an infectious disease caused by the SARS-CoV-2 virus. Various studies exist about the molecular mechanisms of viral infection. However, such information is spread across many publications and it is very time-consuming to integrate, and exploit. We develop CoVex, an interactive online platform for SARS-CoV-2 host interactome exploration and drug (target) identification. CoVex integrates virus-human protein interactions, human protein-protein interactions, and drug-target interactions. It allows visual exploration of the virus-host interactome and implements systems medicine algorithms for network-based prediction of drug candidates. Thus, CoVex is a resource to understand molecular mechanisms of pathogenicity and to prioritize candidate therapeutics. We investigate recent hypotheses on a systems biology level to explore mechanistic virus life cycle drivers, and to extract drug repurposing candidates. CoVex renders COVID-19 drug research systems-medicine-ready by giving the scientific community direct access to network medicine algorithms. It is available at https://exbio.wzw.tum.de/covex/.

[1] Chair of Experimental Bioinformatics, TUM School of Life Sciences, Technical University of Munich, München, Germany. [2] Institute of Virology, TUM School of Medicine, Technical University of Munich, München, Germany. [3] LipiTUM, Chair of Experimental Bioinformatics, TUM School of Life Sciences, Technical University of Munich, München, Germany. [4] Natural Sciences Department, Universidad Autónoma Metropolitana-Cuajimalpa (UAM-C), 05300 Mexico City, Mexico. [5] Computational Biomedicine Lab, Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark. [6] These authors contributed equally: Sepideh Sadegh, Julian Matschinske. ✉email: jan.baumbach@wzw.tum.de

Coronavirus Disease-2019 (COVID-19) is an infectious disease caused by SARS-CoV-2 (severe acute respiratory syndrome coronavirus 2). It was first identified in Wuhan, China and has spread causing an ongoing pandemic[1] with globally 2.4 million confirmed cases and 167 thousand deaths as of April 20, 2020.

Our insights into SARS-CoV-2 infection mechanisms are limited and clinical therapy has largely focused on treating critical symptoms. Therefore, the current pandemic requires fast and freely accessible knowledge to accelerate the development of vaccines, treatments, and diagnostic tests. Research data have been collected in several online platforms, such as the COVID-19 Open Research Dataset and the Dimensions COVID-19 collection[2,3]. In addition, existing databases that collect virus information have responded by integrating new SARS-CoV-2 research[4,5].

As vaccine and drug development may take years, drug repurposing is a potent approach that offers new therapeutic options through the identification of alternative uses of already approved drugs[6]. These drugs have previously undergone clinical and safety trials and, hence, accelerate drug development timelines from a decade to a few years or months. Due to the COVID-19 pandemic, numerous research groups around the world have been joining their efforts to identify drugs that can be repurposed to effectively treat COVID-19. Numerous drugs are already part of clinical trials, including Remdesivir (a less effective ebola drug), Chloroquine, Hydroxychloroquine (antimalarial drugs), Tocilizumab (rheumatoid arthritis drug), Favipiravir (influenza drug), and Kaletra (a combination of Lopinavir and Ritonavir for treating human immunodeficiency virus HIV-1)[7].

Computational systems and network medicine approaches offer a methodological toolbox required to understand molecular virus–host–drug mechanisms and to predict novel drug targets to attack them[8,9]. Few studies on these mechanisms in SARS-CoV-2 exist. Gordon et al.[10] applied affinity purification-mass spectrometry (AP-MS) to reconstruct the SARS-CoV-2-human protein–protein interaction (PPI) network and subsequently employed a chemoinformatics approach to identify potential drugs for repurposing. The data generated from that study is a major advancement in understanding SARS-CoV-2 infection. However, to identify drug candidates, the study mainly considered the direct interactors of the human proteins as putative targets and thus did not take into account the network context of the human interactome. However, viral interactions with human proteins have cascading effects in the human interactome, where key proteins necessary for the viral replication cycle are only indirectly affected. Therefore, downstream host proteins may be additional promising targets for therapeutic intervention, but require thorough data integration and mining to be identified (see Supplementary Methods for details). Figure 1 illustrates the concept of systems medicine-based drug repurposing specifically for SARS-CoV-2.

Gysi et al.[11] integrated the experimentally validated SARS-CoV-2 virus–host interactions with the human interactome and investigated comorbidity and differences of virus–host interactions across 56 tissues. Furthermore, network medicine analysis was applied to compile a list of drug repurposing candidates that target also indirectly affected proteins in the human interactome. However, the combined number of virus–host, host–host, and drug–target interactions goes into the millions such that purely algorithmic approaches to discovering new drug targets and drug repurposing candidates produces a large number of results, many of which lack mechanistic specificity and, hence, are not useful. Thus, to make their results accessible, Gysi et al.[11] worked closely together with clinical experts to narrow down the number of predicted repurposable drugs.

In order to allow for the interactive integration of expert knowledge about virus replication, immune-related biological processes, or drug mechanisms, we developed the interactive systems and network medicine platform CoVex (CoronaVirus Explorer). It integrates experimental virus–human interaction data for SARS-CoV-2 and SARS-CoV-1 with the human interactome as well as drug information to predict novel drug (target) candidates, and it offers biomedical and clinical researchers' interactive and user-friendly access to network medicine algorithms for advanced data mining and hypothesis testing. CoVex follows a human-in-the-loop paradigm and provides an intuitive visualization of virus–host interactions, drug targets, and drugs to enable researchers to examine molecular mechanisms that can be targeted using repurposed drugs. CoVex offers two main actions for which several network medicine algorithms are available: Given a list of user-selected human host proteins, viral proteins, or drugs (referred to as seeds), users can (1) search the human interactome for viable drug targets and (2) identify repurposable drug candidates. In a typical workflow, these two actions are combined, that is, starting from a selection of virus or virus-interacting proteins, users mine the interactome for suitable drug targets for which, in turn, suitable drugs are identified. Additionally, users can leverage expert knowledge by uploading a list of proteins or drugs of interest as seeds to guide the analysis. Such seeds could, for instance, be a list of differentially expressed genes (DEGs), a list of proteins related to a molecular mechanism of interest, or a set of drugs known to be effective.

The remainder of this paper is structured as follows: In the "Methods" section, we first describe the datasets and integration strategy used in CoVex. Next, we introduce the rationales of the systems and network medicine algorithms implemented in CoVex, and briefly describe the overall architecture of the platform. In the "Results" section, we show several application examples to illustrate the flexibility and typical use cases of CoVex. Finally, we will discuss opportunities and limitations in using CoVex for COVID-19 research.

CoVex opens up the systems medicine toolbox for the entire infectious disease research community by providing an easy-to-use web tool enriched with data mining algorithms for drug repurposing. This allows specialists from different fields to bring in expert knowledge to identify the most promising drug targets and drug repurposing candidates for developing effective therapies. We would like to stress that the CoVex platform can and will be adopted and extended to allow exploring other viral–host–drug interactomes, for example, with MERS (Middle East respiratory syndrome), Zika, dengue, and influenza viruses, thereby increasing preparedness for similar future events.

Results

The CoVex platform. The main result is the CoVex platform itself, which renders drug repurposing research systems-medicine-ready. In the following, we first describe how the platform's user interface (Fig. 2) provides the full feature spectrum of CoVex to clinicians and scientists. Afterwards, we demonstrate the use of CoVex in four different application scenarios starting with four hypotheses and ending with different drug repurposing candidates, as well as a short discussion on how to prioritize them (Fig. 3).

Figure 2 shows the CoVex web interface. To find potential drugs, the "Quick Start" analysis will produce a multi-Steiner tree, which considers all viral proteins as seeds and adds a small number of host proteins to connect them. Subsequently, drugs directly targeting these proteins are selected via closeness centrality. After the computation has finished, a click on the corresponding task opens the analysis results, consisting of a table

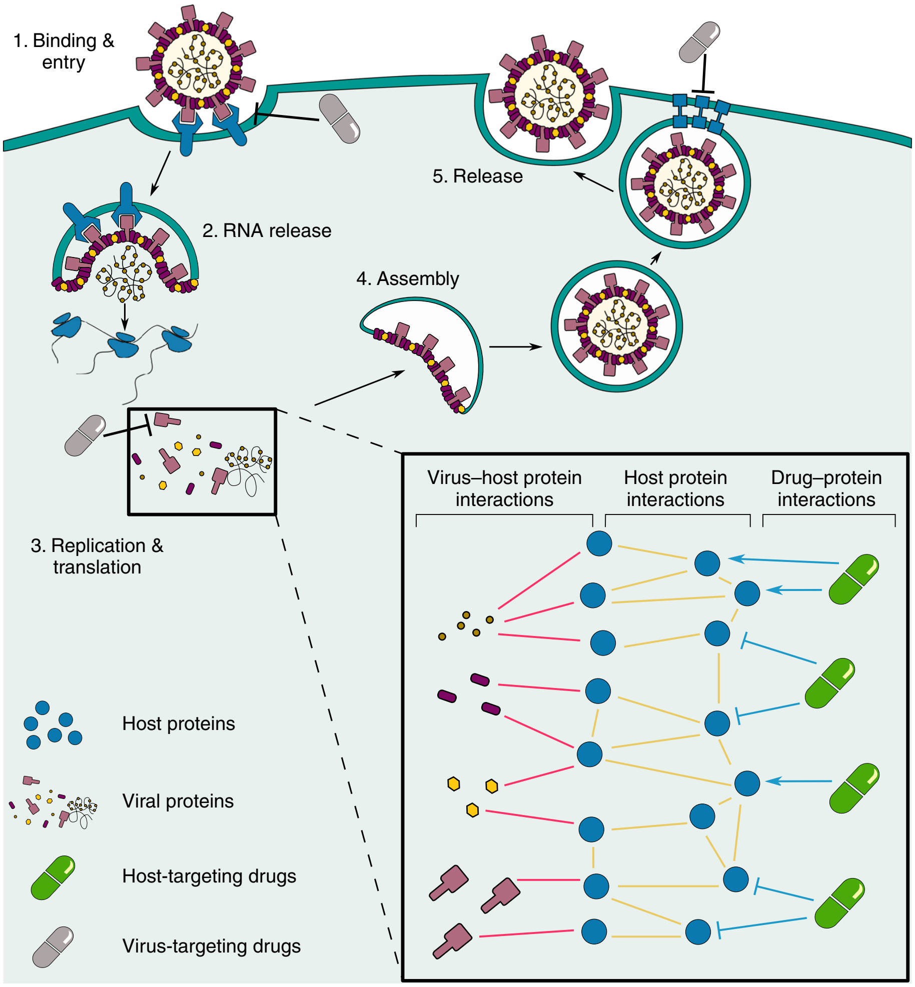


Fig. 1 The SARS-CoV-2 life cycle and the CoVex systems medicine approach of drug repurposing. Most antiviral drugs (gray drugs) target the virus proteins or their direct host interactor proteins to inhibit different stages of the viral life cycle. Our rationale, however, is that viral interactions with human host proteins have a cascading effect to hijack and control key proteins necessary for the virus' life cycle. We aim to identify repurposable drug candidates (green drugs) targeting these key host modulators to interfere with virus replication and disease progression following infection. Besides an increased antiviral drug repertoire, targeting host proteins would make it more difficult for the virus (population) to develop resistance mutations.

view of drugs and proteins, a visualization of the protein–protein and drug–protein interactions, and a list of parameters used for the analysis. In the "Simple Analysis" panel, users can select seed proteins manually and search for drugs targeting them. In the "Advanced Analysis" panel, users can choose from a list of network medicine algorithms (see "Methods" and Supplementary Methods for details) to discover drug targets or drug repurposing candidates. Users can either select proteins from the view, upload a custom list of proteins or drugbank ids, or select proteins expressed in a given tissue. An enrichment analysis of the identified drug target proteins may be performed with g: Profiler[12].

Application scenarios. The utility of CoVex and its integrated systems medicine approaches is outlined in the following four scenarios. More details on each can be found in the Supplementary Notes.

Scenario a: Starting from a selection of viral proteins, we use the PPI network to identify the biological mechanism or pathway utilized by the virus. As an example, we consider the viral proteins E, M, and Spike, which constitute the external structure of the virus and thus mediate entry into the host cells during the infection process[13,14]. We select the interactors of these viral proteins reported for SARS-CoV-2 and use the multi-Steiner tree algorithm to uncover the biological pathway involved. The resulting network (Fig. 4) yields 26 new potential drug targets, including the bradykinin receptor B1 (BDKRB1). Subsequently, we use closeness centrality to find drugs affecting this pathway. Notably, we identify six relevant drugs that target BDKRB1: Ramipril, Captopril, Perindopril, and Enalaprilat (approved), which belong to the angiotensin-converting enzyme (ACE) inhibitor class[15]; Icatibant, an antagonist of the bradykinin receptor B2[16]; and bradykinin, a non-approved drug that is degraded by the ACE[17]. Furthermore, to understand the relationship between BDKRB1 and two proteins known to participate in the entry of the virus (angiotensin-converting enzyme 2 (ACE2) and transmembrane protease serine 2)[18], we use the "custom proteins" option available in CoVex. We found that kininogen 1 and angiotensin proteins connect BDKRB1 with ACE2. These four proteins are functionally related through the

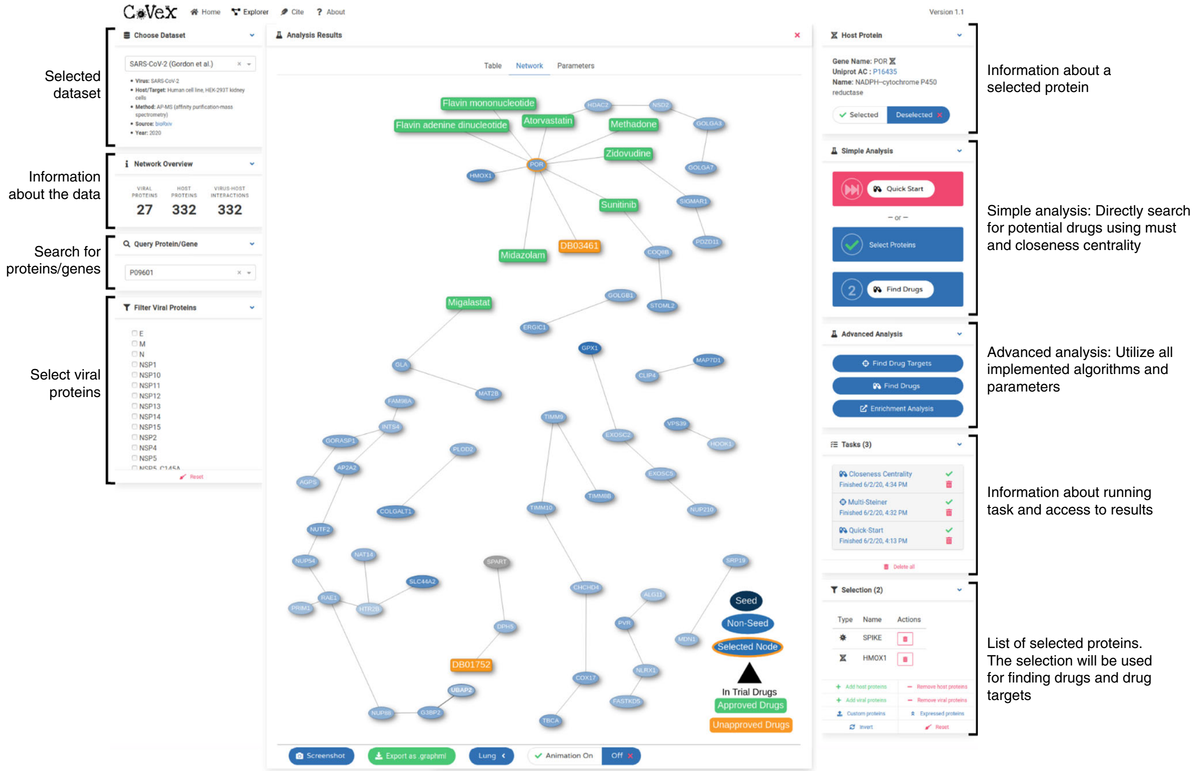


Fig. 2 The CoVex online platform. The network view (middle) shows drug candidates (green nodes) that were found using closeness centrality on a set of proteins (blue nodes), which resulted from a multi-Steiner tree computation with all viral proteins as seeds (not shown here). Therefore, drugs targeting these seeds might be able to interrupt the viral life cycle progression. Here we colored nodes based on lung-tissue-specific median gene expression according to GTEx.

renin–angiotensin system, which is targeted by ACE inhibitors (www.wikipathways.org/instance/WP554). In summary, CoVex identifies the protein BDKRB1, which appears to play a role in SARS-CoV-2 host cell entry and can be targeted by several ACE inhibitors widely used in clinical trials to treat COVID-19. It should be noted that the ACE2 protein is not present in the set of seeds used to start the analysis. Nevertheless, CoVex is capable of identifying the pathway and new protein targets functionally related to ACE2 (Fig. 4).

Scenario b: Starting from both viral proteins and a list of proteins of interest, we can use CoVex to identify a connecting pathway or biological mechanisms that can be targeted by drugs. In this scenario, we are specifically interested in viral proteins that suppress host immunity and the corresponding host immune response pathways. First, we select the viral proteins ORF7a and ORF3a, which are potentially involved in innate immune response and apoptosis as discussed by Gordon et al.[10]. Next, we compile a list of proteins of interest based on the DEGs from the study by Blanco-Melo et al.[19] lung epithelial cells were infected with the SARS-CoV-2 virus, leading to altered expression of immunity-related genes to combat the viral infection. We consider DEGs known to be associated with the host pathway involving infection with the herpes simplex virus, another viral pathogen. These genes include *IFIH1*, *OAS1*, *STAT1*, *DDX58*, *OAS2*, *OAS3*, *IRF7*, *EIF2AK2*, *IFIT1*, and *IRF9*. The selected viral proteins and DEGs (converted to Uniprot ids) were used as seeds for the multi-Steiner tree algorithm to extract a potential immune-related mechanism. As expected, the resulting subnetwork reveals that the viral proteins are close to the DEGs in the host PPI network. Closeness centrality analysis assigned a high rank to Tofacitinib and Ruxolitinib, which are currently being assessed in clinical trials. Tofacitinib and Ruxolitinib exert immunomodulatory effects as Janus kinase inhibitors[20,21]. Thus, administration with these drugs may mitigate immune-mediated lung injury and reduce functional deterioration caused by an overamplified host inflammatory response. This could be especially important in later stages of the disease to prevent an overreaction of the body's immune system and, hence, may further prevent the need for mechanical ventilation in patients suffering from severe COVID-19. Other drugs that target this subnetwork include Masitinib, Erlotinib, and Sorafenib, which could be further examined in downstream analyses. In a similar manner, users may provide a custom list of proteins as seeds to hunt for drugs that can target a putative mechanism of interest.

Scenario c: Starting with a set of drugs of interest, we can follow a top-down approach to extract potential host mechanisms and additional drugs targeting the proteins participating in these mechanisms. As an example, we identify 69 drugs currently in clinical trials for COVID-19 and group them based on their Anatomical Therapeutic Chemical classification (Supplementary Table 5)[22]. We focus on drugs from the immunostimulants class (L03) and their target proteins as starting seeds. We further select the interactors of the immune-related viral proteins ORF9B, ORF6, ORF3B, and ORF3A[10] as end-point seeds. By applying the multi-Steiner tree algorithm, we discover pathways of interacting proteins that connect the selected drugs (and their target

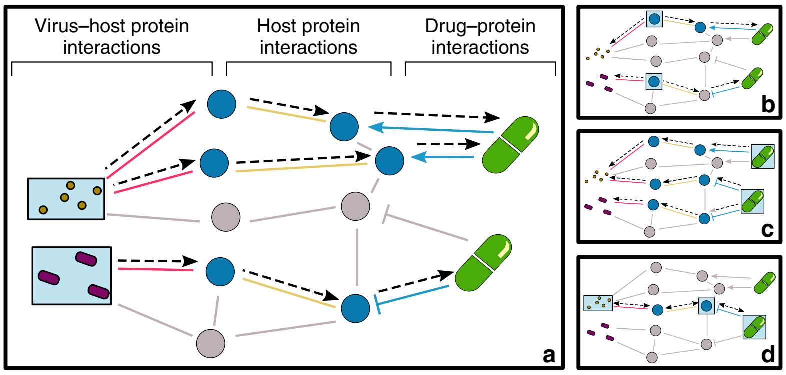

Virus–host protein interactions
Host protein interactions
Drug–protein interactions
a
b
c
d

Fig. 3 CoVex application scenarios. Depending on the starting hypothesis, dedicated systems medicine algorithms will propagate from selected seeds to connect drugs with viral proteins using host proteins as proxies. Essentially, four different strategies apply: **a** Starting with viral proteins, one can identify drugs targeting host proteins that connect the viral seeds. **b** Starting with a set of proteins of interest as proxies, we identify pathways connecting them to (selected or all) viral proteins. Subsequently, we identify drugs targeting this mechanism. **c** Starting with a set of drugs of interest, one may find pathways to (selected or all) viral proteins extracting a potentially druggable host mechanism. **d** Hypothesis-driven, hybrid approach with seeds in different levels to be connected for druggable mechanism extraction. Boxes with light blue background indicate the typical starting points in the respective application scenario.

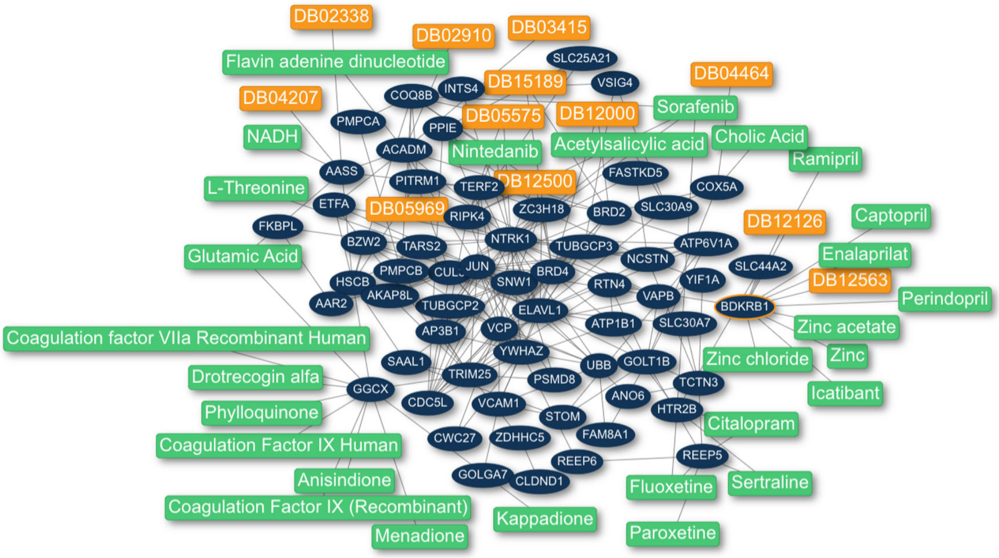


Fig. 4 CoVex result network for application scenario a. Drug–protein–protein interaction network obtained using the viral proteins E, M, and Spike with multi-Steiner tree followed by closeness centrality. Blue nodes are protein targets. Green nodes are approved drugs and orange nodes are non-approved drugs. Lines represent the interactions between proteins and drugs. Note that some ACE inhibitor drugs have been identified, such as Ramipril, Captopril, Perindopril, and Enalaprilat targeting the BDKRB1 protein, which are currently being evaluated in clinical trials.

proteins) with the selected viral proteins. Among these connector proteins, we find five genes associated with cytokine signaling in the immune system according to Reactome Pathways (*CSF2*, NRG1, *NUP188*, *PTPN18*, *SOCS1*)[23]. Notably, *CSF2* is enriched in lung, pancreas, and immune cells (www.proteinatlas.org/ENSG00000164400-CSF2)[24] and can be inhibited by KB002 (DB05194), which is an investigational drug and an engineered human monoclonal antibody treatment for inflammatory and autoimmune processes[25]. In summary, with CoVex, we found a new drug target that may play a key role in the host immune response during viral infection. We also identified a new drug candidate for COVID-19, as it targets the proteins involved in the pathogenic mechanisms triggered by ORF3A, ORF3B, ORF6, and ORF9B viral proteins.

Scenario d: Starting from a hypothesis-driven mixed selection of viral and host proteins, as well as drugs, we seek to utilize PPIs to identify a full mechanism or pathway and to suggest additional drug candidates. As an application case, we follow-up on a recently published hypothesis by Liu and Abrahams concerning the putative interference of SARS-CoV-2 with the formation of hemoglobin in erythrocytes[26,27]. Essentially, the virus is believed to interfere with heme formation causing symptoms of hypoxia. Liu and Abrahams hypothesize that this would also explain why Chloroquine and Favipiravir are effective drugs, as they may prevent the viral proteins from competing with iron for the porphyrin in hemoglobin (NSP1-16, ORF3a, ORF10, and ORF8 targeted by Chloroquine as well as ORF7a targeted by Favipiravir)[26,27]. Based on this hypothesis (discussed in more detail in the Supplementary Notes), we investigate the pathways connecting these viral proteins with the two effective drugs Chloroquine and Favipiravir. To this end, we select two known heme binding host proteins as seeds: cytochrome b5 reductase, which interacts with the viral protein NSP7, and the viral ORF3a, which binds to heme oxygenase 1. Using KeyPathwayMiner for drug target discovery followed by closeness centrality for drug discovery, we identify methylene blue in addition to Chloroquine and Deferoxamine, which are both in COVID-19 clinical trials[28,29]. Notably, methylene blue is approved by the Food and Drug Administration for the treatment of methemoglobinemia, which fits the investigated hypothesis (reduced oxygen-carrying capacity). Also, Deferoxamine is widely used therapeutically as a chelator of ferric ions in disorders of iron overload[30]. However, note that the available scientific evidence for a methemoglobinemia or ferric ion imbalance caused by SARS-CoV-2 is very limited (see Supplementary Notes) and that we use this hypothesis solely to illustrate the potential of CoVex' network medicine investigation and hypothesis testing capabilities.

Discussion

COVID-19 is a threat to our health and our social life, as well as to our healthcare and economic systems around the globe. Since the development of safe and effective vaccines is a time-consuming process, the only alternative to mitigate the damage by the SARS-CoV-2 pandemic is to quickly identify agents for the treatment and control of COVID-19 symptoms. Much attention in biomedical and clinical research is, thus, given to the task of identifying therapeutically exploitable drugs. A particular interest lies in drug repurposing, since already approved drugs can go through shortened clinical trials within months rather than years. While a number of promising drug repurposing candidates are currently being tested, the discovery of such candidates is still unstandardized and mostly unstructured. Systems and network medicine offer alternative approaches, where the process of drug target discovery is driven by computational data mining methods utilizing molecular interaction networks. As recently demonstrated by Gysi et al.[11] for SARS-CoV-2, this data-driven process can produce a list of promising drug candidates targeting host proteins in close proximity and mechanistically related to virus-interacting proteins[11]. Here, we seek to make this network medicine approach widely available to the community.

With CoVex, we present an interactive and user-friendly web platform that integrates published data of SARS-CoV-1 as well as recent data about virus–host interactions in SARS-CoV-2[10] with the human interactome and several drug–target interaction databases. CoVex allows users to mine the integrated virus–host–drug interactome for putative drug targets and drug repurposing candidates with only a few mouse clicks. Through features such as interactive seed protein selection, filtering, and upload of own lists of proteins or drugs of interest, CoVex covers diverse application scenarios ranging from data-driven, hypothesis-free drug target discovery to expert-guided analyses with a clear underlying hypothesis about virus biology. To address the diversity of research questions adequately, CoVex implements several state-of-the-art graph analysis methods. These were specifically tailored to be employed in a network medicine context and include a weighted version of TrustRank as well as a multi-Steiner tree method (Supplementary Material).

While CoVex is a powerful tool for SARS-CoV-1 and -2 research, results uncovered with our platform have to be considered with caution. We stress that CoVex can only suggest putative drug candidates for further investigation and that those candidates are not guaranteed to have an antiviral effect. While the suggested drugs target proteins involved in a putatively important mechanism for the virus, the actual effect of the drug has to be verified through follow-up investigations. The inhibition of a cofactor that prevents the virus from manipulating host proteins, for example, could even have a proviral effect. After validating the target for the suggested drug through appropriate genetic or chemical approaches, the drug candidate, hence, still needs to be properly vetted by clinical experts and tested following established procedures and clinical trials. Current data about virus–host interactions in SARS-CoV-2 is still preliminary and incomplete. For instance, important proteins such as the ACE2 receptor, a known entrypoint for the virus[18], is missing in the SARS-CoV-2 dataset by Gordon et al.[10]. Moreover, we included only drugs that are reported in databases about clinical trials or in the literature if they have a valid entry in DrugBank, possibly excluding some of the drugs currently being investigated. Further, we do not differentiate between different sources of drug–target interactions. The strength of experimental evidence may vary depending on the experimental assay that was used or the type of annotation from the source database, for example, clinical and variant annotations from PharmGKB, which can be interpreted as indirect drug–protein associations. It should also be noted that we do not list drugs that target viral proteins directly, as the goal of CoVex is to unravel novel drug targets further downstream in the human interactome.

We acknowledge that the choice of algorithm and its associated parameters is nontrivial, forcing users to engage in time-consuming explorative analysis. To make this easier, we allow users to queue multiple tasks, which are executed in parallel. As our experience with this platform grows, we also plan to develop guidelines that allow users to choose an appropriate method for a particular research question. We further plan to integrate new data about virus–host interactions and ongoing clinical trials in corona viruses as it becomes available.

In summary, we have presented CoVex, a web-based platform for the interactive exploration and network-based analysis of virus–host interactions, aimed towards drug repurposing for the treatment of COVID-19. CoVex can be easily updated to accommodate the fast-paced data generation in the battle against the global pandemic. CoVex is expected to speed up the discovery of potential therapeutics for COVID-19. For the future, we also plan to extend the CoVex network medicine platform to other viruses in which new drug targets and drug repurposing candidates are urgently sought, including MERS, Zika, influenza, and dengue. We will also add features for the integration of additional molecular data, such as gene expression. Until then users can work with the "add custom protein" functionality of CoVex, allowing them to utilize and filter by any set of genes, including those derived by gene expression pattern analyses.

Methods

Data integration. We integrated virus–host interaction data from several sources. We obtained SARS-CoV-2 AP-MS data reported by Gordon et al.[10], containing 332 high-confidence virus–host interactions for 27 SARS-CoV-2 proteins[10], as well as SARS-CoV-1 interactions from VirHostNet[4] (24 interactions), and Pfefferle et al.[31] (113 interactions existing in our interactome). Human PPIs were obtained from the integrated interactions database[32] filtered based on experimental validation. The resulting interactome consists of 17,666 proteins connected via 329,215 interactions. Drug–target associations were obtained from ChEMBL (2020-03)[33], DrugBank (v. 5.1.5)[25], DrugCentral (2018-08-26)[34], Target Therapeutic Database (2019-07-14)[35], Guide To Pharmacology (2020-01; only approved drugs)[36], PharmGKB (downloaded 2020-04)[37], and BindingDB (2019-08-12)[38]. Where applicable, we considered drugs that have binding affinity values (EC50, IC50, K_d, and K_i) <10 μM[39,40]. Only drugs that were mappable to DrugBank IDs and targeting host proteins were included in the network. Drugs currently undergoing clinical trials and mappable to DrugBank IDs (as of April 4, 2020) for the treatment of COVID-19 were collected from ClinicalTrials.gov (www.ClinicalTrials.gov)[41], the EU Clinical Trials Register (www.clinicaltrialsregister.eu), and the International Clinical Trials Registry Platform (www.who.int/ictrp/). In total, we have 6861 drugs (67 in clinical trials) and 52,860 drug–target associations integrated in our network. We further downloaded per-tissue median gene expression levels from the GTEx data portal (Release V8, dbGaP Accession phs000424.v8.p2, downloaded 2020-05-30) to allow for tissue-specific filtering and visualization of gene expression values. Note that we rely on integrating published data and, thus, on their corresponding quality.

Systems medicine algorithms for drug repurposing prediction. The general idea of CoVex is to provide researchers and clinicians with a tool to visually explore druggable molecular mechanisms that drive the interactions between virus and host. To this end, the integrated virus–human–drug interactions form molecular networks that are modeled as graphs with nodes as proteins or drugs, and edges referring to interactions between them. The goal of CoVex is to explore this network while allowing for the exploitation of expert knowledge. Starting with a selected set of (usually) hypothesis-driven seeds (virus proteins, human proteins, or drugs), the goal is to first identify subnetworks connecting these seeds and, subsequently, to identify drug repurposing candidates associated with these mechanisms. A vast number of methods have been reported in the literature for identifying subnetworks[42]. In CoVex, we have integrated several algorithms (including a dedicated multi-Steiner tree algorithm) with different underlying paradigms to provide specific exploration options to various particular medical, therapeutic, and research questions and hypotheses. CoVex, thus, allows users to choose among the following approaches in the "advanced analysis" procedures.

Degree centrality is the simplest conceivable centrality measure and ranks proteins or drugs interacting with the seeds by their node degree, that is, the number of interactions. Thus, this algorithm yields subnetworks in which seed-connected proteins and/or drugs are preferentially selected if they interact with many other proteins in the network. The only user-selected parameter is the result size, that is, how many of the top-ranked proteins or drugs are included. Notably, centrality measures in CoVex can be used for detecting drug targets and for identifying promising drugs.

Closeness centrality is a node centrality measure that ranks the nodes in a network based on the lengths of their shortest paths to all other nodes in the network. The rationale behind this algorithm is to preferentially select proteins and/or drugs that are a short distance from all other proteins in the network and are thus of central importance. In CoVex, we use a modified version suggested by Kacprowski et al.[43], where only the shortest paths to a set of selected seed nodes are considered. The only algorithm-specific, user-selected parameter is the result size.

Betweenness centrality is another node centrality measure that ranks the nodes in a network based on how many shortest paths pass through them. In CoVex, we use a modified version suggested by Kacprowski et al.[43], which only considers shortest paths between pairs of seed nodes. Hence, nodes receive a high score if they are on many shortest paths between the seeds. Since drugs are not contained in any shortest paths in our integrated interactome (see Fig. 1), betweenness centrality can be used only to find drug targets. The only algorithm-specific, user-selected parameter is the result size.

Guney et al.[44] introduced the network proximity between a drug and a set of seed nodes as the average minimum distance from the drug's targets to all of the seeds. The algorithm computes empirical z-scores by comparing the obtained proximity score to a background distribution obtained by randomly sampling sets of seed nodes and drug targets. In CoVex, network proximity can be employed to find drugs, given a set of host proteins of interest. The user can specify the result size, as well as the number of randomly sampled instances used for computing the background distribution.

TrustRank is conceptually similar to closeness centrality but additionally considers the importance of the seed nodes themselves. In other words, TrustRank ranks nodes in a network based on how well they are connected to a (trusted) set of seed nodes[45]. It is a variant of Google's PageRank algorithm, where "trust" is iteratively propagated from seed nodes to neighboring nodes using the network structure. The node centralities are initialized by assigning uniform probabilities to all seeds and zero probabilities to all non-seed nodes. In CoVex, the TrustRank algorithm can be run starting from a user-defined set of (trusted) seed proteins to obtain a ranked list of proteins in the PPI network that could be prioritized as putative drug targets. Similarly, TrustRank can be executed on the joint protein–drug interactome to identify drug repurposing candidates. User-selected parameters include the result size and the damping factor (range 0–1), which controls how fast "trust" is propagated through the network. A small damping factor results in a conservative behavior of the algorithm (nodes close to the seeds receive much higher scores than distant ones), while a large damping factor makes its behavior more explorative.

The Steiner tree problem is a classical combinatorial optimization problem. It aims at finding a subgraph of minimum cost connecting a given set of seed nodes. For CoVex, we have developed a weighted multi-Steiner tree method that computes approximate weighted multiple Steiner trees and connects them to one subnetwork. The user can select the set of proteins of interest and extract subnetwork(s) that connect the selected seed proteins as candidate mechanism(s) involved in COVID-19 progression. In this mechanistic subnetwork(s), we can then extract essential proteins and, thus, the most promising drug targets and repurposable drugs for COVID-19. User-selected parameters include the number of Steiner trees to be merged as well as the tolerance towards accepting more expensive subnetworks (for speeding up the approximation algorithm; for details see Supplementary Methods).

KeyPathwayMiner is a network enrichment tool that identifies condition-specific subnetworks (key pathways)[46]. In CoVex, we utilize the KeyPathwayMiner web service to extract a maximally connected subnetwork starting from a user-defined set of proteins of interest (seeds). The only user-selected parameter is K, which represents the number of permitted exception nodes, that is, proteins that were not part of the seed proteins but serve to connect them. Since these proteins act as bridges, these may represent key proteins participating in the dysregulated subnetwork even though they are not directly targeted by the virus and are therefore promising candidates for intervention. In its current implementation, exception nodes will only be added if they indeed possess a bridging characteristic and will not be shown otherwise.

Irrespective of the network analysis method used, the extracted solutions have a higher intrinsic probability to contain high-degree nodes (hubs), that is, proteins that have a large number of interactions. While these proteins are key players in the human interactome, they are not necessarily suitable drug targets as perturbing them might lead to severe unintended side effects. Since it is more likely that hub proteins are involved in several mechanisms and are not specific to the mechanism of the disease under study, users can also perform the analysis with the hub penalty, which can potentially favor more specific mechanisms related to COVID-19. To mitigate this bias, users can either select an upper bound to filter out high-degree nodes or, alternatively, penalize high-degree nodes by incorporating the degree of neighboring nodes as edge weights in the optimization. For the latter, values between 0 and 1 can be selected, where higher values correspond to a higher penalty. Both options are available in advanced analyses for all methods except for degree centrality, because its rationale is to identify hubs, and KeyPathwayMiner, which conceptually does not allow for weighted subnetwork extraction.

All network algorithms except multi-Steiner tree and KeyPathwayMiner yield scores for the nodes contained in the returned subnetwork. In the case of degree centrality, closeness centrality, betweenness centrality, and TrustRank, these scores correspond to, respectively, the number of direct interactions with the seeds, the inverse of the mean distance to the seeds, the fraction of shortest paths between the seeds passing through the node, and the "trust" on the node at termination. In all four cases, high scores indicate that the nodes are central with respect to the seeds, but the scores do not carry any intrinsic statistical semantics. In CoVex, we hence display normalized scores for degree centrality, closeness centrality, betweenness centrality, and TrustRank, which we compute by dividing by the obtained maximum. In contrast to that, network proximity yields empirical z-scores, which are smaller the more promising the drugs are for the selected set of seed proteins. Since these z-scores directly translate into empirical p values, we do not normalize them.

Implementation. CoVex consists of five components: (i) Data are stored in a PostgreSQL database (v. 12.2). (ii) The backend is implemented using the Django web framework (v. 3.0.5) with Python (v. 3.6) and the Django REST framework (v. 3.11.0) to build the web API. (iii) The network algorithms (except KeyPathwayMiner) are implemented with graph-tool (v. 2.3.1)[47]. (iv) Background task processing is implemented using Redis Queue (RQ, v. 1.3.0) and the in-memory database Redis (v. 3.4.1). Django enqueues the jobs and RQ processes them in the background while Redis functions as a broker between Django and RQ. (v) The frontend is implemented in Angular (v. 9.0.2) and utilizes the JavaScript libraries vis-data (v. 6.5.1) and vis-network (v. 7.4.2) for network visualization.

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

Data availability

The authors declare that all data supporting the findings of this study are available publicly and their integration is described accordingly within the paper and its supplementary information files. Human protein–protein interactions were obtained from the Integrated Interactions Database (http://iid.ophid.utoronto.ca/). Virus–host interactions were downloaded from VirHostNet (http://virhostnet.prabi.fr/). Drug–target associations were integrated from the following databases: ChEMBL (https://www.ebi.ac.uk/chembl/), DrugBank (https://www.drugbank.ca/), DrugCentral (http://drugcentral.org/), Target Therapeutic Database (http://bidd.nus.edu.sg/group/cjttd/), Guide To Pharmacology (https://www.guidetopharmacology.org/), PharmGKB (https://www.pharmgkb.org/), and BindingDB (https://www.bindingdb.org/bind/index.jsp). Drugs undergoing clinical trials for COVID-19 were collected from ClinicalTrials.gov (https://clinicaltrials.gov/), the EU Clinical Trials Register (https://www.clinicaltrialsregister.eu/), and the International Clinical Trials Registry Platform (https://www.who.int/ictrp/en/). Tissue-specific gene expression levels were obtained from the GTEx data portal (https://www.gtexportal.org/home/, dbGaP Accession phs000424.v8.p2).

Code availability

CoVex is a public online platform software running on a web server. The CoVex code is available from the corresponding author upon reasonable request. The online tool is available at https://exbio.wzw.tum.de/covex/.

Received: 23 April 2020; Accepted: 12 June 2020;
Published online: 14 July 2020

References

1. World Health Organisation. Coronavirus. https://www.who.int/emergencies/diseases/novel-coronavirus-2019 (2020).
2. Dimensions Resources. Dimensions COVID-19 publications, datasets and clinical trials. https://dimensions.figshare.com/articles/dataset/Dimensions_COVID-19_publications_datasets_and_clinical_trials/11961063 (2020).
3. Semantic Scholar. COVID-19 open research dataset (CORD-19). https://pages.semanticscholar.org/coronavirus-research.
4. Guirimand, T., Delmotte, S. & Navratil, V. VirHostNet 2.0: surfing on the web of virus/host molecular interactions data. *Nucleic Acids Res.* **43**, D583–D587 (2015).
5. Guirimand, T., Delmotte, S. & Navratil, V. VirHostNet 2.0: surfing on the web of virus/host molecular interactions data. *Nucleic Acids Res.* **43**, D583–D587 (2014).
6. Sun, P., Guo, J., Winnenburg, R. & Baumbach, J. Drug repurposing by integrated literature mining and drug–gene–disease triangulation. *Drug Discov. Today* **22**, 615–619 (2017).
7. World Health Organisation. 'Solidarity' clinical trial for COVID-19 treatments. https://www.who.int/emergencies/diseases/novel-coronavirus-2019/global-research-on-novel-coronavirus-2019-ncov/solidarity-clinical-trial-for-covid-19-treatments (2020).
8. Casas, A. I. et al. From single drug targets to synergistic network pharmacology in ischemic stroke. *Proc. Natl. Acad. Sci. USA* **116**, 7129–7136 (2019).
9. Baumbach, J. & Schmidt, H. The end of medicine as we know it: introduction to the new journal, systems medicine. *Network Syst. Med.* **1**, 1–2 (2018).
10. Gordon, D. E., Jang, G. M., Bouhaddou, M., Xu, J. & Obernier, K. A SARS-CoV-2-human protein–protein interaction map reveals drug targets and potential drug-repurposing. *BioRxiv* https://doi.org/10.1038/s41586-020-2286-9 (2020).
11. Gysi, D. M. et al. Network medicine framework for identifying drug repurposing opportunities for COVID-19. Preprint at arXiv:2004.07229v1 [q-bio.MN] (2020).
12. Raudvere, U. et al. g:Profiler: a web server for functional enrichment analysis and conversions of gene lists (2019 update). *Nucleic Acids Res.* **47**, W191–W198 (2019).
13. de Haan, C. A. M. & Rottier, P. J. M. in *Advances in Virus Research*, Vol. 64, 165–230 (Academic Press, 2005).
14. Walls, A. C. et al. Structure, function, and antigenicity of the SARS-CoV-2 spike glycoprotein. *Cell* **181**, 281–292.e6 (2020).
15. Piepho, R. W. Overview of the angiotensin-converting-enzyme inhibitors. *Am. J. Health Syst. Pharm.* **57**(Suppl. 1), S3–S7 (2000).
16. HOE 140, JE 049, JE049. Icatibant. *Drugs R D* **5**, 343–348 (2004).
17. Kuoppala, A., Lindstedt, K. A., Saarinen, J., Kovanen, P. T. & Kokkonen, J. O. Inactivation of bradykinin by angiotensin-converting enzyme and by carboxypeptidase N in human plasma. *Am. J. Physiol. Heart Circ. Physiol.* **278**, H1069–H1074 (2000).
18. Hoffmann, M. et al. SARS-CoV-2 cell entry depends on ACE2 and TMPRSS2 and is blocked by a clinically proven protease inhibitor. *Cell* https://doi.org/10.1016/j.cell.2020.02.052 (2020).
19. Blanco-Melo, D. et al. SARS-CoV-2 launches a unique transcriptional signature from in vitro, ex vivo, and in vivo systems. *bioRxiv* https://doi.org/10.1101/2020.03.24.004655 (2020).
20. Elli, E. M., Baratè, C., Mendicino, F., Palandri, F. & Palumbo, G. A. Mechanisms underlying the anti-inflammatory and immunosuppressive activity of Ruxolitinib. *Front. Oncol.* **9**, 1186 (2019).
21. van Vollenhoven, R. et al. Evaluation of the short-, mid-, and long-term effects of Tofacitinib on lymphocytes in patients with rheumatoid. *Arthritis Arthritis Rheumatol.* **71**, 685–695 (2019).
22. WHOCC. WHOCC-ATC/DDD Index. *WHOCC* https://www.whocc.no/atc_ddd_index/ (2016).
23. National Library of Medicine. Cytokine signaling in immune system. *Reactome* https://reactome.org/content/detail/R-HSA-1280215 (2020).
24. Uhlen, M. et al. A pathology atlas of the human cancer transcriptome. *Science* **357**, https://doi.org/10.1126/*science*.aan2507 (2017).
25. Wishart, D. S. et al. DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Res.* **46**, D1074–D1082 (2018).
26. Abrahams, L. Covid-19: acquired acute porphyria hypothesis. Preprint at https://doi.org/10.31219/osf.io/4wkfy (2020).
27. Wenzhong, L. & Hualan, L. COVID-19: attacks the 1-beta chain of hemoglobin and captures the porphyrin to inhibit human heme. *Metabolism* https://doi.org/10.26434/chemrxiv.11938173.v7 (2020).
28. Vincent, M. J. et al. Chloroquine is a potent inhibitor of SARS coronavirus infection and spread. *Virol. J.* **2**, 69 (2005).
29. Wang, M. et al. Remdesivir and chloroquine effectively inhibit the recently emerged novel coronavirus (2019-nCoV) in vitro. *Cell Res.* **30**, 269–271 (2020).
30. Lederman, H. M., Cohen, A., Lee, J. W., Freedman, M. H. & Gelfand, E. W. Deferoxamine: a reversible S-phase inhibitor of human lymphocyte proliferation. *Blood* **64**, 748–753 (1984).
31. Pfefferle, S. et al. The SARS-coronavirus–host interactome: identification of cyclophilins as target for pan-coronavirus inhibitors. *PLoS Pathog.* **7**, e1002331 (2011).
32. Kotlyar, M., Pastrello, C., Malik, Z. & Jurisica, I. IID 2018 update: context-specific physical protein–protein interactions in human, model organisms and domesticated species. *Nucleic Acids Res.* **47**, D581–D589 (2019).
33. Mendez, D. et al. ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Res.* **47**, D930–D940 (2019).
34. Ursu, O. et al. DrugCentral 2018: an update. *Nucleic Acids Res.* **47**, D963–D970 (2019).
35. Wang, Y. et al. Therapeutic target database 2020: enriched resource for facilitating research and early development of targeted therapeutics. *Nucleic Acids Res.* **48**, D1031–D1041 (2020).
36. Armstrong, J. F. et al. The IUPHAR/BPS guide to pharmacology in 2020: extending immunopharmacology content and introducing the IUPHAR/MMV guide to malaria pharmacology. *Nucleic Acids Res.* **48**, D1006–D1021 (2020).
37. Barbarino, J. M., Whirl-Carrillo, M., Altman, R. B. & Klein, T. E. PharmGKB: a worldwide resource for pharmacogenomic information. *Wiley Interdiscip. Rev. Syst. Biol. Med.* **10**, e1417 (2018).
38. Gilson, M. K. et al. BindingDB in 2015: a public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic Acids Res.* **44**, D1045–D1053 (2016).
39. Talevi, A. Multi-target pharmacology: possibilities and limitations of the 'skeleton key approach' from a medicinal chemist perspective. *Front. Pharmacol.* **6**, 673 (2015).
40. Zhang, S., Zhao, H. & John, R. Development of a quantitative relationship between inhibition percentage and both incubation time and inhibitor concentration for inhibition biosensors—theoretical and practical considerations. *Biosens. Bioelectron.* **16**, 1119–1126 (2001).
41. Charatan, F. US launches new clinical trials database. *BMJ* **320**, 668 (2000).
42. Batra, R. et al. On the performance of de novo pathway enrichment. *NPJ Syst. Biol. Appl.* **3**, 6 (2017).
43. Kacprowski, T., Doncheva, N. T. & Albrecht, M. NetworkPrioritizer: a versatile tool for network-based prioritization of candidate disease genes or other molecules. *Bioinformatics* **29**, 1471–1473 (2013).
44. Guney, E., Menche, J., Vidal, M. & Barábasi, A.-L. Network-based in silico drug efficacy screening. *Nat. Commun.* **7**, 10331 (2016).
45. Gyöngyi, Z., Garcia-Molina, H. & Pedersen, J. Combating web spam with TrustRank. In *Proceedings 2004 VLDB Conference* (eds Nascimento, M. A. et al.) 576–587 (Morgan Kaufmann, 2004).
46. Alcaraz, N. et al. Robust de novo pathway enrichment with KeyPathwayMiner 5. *F1000Res* **5**, 1531 (2016).
47. Peixoto, T. P. The graph-tool python library. *Figshare* https://doi.org/10.6084/m9.figshare.1164194 (2014).

Acknowledgements

T.K. and S.S. are grateful for financial support from H2020 project RepoTrial (no. 777111). J.M., J.S., N.K.W., and R.N. received funding from H2020 project FeatureCloud (no. 826078). J.B., T.K., and M.L. are grateful for financial support from BMBF grant Sys_CARE (no. 01ZX1908A) of the Federal German Ministry of Research and Education. J.B.'s BMBF grant SyMBoD (no. 01ZX1910D) also financed parts of this project. M.O. is funded by the Collaborative Research Centre SFB924 of the German Research Foundation. J.B. was partially funded by his VILLUM Young Investigator Grant no. 13154. Contributions by J.K.P. and T.D.R. are funded by the Bavarian State Ministry of Science and the Arts in the framework of the Center Digitization.Bavaria (ZD.B, grant LipiTUM). M.S.-A. is grateful for a Ph.D. fellowship funding from CONACYT (CVU659273) and the German Academic Exchange Service, DAAD (ref. 91693321).

Author contributions

S.S., J.M., J.B., M.L., T.K., J.K.P., A.P., and A.S. conceived and designed the study. S.S. and J.M. were in charge of overall direction, planning, and supervision. S.S., G.G., T.D.R., M.S.-A., and N.K.W. performed the acquisition, integration, and interpretation of data. S.S., D.B.B., M.L., and K.Y. developed and adapted the algorithms for network-based drug repurposing. J.M., R.N., M.O., and J.S. implemented the web platform. All authors provided critical feedback and helped in the interpretation of data, manuscript writing, and the improvement of the platform.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41467-020-17189-2.

Correspondence and requests for materials should be addressed to J.B.

Peer review information *Nature Communications* thanks Alan Talevi, Marcel Müller and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permission 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.

B

sPLINK

Reza Nasirigerdeh et al. “sPLINK: a hybrid federated tool as a robust alternative to meta-analysis in genome-wide association studies”. In: *Genome Biology* 23.1 (2022), p. 32. ISSN: 1474-760X. DOI: 10.1186/s13059-021-02562-1. URL: https://doi.org/10.1186/s13059-021-02562-1

Genome Biology

METHOD Open Access

sPLINK: a hybrid federated tool as a robust alternative to meta-analysis in genome-wide association studies

Reza Nasirigerdeh[1,2*], Reihaneh Torkzadehmahani[1], Julian Matschinske[3], Tobias Frisch[4], Markus List[5], Julian Späth[3], Stefan Weiss[6], Uwe Völker[6], Esa Pitkänen[7,8], Dominik Heider[9], Nina Kerstin Wenke[3], Georgios Kaissis[1,2,12,13], Daniel Rueckert[1,2,12], Tim Kacprowski[10,11†] and Jan Baumbach[3,4†]

*Correspondence:
reza.nasirigerdeh@tum.de
†Tim Kacprowski and Jan Baumbach are joint senior authors.
[1] AI in Medicine and Healthcare, Technical University of Munich, Munich, Germany
[2] Klinikum rechts der Isar, Munich, Germany
Full list of author information is available at the end of the article

Abstract

Meta-analysis has been established as an effective approach to combining summary statistics of several genome-wide association studies (GWAS). However, the accuracy of meta-analysis can be attenuated in the presence of cross-study heterogeneity. We present *sPLINK*, a hybrid federated and user-friendly tool, which performs privacy-aware GWAS on distributed datasets while preserving the accuracy of the results. *sPLINK* is robust against heterogeneous distributions of data across cohorts while meta-analysis considerably loses accuracy in such scenarios. *sPLINK* achieves practical runtime and acceptable network usage for chi-square and linear/logistic regression tests. *sPLINK* is available at https://exbio.wzw.tum.de/splink.

Keywords: sPLINK, PLINK, Federated learning, Genome-wide association studies, GWAS, Meta-analysis, Privacy

Background

Genome-wide association studies (GWAS) test millions of single nucleotide polymorphisms (SNPs) to identify possible associations between a specific SNP and disease [1]. They have led to considerable achievements over the past decade including better comprehension of the genetic structure of complex diseases and the discovery of SNPs playing a role in many traits or disorders [2, 3]. GWAS sample size is an important factor in detecting associations, and larger sample sizes lead to identifying more associations and more accurate genetic predictors [2, 4].

PLINK [5] is a widely used open source software tool for GWAS. The major limitation of *PLINK* is that it can only perform association tests on local data. If multiple cohorts want to conduct collaborative GWAS to take advantage of larger sample sizes, they can pool their data for a joint analysis (Fig. 1a); however, this is close to impossible due to privacy

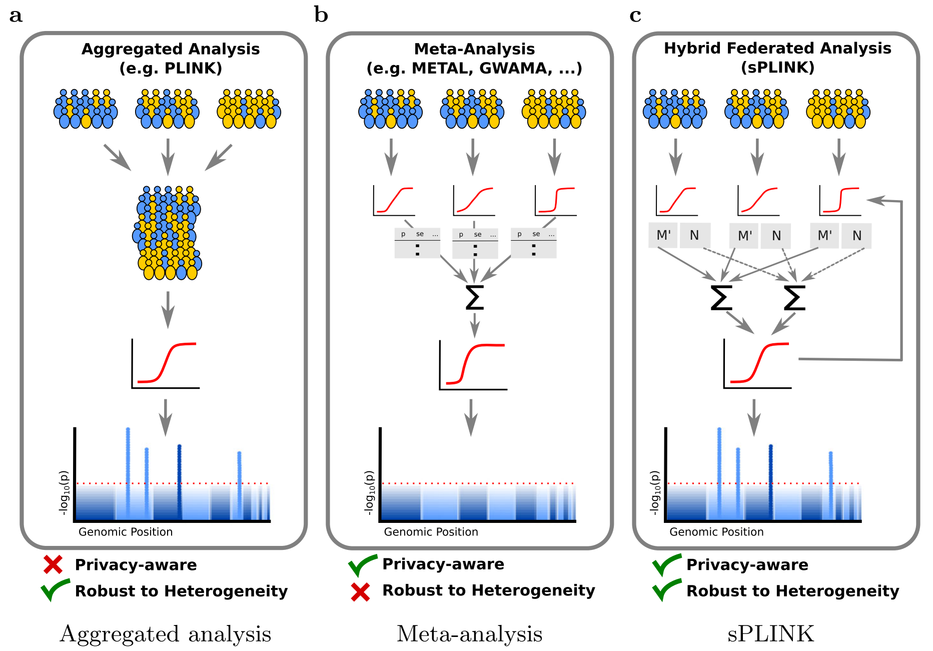


Fig. 1 Comparison of *sPLINK* (**c**), aggregated analysis (**a**), and meta-analysis (**b**) approaches: Aggregated analysis requires cohorts to pool their private data for a joint analysis. The meta-analysis approaches aggregate the summary statistics from the cohorts to estimate the combined *p*-values. In *sPLINK*, the cohorts calculate the model parameters (*M*) from the local data and global model, generate noise (*N*), and make the parameters noisy (*M'*) in an iterative manner. The aggregated noise and noisy parameters are in turn aggregated to update the global model or build the final model. *sPLINK* combines the advantages of the aggregated analysis and meta-analysis, i.e. robustness against heterogeneous data and enhancing the privacy of cohorts' data. Yellow/blue color indicates case/control samples

restrictions and data protection issues, especially concerning genetic and medical data. Hence, the field has established methods for meta-analysis of individual studies, where only the results and summary statistics of the individual analyses have to be exchanged [6] (Fig. 1b).

There are several software packages such as *METAL* [7], *GWAMA* [8], and *PLINK* [5] that implement different meta-analysis models including fixed or random effect models [9]. Although meta-analysis approaches are privacy-aware, i.e. the raw data is not shared with third parities, they suffer from two main constraints: first, they rely on detailed planning and agreement of cohorts on various study parameters such as meta-analysis model (e.g. fixed effect or random effect), meta-analysis tool (e.g., METAL or GWAMA), heterogeneity metric (e.g. Cochran's Q or the I^2 statistic), the covariates to be considered, etc [4]. Second and more importantly, the statistical power of meta-analysis can be adversely affected in the presence of cross-study heterogeneity, leading to inaccurate estimation of the joint results and yielding misleading conclusions [10, 11].

To address the aforementioned shortcomings, privacy-aware collaborative GWAS can be developed using homomorphic encryption (HE) [12], secure multi-party computation (SMPC) [13], and federated learning [14, 15]. In HE, the cohorts encrypt their private data and share it with a single server, which performs operations on the encrypted data from the cohorts to compute the association test results. In SMPC, there are several computing parties and the cohorts extract a separate secret share (anonymized chunk) [16] from the private data and send it to a computing party. The computing parties calculate intermediate results from the secret shares and exchange the intermediate results with each other. Each computing party computes the final results given all intermediate results. In federated learning, the cohorts extract model parameters (e.g. Hessian matrices) from the private data and share the parameters with a central server. The server aggregates the parameters from all cohorts to calculate the association test results.

Kamm et al. [17] and Cho et al. [18] proposed GWAS frameworks based on SMPC. The former developed simple association tests including Cochran–Armitage and chi-square (χ^2) and the latter implemented only the Cochran–Armitage test for trend. Shi et al. [19] presented an SMPC-based logistic regression framework for GWAS. Constable et al. [20] implemented an SMPC-based framework for minor allele frequency and chi-square computation. These frameworks inherit the limitations of SMPC itself: They follow the paradigm of "move data to computation," where they put the processing burden on a few computing parties. Consequently, they are computationally expensive [21] and are not scalable for large-scale GWAS. Moreover, they suffer from the colluding-parties problem [17] in which, if the parties send the secret shares of the cohorts to each other, the whole private data of the cohorts is exposed.

Lu et al. [22], Morshed et al. [23], and Kim et al. [24] developed chi-square, linear regression, and logistic regression tests using HE for GWAS, respectively. Sadat et al. [25] introduced the *SAFETY* framework based on HE and Intel Software Guard Extensions technology, which implements the linkage disequilibrium, Fisher's exact test, Cochran-Armitage test for trend, and Hardy-Weinberg equilibrium statistical tests. Similar to SMPC-based methods, they are not computationally efficient because a single server carries out operations over encrypted data, causing considerable overhead [26]. Additionally, HE-based methods introduce accuracy loss in the association test results [23, 24]. This is because HE only supports addition and multiplication, and as a result, non-linear operations in regression tests should be approximated using those two operations.

To address the computational limitation of HE/SMPC-based methods, the association tests can be implemented in a federated fashion. Federated learning-based methods follow the paradigm of "move computation to data," distributing the heavy computations among the cohorts while performing lightweight aggregation (simple operations such as addition and multiplication of the parameters) at the central server. Wang et al. [27] introduced EXPLORER for distributed logistic regression algorithm. EXPLORER is a model but not a tool for GWAS. Moreover, it does not provide a "guarantee for optimal global solution," implying that its results can be different from the aggregated analysis in general. GLORE [28, 29] implemented a federated logistic regression test but the parameter values computed by each cohort are revealed to the server.

Several hybrid federated frameworks including *HyFed* [30] have been introduced to improve the privacy of federated learning by hiding the local parameters of a cohort from third parties. HyFed is a suitable framework for developing federated GWAS algorithms because it provides enhanced privacy while preserving the accuracy of the results. It also supports federated mode, where different components can run in separate physical machines and securely communicate with each other over the Internet.

In this paper, we present a hybrid federated tool called *sPLINK (safe PLINK)* based on the *HyFed* framework for privacy-aware GWAS. *sPLINK* consists of four main components (Fig. 2): *Web application (WebApp)* to configure the parameters (e.g. association

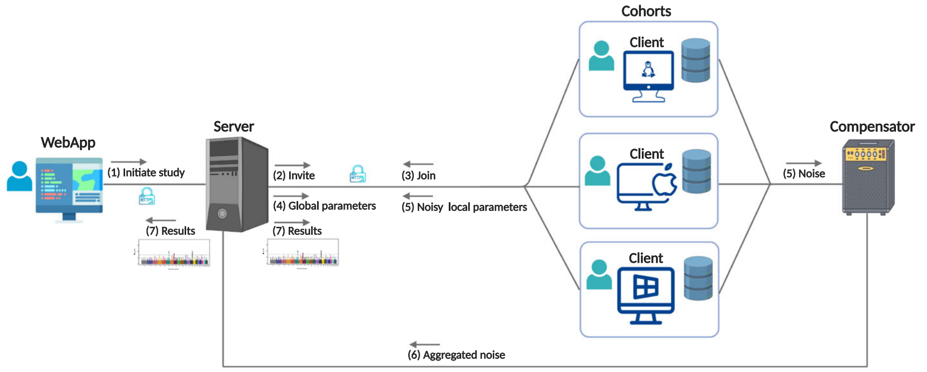


Fig. 2 *Architecture of sPLINK* : (1) The coordinator creates a new project through the WebApp component and (2) invites a set of cohorts to join the project; (3) the cohorts join the project and select the dataset using the client component. The project is started automatically, when all cohorts joined. The computation of the test results is performed in a an iterative manner, where the clients (4) obtain the global parameters from the server, (5) compute the local parameters, mask them with noise, and share the noise and noisy local parameters with the compensator and server, respectively; (6) the compensator aggregates the noise values and sends the aggregated noise to the server; the server calculates the global parameters by aggregating the noisy local parameters and the negative of the aggregated noise; (7) after the computation is done, the cohorts and coordinator can access the results. All communications are performed in a secure channel over HTTPS protocol. The cohorts can use Linux distributions, Microsoft Windows, or MacOS to run the client component

test) of the new study; *client* to compute the local parameters, mask them with noise, and share the noise with *compensator* and noisy local parameters with *server*; *compensator* to aggregate the noise values of the clients and send the aggregated noise to the *server*; *server* to compute the global parameters by adding up the noisy local parameters and the negative of the aggregated noise. Notice that the utility of the global model is preserved because the aggregated noise from the compensator cancels out the accumulated noise from the noisy local parameters during the aggregation.

Unlike *PLINK*, *sPLINK* is applicable to distributed data in a privacy-aware fashion. In *sPLINK*, neither the private data of cohorts leaves the site nor the original values of the local parameters are revealed to the other parties (Fig. 1c). Contrary to the existing HE/SMPC-based methods, *sPLINK* is computationally efficient because heavy computations are distributed across the cohorts while simple aggregation is performed on the server and compensator. Compared to the current federated tools like GLORE, *sPLINK* not only provides enhanced privacy but also supports multiple association tests including logistic and linear regression [31], and chi-square [32] for GWAS.

The advantage of *sPLINK* over the meta-analysis approaches is twofold: usability and robustness against heterogeneity. *sPLINK* is easier to use for collaborative GWAS compared to meta-analysis. In *sPLINK*, a coordinator initiates a collaborative study and invites the cohorts. The only decision the cohorts make is whether or not to join the study. After accepting the invitation, the cohorts just select the dataset they want to employ in the study. More importantly, *sPLINK* is robust to data heterogeneity (phenotype and confounding factors). It gives the same results as aggregated analysis even if the phenotype distribution is imbalanced or if confounding factors are distributed heterogeneously across cohorts. In contrast, meta-analysis tools typically lose statistical power in such imbalanced or heterogeneous scenarios (details in the "Results" section).

Results

We first verify *sPLINK* by comparing its results with those from aggregated analysis conducted with *PLINK* for all three association tests on a real GWAS dataset from the SHIP study [33]. We refer to this dataset as the *SHIP* dataset, which comprises the records of 3699 individuals with *serum lipase activity* as phenotype. The quantitative version represents the square root transformed serum lipase activity, while the dichotomous (binary) version indicates if the serum lipase activity of an individual is above or below the 75th percentile. The *SHIP* dataset contains around 5 million SNPs as well as sex, age, smoking status (current-, ex-, or non-smoker), and daily alcohol consumption (in g/day) as confounding factors (Table 1).

We employ the binary phenotype for logistic regression and the chi-square test, and the quantitative phenotype for linear regression. We incorporate all four confounding factors in the regression models and no confounding factor in the chi-square test. We horizontally (sample-wise) split the dataset into four parts, simulating four different cohorts (Additional file 1: Table S1). *PLINK* computes the statistics for each association test using the whole dataset while *sPLINK* does it in a federated manner using the splits of the individual cohorts. To be consistent with *PLINK*, *sPLINK* calculates the same statistics as *PLINK* for the association tests.

We compute the difference between the p-values as well as the Pearson correlation coefficient (ρ) of p-values from *sPLINK* and *PLINK*. We use $-log_{10}(p\text{-value})$ because the p-values are typically small and $-log_{10}(p\text{-value})$ can be a better indicator of small p-value differences. According to Fig. 3a–c, the p-value difference is zero for most of the SNPs. We also observe that the maximum difference is 0.162 for a SNP in the linear regression. *sPLINK* and *PLINK* report 4.441×10^{-16} and 3.058×10^{-16} as p-values for the SNP, respectively. This negligible difference can be attributed to inconsistencies in floating point precision.

The correlation coefficient of p-values from *sPLINK* and *PLINK* for all three tests is $0.\overline{99}$, which is consistent with the results of p-value difference from Fig. 3a–c. We investigate the overlap of significantly associated SNPs between *sPLINK* and *PLINK*. We

Table 1 Description of datasets

Dataset	# Samples	# SNPs	Adjustments	Phenotype
SHIP[a]	3699	∼5M	Sex, age, smoking status, daily alcohol consumption	SLA[b], dichotomous (75th percentile, 934 cases, 2765 controls)
				SLA, quantitative, Mean±SD[c] 1.23±0.3
COPDGene[d]	5343	∼600K	Sex, age, smoking status, pack years of smoking	COPD[e], dichotomous, (2811 cases, 2532 controls)
				FEV1[f], quantitative, Mean±SD 2.993±0.635
FinnGen	135,615	∼ 1M	Sex and age	Hypertension, dichotomous, (34,257 cases, 101,358 controls)

[a] Study of Health in Pomerania
[b] Serum lipase activity
[c] Standard deviation
[d] Genetic Epidemiology of chronic obstructive pulmonary disease
[e] Chronic obstructive pulmonary disease
[f] Forced expiratory volume in one second

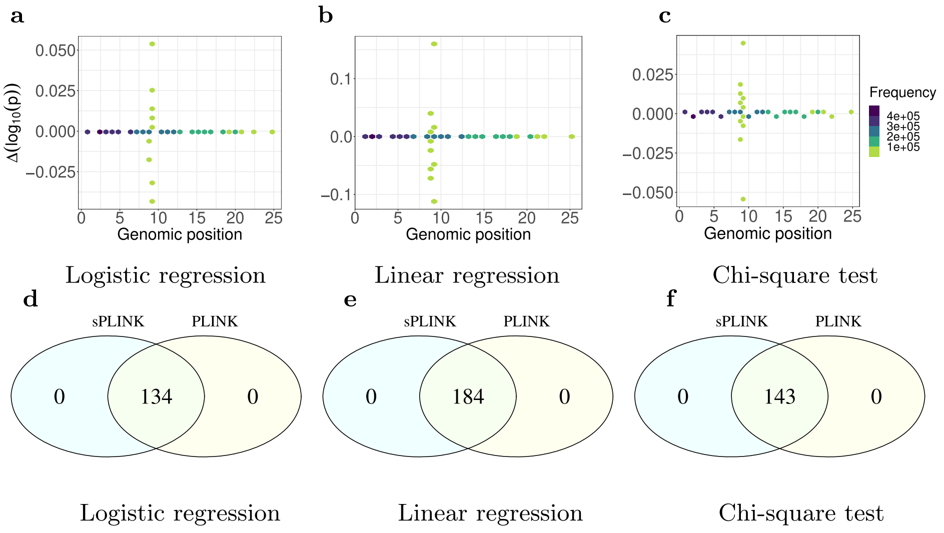


Fig. 3 Δlog_{10}(*p*-value) between *sPLINK* and *PLINK* as well as the set of SNPs identified by *sPLINK* and *PLINK* as significant for logistic regression (**a**, **d**), linear regression (**b**, **e**), and chi-square test (**c**, **f**), respectively. For most of the SNPs, the difference is zero, indicating that *sPLINK* gives the same *p*-values as *PLINK*. The negligible difference between *p*-values for the other SNPs can be attributed to differences in floating point precision. The spikes in some genomic positions are due to the strong association of the corresponding SNPs, which result in higher absolute error. *sPLINK* and *PLINK* also recognize the same set of SNPs as significant. Genomic positions (ticks in **a**–**c**) indicate chromosome numbers. The details of the experiments are available in Additional file 1: Table S1

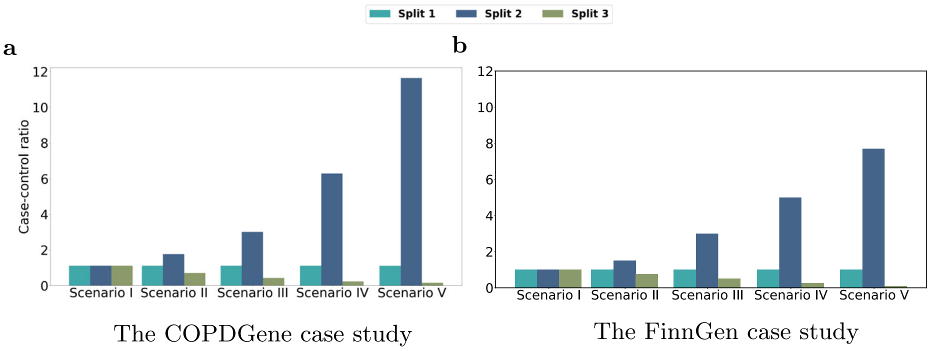
consider a SNP as significant if its *p*-value is less than 5×10^{-8} (genome-wide significance). *PLINK* and *sPLINK* recognize the same set of SNPs as significant (Fig. 3d–f). Notably, the identified SNPs, e.g. rs8176693 and rs632111, lying in genes ABO (intronic) and FUT2 (3-UTR), respectively, have also been implicated in a previous analysis of this dataset [34]. We also leverage the Bonferroni significance threshold (which is $\approx 1 \times 10^{-8}$ for our tests) to compare the overlapping significant SNPs from *sPLINK* and *PLINK*. The results remain similar and the associated plot is available at Additional file 1: Fig. S1. These results indicate that *p*-values computed by *sPLINK* in a federated manner are the same as those calculated by *PLINK* on the aggregated data (ignoring negligible floating point precision error). In other words, the federated computation in *sPLINK* preserves the accuracy of the results of the association tests.

Next, we compare *sPLINK* with some existing meta-analysis tools, namely *PLINK*, *METAL*, and *GWAMA*. We leverage the *COPDGene* (non-hispanic white ethnic group) [35] and *FinnGen* (data release 3) [36] datasets. The *COPDGene* dataset has an equal distribution of case and control samples unlike the *SHIP* dataset. It contains 5343 samples (ignoring 1327 samples with missing phenotype value) and around 600K SNPs. We utilize chronic obstructive pulmonary disease (COPD) as the binary phenotype and include sex, age, smoking status, and pack years of smoking as confounding factors [37]. *FinnGen* is much larger dataset (in terms of sample size) compared to the *SHIP* and *COPDGene* datasets. It consists of 135,615 samples (ignoring 23 samples with missing phenotype value) and about 1 million SNPs. We use *Hypertension* as the (binary) phenotype and adjust for sex and age as confounding factors (Table 1).

To simulate cross-study heterogeneity [38] on the *COPDGene* dataset, we consider six different scenarios: *Scenario I* (*Balanced*), *Scenario II* (*Slightly Imbalanced*), *Scenario III*

(*Moderately Imbalanced*), *Scenario IV* (*Highly Imbalanced*), *Scenario V* (*Severely Imbalanced*), and *Scenario VI* (*Heterogeneous Confounding Factor*) (Figs. 4a and 5). In each scenario, we partition the dataset into three splits with the same sample size (more details in Additional file 1: Table S2). The distribution of all four confounding factors is homogeneous (similar) across the splits for the first five scenarios. The splits have the same (and balanced) case-control ratio in *Scenario I* and *Scenario VI* but their case-control ratio is different for the imbalanced scenarios (Fig. 4a). In *Scenario VI*, the values of two confounding factors (i.e. smoking status and age) are homogeneously distributed among the splits; however, the distribution of sex and pack years of smoking is slightly and highly heterogeneous across the splits, respectively (Fig. 5). We obtain the summary statistics (e.g. minor allele, odds ratio, and standard error) for each split to conduct meta-analyses. The results are then compared to the federated analysis employing *sPLINK*. Figure 6a shows the Pearson correlation coefficient of $-log_{10}$(p-value) between each tool and the aggregated analysis for all six scenarios. Figure 6c depicts the number of SNPs correctly identified as significant by the tools (true positives).

According to Fig. 6a, the correlation of p-values between *sPLINK* and the aggregated analysis is ∼ 1.0 for all six scenarios, implying that *sPLINK* gives the same p-values as the aggregated analysis regardless of how phenotypes or confounding factors have been distributed across the cohorts. In contrast, the correlation coefficient for the meta-analysis tools shrinks with increasing imbalance/heterogeneity, indicating loss of accuracy. Figure 6c illustrates that *sPLINK* correctly identifies all four significant SNPs in all scenarios. In the balanced scenario, almost all meta-analysis tools perform well and recognize all significant SNPs. An exception is *METAL*, which misses one of them. However, they miss more and more significant SNPs as the phenotype imbalance across the splits increases. In the *Highly Imbalanced* and *Severely Imbalanced* scenarios, the meta-analysis tools cannot recognize any significant SNP. This is also the case if the distribution of some confounding factors becomes heterogeneous across the cohorts (*Scenario VI*). We checked the number of SNPs wrongly identified as significant by the tools (false positives) too. *sPLINK* has no false positive in any of the scenarios and the meta-analysis tools introduce zero or one false positive depending on the scenario.



The COPDGene case study

The FinnGen case study

Fig. 4 *Scenario I-V*: The case-control ratio is the same for all splits in the balanced scenario (I) while the splits have different case-control ratios in the imbalanced scenarios (II–V). All three splits have the same sample size in the *COPDGene* dataset as well as the balanced scenario in the *FinnGen* dataset. For the imbalanced scenarios in the *FinnGen* dataset, the splits have different sample sizes

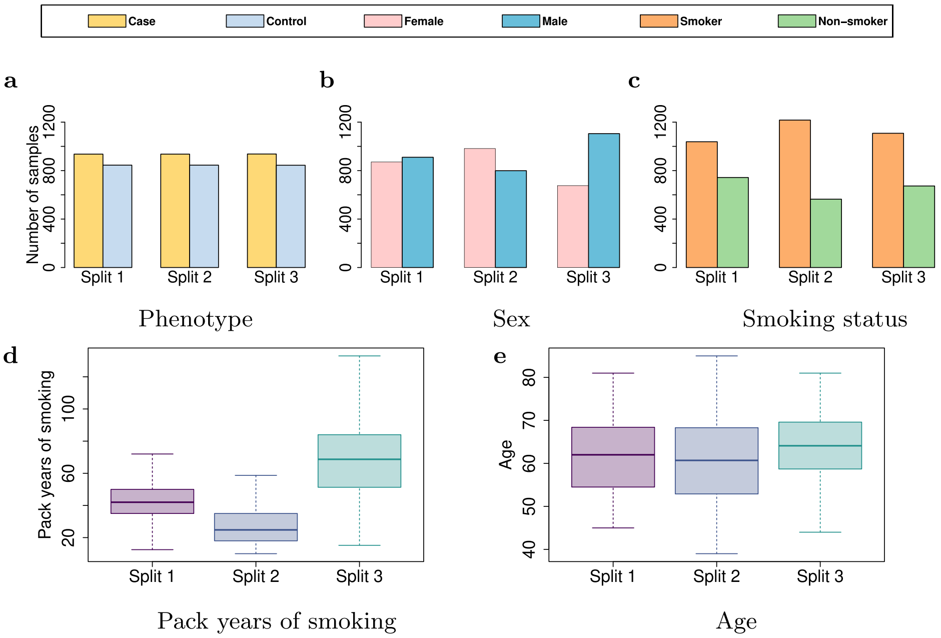


Fig. 5 *Scenario VI (Heterogeneous Confounding Factor)* for the *COPDGene* case study: The phenotype distribution is the same and balanced; the values of smoking status and age are homogeneously distributed; the distribution of sex and pack years of smoking are slightly and highly heterogeneous across the splits, respectively

To show that our findings on the *COPDGene* dataset also hold true for a much larger dataset, we repeat the simulations on the *FinnGen* dataset (more details in Additional file 1: Table S3). Similar to the *COPDGene* case study, we divide the dataset into three splits and define *Scenario I* to *Scenario V*, where the splits have the same case-control ratio (1.0) and sample size (22,838) as in *Scenario I* but different case-control ratios in the remaining scenarios (Fig. 4b); Unlike the *COPDGene* case study in which the sample size of the splits are equal for all scenarios including the imbalanced ones, the splits have different number of samples in the imbalanced scenarios of the *FinnGen* case study. For instance, split1, split2 and split3 have 22,838, 12,561, and 99,345 samples in *Scenario V*, respectively (a split with lower case-control ratio has larger sample size). It implies that the aggregated datasets have different number of samples in the scenarios, and as a result, there are different set of significant SNPs in each scenario of the *FinnGen* case study (total of 110, 116, 199, 304, and 446 significant SNPs in *Scenario I* to *Scenario V*, respectively).

Figures 6b and 6d illustrate the Pearson correlation coefficient and percentage of correctly identified significant SNPs for each scenario on the *FinnGen* case study, respectively. According to Fig. 6b, the correlation coefficient diminishes for the meta-analysis tools as the scenario becomes more and more imbalanced. This is also the case for the percentage of the SNPs correctly identified as significant by each meta-analysis tool (Fig. 6d). These results are consistent with those from the *COPDGene* case study. Moreover, we observed that the meta-analysis tools report high number of false positives (14–88) in *Scenario IV*. Thus, the limitations of meta-analysis tools towards class imbalance observed in the *COPDGene* dataset can be reproduced on a large dataset. However, sPLINK always provides the same results as PLINK with the aggregated analysis (the "Methods" section, Figs. 3 and 6a, c).

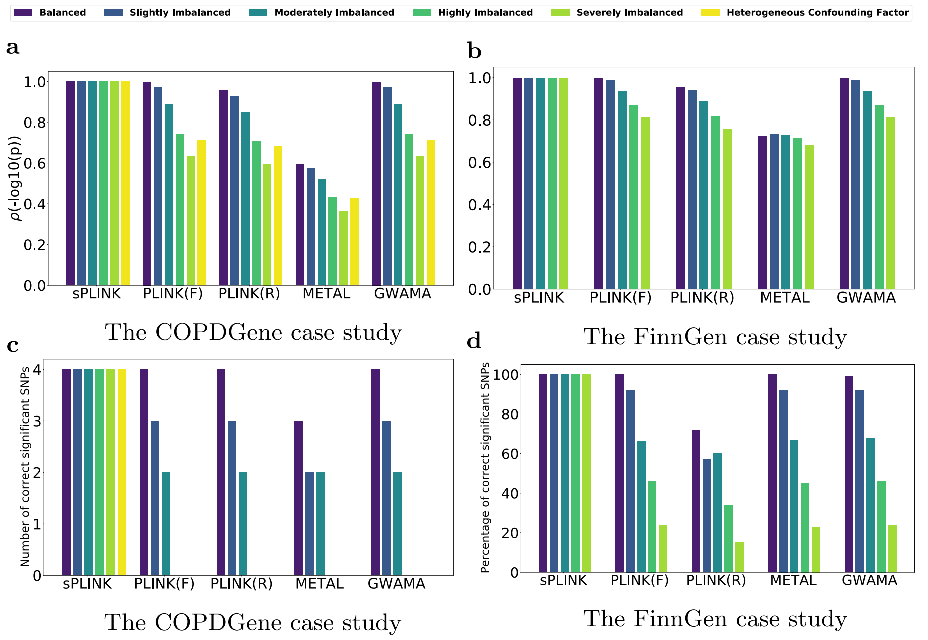


Fig. 6 The Pearson correlation coefficient (ρ) of $-log_{10}$(p-value) between each tool and aggregated analysis (**a**, **b**) and the number (**c**) and the percentage (**d**) of SNPs correctly identified as significant (true positives) by each tool. *F* and *R* stand for fixed-effect and random-effect, respectively. The details of the experiments are available in Additional file 1: Table S2, and Table S3

We also leverage the Spearman correlation to check whether or not the meta-analysis tools maintain the ordering of significance compared to the aggregated analysis. Our results show that this is not the case, and the Spearman correlation values for the meta-analysis tools reduce as the phenotype imbalance across the splits increases, similar to the results from Fig. 6, where the Pearson correlation is used. The corresponding plot can be found in Additional file 1: Figure S2.

Table 2 shows a concise comparison between *sPLINK* and the state-of-the-art approaches. Unlike *PLINK*, *sPLINK* is privacy-aware, where the private data never leaves the cohorts. *sPLINK* is also robust against the imbalance/heterogeneity of phenotype/confounding factor distributions across the cohorts. *sPLINK* always delivers the same p-values as aggregated analysis and correctly identifies all significant SNPs independent of the phenotype or confounding factor distribution in the cohorts. In contrast, meta-analysis tools lose their statistical power in imbalanced phenotype scenarios, missing some or all significant SNPs. This is also the case if the phenotype distribution is balanced but the values of confounding factor(s) have heterogeneously been distributed across the datasets. Compared to the existing SMPC/HE-based approaches, *sPLINK* is computationally efficient and supports multiple association tests including chi-square and linear/logistic regression. *sPLINK* provides enhanced privacy by hiding the model parameters of each cohort from the third parties while federated learning-based frameworks such as GLORE reveal them to the server.

Finally, we measure the runtime and network bandwidth usage of *sPLINK* for each association test using the COPDGene dataset partitioned into three splits of the same sample

Table 2 Comparison between *sPLINK* and the state-of-the-art approaches

Tool/Study	Privacy-aware	Robust to heterogeneity	Computationally efficient	Linear regression	Logistic regression
PLINK	✗	✓	✓	✓	✓
Meta-analysis	✓	✗	✓	✓	✓
Kamm et al. [17]	✓	✓	✗	*	✗
Cho et al. [18]	✓	✓	✗	*	✗
Morshed et al. [23]	✓	✗	✗	✓	✗
Kim et al. [24]	✓	✗	✗	✗	✓
GLORE [28]	✓	✓	✓	✗	✓
sPLINK	✓	✓	✓	✓	✓

*The study supports the Cochran–Armitage test, which is computationally comparable to linear regression

size. We use *COPD* in chi-square as well as logistic regression and *FEV1* in linear regression as phenotype. We include age, sex, smoking status, and pack years of smoking as confounding factors only for the regression tests. The server and WebApp packages are installed on a physical machine located at *Freising* (*Germany*) while the compensator is running on a machine at *Odense* (*Denmark*). Three commodity laptops located at *Munich* or *Freising* are running the client package and host the splits. They communicate with the server and compensator through the Internet. The system specification of the machines and laptops as well as the details of the experiments can be found in Additional file 1: Table S4 and S5.

Figure 7a plots the *sPLINK's* runtime for each association test. *sPLINK* computes the results for chi-square, linear regression, and logistic regression in 8 min, 20 min, and 75 min, respectively. Sending parameters from the clients to the server and compensator contributes the most in sPLINK's runtime. Compared to Kamm et al. [17], *sPLINK* is almost 13 times faster for chi-square test (8 min vs. 110 min[1]) with less powerful hardware, larger sample size (5343 vs. 1080), and more number of SNPs (∼ 580K vs. ∼ 263K).

Figure 7b depicts the network usage of *sPLINK*. The clients, server, and compensator exchange total of 0.967 GB, 2.49 GB, and 11.06 GB traffic in chi-square, linear regression, and logistic regression, respectively. Logistic regression has higher volume of traffic

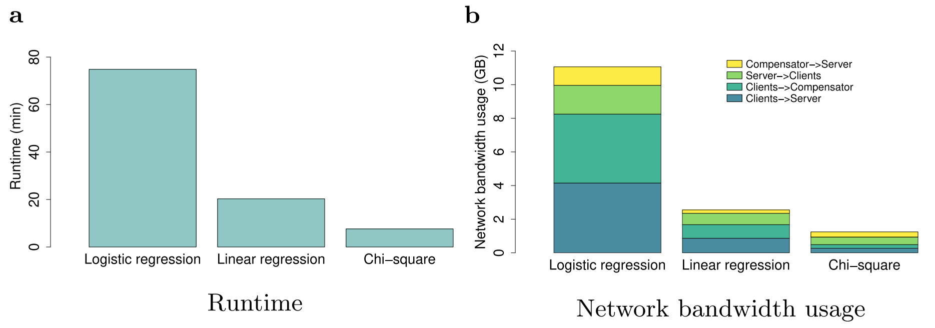


Fig. 7 Runtime and network bandwidth consumption of *sPLINK*. Logistic regression is the most time-consuming association test and exchanges the highest traffic over the network due to the iterative nature of the algorithm. The experimental setup can be found in Additional file 1: Table S5

[1] The best result from Kamm et al. [17] has been considered.

exchange because the computation of beta coefficients are performed in an iterative fashion. A fair comparison between *sPLINK* and SMPC-based frameworks from the network communication aspect is tricky. However, in general, (hybrid) federated learning-based approaches consume more network bandwidth than SMPC-based ones.

We also conduct a set of experiments to investigate how the runtime and network bandwidth consumption of *sPLINK* change with varying number of samples, SNPs, and clients. The results demonstrate that the traffic exchanged over the network is independent of the sample size and linearly increases with the number of SNPs and clients (as expected). Moreover, runtime is not affected much by the sample size thanks to the multi-threading capability of *sPLINK*'s client package, and linearly/non-linearly increases with the number of SNPs/clients. The corresponding plots are available in Additional file 1: Fig. S3, S4, and S5.

Discussion

We first provide a general discussion on the privacy of the existing tools for collaborative GWAS including *sPLINK*. To be more accurate, we draw a distinction between the privacy-aware and privacy-preserving definitions [39]. In a privacy-aware approach, it is not required to share the private data with a third party. A privacy-aware approach is privacy-preserving if the approach offers a privacy guarantee that captures the privacy risk associated with individual samples in the dataset. Given that, meta-analysis, SMPC, HE, federated learning, and hybrid federated learning based on SMPC are privacy-aware because they do not share the raw data with a third party. In meta-analysis/federated learning, the summary statistics/model parameters of each cohort are shared with a third party. In SMPC-based hybrid federated learning, the aggregated (global) parameters are revealed to the server and cohorts. These approaches, including HE and SMPC, reveal the final model too. However, these methods are not privacy-preserving because none of them provides a privacy guarantee indicating to what extent the revealed information leaks the private data of a particular sample in the dataset. To our knowledge, differential privacy (DP) [40] and DP-based hybrid federated learning can offer such a guarantee at the cost of the utility of the model and are considered as privacy-preserving approaches.

While privacy-aware approaches do not offer a privacy guarantee, they might provide stronger/weaker privacy compared to each other based on the amount and nature of the information they share with third parties. For instance, HE-based methods provide stronger privacy because they only reveal the final model (results) while other privacy-aware approaches disclose not only the final results but also other information such as summary statistics or local parameters. Similarly, *sPLINK* provides enhanced privacy in comparison with existing federated learning based tools such as GLORE. This is because GLORE discloses the local parameters of each cohort to the server, which is not revealed in *sPLINK*.

sPLINK is a privacy-aware tool, assuming honest-but-curious server, compensator, and clients, which (I) follow the protocol as it is; for instance, the server always sends the global beta values resulted from the aggregation but not the beta values tampered with such as all zeros to the clients, and (II) do not collude with each other, e.g. the compensator never shares the individual noise values of the clients with the server and similarly, the server does not send the noisy local parameters to the compensator, but (III) they try to reconstruct the raw data using the model parameters. Additionally, (IV) there are at least three different cohorts participating in the study, and their client components as well as the server and compensator components are running in separate physical machines.

Given these assumptions, we discuss the privacy of the masking mechanism of *sPLINK* (inherited from *HyFed*) for the supported association tests. To this end, we use the information theoretic criterion called *mutual information* between two random variables X and Y [30, 41]:

$$I(X, Y) = H(X) - H(X|Y)$$

where $H(X)$ and $H(X|Y)$ indicate the entropy of X and the conditional entropy of X given Y, respectively. The mutual information measures (in bits) the decrease in uncertainty about X having the knowledge of Y. In *sPLINK*, the noisy local parameter M'_L is a secret share from the local parameter M_L (the secret), and random variables X and Y indicate the distributions of M_L and M'_L, respectively.

The local parameter M_L of a client is either a non-negative integer (e.g. sample count, allele count, or contingency table) or floating-point number (e.g. Hessian or covariance matrix) in the association tests. For non-negative integers, *sPLINK* capitalizes on *additive secret sharing* based on *modular arithmetic* over the finite field $\mathbb{Z}_p$={0, 1, $p - 1$}, in which p is a *prime* number [13]. For floating-point numbers, *sPLINK* employs *real value secret sharing* based on Gaussian (Normal) distribution [42, 43] (more details in "Methods" section).

For non-negative integers, noise N_L is generated from a uniform distribution over $\mathbb{Z}_p$, and M'_L is the modular addition of M_L and N_L: $M'_L = (M_L + N_L) \mod p$. For this scheme, it has been shown that the knowledge of Y (noisy local parameter) provides no information about X (local parameter), which means the mutual information between them is zero: $I(X, Y) = 0$ [13, 16]. Notice that this is the case for any value of prime number p.

For floating-point numbers, noise N_L is generated using Gaussian distribution with variance of σ_N^2. Assuming that the variance of X is $\sigma_{M_L}^2$, the mutual information between X and Y is maximum if Y follows the Gaussian distribution (variance $\sigma_{M_L}^2 + \sigma_N^2$) [43]. Thus, the upper bound on the mutual information between X and Y is:

$$I(X, Y) = \frac{1}{2} \log_2(1 + \frac{\sigma_{M_L}^2}{\sigma_N^2})$$

That is, the amount of reduction in uncertainty about the local parameters having the knowledge of the noisy local parameters depends on the relative variance of the corresponding distributions. Therefore, using larger values for variance in the Gaussian random generator will provide lower information leakage. The value of mean for the Gaussian random generator does not remarkably impact the privacy and can be set to zero [43], which is the case for *sPLINK*. The default value of σ_N^2 is 10^{12} for *sPLINK*, which is large enough for typical GWAS, but it can be set to higher values if needed to ensure that $\frac{\sigma_{M_L}^2}{\sigma_N^2}$ remains small.

Notice that although *sPLINK* significantly enhances the privacy of data compared to existing federated learning tools by hiding the local parameters of clients from a third party, it does not eliminate the possibility of data reconstruction using the aggregated parameters or final results. For example, the $X^T X$ parameter (covariance matrix) in the linear regression algorithm can be exploited to determine the sex of the patients if the total number of samples across all cohorts is comparable to the number of the confounding factors. However, for a reliable GWAS study, the total sample size is considerably larger than the number of confounding factors, and therefore, the reconstruction of the cohorts' private data from the aggregated parameters can be difficult (but still possible) in practice. A similar argument is also applicable to meta-analysis approaches, which reveal the summary statistics of each cohort to a third party.

The value of prime number p impacts the correctness of the masking mechanism. To ensure the correctness, overflow must not occur in $\sum_{i=1}^{i=K} N_{L_i}$ and $\sum_{i=1}^{i=K} M'_{L_i}$ calculations, and $\sum_{i=1}^{i=K} M_{L_i} < p$. *sPLINK* uses the default value of $p = 2^{54} - 33$, which is the largest prime number than can fit in 54-bit integer. A higher value of p can be employed to handle larger integer values but at the expense of a lower number of clients [30]. Likewise, too large values of variance σ_N^2 (e.g. 10^{30}) can impact the precision of the results. With default values of p and σ_N^2, however, our experiments indicate that there are no statistically significant differences between the results from *sPLINK* with and without the masking mechanism for all three association tests (the experimental setup of Fig. 7 is used in the experiments).

sPLINK currently supports chi-square and linear/logistic regression tests, but it can be extended to compute other useful statistics in GWAS such as minor allele frequency (MAF), Hardy-Weinberg equilibrium (HWE), and linkage disequilibrium (LD) between SNPs in a privacy-aware manner. The federated computation of the aforementioned statistics in *sPLINK* is expected to be straightforward because they are based on the allele frequencies, and *sPLINK* already calculates the minor and major allele counts in the *Non-missing count* step of its computational workflow (the "Methods" section). Moreover, population stratification using the principal component analysis (PCA) will be addressed in the future version of *sPLINK* due to the complexity of the problem. *sPLINK*'s implementation of the association tests is horizontally-federated, where the datasets have different samples but the same features (i.e. SNP and confounding factors). However, correcting for population structure using *sPLINK* requires a vertically-federated [44] PCA algorithm because the eigenvectors should be computed from the sample by sample covariance matrix, and therefore, the samples and features swap roles in the federated PCA (SNPs are considered as samples and patients as features) [45]. Vertical federated learning algorithms are still understudied, and they are considered more complicated than the horizontal algorithms.

Additionally, the federated PCA algorithm should be an iterative, randomized algorithm [46] so that it can handle large GWAS datasets with a practical amount of main memory. The iterative nature of the algorithm will present network and runtime challenges because it might need dozens or hundreds of iterations and exchange huge traffic over the network to converge to the final eigenvectors. From the privacy perspective, a recent study [45] demonstrates that even if we assume the federated PCA and linear regression algorithms individually provide perfect privacy, federated population stratification in GWAS, where the eigenvectors are used as the confounding factors in the association test, does not necessarily offer perfect privacy. Consequently, the server can reconstruct the SNP or binary confounding factor values in polynomial time. To tackle this issue, they suggested that the final eigenvectors should be computed at the clients and the model parameter values should be hidden from the server. The federated population stratification in *sPLINK* should be implemented taking into account those suggestions.

We showed that *sPLINK* is robust against an important source of data heterogeneity, namely the heterogeneous distribution of the phenotype or confounding factor values across the distributed datasets of the cohorts. Population heterogeneity across the cohorts is another source of data heterogeneity in GWAS, which is commonly tackled by population stratification using the PCA algorithm. *sPLINK* currently does not address this kind of data heterogeneity but the future versions of the tool will support population stratification to this end.

Conclusions

We introduce *sPLINK*, a user-friendly, hybrid federated tool for GWAS. *sPLINK* enhances the privacy of the cohorts' data without sacrificing the accuracy of the test results. It supports multiple association tests including chi-square, linear regression, and logistic regression. *sPLINK* is consistent with *PLINK* in terms of the input data formats and results. We compare *sPLINK* to aggregated analysis with *PLINK* as well as meta-analysis with *METAL*, *GWAMA*, and *PLINK*. While *sPLINK* is robust against the heterogeneity of phenotype or confounding factor distributions across separate datasets, the statistical power of the meta-analysis tools is declined in imbalanced/heterogeneous scenarios. We argue that *sPLINK* is easier to use for collaborative GWAS compared to meta-analysis approaches thanks to its straightforward functional workflow. We also show that *sPLINK* achieves practical runtime, in order of minutes or hours, and acceptable network usage. *sPLINK* is an open-source tool and its source code is publicly available under the Apache License Version 2.0. *sPLINK* is a novel and robust alternative to meta-analysis, which performs collaborative GWAS in a privacy-aware manner. It has the potential to immensely impact the statistical genetics community by addressing current challenges in GWAS including cross-study heterogeneity and, thus, to replace meta-analysis as the gold standard for collaborative GWAS.

Methods

Federated learning [14, 15] is a type of distributed learning, where multiple cohorts collaboratively learn a joint (global) model under the orchestration of a central server [47]. The cohorts never share their private data with the server or the other cohorts. Instead, they extract local parameters from their data and send them to the server. The server aggregates the local parameters from all cohorts to compute the global model parameters (or global results), which in turn, are shared with all cohorts. While federated learning is privacy-aware, where the private data of the cohorts is not shared with the server, studies [48, 49] have shown that for some models such as deep neural networks, the raw data can be reconstructed from the parameters shared by the cohorts.

To improve the privacy of federated learning, privacy-enhancing technologies (PETs) such as DP, HE, or SMPC can be combined with federated learning to avoid revealing the original values of the local parameters to third parties including the server [50]. DP-based hybrid federated learning approaches can provide a privacy guarantee but their final results might be considerably impacted by the random noise employed for the perturbation of the model. HE-based aggregation methods can incur remarkable computational overhead because they require the cohorts to encrypt/decrypt the local/global model parameters and the server to perform the aggregation over the encrypted parameters. SMPC-based hybrid federated learning methods [30, 51] increase the network bandwidth usage but does not adversely affect the final results. *HyFed* is an open-source hybrid federated framework, which combines federated learning with additive secret sharing-based SMPC to enhance the privacy of the federated algorithms while preserving the utility (performance) of the global model. HyFed provides a generic API (application programming interface) to develop federated machine learning algorithms. It supports the federated mode of operation, where different components of the framework can be installed in separate physical machines and securely communicate with each other through the Internet.

sPLINK implements a hybrid federated approach using the *HyFed* API to enhance the privacy of data. *sPLINK* works with distributed GWAS data, where samples are individuals and features are SNPs and categorical or quantitative phenotypic variables. While the samples are different across the cohorts, the feature space is the same because *sPLINK* only considers SNPs and phenotypic variables that are common among all datasets (horizontal or sample-based federated learning)[44]. The client package of *sPLINK* is installed on the local machine of each cohort with access to the private data. The compensator is running in a separate machine. sPLINK's server and WebApp packages are installed on a central server.

In *sPLINK*, the original values of the parameters computed from the private data in one cohort is not revealed to the server, compensator, or other cohorts, improving the privacy of the cohorts' data. *sPLINK* provides the chunking capability to handle large datasets containing millions of SNPs. The chunk size (configured by the coordinator) specifies how many SNPs should be processed in parallel. Larger chunk sizes allow for more parallelism, and therefore less running time in general but require more computational resources (e.g. CPU and main memory) from the local machines of the cohorts, the server, and compensator. *sPLINK*'s client package is multi-threaded, where the number of cores is configurable by the participants. This makes the computation of the model parameters in the cohorts very fast, especially for large datasets. While we provide a readily usable web service running at *exbio server* (https://exbio.wzw.tum.de/splink) and online compensator at *compbio server* (https://compensator.compbio.sdu.dk), the server, WebApp, and compensator packages can, of course, be deployed on customized physical machines.

The *functional workflow* of *sPLINK* is comprised of the following steps:

1. **Project creation**: The coordinator creates the project (new study) through the Web interface. To this end, she/he first specifies the project name, association test name, chunk size, and the list of confounding features (only for regression tests), and then, generates a unique project token for each cohort.
2. **Cohort invitation**: The coordinator sends the project ID (automatically generated) and token to each participant (a human entity interacting with the client package in a cohort) through a secure channel such as email for inviting the cohorts to the project.
3. **Cohort joining**: The participants use their corresponding username, password, project ID, and token to join the project. After joining, they can view the general information of the project such as the coordinator, server/compensator name/URL, and etc. If they agree to proceed, they choose the dataset they want to employ in the study. To be consistent with *PLINK*, *sPLINK* supports *.bed* (value of SNPs), *.fam* (sample IDs as well as sex and phenotype values), *.bim* (chromosome

number, name, and base-pair distance of each SNP), *.cov* (value of confounding factors), and *.pheno* (phenotype values that should be used instead of those in *.fam* file) file formats as specified in the *PLINK* manual [52]. For linear regression, phenotype values must be quantitative while for logistic regression and chi-square, phenotype values have to be binary (control/case are encoded as 1/2).

4. **Federated computation**: In *sPLINK*, the association test results are computed by the client package (running on the local machines of cohorts), server package (running in the central server), and compensator (running in its own machine) in a federated manner. The computation is iterative and consists of six general steps:
   (a) **Get global parameters**: All clients obtain the required global parameters M_G from the server.
   (b) **Compute local parameters**: Each client i computes the local parameters M_{L_i} using the local data and global parameters.
   (c) **Mask local parameters**: Each client i generates random noise N_{L_i} with the same shape as M_{L_i}, and masks M_{L_i} with N_{L_i} to obtain the noisy local parameters M'_{L_i}.
   (d) **Share noisy local parameters and noise**: Each client i shares M'_{L_i} and N_{L_i} with the server and compensator, respectively.
   (e) **Aggregate noise**: The compensator computes the aggregated noise N given the noise values from the clients and sends the aggregated noise N to the server.
   (f) **Compute global parameters**: The server calculates (unmasks) the global parameters given the noisy local parameters and the negative of the aggregated noise.

5. **Result download**: The final results are automatically downloaded for the cohorts but the coordinator needs to download them manually through the web interface. Similar to *PLINK*, *sPLINK* reports minor allele name ($A1$) and p-value (P) for all three association tests, chi-square ($CHISQ$), odds ratio (OR), minor allele frequency in cases (F_A), and minor allele frequency in controls (F_U) for chi-square test, and the number of non-missing samples ($NMISS$), beta ($BETA$), and t-statistic ($STAT$) for linear and logistic regression tests.

sPLINK inherits its masking mechanism from *HyFed*, which masks the local parameters with non-negative integer and floating-point values in different ways. For a local parameter with a non-negative integer value, *sPLINK* considers a finite field $\mathbb{Z}_p$={0, 1, $p - 1$} (p is a *prime* number) [13], where each client i generates a uniform random integer from $\mathbb{Z}_p$ as noise N_{L_i} and masks its local parameter M_{L_i} with N_{L_i} by performing the *modular addition* over $\mathbb{Z}_p$: $M'_{L_i} = (M_{L_i} + N_{L_i}) \mod p$. Notice that $M_{L_i}, N_{L_i}, M'_{L_i} \in \mathbb{Z}_p$. For M_{L_i} with a floating-point value, each client i generates noise N_{L_i} using Gaussian random generator with zero-mean and variance σ_N^2, and masks M_{L_i} with N_{L_i} using the ordinary addition: $M'_{L_i} = M_{L_i} + N_{L_i}$.

The compensator computes the aggregated noise N by taking sum over the noise values of the clients using the modular or ordinary addition depending on the data type of the noise: if N_{L_i} is non-negative integer, then $N = (\sum_{i=1}^{i=K} N_{L_i}) \mod p$; if N_{L_i} is floating-point type, then $N = \sum_{i=1}^{i=K} N_{L_i}$. To calculate the global parameters with non-negative integer values, the server first computes the aggregated noisy parameter by taking sum over the noisy local parameters using the modular addition, and then subtracts the aggregated noise from the aggregated noisy parameter using the modular subtraction: $M_G = (((\sum_{i=1}^{i=K} M'_{L_i}) \mod p) - N) \mod p$. For model parameters with floating-point values, the server adds up the noisy local parameters and the negative of the aggregated noise using the ordinary addition: $M_G = \sum_{i=1}^{i=K} M'_{L_i} - N$.

The *computational workflow* of *sPLINK* involves seven steps common among all association tests as well as a couple of steps specific to each association test (Fig. 8). In the first three steps (i.e. *Init*, *SNP name*, and *Allele name*) as well as the sixth step (*Minor allele*), the clients only communicate with the server, where the name of the SNPs and alleles (which are not considered private) are directly shared with the server. In the remaining steps, the compensator is involved and clients mask the local parameters with noise to hide their original values from the server. The formulas associated with the steps indicate how the clients compute local parameters and how the server calculates the global parameters using the noisy local parameters of the clients and the aggregated noise from the compensator. In the following, we provide an overview of each step:

1. **Init**: Each client i opens the files of the dataset selected by the participant to be employed in the study and creates its phenotype vector (Y_i) and feature matrix (X_i), which includes the value of SNPs and confounding factors. It is worth noting that there is a separate feature matrix for each SNP but the phenotype vector is the same for all SNPs. Assume a dataset containing three SNPs named *SNP1*, *SNP2*, and *SNP3* and *age* and *sex* as confounding features. There will be three different feature matrices, one feature matrix per SNP. For instance, the feature matrix of *SNP1* has three columns including *SNP1*, *age*, and *sex* values. Phenotype vector and feature matrix are the private data of the cohorts. They cannot be shared with the server, compensator, or the other cohorts. The aggregation process in the server just makes sure that all clients successfully initialized their data.
2. **SNP name**: Each client shares the SNP names with the server. In the aggregation process, the server computes the intersection of all SNP names. Only common SNPs are considered in the computation of the association test results.
3. **Allele name**: Each client sends the allele names (e.g. G,A) of each SNP to the server. In the aggregation process, the server ensures that all cohorts employ the same allele names for the SNPs. Notice that the clients sort the allele names to avoid revealing which one is minor or major allele.
4. **Sample count**: Each client i calculates its local sample count T_i (number of samples in its dataset including missing samples, which is the size of vector Y_i). The server computes the corresponding global sample count: $T = (((\sum_{i=1}^{i=K} T'_i) \mod p) - N_T) \mod p$, where T'_i is the noisy local sample count of client i: $T'_i = (T_i + N_i) \mod p$ and N_T is the aggregated noise from the compensator: $N_T = (\sum_{i=1}^{i=K} N_i) \mod p$.
5. **Non-missing count**: In this step, SNPs are split into chunks which can be processed in parallel. The chunking capability is provided to handle very large datasets containing millions of SNPs. The clients compute the non-missing sample count by filtering out the missing samples (value of -9 is considered as missing). Likewise, they calculate the local allele count by counting the number of alleles in

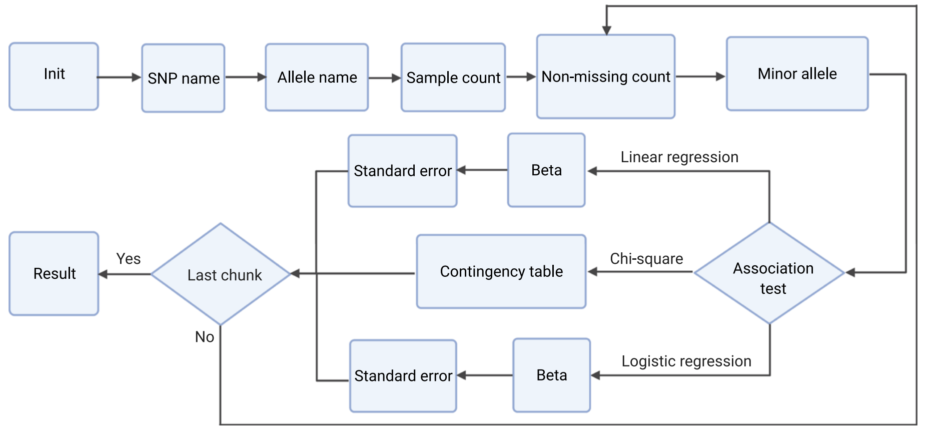


Fig. 8 Computational workflow of *sPLINK*: The first six steps and the last step are common among all association tests. Contingency table is specific to the chi-square test while Beta and Standard error are regression test related steps

each SNP. In the aggregation process, the server computes the global non-missing sample count (n) and allele count using the corresponding noisy parameters and the aggregated noise similar to the sample count step. Finally, the server determines the global minor allele based on the values of the global allele counts.

6. **Minor allele**: The clients compare their local minor allele with the global minor allele. If they are the same, they do nothing. Otherwise, they update the mapping of SNP values read from .bed file. Each SNP value can be 0, 1, 2, or 3 (missing value). These values are encoded based on the minor allele name. If the minor allele is changed, the value of the SNP needs to be swapped if it is 0 or 2. Thus, if a client's minor allele is different from global minor allele, it inverses the mapping of SNP values ($0 \rightarrow 2$ and $2 \rightarrow 0$). The aggregation in the server makes sure that all clients successfully completed this step.
7. **Association test specific steps**: In the following, we elaborate on the steps specific to each association test. Regarding regression tests, *sPLINK* implements the federated versions of ordinary least squares linear regression and Newton-Raphson method based logistic regression.

   Chi-square: The only test-specific step for the chi-square test is *Contingency table*, where each client i computes its local contingency table containing minor allele frequency for cases (t_i), minor allele frequency for controls (r_i), major allele frequency for cases (q_i), and major allele frequency for controls (s_i). The server aggregates the noisy contingency tables from the clients (t'_i, r'_i, q'_i, and s'_i are the elements of the table) and the corresponding aggregated noise from the compensator (N_t, N_r, N_q, and N_s) to compute the global (observed) contingency table (Table 3). It also calculates the expected contingency table based on the observed contingency table (Table 4).

   Given the observed contingency table (O) and the expected contingency table (E), the server computes odds ratio (OR), χ^2, and p-value (P) as follows:

$$\text{OR} = \frac{t \times s}{q \times r} \tag{1}$$

Table 3 Global (observed) contingency table

	Minor allele	Major allele	Total
Case	$t = (((\sum_{i=1}^{i=K} t'_i) \mod p) - N_t) \mod p$	$q = (((\sum_{i=1}^{i=K} q'_i) \mod p) - N_q) \mod p$	$t + q$
Control	$r = (((\sum_{i=1}^{i=K} r'_i) \mod p) - N_r) \mod p$	$s = (((\sum_{i=1}^{i=K} s'_i) \mod p) - N_s) \mod p$	$r + s$
Total	$t + r$	$q + s$	2n

$$\chi^2 = \sum \frac{(E - O)^2}{E} \tag{2}$$

$$P = 1 - F_t(\chi^2, 1) \tag{3}$$

where F_t is the cumulative distribution function (CDF) of χ^2 distribution (degree of freedom is 1).

Linear regression: *Beta* and *Standard error* are two steps specific to linear regression test. In the *Beta* step, each client i computes $X_i^T X_i$ and $X_i^T Y_i$, where X_i^T is the transpose of X_i. In the aggregation process, the server performs the following calculations (K is the number of clients):

$$X^T X = \sum_{i=1}^{i=K} (X_i^T X_i)' - N_{X^T X} \tag{4}$$

$$X^T Y = \sum_{i=1}^{i=K} (X_i^T Y_i)' - N_{X^T Y} \tag{5}$$

$$\beta = (X^T X)^{-1} (X^T Y) \tag{6}$$

where $(X_i^T X_i)'$ and $(X_i^T Y_i)'$ are the noisy local parameters from the clients, $N_{X^T X}$ and $N_{X^T Y}$ are the corresponding aggregated noise from the compensator, and $()^{-1}$ indicates the inverse matrix.

In the *Standard error* step, each client i calculates the local sum square error (SSE) E_i by having the global β vector.

$$\hat{Y}_i = X_i \beta \tag{7}$$

$$E_i = \sum (Y_i - \hat{Y}_i)^2 \tag{8}$$

and then the server calculates the global standard error vector (SE) as follows:

$$E = \sum_{i=1}^{i=K} E'_i - N_E \tag{9}$$

$$\mathrm{VAR} = (\frac{E}{n - m - 1})(X^T X)^{-1} \tag{10}$$

$$\mathrm{SE} = \sqrt{\mathrm{diag}(\mathrm{VAR})} \tag{11}$$

Table 4 Expected contingency table

	Minor allele	Major allele
Case	$\frac{(t+q)\times(t+r)}{2n}$	$\frac{(t+q)\times(q+s)}{2n}$
Control	$\frac{(r+s)\times(t+r)}{2n}$	$\frac{(r+s)\times(q+s)}{2n}$

where E_i' and N_E are the noisy SSE values and the corresponding aggregated noise, respectively; n is the global non-missing sample count, m is the number of features (1 + number of confounding factors), and *diag* is the main diagonal of the matrix. Given the standard error vector, the server computes the *T statistic* (T) and *p-value* (P) as follows:

$$T = \frac{\beta}{\mathrm{SE}} \tag{12}$$

$$\mathrm{DF} = n - m - 1 \tag{13}$$

$$P = 2 \times (1 - F_t(|T|, \mathrm{DF})) \tag{14}$$

in which DF is degree of freedom and F_t is the CDF of T distribution.

Logistic regression: Similar to linear regression, logistic regression has two specific steps: *Beta* and *Standard error*. However, the *Beta* step is iterative in logistic regression (maximum number of iterations is specified by the coordinator and its default value is 20). In each iteration, each client i computes local gradient (∇_i), Hessian matrix (H_i) and log-likelihood (L_i) as follows:

$$\hat{Y}_i = \frac{1}{1 + e^{-X_i\beta}} \tag{15}$$

$$\nabla_i = X_i^T(Y_i - \hat{Y}_i) \tag{16}$$

$$H_i = (X_i^T \circ (\hat{Y}_i \circ (1 - \hat{Y}_i))^T)X_i \tag{17}$$

$$L_i = \sum(Y_i \circ \log \hat{Y}_i + (1 - Y_i) \circ \log(1 - \hat{Y}_i)) \tag{18}$$

where β is the global beta vector from the previous iteration and $\circ$ indicates element-wise multiplication.

The server aggregates the noisy local gradients (∇_i'), Hessian matrices (H_i') and log-likelihood values (L_i') from K clients and the associated aggregated noise values N_∇, N_H, N_L as follows:

$$\nabla = \sum_{i=1}^{i=K} \nabla_i' - N_\nabla \tag{19}$$

$$H = \sum_{i=1}^{i=K} H_i' - N_H \tag{20}$$

$$L = \sum_{i=1}^{i=K} L_i' - N_L \tag{21}$$

Then, it updates the β values accordingly:

$$\beta_{\mathrm{new}} = \beta_{\mathrm{old}} + H^{-1}\nabla \tag{22}$$

where β_{old} is the β value from the previous iteration. The server also compares the newly computed log-likelihood value (L) with the one from previous iteration (L_{old}). If their difference is less than a pre-specified threshold, β values converged, and therefore, it stops updating beta.

In the *Standard error* step, the server shares the global β values with the clients. Each client i computes its local Hessian matrix (H_i) using the global β. The server gets the noisy local Hessian matrices from K clients and the aggregated noise from the compensator and applies the following formula to obtain the global standard error vector (SE):

$$\mathrm{SE} = \sqrt{\mathrm{diag}\left(\left(\sum_{i=1}^{i=K} H_i' - N_H \right)^{-1} \right)} \tag{23}$$

Having standard error values, the server calculates T statistics and p-value (P) as follows:

$$T = \frac{\beta}{\mathrm{SE}} \tag{24}$$

$$P = 1 - F_t(|T|^2, 1) \tag{25}$$

where F_t is CDF of χ^2 distribution (degree of freedom is 1).

8. **Result**: The computation of association test results have been completed for all chunks and the results are shared with all cohorts.

The client and server components of *sPLINK* has been written using the Python API of the HyFed framework [53]. The WebApp component has been implemented using Angular and HTML/CSS. *sPLINK* employs the algorithm-agnostic compensator of the HyFed framework. The *pandas* package [54] is used in the client component to open the dataset files while *NumPy* [55] is leveraged to pre-process the data and to compute the local parameters. In the server component, the *NumPy* and *SciPy* [56] packages are used for aggregation and computing p-values.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13059-021-02562-1.

Additional file 1: Experimental details. Table S1. The SHIP case study. **Table S2.**. The COPDGene case study. **Table S3.** The FinnGen case study. **Supplementary results. Figure S1.** The significant SNPs overlapped between sPLINK and PLINK for the SHIP case study considering Bonferroni significance threshold. **Figure S2.** The Spearman rank correlation coefficient between the *p*-values from each tool and the aggregated analysis for the COPDGene and FinnGen case studies. **Figure S3.** Runtime and network bandwidth usage of sPLINK with varying number of SNPs. **Figure S4.** Runtime and network bandwidth usage of sPLINK with varying number of samples. **Figure S5.** Runtime and network bandwidth usage of sPLINK with varying number of clients. **Experimental setup. Table S4.** The system specification of the physical machines and laptops used to measure the runtime and network bandwidth usage of sPLINK. **Table S5.** The experimental setup used for measuring the runtime and network bandwidth usage of sPLINK.

Additional file 2: Review history.

Acknowledgements

We would like to thank Anne Hartebrodt and Richard Röttger for providing and setting up the workstation for the compensator component at the University of Southern Denmark. SHIP is part of the Community Medicine Research net of the University of Greifswald, Germany (https://www.community-medicine.de), which is funded by the Federal Ministry of Education and Research (grants no. 01ZZ9603, 01ZZ0103, and 01ZZ0403), the Siemens AG, the Ministry of Cultural Affairs as well as the Social Ministry of the Federal State of Mecklenburg-West Pomerania, and the network 'Greifswald Approach to Individualized Medicine (GANI_MED)' funded by the Federal Ministry of Education and Research (grant 03IS2061A). ExomeChip data have been supported by the Federal Ministry of Education and Research (grant no. 03Z1CN22) and the Federal State of Mecklenburg-West Pomerania. Patients and control subjects in FinnGen provided informed consent for biobank research, based on the Finnish Biobank Act. Alternatively, older research cohorts, collected prior the start of FinnGen (in August 2017), were collected based on study-specific consents and later transferred to the Finnish biobanks after approval by Fimea, the National Supervisory Authority for Welfare and Health. Recruitment protocols followed the biobank protocols approved by Fimea. The Coordinating Ethics Committee of the Hospital District of Helsinki and Uusimaa (HUS) approved the FinnGen study protocol Nr HUS/990/2017. The FinnGen project is approved by Finnish Institute for Health and Welfare (THL), approval number THL/2031/6.02.00/2017, amendments THL/1101/5.05.00/2017, THL/341/6.02.00/2018, THL/2222/6.02.00/2018, THL/283/6.02.00/2019 Digital and population data service agency VRK43431/2017-3, VRK/6909/2018-3, the Social Insurance Institution (KELA) KELA 58/522/2017, KELA 131/522/2018 and Statistics Finland TK-53-1041-17. The Biobank Access Decisions for FinnGen samples and data utilized in FinnGen Data Freeze 3 include: THL Biobank BB2017_55, BB2017_111, BB2018_19, BB_2018_34, BB_2018_67, BB2018_71, Red Cross Blood Service Biobank 7.12.2017, Helsinki Biobank HUS/359/2017, Auria Biobank AB17-5154, Biobank Borealis of Northern Finland_2017_1013, Biobank of Eastern Finland 1186/2018, Finnish Clinical Biobank Tampere MH0004, Central Finland Biobank 1-2017. The FinnGen project is funded by two grants from Business Finland (HUS 4685/31/2016 and UH 4386/31/2016) and eleven industry partners (AbbVie Inc, AstraZeneca UK Ltd, Biogen MA Inc, Celgene Corporation, Celgene International II Sàrl, Genentech Inc, Merck Sharp & Dohme Corp, Pfizer Inc., GlaxoSmithKline, Sanofi, Maze Therapeutics Inc., Janssen Biotech Inc). Following biobanks are acknowledged for collecting the FinnGen project samples: Auria Biobank (https://www.auria.fi/biopankki), THL Biobank (https://www.thl.fi/biobank), Helsinki Biobank (https://www.helsinginbiopankki.fi), Biobank Borealis of Northern Finland (https://www.ppshp.fi/Tutkimus-ja-opetus/Biopankki/Pages/Biobank-Borealis-briefly-in-English.aspx), Finnish Clinical Biobank Tampere (https://www.tays.fi/en-US/Research_and_development/Finnish_Clinical_Biobank_Tampere), Biobank of Eastern Finland (https://www.ita-suomenbiopankki.fi/en), Central Finland Biobank (https://www.ksshp.fi/fi-FI/Potilaalle/Biopankki), Finnish Red Cross Blood Service Biobank (https://www.veripalvelu.fi/verenluovutus/biopankkitoiminta) and Terveystalo Biobank (https://www.terveystalo.com/fi/Yritystietoa/Terveystalo-Biopankki/Biopankki/). All Finnish Biobanks are members of BBMRI.fi infrastructure (https://www.bbmri.fi). Figures 2 and 8 were created with BioRender.com.

Peer review information

Barbara Cheifet was the primary editor of this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Review history

The review history is available as Additional file 2.

Authors' contributions

R.N., R.T., T.F., T.K., and J.B. conceived and designed the study. R.N. and R.T. developed the federated algorithms. R.N., R.T., and J.M. implemented the client and server components. J.M., R.N., R.T., T.F., and J.S. implemented the WebApp component. T.K. and R.N. performed the aggregated and federated association tests on the SHIP dataset. R.N., T.F., T.K., M.L., J.B., and S.W. conducted the meta-analysis on the COPDGene case study. R.N. and E.P. performed the meta-analysis on the FinnGen dataset. R.N., J.S., J.M., and R.T. conducted the performance measurements. R.N. and R.T. prepared the original draft. G.K. and D.R. provided critical feedback on the design and implementation of the tool from the privacy perspective. M.L., T.K., N.K.W., D.H., U.V., and J.B. helped with the manuscript revising. T.K., J.B., and M.L. assisted in the improvement of the tool. The authors read and approved the final manuscript.

Funding

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078. This reflects only the author's view and the European Commission is not responsible for any use that may be made of the information it contains. This work was supported by the BMBF-funded de.NBI Cloud within the German Network for Bioinformatics Infrastructure (de.NBI) (031A537B, 031A533A, 031A538A, 031A533B, 031A535A, 031A537C, 031A534A, 031A532B). Open Access funding enabled and organized by Projekt DEAL.

Availability of data and materials

The SHIP dataset [33] is accessible to researchers after completing a web-based request form at http://ship.community-medicine.de and approval. The COPDGene dataset [35] is publicly available (dbGaP accession number phs000179.v1.p1). The FinnGen dataset [36] is available for researchers by requesting access to the FinnGen Sandbox environment, and after completing Sandbox training on how to deal with personal data, and passing an exam about data security (https://www.finngen.fi/en). The sPLINK tool is available online at https://exbio.wzw.tum.de/splink. The source code of sPLINK is publicly available at GitHub (https://github.com/tum-aimed/splink) and Zenodo (DOI: 10.5281/zenodo.5735472) [57] under the Apache License Version 2.0.

Declarations

Ethics approval and consent to participate

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]AI in Medicine and Healthcare, Technical University of Munich, Munich, Germany. [2]Klinikum rechts der Isar, Munich, Germany. [3]Chair of Computational Systems Biology, University of Hamburg, Hamburg, Germany. [4]Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark. [5]Chair of Experimental Bioinformatics, TUM School of Life Sciences, Technical University of Munich, Munich, Germany. [6]Department of Functional Genomics, University Medicine Greifswald, Greifswald, Germany. [7]Institute for Molecular Medicine Finland (FIMM), Helsinki Institute of Life Science (HiLIFE), University of Helsinki, Helsinki, Finland. [8]Applied Tumor Genomics Research Program, Research Programs Unit, Faculty of Medicine, University of Helsinki, Helsinki, Finland. [9]Department of Mathematics and Computer Science, University of Marburg, Marburg, Germany. [10]Division Data Science in Biomedicine, Peter L. Reichertz Institute for Medical Informatics of TU Braunschweig and Hannover Medical School, Brunswick, Germany. [11]Braunschweig Integrated Centre of Systems Biology (BRICS), Brunswick, Germany. [12]Biomedical Image Analysis Group, Imperial College London, London, UK. [13]OpenMined, Oxford, UK.

Received: 25 November 2020 Accepted: 2 December 2021
Published online: 24 January 2022

References

1. Fareed M, Afzal M. Single nucleotide polymorphism in genome-wide association of human population: A tool for broad spectrum service. Egypt J Med Human Genet. 2013;14(2):123–34.
2. Visscher PM, Wray NR, Zhang Q, Sklar P, McCarthy MI, Brown MA, Yang J. 10 years of gwas discovery: biology, function, and translation. Am J Hum Genet. 2017;101(1):5–22.
3. Visscher PM, Brown MA, McCarthy MI, Yang J. Five years of gwas discovery. Am J Hum Genet. 2012;90(1):7–24.
4. De R, Bush W, Moore J. Bioinformatics challenges in genome-wide association studies (gwas). Methods Mol Biol (Clifton, NJ). 2014;1168:63–81.
5. Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MA, Bender D, Maller J, Sklar P, De Bakker PI, Daly MJ, et al. Plink: a tool set for whole-genome association and population-based linkage analyses. Am J Hum Genet. 2007;81(3): 559–75.
6. Evangelou E, Ioannidis JP. Meta-analysis methods for genome-wide association studies and beyond. Nat Rev Genet. 2013;14(6):379–89.
7. Willer CJ, Li Y, Abecasis GR. Metal: fast and efficient meta-analysis of genomewide association scans. Bioinformatics. 2010;26(17):2190–1.
8. Mägi R, Morris AP. Gwama: software for genome-wide association meta-analysis. BMC Bioinformatics. 2010;11(1):288.
9. Lunetta KL. Methods for meta-analysis of genetic data. Curr Protoc Human Genet. 2013;77(1):1–24.
10. Cantor RM, Lange K, Sinsheimer JS. Prioritizing gwas results: a review of statistical methods and recommendations for their application. Am J Hum Genet. 2010;86(1):6–22.
11. de Vlaming R, Okbay A, Rietveld CA, Johannesson M, Magnusson PK, Uitterlinden AG, van Rooij FJ, Hofman A, Groenen PJ, Thurik AR, et al. Meta-gwas accuracy and power (metagap) calculator shows that hiding heritability is partially due to imperfect genetic correlations across studies. PLoS Genet. 2017;13(1):e1006495.
12. Gentry C. Fully homomorphic encryption using ideal lattices. In: Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing; 2009. p. 169–78.
13. Cramer R, Damgård IB, Nielsen JB. Secure Multiparty Computation and Secret Sharing. Cambridge: Cambridge University Press; 2015.
14. McMahan B, Moore E, Ramage D, Hampson S, y Arcas BA. Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics. Fort Lauderdale: PMLR; 2017. p. 1273–82.
15. Konečnỳ J, McMahan HB, Yu FX, Richtárik P, Suresh AT, Bacon D. Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492. 2016. https://arxiv.org/abs/1610.05492.
16. Shamir A. How to share a secret. Commun ACM. 1979;22(11):612–3.
17. Kamm L, Bogdanov D, Laur S, Vilo J. A new way to protect privacy in large-scale genome-wide association studies. Bioinformatics. 2013;29(7):886–93.
18. Cho H, Wu DJ, Berger B. Secure genome-wide association analysis using multiparty computation. Nat Biotechnol. 2018;36(6):547–51.
19. Shi H, Jiang C, Dai W, Jiang X, Tang Y, Ohno-Machado L, Wang S. Secure multi-party computation grid logistic regression (smac-glore). BMC Med Inf Dec Making. 2016;16(3):89.
20. Constable SD, Tang Y, Wang S, Jiang X, Chapin S. Privacy-preserving gwas analysis on federated genomic datasets. BMC Med Inf Dec Making. 2015;15:1–9.
21. Alexandru AB, Pappas GJ. Secure multi-party computation for cloud-based control. In: Privacy in Dynamical Systems. Singapore: Springer; 2020. p. 179–207.
22. Lu W-J, Yamada Y, Sakuma J. Privacy-preserving genome-wide association studies on cloud environment using fully homomorphic encryption. BMC Med Inf Dec Making. 2015;15:1–8.
23. Morshed T, Alhadidi D, Mohammed N. Parallel linear regression on encrypted data. In: 2018 16th Annual Conference on Privacy, Security and Trust (PST). Los Alamitos: IEEE Computer Society; 2018. p. 1–5.
24. Kim M, Song Y, Wang S, Xia Y, Jiang X, et al. Secure logistic regression based on homomorphic encryption: Design and evaluation. JMIR Med Inf. 2018;6(2):8805.
25. Sadat MN, Al Aziz MM, Mohammed N, Chen F, Jiang X, Wang S. Safety: secure gwas in federated environment through a hybrid solution. IEEE/ACM Trans Comput Biol Bioinforma. 2018;16(1):93–102.
26. Chialva D, Dooms A. Conditionals in homomorphic encryption and machine learning applications. arXiv preprint arXiv:1810.12380. 2018. https://arxiv.org/abs/1810.12380.
27. Wang S, Jiang X, Wu Y, Cui L, Cheng S, Ohno-Machado L. Expectation propagation logistic regression (explorer): distributed privacy-preserving online model learning. J Biomed Inf. 2013;46(3):480–96.
28. Wu Y, Jiang X, Kim J, Ohno-Machado L. Grid binary logistic regression (glore): building shared models without sharing data. J Am Med Inf Assoc. 2012;19(5):758–64.
29. Jiang W, Li P, Wang S, Wu Y, Xue M, Ohno-Machado L, Jiang X. Webglore: a web service for grid logistic regression. Bioinformatics. 2013;29(24):3238–40.
30. Nasirigerdeh R, Torkzadehmahani R, Matschinske J, Baumbach J, Rueckert D, Kaissis G. HyFed: A Hybrid Federated Framework for Privacy-preserving Machine Learning. arXiv preprint arXiv:2105.10545. 2021. https://arxiv.org/abs/2105.10545.
31. Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning. Cambridge: Springer; 2009.
32. McHugh ML. The chi-square test of independence. Biochemia Med Biochemia Med. 2013;23(2):143–9.
33. Völzke H, Alte D, Schmidt CO, Radke D, Lorbeer R, Friedrich N, Aumann N, Lau K, Piontek M, Born G, et al. Cohort profile: the study of health in pomerania. Int J Epidemiol. 2011;40(2):294–307.
34. Weiss FU, Schurmann C, Guenther A, Ernst F, Teumer A, Mayerle J, Simon P, Völzke H, Radke D, Greinacher A, et al. Fucosyltransferase 2 (fut2) non-secretor status and blood group b are associated with elevated serum lipase activity in asymptomatic subjects, and an increased risk for chronic pancreatitis: a genetic association study. Gut. 2015;64(4):646–56.

35. COPDGene. http://www.copdgene.org/. Accessed 30 Nov 2021.
36. FinnGen Documentation of R3 release. https://r3.finngen.fi/about. Accessed 30 Nov 2021.
37. Pillai SG, Ge D, Zhu G, Kong X, Shianna KV, Need AC, Feng S, Hersh CP, Bakke P, Gulsvik A, et al. A genome-wide association study in chronic obstructive pulmonary disease (copd): identification of two major susceptibility loci. PLoS Genet. 2009;5(3):e1000421.
38. Pei Y-F, Tian Q, Zhang L, Deng H-W. Exploring the major sources and extent of heterogeneity in a genome-wide association meta-analysis. Ann Hum Biol. 2016;80(2):113–22.
39. Lyu L, Yu H, Yang Q. Threats to federated learning: A survey. arXiv preprint arXiv:2003.02133. 2020. https://arxiv.org/abs/2003.02133.
40. Dwork C. Differential privacy. In: International Colloquium on Automata, Languages, and Programming. Berlin: Springer; 2006. p. 1–12.
41. Cover TM. Elements of Information Theory. New York: John Wiley & Sons; 1999.
42. Dibert A, Csirmaz L. Infinite secret sharing–examples. J Math Cryptol. 2014;8(2):141–68.
43. Tjell K, Wisniewski R. Privacy in Distributed Computations based on Real Number Secret Sharing. arXiv preprint arXiv:2107.00911. 2021. https://arxiv.org/abs/2107.00911.
44. Yang Q, Liu Y, Chen T, Tong Y. Federated machine learning: Concept and applications. ACM Trans Intell Syst Technol (TIST). 2019;10(2):1–19.
45. Nasirigerdeh R, Torkzadehmahani R, Baumbach J, Blumenthal DB. On the privacy of federated pipelines. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21). New York: Association for Computing Machinery; 2021.
46. Galinsky KJ, Bhatia G, Loh P-R, Georgiev S, Mukherjee S, Patterson NJ, Price AL. Fast principal-component analysis reveals convergent evolution of adh1b in europe and east asia. Am J Hum Genet. 2016;98(3):456–472.
47. Kairouz P, McMahan HB, Avent B, Bellet A, Bennis M, Bhagoji AN, Bonawitz K, Charles Z, Cormode G, Cummings R, et al. Advances and open problems in federated learning. arXiv preprint arXiv:1912.04977. 2019. https://arxiv.org/abs/1912.04977.
48. Zhu L, Han S. Deep leakage from gradients. In: Federated Learning. Cham: Springer; 2020. p. 17–31.
49. 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). Manhattan: IEEE; 2019. p. 691–706.
50. Torkzadehmahani R, Nasirigerdeh R, Blumenthal DB, Kacprowski T, List M, Matschinske J, Späth J, Wenke NK, Bihari B, Frisch T, et al. Privacy-preserving Artificial Intelligence Techniques in Biomedicine. arXiv preprint arXiv:2007.11621. 2020. https://arxiv.org/abs/2007.11621.
51. Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan HB, Patel S, Ramage D, Segal A, Seth K. Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. New York: Association for Computing Machinery; 2017. p. 1175–91.
52. PLINK data formats. http://zzz.bwh.harvard.edu/plink/data.shtml. Accessed 30 Nov 2021.
53. HyFed API. https://github.com/tum-aimed/hyfed. Accessed 30 Nov 2021.
54. pandas: Python Data Analysis Library. https://pandas.pydata.org/. Accessed 30 Nov 2021.
55. Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, Smith NJ, Kern R, Picus M, Hoyer S, van Kerkwijk MH, Brett M, Haldane A, del Río JF, Wiebe M, Peterson P, Gérard-Marchant P, Sheppard K, Reddy T, Weckesser W, Abbasi H, Gohlke C, Oliphant TE. Array programming with NumPy. Nature. 2020;585(7825):357–62.
56. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey CJ, Polat İ, Feng Y, Moore EW, VanderPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P, SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nat Methods. 2020;17:261–272.
57. Nasirigerdeh R, Torkzadehmahani R, Matschinske J, Frisch T, List M, Späth J, Weiss S, Völker U, Pitkänen E, Heider D, Wenke NK, Kaissis G, Rueckert D, Kacprowski T, Baumbach J. splink: a hybrid federated tool as a robust alternative to meta-analysis in genome-wide association studies. Zenodo. 2021. https://doi.org/10.5281/zenodo.5735472.

Publisher's Note

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

C

FeatureCloud

Julian Matschinske et al. “The FeatureCloud Platform for Federated Learning in Biomedicine: Unified Approach”. English. In: *Journal of Medical Internet Research* 25 (July 2023). ISSN: 1438-8871. DOI: 10.2196/42621

Original Paper

The FeatureCloud Platform for Federated Learning in Biomedicine: Unified Approach

Julian Matschinske[1*], MSc; Julian Späth[1*], MSc; Mohammad Bakhtiari[1], MSc; Niklas Probul[1], MSc; Mohammad Mahdi Kazemi Majdabadi[1], MSc; Reza Nasirigerdeh[2], MSc; Reihaneh Torkzadehmahani[2], MSc; Anne Hartebrodt[3], MSc, PhD; Balazs-Attila Orban[4], MSc; Sándor-József Fejér[4], MSc; Olga Zolotareva[2], PhD; Supratim Das[1], MSc; Linda Baumbach[5], PhD; Josch K Pauling[2], PhD; Olivera Tomašević[6], MSc; Béla Bihari[4], MSc; Marcus Bloice[7], MSc; Nina C Donner[8], PhD; Walid Fdhila[9], PhD; Tobias Frisch[3], PhD; Anne-Christin Hauschild[10], Prof Dr; Dominik Heider[11], Prof Dr; Andreas Holzinger[7], Prof Dr; Walter Hötzendorfer[12], Dr; Jan Hospes[12], Mag iur; Tim Kacprowski[13], Prof Dr; Markus Kastelitz[12], PhD; Markus List[2], PhD; Rudolf Mayer[9], MSc; Mónika Moga[4], PhD; Heimo Müller[7], PhD; Anastasia Pustozerova[9], MSc; Richard Röttger[3], Prof Dr; Christina C Saak[1], PhD; Anna Saranti[7], PhD; Harald H H W Schmidt[14], Prof Dr; Christof Tschohl[12], Dr; Nina K Wenke[1], PhD; Jan Baumbach[1], Prof Dr

[1]University of Hamburg, Hamburg, Germany
[2]Technical University Munich, Munich, Germany
[3]University of Southern Denmark, Odense, Denmark
[4]Gnome Design SRL, Sfântu Gheorghe, Romania
[5]University Medical Center Hamburg-Eppendorf, Hamburg, Germany
[6]University of Novi Sad, Novi Sad
[7]Medical University of Graz, Graz, Austria
[8]Concentris Research Management gGmbH, Fürstenfeldbruck, Germany
[9]SBA Research gGmbH, Vienna, Austria
[10]University Medical Center Göttingen, Göttingen, Germany
[11]Philipps-University of Marburg, Marburg, Germany
[12]Research Institute AG & Co KG, Vienna, Austria
[13]Technical University Braunschweig and Hannover Medical School, Brunswick, Germany
[14]Maastricht University, Maastricht, Netherlands
[*]these authors contributed equally

Corresponding Author:
Julian Matschinske, MSc
University of Hamburg
Notkestrasse 9
Hamburg, 22607
Germany
Phone: 49 40 42838 ext 7640
Email: julian.matschinske@uni-hamburg.de

Abstract

Background: Machine learning and artificial intelligence have shown promising results in many areas and are driven by the increasing amount of available data. However, these data are often distributed across different institutions and cannot be easily shared owing to strict privacy regulations. Federated learning (FL) allows the training of distributed machine learning models without sharing sensitive data. In addition, the implementation is time-consuming and requires advanced programming skills and complex technical infrastructures.

Objective: Various tools and frameworks have been developed to simplify the development of FL algorithms and provide the necessary technical infrastructure. Although there are many high-quality frameworks, most focus only on a single application case or method. To our knowledge, there are no generic frameworks, meaning that the existing solutions are restricted to a particular type of algorithm or application field. Furthermore, most of these frameworks provide an application programming interface that needs programming knowledge. There is no collection of ready-to-use FL algorithms that are extendable and allow users (eg, researchers) without programming knowledge to apply FL. A central FL platform for both FL algorithm developers and users does not exist. This study aimed to address this gap and make FL available to everyone by developing FeatureCloud, an all-in-one platform for FL in biomedicine and beyond.

Methods: The FeatureCloud platform consists of 3 main components: a global frontend, a global backend, and a local controller. Our platform uses a Docker to separate the local acting components of the platform from the sensitive data systems. We evaluated our platform using 4 different algorithms on 5 data sets for both accuracy and runtime.

Results: FeatureCloud removes the complexity of distributed systems for developers and end users by providing a comprehensive platform for executing multi-institutional FL analyses and implementing FL algorithms. Through its integrated artificial intelligence store, federated algorithms can easily be published and reused by the community. To secure sensitive raw data, FeatureCloud supports privacy-enhancing technologies to secure the shared local models and assures high standards in data privacy to comply with the strict General Data Protection Regulation. Our evaluation shows that applications developed in FeatureCloud can produce highly similar results compared with centralized approaches and scale well for an increasing number of participating sites.

Conclusions: FeatureCloud provides a ready-to-use platform that integrates the development and execution of FL algorithms while reducing the complexity to a minimum and removing the hurdles of federated infrastructure. Thus, we believe that it has the potential to greatly increase the accessibility of privacy-preserving and distributed data analyses in biomedicine and beyond.

(J Med Internet Res 2023;25:e42621) doi: 10.2196/42621

KEYWORDS

privacy-preserving machine learning; federated learning; interactive platform; artificial intelligence; AI store; privacy-enhancing technologies; additive secret sharing

Introduction

The Problem of Scattered Data

Machine learning (ML) and artificial intelligence (AI) have increased in popularity over the last decade, leading to discoveries in various fields, including biomedicine [1-3]. The utility of ML and AI models depends on the size and quality of the available training data. However, data sources are often scattered across multiple facilities, and privacy regulations restrict data sharing, rendering large-scale, centralized ML infeasible. Particularly in biomedicine, the collection of molecular and clinical data is becoming ubiquitous with the successful applications of ML in diagnostics [4] or drug discovery [5]. Privacy concerns hinder even faster advances because of the small sample size of the individual data sets available, such as in the case of rare diseases.

Federated Learning and Privacy-Enhancing Technologies

One way to overcome these challenges is federated learning (FL). FL allows distributed data analysis by only exchanging model parameters and local models instead of sensitive raw data [6]. Hence, analyses can benefit from considerably larger data sets and be exploited with a lower risk of revealing primary data. FL can be divided into several subcategories that address different problems in decentralized computation and differ in their requirements [7]. First, FL can be categorized according to how the data are distributed among the clients. Horizontal FL addresses the training of a model on distributed data that has the same features but different samples. Vertical FL, in contrast, trains a model for the same samples but distributed features. Second, FL is distinguished by the number of clients that participate. Training a model on decentralized data from several organizations or data silos, such as hospitals or companies, is called cross-silo FL. If model training involves thousands or millions of clients, such as mobile phones or internet of things devices, we speak of cross-device FL. A typical FL setup consists of several clients and a central aggregator. Each client updates a local model based on its local data and sends it to a central aggregator. Here, the local models are aggregated into a common global model by an aggregation function, such as federated average [6]. This global model is then broadcasted to each client again. The entire process is repeated for the iterative algorithms.

Although other techniques, such as homomorphic encryption (HE), also allow for the analysis of distributed data by enabling calculations on encrypted data directly, they are computationally expensive compared with FL. In addition, they often require drastic changes to their original ML algorithm. In contrast, FL alone cannot always fulfill strict privacy requirements [8,9]. Therefore, to improve data privacy, FL can be combined with privacy-enhancing technologies (PETs) [10], such as secure aggregation [11] or differential privacy (DP) [12,13]. A recent study demonstrated that federated algorithms could achieve comparable or identical results compared with centralized ML [14-18].

Prior Work

Several frameworks have recently been developed to make FL available for a broader user group. Backend frameworks provide developers with methods to simplify the implementation of federated and privacy-aware algorithms [19-22]. They are limited to users with a strong background in software development or programming experience. Such skills are usually not expected from clinical experts and researchers, which considerably restricts their usability. All-in-one frameworks bring privacy-aware analyses to users without in-depth programming skills by providing a graphical user interface (GUI) [23-26]. However, most existing all-in-one frameworks are either not extendible or highly specific, focusing on a certain type of algorithm (eg, deep learning [DL] only) or application (eg, neuroimaging and genomics).

Existing Shortcomings

Although the available frameworks demonstrate that FL is applicable and accelerates research in health care or biomedicine, the focus on 1 specific application or algorithm is also a huge restriction, especially in the collaboration of different fields. To the best of our knowledge, a generic, low-code, and open-source platform that can be driven and extended openly by the community to cover different algorithms and fields has been unavailable. However, such a platform is needed to enable FL across different applications and to make it applicable for users without technical knowledge of FL infrastructure or coding skills.

Goal of This Work

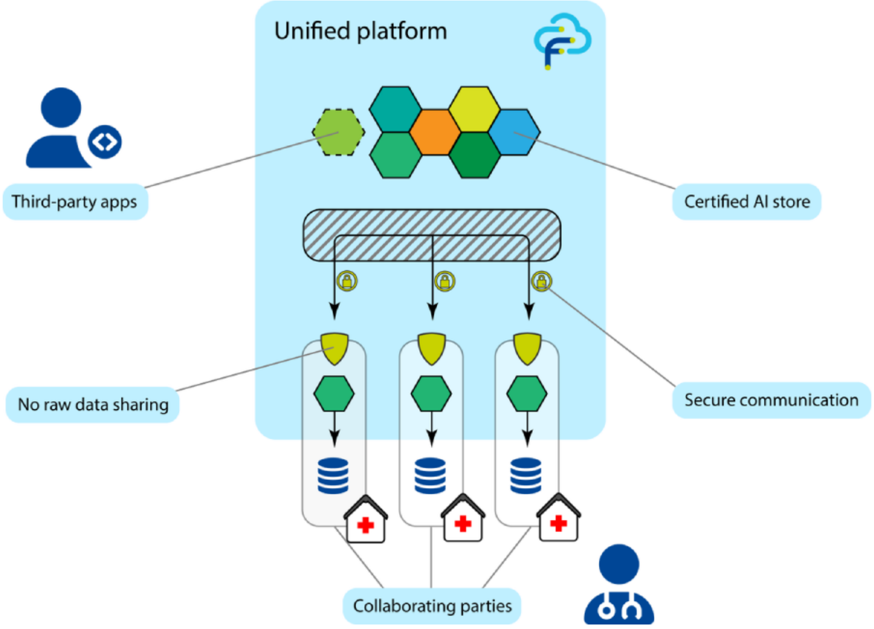
To close this gap, we present FeatureCloud, a comprehensive platform covering all the required steps from project coordination and workflow execution for the development of algorithms for cross-silo FL [27]. It incorporates and facilitates the development and deployment of federated algorithms and alleviates the technical difficulties of end users by providing a complete and ready-to-use infrastructure. Contrary to existing programming frameworks, FeatureCloud provides a running all-in-one platform that eliminates the need for developers and users to arrange a server deployment to conduct a federated study.

Methods

Overview

FeatureCloud was developed as a unified platform to increase the accessibility of FL for two large user groups as follows: (1) end users running FL algorithms to train ML models on distributed data sets and (2) developers implementing federated algorithms for statistics or ML that are not easily accessible in federated environments yet. As illustrated in Figure 1, the interface between developers and end users is our integrated AI store. Application developers can easily implement their own applications and publish them in the AI store, making them easily accessible to end users. Out of a broad collection of applications in the AI store, end users can assemble tailored workflows, invite collaborators, and perform FL on geographically distributed data. Therefore, FeatureCloud provides a complete infrastructure, including secure state-of-the-art communication, no raw data sharing, and several mechanisms to keep the actual data private.

Figure 1. Outline of the FeatureCloud system. Medical institutions collaborate in a federated study with all primary or raw data remaining at their original location. FeatureCloud handles the distribution, execution, and communication of certified artificial intelligence (AI) applications from the FeatureCloud AI store and addresses developers and end users.



Implementation

In this section, we present our implementation of the FeatureCloud platform: its system architecture, the FeatureCloud application programming interface (API) for developers, and the FL scheme and PETs used. Furthermore, we present the FL algorithms used for the evaluation of our platform.

System Architecture

FeatureCloud was developed as a system consisting of several interacting parts distributed between the participants and a central server. The central components include the backend (Python and Django), frontend (Angular), and Docker registry. The local components include the controller (Golang), the Docker engine, and the application instances (Docker images). Figure 2 shows the system components and the communication channels between them. Further details regarding their implementation and technology used can be found in Multimedia Appendix 1.

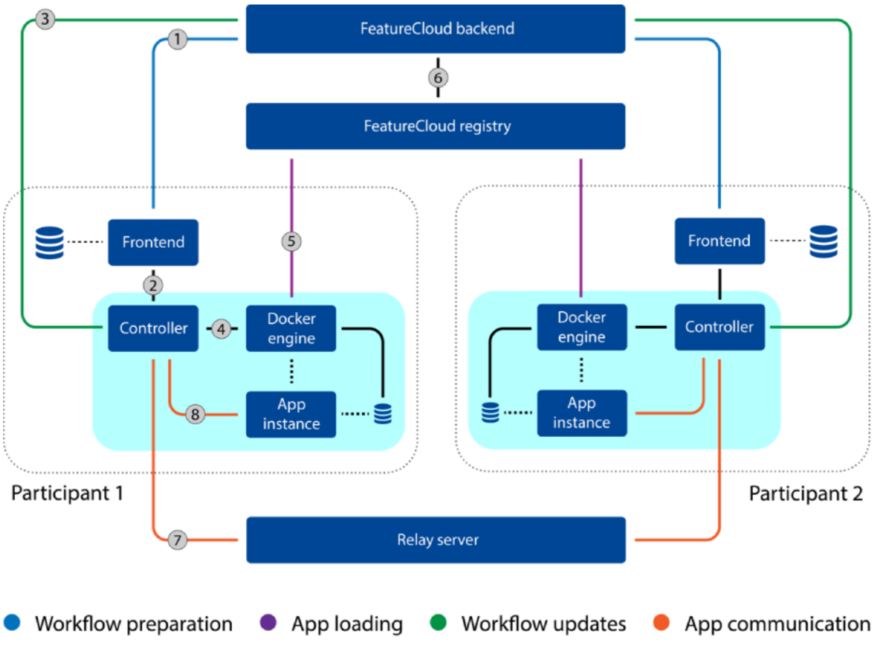
The frontend is a web application running on a web browser. It uses the FeatureCloud backend API (link 1 in Figure 2) to offer all the features of the AI store and for collaborative project management. It is also connected to the controller to allow for monitoring and handing over data for workflow runs (link 2 in Figure 2).

The controller is responsible for orchestrating the local part of the workflow execution. It receives information via the FeatureCloud backend API (link 3 in Figure 2), indicating which applications to execute next, and reports about the progress. Contrary to the relay server traffic, this traffic only contains metainformation about the execution and no data used in the algorithms themselves. It uses the Docker API (link 4 in Figure 2) to instruct the Docker engine to manage containers that serve as isolated application instances and pulls the images of the required applications for a workflow from the Docker registry (link 5 in Figure 2). When pushing new application versions, the Docker registry ensures that the user is entitled to do so by verifying their credentials through the backend (link 6 in Figure 2). In addition, the controller is an integral part of the security and privacy system of FeatureCloud. It handles local data processing and is the only part of FeatureCloud that has access to the local computer system. The controller runs in a Docker container to prevent random access to data on the system. Therefore, it only has access to selected data sets that were actively chosen by a system administrator or a user through a FeatureCloud application.

The participants of a federated workflow must also agree on a common relay server. The relay server, implemented in Go, is responsible for transmitting all traffic of the federated algorithms via a secure socket connection (link 7 in Figure 2). This central communication hub is aware of all the participants and their roles in the federated execution. It follows the required communication pattern, sending aggregated models to all the participants and local model parameters to the coordinating party only. Although FeatureCloud provides a relay server instance used by default, it is possible to use a private instance to completely shield the traffic from anyone outside the collaboration by adjusting the configuration file for the controller.

As FeatureCloud applications are a dynamic system component, partly contributed by external developers, it is necessary to isolate their implementation. This is achieved by using Docker, which ensures that they cannot access system resources other than required, especially the filesystem and network, and allows for limiting resource use, such as central processing unit or memory. They receive their input data inside a Docker volume and communicate with the *controller* through a defined API (link 8 in Figure 2). This API is the main interface between externally developed applications and the FeatureCloud system. It is http based and requires the application to act as a web server, which means that it needs to wait for the controller to query for data and cannot actively send data by itself; thus, active network access can be forbidden.

Figure 2. System architecture of FeatureCloud with 2 participants. The controller, frontend, Docker engine, and application instances run locally at each participant's site. The FeatureCloud backend and Docker registry are running on FeatureCloud servers. The relay server can be run on a separate server, or participants can use a provided instance from FeatureCloud. The components are connected via transmission control protocol/IP connections (straight lines). All links are http based, except for link 7, which uses a raw socket connection. Links 1 to 3 use JSON for serialization, and links 4 to 6 use the Docker application programming interface.



The FeatureCloud API for Developers

To avoid restricting end users to the current selection of applications, FeatureCloud invites external developers to implement their own federated applications and publish them in our AI store. A FeatureCloud application is a program isolated inside a Docker container that communicates with other instances using the FeatureCloud API [28]. Several templates and example applications are provided to further facilitate the implementation by directly explaining the API with code.

In addition to the AI store and the API, FeatureCloud provides tools to accelerate the development of federated applications. When developing a new federated method, application developers can directly start with the federation of the AI logic by using an existing template. To verify that the API has been implemented correctly, a simulation tool aids the developer in testing their application before publishing. Each test run specifies the number of participants, test data, and communication channels and subsequently starts the corresponding instances, simulating a real-world execution on multiple machines. During the test run, it shows logs and results for each participant and the network traffic to monitor the execution and identify bugs and potential communication bottlenecks.

After the development phase, applications can be published in the FeatureCloud AI store. Developers need to fill out a form prompting all relevant information about the application, which is displayed to the end users and used for the search and filter functions. Subsequently, they can push their Docker image into the Docker registry of the FeatureCloud platform. For end users collaborating with the developer, who explicitly enables uncertified applications, it is already usable and can be tested in a real-world scenario. For other end users, we enforce a certification process to increase the hurdle for malicious applications and maintain high privacy standards in the AI store. To this end, the developer must provide the necessary documentation and details regarding the implemented privacy mechanism. Furthermore, the application's source code must be accessible so that the application can be exhaustively tested and vetted by the FeatureCloud team and community for possible privacy leaks. When the certification process has been successfully completed by a member of the FeatureCloud consortium according to a defined checklist (Multimedia Appendix 1), the application will be displayed in the AI store and can be used by all end users. If the certification process is unsuccessful, the developer is notified and requested to address the issues raised. Upon each update of an application, a new certification procedure is triggered.

As FeatureCloud does not impose restrictions on the types of algorithms it supports, the running environment of the federated applications is kept very general. It allows the implementation of any type of ML algorithm and an optional custom GUI for user interaction in the form of a web-based frontend. This GUI can be used to receive input parameters, indicate the current progress, or display the results. No direct internet access is granted to the applications to avoid security risks.

FL Scheme and PETs

FL generally involves two possibly alternating operations as follows: (1) local optimization and (2) global aggregation. In FeatureCloud, all running instances of a federated application have 1 of 2 roles (participant and coordinator) performing the respective federated operation. FeatureCloud expects precisely 1 coordinator and an arbitrary number of participants, leading to a star-based architecture. We chose this architecture over others because it mirrors the general design of a FL scheme with a central aggregator and clients with local data sets.

After the local learning operation has been completed by a participant, it sends the local parameters to the coordinator. The coordinator collects these parameters and aggregates them into a collective (global) model, which is shared with the participants again. Depending on the type of ML algorithm, these 2 operations can alternate multiple times, for example, until convergence or a predefined number of iterations has been reached (Figure S1 in Multimedia Appendix 1). For some algorithms (eg, random forest [RF] and linear regression), only 1 iteration is necessary. However, this strict separation between optimization and aggregation is not actively enforced by FeatureCloud. In many cases, aggregation can start after the first parameters have been received, thereby increasing efficiency through parallelization of the computation. During the implementation of a federated application, the distinction between the coordinator and the participant is of conceptual relevance. However, in practice, the coordinator can also obtain local data that can be used for training. Therefore, FeatureCloud allows the coordinator to simultaneously adopt the role of a participant.

Although FL improves privacy, it can still leak information to the coordinator, who can see all individual models before aggregating them. Local updates of the model based on a previously distributed global model may reveal information regarding the primary data [29]. Secure aggregation techniques can address this problem. In FeatureCloud, we integrated additive secret sharing as a mitigation method to obtain the global sum without revealing the local submodels. Application developers can use this method with minimal or no added complexity to their algorithms. More details can be found in Multimedia Appendix 1.

Federated Algorithms

Comparing Federated Algorithms

As there are unique challenges for federating individual algorithms, each ML model needs to be developed independently and, therefore, needs to be based on a different underlying federation mechanism. This means that each algorithm has challenges regarding effectiveness, privacy, or scalability that need to be solved by the application developers. For the evaluation of our platform in this work, we used 4 FeatureCloud FL applications: the linear and logistic regression applications, a RF, and a DL application.

Federated Linear and Logistic Regression

For the implementation of the linear and logistic regression applications, the methods introduced by Nasirigerdeh et al [17] have been adapted from genome-wide association studies (GWAS) to a general ML use case. For linear regression, the local X^TX and X^TY matrices are computed by each participant individually, where X is the feature matrix and Y is the label vector. Then, they are sent to the coordinator, aggregating the local matrices to the global matrices by adding them. Using these global matrices, the coordinator can calculate the beta vector through the federated method in such a way that it is identical to the beta vector calculated through the nonfederated method.

Logistic regression was implemented as an iterative approach. On the basis of the current beta vector, the local gradient and Hessian matrices of each participant are calculated and shared with the coordinator in each iteration. The coordinator aggregates the matrices again by adding them, updates the beta vector, and broadcasts it back to the participants. This process is repeated until convergence or the maximum number of iterations (prespecified for each execution) is achieved.

Internally, the scikit-learn model API has been used to implement the applications [30,31]. In the performance evaluation, we used the default scikit-learn hyperparameters for the linear regression models. For logistic regression, the penalty was set to none; the maximum number of iterations was set to 10,000; and the “lbgs” solver was used to fit the models.

Federated RF

We used the popular RF classifier and RF regressor as the second algorithm for our evaluation. As an ensemble algorithm, RF can be easily federated in a naive manner [32]. Our implementation trains multiple classification or regression decision trees on the local primary data of each participant. The fitted trees are then transmitted to the coordinator and merged into a global RF. To account for the different number of samples for each participant, each of them contributes a portion of the merged RF proportional to the number of samples. To achieve a similar behavior as the centralized implementation, the size of the merged RF is kept constant, meaning that an increasing number of participants decreases the number of required trees per participant. The federated computation occurs in three steps, each involving data exchange as follows: (1) participants indicate the number of samples and receive the total number of samples; (2) participants train the required number of trees, and the aggregator merges them into a global RF; and (3) participants receive the aggregated model to evaluate its performance on their data and share the results to obtain a global summary.

As the aim is not to achieve the highest possible accuracy but to compare the federated version with the nonfederated version, the hyperparameters were set to the default values of sklearn, namely, 100 decision trees, Gini impurity minimization as the splitting rule, and feature sampling equal to the square root of the features. Prepruning parameters such as maximum depth, minimum samples per node, and other constraints were not applied.

Federated DL

Our federated DL application is based on the federated average algorithm [6]. In the training phase, the weights and biases update is performed iteratively, where each iteration implies the parameter aggregation performed in three steps as follows: (1) the local weights and biases are computed by every participant individually and shared with the coordinator, (2) the coordinator averages the parameters and broadcasts them back to participants, and (3) the participants receive the new values of weights and biases and update the weights and biases of their model accordingly. After the final number of iterations is reached, the model performance of each participant is independently assessed using their data. The local weights and biases update is performed with the back-propagation algorithm, applied to data batches of a specified size. The neural network model architecture and training were implemented using the PyTorch library [33]. The application enables the implementation of any architecture and provides a centralized version of a PyTorch code. The application also enables federated transfer learning to be applied to a pretrained model, whose specified layers are trained in the same federated fashion.

Results

The results comprise the unified platform and an evaluation demonstrating the technical capabilities of FeatureCloud to run different workflows. The platform consists of the open AI store, development and debugging tools, and an execution environment for federated workflows.

Unified Platform

The unified platform (Figure 1) provides developers with an API to quickly develop privacy-enhancing FL applications. This supports a hybrid communication scheme for FL and secure aggregation (additive secret sharing). The integrated AI store is the interface between developers and end users, displaying and describing all available applications. Developers can publish (deploy) their applications in the AI store that are then available for use in federated workflows for the end users, for example, biomedical researchers. They can quickly create projects, assemble federated workflows with the applications from the AI store, invite other sites to the study, and view and download the results of each run. The interface of end users with the complicated federated architecture is reduced to only a web frontend and the FeatureCloud controller, running in the background and responsible for the local processing of sensitive data. Moreover, all applications and the entire architecture of FeatureCloud are open source, making it the first unified and open-source FL platform that considers all steps including development, deployment, and execution.

AI Store

The integrated AI store provides an intuitive and user-friendly interface for biomedical researchers and developers. It offers a variety of applications and displays basic information about them, including short descriptions, keywords, end-user ratings, and certification status. Users can easily find applications of interest via a textual search and filter them by type (preprocessing, analysis, and evaluation) and their privacy-enhancing techniques (FL, DP, and HE). End users can review the applications and provide feedback. The application pages display a method summary, description, user reviews, developer name, and contact details to report bugs. Each application provides either a GUI or a configuration file to set the application parameters and adapt them to different contexts. This reduces technical details and makes applications user friendly for end users, independent of their background. When users add applications to their library, they can assemble them into a workflow and manage the execution with other collaborators on the FeatureCloud website without having to download any additional software.

The AI store has a broad selection of popular ML models, as listed in Table 1. The applications are categorized into preprocessing, analysis, and evaluation. Some analysis applications, such as linear regression and RF, are generic and suitable for different data types and application scenarios. These applications can be easily integrated into a federated workflow with preprocessing and evaluation applications, such as a federated standardization of the input data and a final evaluation of the trained classifier with several performance metrics. Other applications, such as the sPLINK [17] application for federated GWAS, integrate all the necessary steps of an application-specific workflow and do not require combination with other applications.

Table 1. Applications in the FeatureCloud artificial intelligence (AI) store[a].

Application	Type	Description
Ada boost	Machine learning	Classification model based on boosting trees
CACS[b] forest	Machine learning	Random forest classifying patients into their CACS
Cox PH[c] model	Survival analysis	Survival regression based on the lifelines library
Cross-validation	Preprocessing	Local splits for a k-fold cross-validation
Deep learning	Machine learning	Deep neural networks implemented in PyTorch
Evaluation (Classification)	Evaluation	Evaluation with various classification metrics (eg, accuracy)
Evaluation (Regression)	Evaluation	Evaluation with various regression metrics (eg, mean squared error)
Evaluation (survival)	Evaluation	Evaluation of survival or time-to-event predictions
Flimma	Differential expression	Differential expression analysis based on limma-voom
Graph-guided random forest	Machine learning	Random forest classification, regression, and survival based on graphs
Kaplan-Meier estimator	Survival analysis	Survival function estimation and log-rank test
Linear regression	Machine learning	Regression model
Logistic regression	Machine learning	Classification model
Nelson-Aalen estimator	Survival analysis	Hazard function estimation and log-rank test
Normalization	Preprocessing	Standardizing input data
One-hot encoder	Preprocessing	One-hot encoding for categorical variables
Random forest	Machine learning	Classification and regression model based on decision trees
Random survival forest	Survival analysis	Survival prediction based on scikit-survival
SVD[d]	Machine learning	SVD for dimensionality reduction
sPLINK[e]	GWAS[f]	GWAS based on PLINK
Survival SVM[g]	Survival analysis	Survival prediction based on scikit-survival

[a] The growing list of applications available in the AI store covers preprocessing, analysis, and evaluation. All-in-one applications cover the entire workflow for a more specific domain and can be executed without other applications.

[b] CACS: coronary artery calcification score.

[c] PH: proportional hazard.

[d] SVD: singular value decomposition.

[e] sPLINK: secure PLINK.

[f] GWAS: genome-wide association studies.

[g] SVM: support vector machine.

Multi-institutional Federated Workflows

FeatureCloud offers easy project management for the execution of FL workflows. In these workflows, users can select from a large variety of applications in the AI store and connect them to the entire workflow. Before collectively running a federated workflow, all collaborating sites (participants) must download and start the client-side FeatureCloud controller on their machines. It only requires Docker, which is freely available for all the major operating systems. Users also need to create an account on the FeatureCloud website, which serves as a web frontend and is used to coordinate the FeatureCloud system (refer to the *Methods* section and Multimedia Appendix 1 for details on the architecture). Each collaborative execution of applications is organized into so-called projects on the web frontend. They contain a description of the planned analysis, connect the collaborating partners by allowing invited participants to join, and show the current status of the workflow (Figure S2 in Multimedia Appendix 1).

Workflows are composed of 1 or multiple applications from the AI store that are to be executed consecutively. Each application produces intermediate results that serve as input for the consecutive application. Intermediate results are maintained on the respective machines and are not shared with other participants. The last application produces the final results, which are then shared with all the project participants. During the execution of a workflow, its progress can be monitored on the FeatureCloud website, showing the current stage, computational progress, and intermediate results from each application. Applications can provide their own user interface, allowing for user interaction if necessary and for showing specific reports. Users can monitor application logs and react in case something unexpected occurs (eg, stop and rerun the workflow with other data or a different configuration). When the last application in the workflow successfully completes its computation, the final results are automatically shared with all project participants. Intermediate results and application logs remain available on the local machines to allow for later verification. For example, the results may include a report showing the effectiveness of the trained model and the model itself. The latter can also be used outside of FeatureCloud. For example, if a project fails because a participant drops out, it can be restarted quickly after the problem has been solved. During the entire process, no programming knowledge or command-line interaction is required, making the system especially suited for medical personnel without technical education.

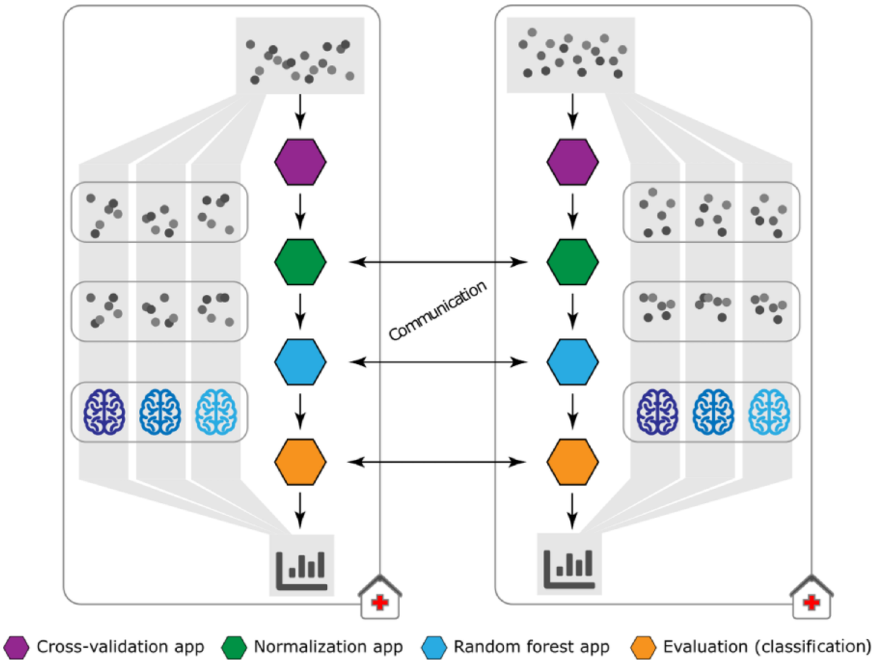
Evaluation

Methods and Data Sets

To evaluate the practical applicability of FeatureCloud, multiple workflows operating on different data sets were created. Except for DL, each workflow consists of a cross-validation (CV) application (10-fold CV), a standardization application, a model training application, and a final evaluation application (Figure 3). For DL, we evaluated a 20% test set, as this is more common for big data to reduce the training time. Individual applications are data-type agnostic and are suitable for various applications. Classification analyses were performed on the Indian Liver Patient Dataset [34] with 579 samples and 10 features and the Cancer Genome Atlas Breast Invasive Carcinoma [35] data set with 569 samples and 20 features. For regression analyses, they were evaluated on the Diabetes [36] data set with 442 samples and 10 features and the Boston [37] house prices data set with 506 samples and 13 features, both provided by scikit-learn [30]. Finally, for DL regression, we used a large data set from the Survey of Health, Aging, and Retirement in Europe [38], with 12 questionnaire variables and the target 12-item critical assessment of protein structure prediction quality of life score. After dropping samples with “Refusal” and “Don’t know” type values in those 12 variables and nonavailable 12-item critical assessment of protein structure prediction quality of life score, we were left with 42,894 (91.79%) out of 46,733 samples. Further details regarding the network architecture are provided in Multimedia Appendix 1.

For each workflow, we split the central data set into 5 participants with uneven data distribution. Participants 1, 2 and 3, and 4 and 5 each had 10% (4289/42,894), 15% (6434/42,894), and 30% (12,868/42,894) of the samples, respectively. We used the F_1-score to evaluate the classification models and the root mean squared error for the regression models, as both are common metrics used to evaluate ML models. Furthermore, we also investigated the scalability concerning runtime and network traffic for 2 to 8 participants as well as a larger number of participants and iterations.

Figure 3. Workflow structure used for evaluation. The first application (purple—Cross-Validation) creates splits for cross-validation (CV). All following applications perform their tasks on each split individually, in a federated fashion, only transmitting model parameters. The gray dots represent intermediate training and test data. The second application (green—"Normalization") performs normalization, and the third application (blue—"Random Forest") trains the models, generating a global model based on the output of the normalization application. The resulting global model is evaluated in the evaluation application (orange—"Evaluation [Classification]"). The evaluation results are finally aggregated to obtain an evaluation report based on the initial CV splits.



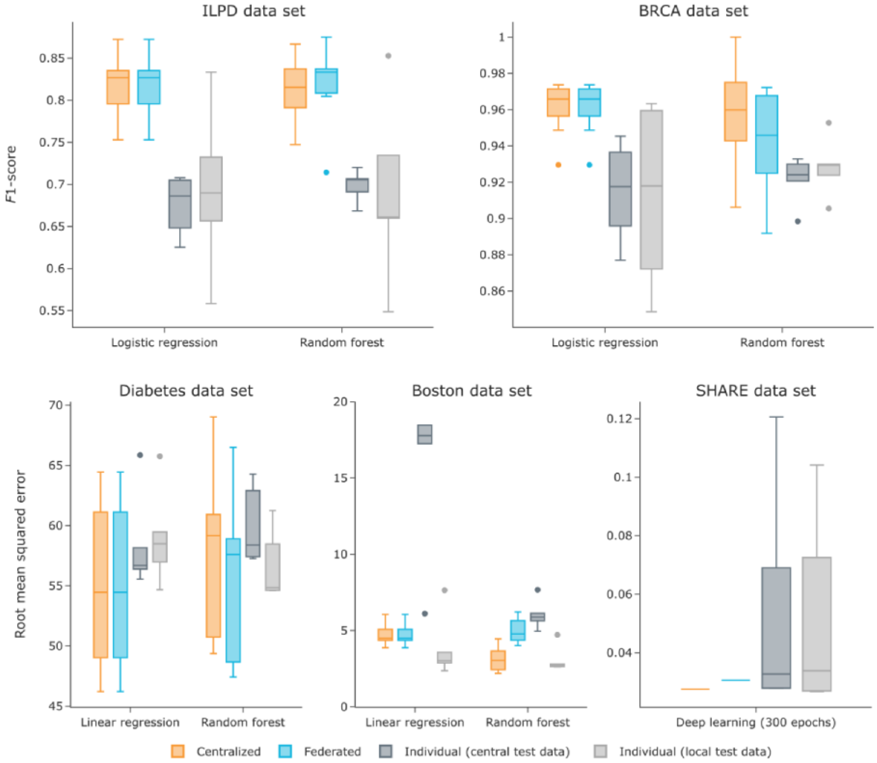
Performance

Previous studies have shown that FL can achieve similar performance to centralized learning in many scenarios [14,15,39]. To verify the approach used in FeatureCloud, we compared the performance of 4 federated FeatureCloud applications integrated into an ML workflow with their corresponding centralized scikit-learn [30] models. The results are shown in Figure 4. For logistic regression and linear regression, the FeatureCloud workflow achieved a performance identical to that of scikit-learn, which is consistent with the previous results of federated linear and logistic regression applications [17,40]. A similar performance was achieved for the RF regression and classification models. Owing to the simple aggregation method that combines the local trees into 1 global tree, identical results were not obtained or expected. Owing to the bootstrapping mechanism and its attached randomness, the federated RF sometimes performs slightly better than the centralized approach. As a final example, our federated DL model trained in 300 epochs produced a very close root mean squared error compared with the centralized model.

Furthermore, we compared the federated models with the individual models trained and evaluated by each participant (10-fold CV, except DL). Here, we distinguish between the central evaluation of the models on the overall test splits (central test data), identical to the test splits for the centralized and federated models, and the local evaluation of the models on the local test splits only (local test data). As shown in Figure 4, the local evaluation performance varies widely but is worse on average than the federated models. For classification, the local evaluation performed worse than the federated models. However, for the regression models, the locally evaluated models of the individual participants sometimes outperformed the centralized model. Nevertheless, compared with the central test data, it is obvious that these models did not generalize well and only performed well for the individual participants with a very small test set. This can be deceptive, as in this case, even the 10-fold CV cannot be trusted. Furthermore, our DL model evaluated on a 20% test set performs much more reliably than individual client models, which can have drastically worse results than the federated or centralized models. This highlights the effectiveness of FL, as these models use more training and test data, resulting in more generalized models. Our RF application is based on a previously published implementation [32] and confirms that our platform yields comparable results, including scenarios in which the data are neither independent nor identically distributed (nonindependent and identically distributed). It performed much more reliably than only using individual client data.

As an additional example of clinical data analysis, we evaluated the Kaplan-Meier estimator application that implements an already published approach for federated survival curves and a log-rank test for multi-institutional time-to-event analyses [18]. The application, implemented and run in FeatureCloud, produced identical results to the centralized analysis (Table S1 in Multimedia Appendix 1) on the lung cancer data set of the North Central Cancer Treatment Group [41]. Similarly, we evaluated the Flimma application for differential gene expression analysis [16] as an example of biomedical data on a subset of 152 breast cancer expressions from the Cancer Genome Atlas repository [42] with 20,536 features. Our Flimma application produced highly similar results to those of the centralized analysis (Figure S3 in Multimedia Appendix 1). These 2 examples further show that FeatureCloud has the capability of implementing and running different approaches and bringing them into a production system.

Figure 4. Performance evaluation of federated artificial intelligence methods. The box plots show the results of a 10-fold cross-valuation for the different classification and regression models and data sets in multiple settings. Only the deep learning model was evaluated on a test set. The centralized results are shown in orange, the corresponding federated results in blue, and the individual results obtained locally at each participant in gray. Each model was evaluated on the entire test set (dark gray) such as the centralized and federated models and on the individual (local) parts of the test set (light gray). The federated logistic and linear regressions perform in identical fashion to their centralized versions, and the federated random forest and deep learning models perform in similar fashion to their centralized versions. BRCA: Breast Invasive Carcinoma; ILDP: Indian Liver Patient Dataset; SHARE: Survey of Health, Aging and Retirement in Europe.



Runtime and Network Traffic

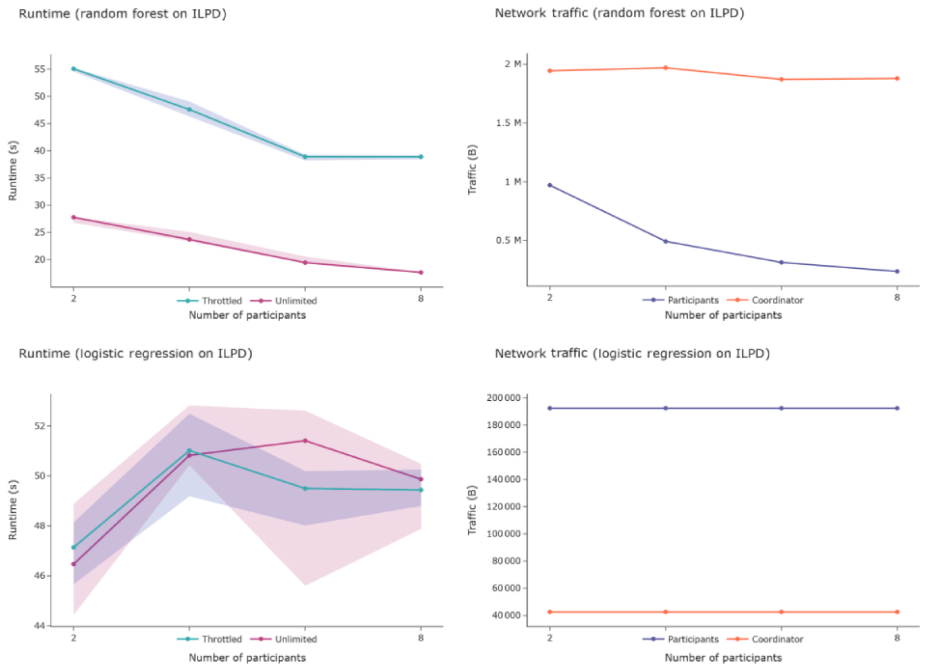
Multiple executions with varying numbers of clients were performed to assess the scalability of the FeatureCloud platform and the federated methods. RF and linear regression classifiers were chosen as the iterative and noniterative methods, respectively, and both were applied to the Indian Liver Patient Dataset. Both were tested with 2, 4, 6, and 8 clients and the same number of samples to ensure comparability across the executions. To investigate the impact of network bandwidth on runtime, all executions were performed on a normal and throttled internet connection with a maximum transmission of 100 kB per second.

Figure 5 shows that runtime mildly increases for logistic regression but decreases for RF. This is because the logistic regression models are of equal size for all clients, whereas the size of the RF models depends on the number of trees. In our implementation of federated RF, the global model is of a fixed size (100 trees), which means that each client contributes a portion that decreases with a higher number of participants. The throttling bandwidth significantly increases the runtime for RF but leaves the runtime for logistic regression almost unaffected. This is because the transmitted data for RF are more extensive and come in 1 chunk, whereas logistic regression requires approximately 10 iterations, each exchanging a few parameters. The centralized versions take 2 to 3 seconds to complete for both logistic regression and RF, implying that their federated versions take 10 to 20 times longer to complete.

In this setting, an increasing number of participating parties has a weak impact on the duration of the aggregation part for these methods, compared with the total runtime. The local computations occur in parallel such that an increasing number of participants does not have a huge impact. However, because the aggregation step cannot be completed before all participants send their models, the runtime of each aggregation step depends on the slowest participant, which poses a potential problem for large federations. FeatureCloud primarily focuses on being used in a tightly regulated medical research environment. Therefore, there is currently no automatic “matchmaking” in place, but all participants must join each project actively. In this context, running an analysis with data sets of >8 participants is still an uncommon scenario. To demonstrate its scalability and robustness for more sophisticated scenarios, we evaluated the

FeatureCloud platform using the logistic regression application for 1, 5, 10, 15, 20, 25, and 30 clients on simulated data, with each client containing 1000 samples and 1, 5, and 10 iterations. Our analysis shows that the FeatureCloud platform is also computationally suitable for larger numbers of clients and higher numbers of iterations, confirming the results of our runtime analysis for a small number of clients (Figure S4 in Multimedia Appendix 1).

Figure 5. Runtime and network traffic. The left plots show runtime for unlimited and throttled connections, the right plots show network traffic for the coordinator and participants evaluated on the ILPD. The lines represent the median values measured over 10 executions. The areas show the 25% and 75% quartiles to illustrate variance across the executions. ILPD: Indian Liver Patient Dataset; s:second; B: byte; M: million.



Discussion

In this section, we summarize our main findings and provide a discussion about its comparison with prior work, its limitations, the potential for future work, and conclusions of our work.

Summary of Results

In this study, we presented the FeatureCloud platform, a comprehensive platform for the application and development of privacy-preserving FL workflows in biomedicine and beyond. Through its high generalization, it allows the application of various ML workflows to a variety of data types. In addition, it offers prebuilt solutions for common-use cases in the form of applications in the AI store or application templates for developers. The concept of freely composing applications in a workflow is challenging because of the need for a standard data format, which is not always available and can reduce flexibility. The same applies to the initial data, which need to be provided in a form that is processable and understandable by the desired application. As FL adaptation is still in its early stages, it is necessary to understand which functionality and types of data will be used, which ML techniques prove to be most prevalent in federated settings, and which challenges arise when using the platform. Therefore, several assumptions can be made in advance.

Comparison With Prior Work

One main goal of FeatureCloud was to keep the platform as flexible and extensible as possible, to align new functionality closely to the demand of its users. The possibility of integrating additional PETs, such as DP or additive secret sharing, on the application layer of the API demonstrates the versatility of this approach. Although the current implementation of additive secret sharing has a quadratic increase in network traffic, it shows that flexible communication can be achieved through asymmetrical encryption and can serve as a blueprint for similar scenarios and future developments.

The prediction performance of our FL workflows is consistent with the current research, with some performing equally well compared with the central implementations (linear and logistic regression and normalization) or highly similar (RF). Computational and communication overheads are acceptable for an ordinary FL. In our opinion, it plays a smaller role than the additional overhead related to human-to-human coordination of federated projects. We demonstrated that the currently available applications and the platform scale well for up to 8 participants.

The main novelty, in contrast to prior work, is the high flexibility of the AI store, ranging from prebuilt task-centered applications, such as GWAS, to generic method-centered applications, such as RF. Therefore, we address a broad spectrum of end users and developers. Less experienced users without deeper methodological or statistical knowledge benefit from the ease of use of a task-centered application. Advanced users can tailor the workflow to their needs. In contrast, application developers can use our API to develop FL applications that can be easily deployed into the AI store and reach a broad user base. They are incentivized to build their applications to be compatible with existing ones (eg, a new AI method that processes data preprocessed by an existing normalization application) to maximize their utility. Thus, the FeatureCloud AI store aims to become an ecosystem for FL, driving collaborative research.

Limitations

In addition to the huge potential of FeatureCloud, some issues still need to be addressed. Our secure aggregation approach, directly implemented into the developer API, only applies to ≥3 participants. Its application on workflows with only 2 participants would allow the coordinator to reveal the local parameters of the other participant and therefore has no benefit. In addition, as it is currently implemented, our additive secret-sharing approach only supports addition and multiplication and is, therefore, not applicable to more complex types of calculations. Although the open AI store accelerated the development and deployment of FL applications and workflows, it is the responsibility of the application developers to provide proof that their implementations provide accurate results. FeatureCloud certifies applications that provide a reasonable amount of privacy and security measures but cannot check the prediction quality of every application. However, through its open-source design, the community can exchange experiences, provide feedback, and enhance applications and algorithms to keep them up to date with the current state of the art.

Future Work

The generic and extendable design of FeatureCloud makes it highly interesting for future studies. FeatureCloud envisions being driven by an emerging community whose features are closely aligned to their needs. As FeatureCloud is entirely open source, it can be quickly maintained and extended and it can accelerate the development, deployment, and execution of privacy-preserving FL workflows in biomedicine and other areas. FeatureCloud applications can be developed by anyone using the developer API and easy-to-start templates. One part could focus on integrating more PETs into the API for the application developers to ease their use and increase adoption in federated algorithms. Although FeatureCloud already integrates an additive secret-sharing scheme, there are many more PETs, such as DP or HE schemes, that can be implemented. Other potential enhancements could focus on nonlinear workflows, the integration of the AIMe registry [43] into the certification process of FeatureCloud applications, and reducing Docker dependency by also supporting other secure containerization systems such as Singularity [44]. To address the problem of data harmonization and preprocessing of different formats at different sites, it may be useful to add a federated database with a common ontology to the FeatureCloud controller [45]. Through this, the problem of different data formats between sites is solved, as the input data for workflows can be directly created from the database. Integrating local data into this database can be performed using predefined Extract-Transform-Load scripts for the most common data formats and standards.

Conclusions

In conclusion, FeatureCloud provides an all-in-one platform for privacy-preserving FL. In contrast to other FL frameworks, FeatureCloud considers every aspect of FL from development and deployment to the execution and project planning of federated analyses. Furthermore, it is highly generic to support all types of algorithms and is not restricted to only DL or a certain application. Thus, we believe that it has a huge potential to accelerate the development of FL workflows and the application of federated analyses in biomedicine.

Acknowledgments

The FeatureCloud project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement 826078. This publication reflects only the authors' view, and the European Commission is not responsible for any use that may be made of the information it contains. This work was further supported by the German Federal Ministry of Education and Research (BMBF)–funded German Network for Bioinformatics Infrastructure (de.NBI) Cloud within the de.NBI (031A532B, 031A533A, 031A533B, 031A534A, 031A535A, 031A537A, 031A537B, 031A537C, 031A537D, and 031A538A). This work was further supported by the German BMBF within the framework of "CLINSPECT-M" (grant FKZ161L0214A). This work was partly developed as part of the Center for Female Artificial Intelligence project funded by the German BMBF under grant 01IS21079. This work was further funded by the German BMBF under grant 16DTM100A. JKP was funded by the Bavarian State Ministry of Education and the Arts in the framework of the Bavarian Research Institute for Digital Transformation (bidt, grant LipiTUM). This study uses data from the Survey of Health, Aging and Retirement in Europe (SHARE) Wave 8 (refer to the study by Börsch-Supan et al [38] for methodological details). The SHARE data collection has been funded by the European Commission, Directorate-General for Research and Innovation (DG RTD) through FP5 (QLK6-CT-2001-00360), FP6 (SHARE-I3: RII-CT-2006-062193, COMPARE: CIT5-CT-2005-028857, and SHARELIFE: CIT4-CT-2006-028812), FP7 (SHARE-PREP: GA N°211909, SHARE-LEAP: GA N°227822, SHARE M4: GA N°261982, and Data Service Infrastructure for the Social Sciences and Humanities (DASISH): GA N°283646), and Horizon 2020 (SHARE-DEV3: GA N°676536, SHARE-COHESION: GA N°870628, SERISS: GA N°654221, SSHOC: GA N°823782, and SHARE-COVID19: GA N°101015924); and by DG Employment, Social Affairs & Inclusion through VS 2015/0195, VS 2016/0135, VS 2018/0285, VS 2019/0332, and VS 2020/0313. Additional funding from the German Ministry of Education and Research, the Max Planck Society for the Advancement of Science, the US National Institute on Aging (U01_AG09740-13S2, P01_AG005842, P01_AG08291, P30_AG12815, R21_AG025169, Y1-AG-4553-01, IAG_BSR06-11, OGHA_04-064, HHSN271201300071C, and RAG052527A), and various national funding sources is gratefully acknowledged [46].

Data Availability

The Survey of Health, Aging and Retirement in Europe (SHARE) data are distributed by SHARE-European Research Infrastructure Consortium (ERIC) to registered users through the SHARE Research Data Center. We used only data from the 8 waves [38]. Except for the SHARE data, all our data sets, including the Indian Liver Patient Dataset [34], Breast Invasive Carcinoma data set [35], Boston data set [37], and Diabetes data set [36], and scripts used for our evaluation results are available in our GitHub repository [47]. To increase interpretation and reproducibility, we followed the minimum information about clinical artificial intelligence modeling (ML-CLAIM) reporting standard (Norgeot et al [48]). The filled-out ML-CLAIM clinical checklist is also available in our GitHub repository.

Conflicts of Interest

None declared.

Multimedia Appendix 1

Additional information containing figures and descriptions related to software architecture and implementation.
[DOCX File , 1262 KB-Multimedia Appendix 1]

References

1. Mamoshina P, Vieira A, Putin E, Zhavoronkov A. Applications of deep learning in biomedicine. Mol Pharm 2016 May 02;13(5):1445-1454 [doi: 10.1021/acs.molpharmaceut.5b00982] [Medline: 27007977]
2. Yu KH, Beam AL, Kohane IS. Artificial intelligence in healthcare. Nat Biomed Eng 2018 Oct;2(10):719-731 [doi: 10.1038/s41551-018-0305-z] [Medline: 31015651]
3. Malle B, Giuliani N, Kieseberg P, Holzinger A. The more the merrier - federated learning from local sphere recommendations. In: Proceedings of the 1st International Cross-Domain Conference on Machine Learning and Knowledge Extraction. 2017 Presented at: CD-MAKE '17; August 29-September 1, 2017; Reggio, Italy p. 367-373 URL: https://link.springer.com/chapter/10.1007/978-3-319-66808-6_24 [doi: 10.1007/978-3-319-66808-6_24]
4. Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, et al. Corrigendum: dermatologist-level classification of skin cancer with deep neural networks. Nature 2017 Jun 28;546(7660):686 [doi: 10.1038/nature22985] [Medline: 28658222]
5. Chan HC, Shan H, Dahoun T, Vogel H, Yuan S. Advancing drug discovery via artificial intelligence. Trends Pharmacol Sci 2019 Oct;40(10):801 [doi: 10.1016/j.tips.2019.07.013] [Medline: 31451243]
6. McMahan HB, Moore E, Ramage D, Hampson S, Arcas BA. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. 2017 Presented at: AISTATS '17; April 20-22, 2017; Ft. Lauderdale, FL, USA p. 1273-1282 URL: http://proceedings.mlr.press/v54/mcmahan17a?ref=https://githubhelp.com
7. Kairouz P, McMahan HB, Avent B, Bellet A, Bennis M, Nitin Bhagoji A, et al. Advances and open problems in federated learning. Found Trends Mach Learn 2021 Jun 23;14(1–2):1-210 [FREE Full text] [doi: 10.1561/2200000083]
8. Tomsett R, Chan KS, Chakraborty S. Model poisoning attacks against distributed machine learning systems. In: Proceedings of the 2019 SPIE Defense and Commercial Sensing. 2019 Presented at: SDCS '19; April 14-18, 2019; Baltimore, MD, USA URL: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11006.toc [doi: 10.1117/12.2520275]
9. Usynin D, Ziller A, Makowski M, Braren R, Rueckert D, Glocker B, et al. Adversarial interference and its mitigations in privacy-preserving collaborative machine learning. Nat Mach Intell 2021 Sep 17;3(9):749-758 [FREE Full text] [doi: 10.1038/s42256-021-00390-3]
10. Acar A, Aksu H, Uluagac AS, Conti M. A survey on homomorphic encryption schemes: theory and implementation. ACM Comput Surv 2019 Jul 31;51(4):79 [FREE Full text] [doi: 10.1145/3214303]
11. Bonawitz KA, Ivanov V, Kreuter B, Marcedone A, McMahan HB, Patel S, et al. Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017 Presented at: CCS '17; October 30-November 3, 2017; Dallas, TA, USA p. 1175-1191 URL: https://dl.acm.org/doi/10.1145/3133956.3133982 [doi: 10.1145/3133956.3133982]
12. Dwork C, McSherry F, Nissim K, Smith A. Calibrating noise to sensitivity in private data analysis. In: Proceedings of the 3rd Theory of Cryptography Conference. 2006 Presented at: TCC '06; March 4-7, 2006; New York, NY, USA p. 265-284 URL: https://link.springer.com/chapter/10.1007/11681878_14 [doi: 10.1007/11681878_14]
13. Dwork C, Roth A. The algorithmic foundations of differential privacy. Found Trends Theor Comput Sci 2014 Aug 11;9(3-4):211-407 [FREE Full text] [doi: 10.1561/0400000042]

14. Nilsson A, Smith S, Ulm G, Gustavsson E, Jirstrand M. A performance evaluation of federated learning algorithms. In: Proceedings of the 2nd Workshop on Distributed Infrastructures for Deep Learning. 2018 Presented at: DIDL '18; December 10-11, 2018; Rennes, France p. 1-8 URL: https://dl.acm.org/doi/10.1145/3286490.3286559 [doi: 10.1145/3286490.3286559]
15. Lee GH, Shin SY. Federated learning on clinical benchmark data: performance assessment. J Med Internet Res 2020 Oct 26;22(10):e20891 [FREE Full text] [doi: 10.2196/20891] [Medline: 33104011]
16. Zolotareva O, Nasirigerdeh R, Matschinske J, Torkzadehmahani R, Bakhtiari M, Frisch T, et al. Flimma: a federated and privacy-aware tool for differential gene expression analysis. Genome Biol 2021 Dec 14;22(1):338 [FREE Full text] [doi: 10.1186/s13059-021-02553-2] [Medline: 34906207]
17. Nasirigerdeh R, Torkzadehmahani R, Matschinske J, Frisch T, List M, Späth J, et al. sPLINK: a hybrid federated tool as a robust alternative to meta-analysis in genome-wide association studies. Genome Biol 2022 Jan 24;23(1):32 [FREE Full text] [doi: 10.1186/s13059-021-02562-1] [Medline: 35073941]
18. Späth J, Matschinske J, Kamanu FK, Murphy SA, Zolotareva O, Bakhtiari M, et al. Privacy-aware multi-institutional time-to-event studies. PLOS Digit Health 2022 Sep;1(9):e0000101 [FREE Full text] [doi: 10.1371/journal.pdig.0000101] [Medline: 36812603]
19. Ryffel T, Trask A, Dahl M, Wagner B, Mancuso J, Rueckert D, et al. A generic framework for privacy preserving deep learning. arXiv Preprint posted online on November 9, 2018 [FREE Full text] [doi: 10.48550/arXiv.1811.04017]
20. Konczyk J. Federated Learning with TensorFlow. Birmingham, UK: Packt Publishing; 2019.
21. Train on the edge with federated learning. XayNet. URL: https://www.xaynet.dev/ [accessed 2023-05-12]
22. Yang L, Tao F, Tianjian C, Qian X, Qiang Y. An industrial grade federated learning framework. The Journal of Machine Learning Research 2021 Aug:10320-10325 [FREE Full text]
23. Gazula H, Kelly R, Romero J, Verner E, Baker BT, Silva RF, et al. COINSTAC: collaborative informatics and neuroimaging suite toolkit for anonymous computation. J Open Source Softw 2020 Oct 25;5(54):2166-2169 [FREE Full text] [doi: 10.21105/joss.02166]
24. Silva S, Altmann A, Gutman B, Lorenzi M. Fed-BioMed: a general open-source frontend framework for federated learning in healthcare. In: Proceedings of the 2nd MICCAI Workshop on Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning. 2020 Presented at: DART '20 and DCL '20; October 4–8, 2020; Lima, Peru p. 201-210 URL: https://link.springer.com/chapter/10.1007/978-3-030-60548-3_20 [doi: 10.1007/978-3-030-60548-3_20]
25. Owkin. URL: https://owkin.com/ [accessed 2023-05-12]
26. Melloddy. URL: https://www.melloddy.eu/ [accessed 2023-05-12]
27. FeatureCloud - Privacy-Preserving AI. URL: https://featurecloud.ai [accessed 2023-06-02]
28. FeatureCloud AI developer API (1.1.0). FeatureCloud AI. URL: https://featurecloud.ai/assets/api/redoc-static.html [accessed 2023-05-12]
29. Lyu L, Yu H, Yang Q. Threats to federated learning: a survey. arXiv Preprint posted online on March 4, 2020 [FREE Full text] [doi: 10.48550/arXiv.2003.02133]
30. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: machine learning in Python. J Mach Learn Res 2011;12(85):2825-2830 [FREE Full text]
31. Buitinck L, Louppe G, Blondel M, Pedregosa F, Mueller A, Grisel O, et al. API design for machine learning software: experiences from the scikit-learn project. In: Proceedings of the 2013 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases. 2013 Presented at: ECML PKDD '13; September 23-27, 2013; Prague, Czech Republic p. 108-122
32. Hauschild AC, Lemanczyk M, Matschinske J, Frisch T, Zolotareva O, Holzinger A, et al. Federated random forests can improve local performance of predictive models for various healthcare applications. Bioinformatics 2022 Apr 12;38(8):2278-2286 [doi: 10.1093/bioinformatics/btac065] [Medline: 35139148]
33. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: an imperative style, high-performance deep learning library. In: Proceedings of the 32nd Conference on Neural Information Processing Systems. 2019 Presented at: NeurIPS '19; December 8-14, 2019; Vancouver, Canada URL: https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html
34. UC Irvine Machine Learning Repository. URL: https://doi.org/10.24432/C5D02C [accessed 2023-05-12]
35. Street WN, Wolberg WH, Mangasarian OL. Nuclear feature extraction for breast tumor diagnosis. In: Proceedings of the 1993 IS&T/SPIE'S Symposium on Symposium on Electronic Imaging: Science and Technology. 1993 Presented at: IS&T '93; January 31-February 5, 1993; San Jose, CA, USA URL: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/1905/1/Nuclear-feature-extraction-for-breast-tumor-diagnosis/10.1117/12.148698.short [doi: 10.1117/12.148698]
36. Efron B, Hastie T, Johnstone I, Tibshirani R. Least angle regression. Ann Stat 2004 Apr;32(2):407-451 [FREE Full text] [doi: 10.1214/009053604000000067]
37. Harrison Jr D, Rubinfeld DL. Hedonic housing prices and the demand for clean air. J Environ Econ Manage 1978 Mar;5(1):81-102 [FREE Full text] [doi: 10.1016/0095-0696(78)90006-2]
38. Börsch-Supan A. Survey of Health, Ageing and Retirement in Europe (SHARE) Wave 8. COVID-19 Survey 1. Release version: 8.0.0. Survey of Health, Ageing and Retirement in Europe (SHARE). 2022 Feb 10. URL: https://share-eric.eu/data/data-set-details/share-corona-survey-1 [accessed 2023-05-12]

39. Sheller MJ, Edwards B, Reina GA, Martin J, Pati S, Kotrotsou A, et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. Sci Rep 2020 Jul 28;10(1):12598 [FREE Full text] [doi: 10.1038/s41598-020-69250-1] [Medline: 32724046]
40. McMahan B, Ramage D. Federated learning: collaborative machine learning without centralized training data. Google Research. 2017 Apr 06. URL: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html [accessed 2023-05-12]
41. Loprinzi CL, Laurie JA, Wieand HS, Krook JE, Novotny PJ, Kugler JW, et al. Prospective evaluation of prognostic variables from patient-completed questionnaires. north central cancer treatment group. J Clin Oncol 1994 Mar;12(3):601-607 [doi: 10.1200/JCO.1994.12.3.601] [Medline: 8120560]
42. Liu J, Lichtenberg T, Hoadley KA, Poisson LM, Lazar AJ, Cherniack AD, Cancer Genome Atlas Research Network, et al. An integrated TCGA pan-cancer clinical data resource to drive high-quality survival outcome analytics. Cell 2018 Apr 05;173(2):400-16.e11 [FREE Full text] [doi: 10.1016/j.cell.2018.02.052] [Medline: 29625055]
43. Matschinske J, Alcaraz N, Benis A, Golebiewski M, Grimm DG, Heumos L, et al. The AIMe registry for artificial intelligence in biomedical research. Nat Methods 2021 Oct;18(10):1128-1131 [FREE Full text] [doi: 10.1038/s41592-021-01241-0] [Medline: 34433960]
44. Kurtzer GM, Sochat V, Bauer MW. Singularity: scientific containers for mobility of compute. PLoS One 2017 May 11;12(5):e0177459 [FREE Full text] [doi: 10.1371/journal.pone.0177459] [Medline: 28494014]
45. Sheth AP, Larson JA. Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Comput Surv 1990 Sep 01;22(3):183-236 [FREE Full text] [doi: 10.1145/96602.96604]
46. Survey of Health, Aging and Retirement in Europe. URL: https://share-eric.eu/ [accessed 2023-05-12]
47. Matschinske J, Späth J. Evaluation - FeatureCloud. GitHub. URL: https://github.com/FeatureCloud/evaluation [accessed 2023-05-12]
48. Norgeot B, Quer G, Beaulieu-Jones BK, Torkamani A, Dias R, Gianfrancesco M, et al. Minimum information about clinical artificial intelligence modeling: the MI-CLAIM checklist. Nat Med 2020 Sep;26(9):1320-1324 [FREE Full text] [doi: 10.1038/s41591-020-1041-y] [Medline: 32908275]

Abbreviations

AI: artificial intelligence
API: application programming interface
CV: cross-validation
DL: deep learning
DP: differential privacy
FL: federated learning
GUI: graphical user interface
GWAS: genome-wide association studies
HE: homomorphic encryption
ML: machine learning
PET: privacy-enhancing technology
RF: random forest

Edited by T Leung; submitted 12.09.22; peer-reviewed by A Brauneck, P Mitrovic, A Wang, Z Zrubka, J Seth; comments to author 20.12.22; revised version received 13.01.23; accepted 26.02.23; published 12.07.23

Please cite as:
Matschinske J, Späth J, Bakhtiari M, Probul N, Kazemi Majdabadi MM, Nasirigerdeh R, Torkzadehmahani R, Hartebrodt A, Orban BA, Fejér SJ, Zolotareva O, Das S, Baumbach L, Pauling JK, Tomašević O, Bihari B, Bloice M, Donner NC, Fdhila W, Frisch T, Hauschild AC, Heider D, Holzinger A, Hötzendorfer W, Hospes J, Kacprowski T, Kastelitz M, List M, Mayer R, Moga M, Müller H, Pustozerova A, Röttger R, Saak CC, Saranti A, Schmidt HHHW, Tschohl C, Wenke NK, Baumbach J
The FeatureCloud Platform for Federated Learning in Biomedicine: Unified Approach
J Med Internet Res 2023;25:e42621
URL: https://www.jmir.org/2023/1/e42621
doi: 10.2196/42621
PMID:

D

AIMe

Julian Matschinske et al. “The AIMe registry for artificial intelligence in biomedical research”. In: *Nature Methods* 18.10 (2021), pp. 1128–1131. ISSN: 1548-7105. DOI: 10.1038/s41592-021-01241-0. URL: https://doi.org/10.1038/s41592-021-01241-0

The AIMe registry for artificial intelligence in biomedical research

We present the AIMe registry, a community-driven reporting platform for AI in biomedicine. It aims to enhance the accessibility, reproducibility and usability of biomedical AI models, and allows future revisions by the community.

Julian Matschinske, Nicolas Alcaraz, Arriel Benis, Martin Golebiewski, Dominik G. Grimm, Lukas Heumos, Tim Kacprowski, Olga Lazareva, Markus List, Zakaria Louadi, Josch K. Pauling, Nico Pfeifer, Richard Röttger, Veit Schwämmle, Gregor Sturm, Alberto Traverso, Kristel Van Steen, Martiela Vaz de Freitas, Gerda Cristal Villalba Silva, Leonard Wee, Nina K. Wenke, Massimiliano Zanin, Olga Zolotareva, Jan Baumbach and David B. Blumenthal

Overcoming the reporting deficit in biomedical AI

The past two decades have seen massive advances and rapidly declining costs in high-throughput technologies that produce enormous amounts of biomedical data. This development has been accompanied by breakthroughs in the field of artificial intelligence (AI). With the help of AI, high-dimensional data can now be modeled in a mathematically robust and accurate way, which has led to numerous applications in biomedical research. For example, AI has been successfully used to determine particles in cryogenic electron microscopy projection images[1], to infer proteins from mass spectrometry data[2], to conduct exploratory analysis of single-cell data[3] and to predict incipient circulatory failure in the intensive care unit[4].

In spite of the obvious potential of AI in biomedical research, we observe trends that are detrimental to the development of new, improved AI methods and also constitute major hurdles in applying biomedical AIs in basic or translational biomedical research. Best practices of machine learning are not always adhered to, and often only selected aspects of the AI models and their evaluation are reported[5]. Because of this, the decisions of biomedical AIs are often opaque, difficult to explain and not fully reproducible[6–12]. In clinical research in particular, it is crucial to instill trust in AI models and to report on them in an explicit and transparent fashion that adheres to commonly used standards[5,12,13]. Or, as put by Davenport et al.[10]: "For widespread adoption to take place, AI systems must be approved by regulators [and] standardised to a sufficient degree [...]."

To address this problem, several checklists and guidelines for reporting AI methodology and results in biomedical and clinical research have been proposed recently[14–21]. This, however, is only a first step toward resolving the reporting deficit because mere guidelines and checklists do not make biomedical AI reports accessible to the scientific community. Moreover, guidelines and checklists provide no practical means to identify biomedical AIs that do not adhere to the recommended best practices. We believe that what is needed is a community-driven registry that allows authors of new biomedical AIs to easily generate accessible, browsable and citable reports that can be scrutinized and reviewed by the scientific community.

In view of this, we present the AIMe registry for artificial intelligence in biomedical research: https://aime-registry.org. It consists of a user-friendly web service that guides authors of new AIs through the AIMe standard, a generic minimal information standard that allows reporting of any biomedical AI system. Once the AIMe standard has been reported, a database entry and an HTML report along with a unique AIMe identifier are created. The latter serves to keep the entry openly accessible and can be disseminated by the authors, for example by inclusion in a manuscript.

We have designed the AIMe registry as a community-driven platform for AI in biomedicine. It allows users to raise issues related to existing entries if they have doubts concerning their adequacy or informativeness. Moreover, we will update the reported AIMe standard each year based on feedback from the scientific community. Interested researchers are invited to join the AIMe steering committee, which consolidates the feedback into an updated version of the AIMe standard.

The remainder of this paper is organized as follows: first, we present the first version of the AIMe standard. We then present the AIMe registry and detail how it incorporates feedback from the scientific community. In the section on governance, we formulate the mission of the AIMe initiative and provide details on the structure of the organization as well as the yearly revision process. Finally, we present conclusions in the last section of the paper.

The AIMe2021 standard

Here, we present the first version of the AIMe standard, the AIMe2021 standard. To design the AIMe2021 standard, we proceeded as follows: as a first step, the initial AIMe steering committee composed of the co-authors affiliated with the Chair of Experimental Bioinformatics of the Technical University of Munich, with the University of Hamburg and with the Department of Mathematics and Computer Science of the University of Southern Denmark compiled a draft of the AIMe2021 standard. We then shared a call for contributions via social media and mailing lists, in which we asked interested researchers to provide feedback and to join the AIMe steering committee. All other co-authors of this paper responded to this call. Finally, we consolidated the feedback into the AIMe2021 standard via a collaborative document editing effort coordinated by the first and last authors of this paper.

The AIMe2021 standard is divided into five sections: Metadata, Purpose, Data, Method and Reproducibility. The formal YAML specification of the AIMe2021 standard is available at https://aime-registry.org/specification/. Examples of AIMe reports are available at https://aime-registry.org/database/.

Metadata. The AIMe standard asks authors of biomedical AIs to report basic metadata for their methods (Supplementary Fig. 1). In a first series of questions, the authors are asked to provide metadata about the paper and the corresponding author(s) (MD.1–MD.6). They should also disclose funding sources (MD.7) and specify whether the entry should appear among the results when searching the AIMe database (MD.8). Temporarily excluding a report from the search might be useful if the reported AI has not been published yet. However, all created reports are always publicly accessible via their unique AIMe identifiers and automatically become searchable once a paper ID or URL is added in (MD.4). Moreover, authors can upload other checklists or reports they might have filled in (MD.9) (e.g., the MI-CLAIM checklist[18]).

Purpose. In this section, authors are requested to elaborated on the purpose of their biomedical AI (Supplementary Fig. 2). They should state what their AI is designed to learn or predict (P.1) and whether it predicts a surrogate marker rather than a directly measurable response variable (P.2). Furthermore, AIMe requests that the authors specify a category to which their AI problem belongs (P.3). Typical categories are classification (assign discrete labels to all items), regression (predict a real-valued number for all items), clustering (partition a set of items into subsets of homogeneous groups), ranking (learn an ordering for a set of items), dimensionality reduction (compress all items' initial high-dimensional representations) and data generation.

Data. In biomedical research, it is common practice to include multiple datasets in the same pipeline to gain insights into complex biological processes. The AIMe standard therefore ask authors of new AIs to add separately each dataset employed and then characterize it in terms of data availability, possible biases and applied transformations (Supplementary Fig. 3).

For each dataset x, the authors should report the type of data (D.x.1)—e.g., expression, methylation or phenotype data. For instance, if an AI uses gene expression data to predict the body mass index (BMI), then the authors should add one dataset for the BMI data and a separate dataset for the expression data. Because there are often no gold-standard data for biomedical AI problems, new AIs are often evaluated on simulated data. In view of this, AIMe asks the authors to specify whether their data is real or simulated (D.x.2). Moreover, the authors should report whether the dataset is publicly available (D.x.3) and specify whether it was used for training the AI method (D.x.4).

Biomedical data are often subject to various biases[22–24]. Even if these biases can be addressed appropriately, readers should be aware of them to avoid possible misinterpretations. Therefore, AIMe asks the authors if, and if so how, they have checked whether their data is subject to biases (D.x.5). AIMe also requests that authors report the dimensionality of their data, i.e., specify the number of samples and features (D.x.6). This is especially important because high-dimensional data often exhibits multicollinearity and sparsity[25], which in turn tends to negatively affect the efficiency of AI systems[26] and often leads to overfitting. As most AI methods are not scale invariant, the data usually need to be normalized during pre-processing. Consequently, AIMe asks the authors if, and if so how, they have pre-processed their data (D.x.7).

Method. The next series of questions addresses the specific AI methods (Supplementary Fig. 4). The first question AIMe asks in this regard is which AI or mathematical methods (e.g., logistic regression, random-forest classification, deep neural networks, ant colony optimization, genetic programming) were used (M.1). Next, the authors must specify how they selected the method's hyper-parameters (e.g., number of trees in random-forest models) (M.2). This is important because hyper-parameters typically have an enormous impact on method performance but are often not reported in the publications[27,28].

The AIMe standard also contains questions related to the validation and verification of the AI method used. The initial questions ask which test metrics (e.g., Gini coefficient, running time, mean squared error) were used to evaluate the method (M.3). Later, the authors are asked to report how they prevented overfitting—i.e., how they ensured that their AI model does not merely memorize the training data but can generalize to unseen, independent data (M.4). Overfitting can be prevented by using various techniques such as ensemble learning, cross-validation and regularization.

Moreover, AIMe asks the authors to clarify whether they have checked if there are trigger situations that induce their method to fail in its task (M.5). A possible trigger situation is the presence of confounding factors: i.e., variables that influence both the model input and output variables and, as a result, potentially distort the results[29]. The authors are also required to report whether they have checked if randomized steps in their AI affect the stability of the results (M.6). Moreover, they should specify whether they have compared their AI method to simple baseline models (M.7), as well as to state-of-the-art competitors (M.8).

Reproducibility. The last four questions help increase the reproducibility of the experiments that validate the proposed AI (Supplementary Fig. 5). First, the authors are asked whether they provide all means to easily re-run their AI, e.g., by providing conda or pip packages, Dockerfiles, language-specific build system files or detailed README files (R.1). They are also required to provide information about the source code availability of the main AI method, the data simulator (if applicable) and the pre-processing pipeline (R.2). Next, AIMe asks the authors whether they provide a pre-trained model, e.g., by uploading it to repositories such as Kipoi[30] (R.3). Finally, the authors should elaborate on the software and hardware environments required to run their AI method (R.4).

The AIMe registry

The AIMe registry provides three main services: add a new report, query the database and contribute to AIMe (Fig. 1).

Creating a new report. During the creation of a new report, AIMe guides authors of new AIs through the current version of the AIMe standard (as discussed earlier in the description of the standard). To ensure that the standard is generically applicable, the system allows authors to skip some of the questions if the information required to answer them is not available. To encourage authors to skip as few questions as possible, a validation and a reproducibility score are computed for each report. The scores range from 0 to 10: the higher the scores, the fewer questions concerning validation and reproducibility of the reported AI have been skipped. Authors of AIMe reports can edit previously created reports at any time, but all previous versions will remain visible in the HTML report.

Querying the AIMe database. Users can find existing reports in the AIMe database via their unique AIMe IDs, or search the database for reports of interest via full-text or keyword search. If users identify answers in the reports they deem inappropriate, uninformative or misleading, they can raise issues after providing their personal information (name and email address). The reports' corresponding authors can reply to the issues, and they are allowed two weeks to notify AIMe's executive board about offensive or otherwise inappropriate issues. If the authors raise no complaints or the executive board classifies the complaints as

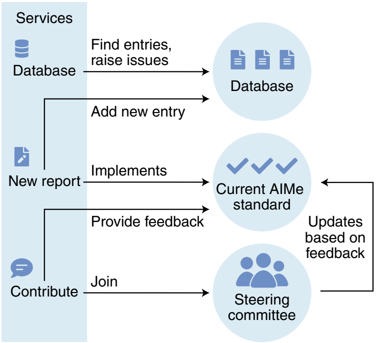


Fig. 1 | Overview of the AIMe registry. Users can create a new report, query the database to find existing entries and raise issues, and contribute to AIMe by joining the AIMe steering committee or providing feedback that will be incorporated into the next version of the standard.

unwarranted, the issues and the personal information of the users who raised them, as well as the authors' replies, are appended to the reports. Note that, because AIMe is committed to open peer review, issues that are due to misunderstandings but do not contain any insulting or off-topic elements will not be classified as inappropriate. Hence, by raising issues, members of the scientific community can review existing AIMe reports. This is important because it helps reveal reports in which questions are answered inadequately.

Contributing to AIMe. The Contribute functionality of the AIMe registry allows interested members of the scientific community to actively shape future versions of the AIMe standard by providing suggestions for improvement and requesting membership in the steering committee (as discussed below in the section on governance). All versions of the AIMe standard are formally specified in a YAML-based language. This ensures that the structure of old reports will remain well defined even after the current standard is updated at the beginning of each year. The YAML specifications are available at https://aime-registry.org/specification/.

AIMe governance

Mission. The mission of the AIMe initiative is to promote open, transparent and reproducible biomedical AI research. For this, we provide a community-driven registry, where biomedical AI researchers can report their AI models in a standardized fashion, search the AIMe database for AI systems related to their work and comment on existing reports as well as the AIMe standard itself (see "The AIMe Registry" above). The AIMe initiative is committed to the following principles of open science[31,32].

- **Open peer review:** Registry users who raise an issue on an existing entry are required to provide personal information, and all issues are appended to the reports and hence visible in the database (unless they are deemed by the AIMe executive board to be offensive or off-topic).
- **Open methodology:** The openly accessible YAML specification of the AIMe standard clearly states how the reproducibility and validation scores are computed based on the answers provided in the reports.
- **Openness to diversity of knowledge:** Biomedical AI researchers with diverse professional and cultural backgrounds are invited to join the steering committee and help shaping future versions of the AIMe standard.
- **Open source code:** The source code of the AIMe registry is freely available under the terms of a widely used open source license (see "Code availability" below).

Organization structure. There are three different roles in which scientists from the field of biomedical AI can participate in and contribute to the AIMe initiative: as a registry user, as a steering committee member and as an executive board member. These roles can be described as follows.

Registry user. Registry users can contribute to the AIMe initiative as described in the registry section above: i.e., by providing new entries, raising issues related to existing entries and commenting on the AIMe standard. Moreover, if they wish to play a more active role in the AIMe community, they can request membership in the steering committee.

Steering committee. The steering committee is responsible for maintaining and updating the specification of the AIMe standard. Its members are professional researchers working at the interface of AI, biomedicine, bioinformatics, computational biology and digital health. The founding steering committee consists of all co-authors of this paper. Supplementary Fig. 6 provides an overview of its members' professional backgrounds and expertises in biomedical AI. The founding steering committee covers all academic career levels from PhD student to full professor and reflects the internationality of the biomedical AI community in that its members work at research institutions in eight different countries in Europe, Asia, and the Americas.

Executive board. The executive board is responsible for coordinating the yearly reviews of the AIMe standard, for hosting and technical maintenance of the AIMe platform, for reviewing complaints on raised issues (i.e., deciding if issues qualify as offensive or off-topic) and for managing requests for membership in the steering committee. Such requests will be answered positively if the requester (a) provides plausible indication that they are a professional researcher with expertise in biomedical AI and (b) commits to actively participating in the yearly revision process. The founding executive board consists of the first and the senior authors of this paper.

Yearly revision process. Because biomedical AI is a rapidly evolving field, it is crucial that the AIMe standard continuously adapt to new developments in order to ensure that it will continue to reflect the needs of the research community. Therefore, AIMe foresees a yearly revision process, which is divided into two phases: a feedback phase from January 1 to September 30 of each year and a consolidation phase from October 1 to December 31.

During the feedback phase, users of the AIMe registry can provide feedback on the current version of the AIMe standard. Moreover, the steering committee members will actively reach out to influential representatives of the biomedical AI community and also submit their own proposals for improvements based on novel trends and developments in biomedical AI. During the consolidation phase, the steering committee will consolidate the collected feedback into a new version of the AIMe standard, coordinated by the executive board. On January 1, the new version of the AIMe standard will replace the old one.

Conclusions

AI is on the rise in biology and medicine and demonstrates utility in numerous application scenarios. However, basic information about data, methods and implementation of AI is often incomplete in the respective publications. This makes it difficult to judge, comprehensively compare and reproduce the results of biomedical AIs, a situation that, in turn, constitutes a major hurdle for developing new AI methods and for applying AI in research and practice. To address this problem and thereby improve the quality, reliability and reproducibility of biomedical AIs, we have developed the community-driven AIMe registry presented in this paper. This allows authors to easily register their AIs and assists researchers and practitioners in finding existing AI systems that are relevant for their application scenarios.

Code availability

The AIMe web service is available at https://aime-registry.org. The source code is available at https://github.com/aime-registry/aime-frontend/ and https://github.com/aime-registry/aime-backend/. It is licensed under the GNU General Public License, Version 3 (https://www.gnu.org/licenses/gpl-3.0.en.html). ❐

Julian Matschinske[1,2], Nicolas Alcaraz[3], Arriel Benis[4,5], Martin Golebiewski[6], Dominik G. Grimm[7,8,9], Lukas Heumos[10,11,12], Tim Kacprowski[1,13,14], Olga Lazareva[1], Markus List[1], Zakaria Louadi[1,2], Josch K. Pauling[15], Nico Pfeifer[16,17], Richard Röttger[18], Veit Schwämmle[19], Gregor Sturm[20], Alberto Traverso[21,22], Kristel Van Steen[23,24], Martiela Vaz de Freitas[25,26], Gerda Cristal Villalba Silva[25,26], Leonard Wee[21,22], Nina K. Wenke[1,2], Massimiliano Zanin[27], Olga Zolotareva[1,2], Jan Baumbach[2,18,29] ✉ and David B. Blumenthal[1,28,29] ✉

[1]Chair of Experimental Bioinformatics, TUM School of Life Sciences, Technical University of Munich, Freising, Germany. [2]Chair of Computational Systems Biology, University of Hamburg, Hamburg, Germany. [3]The Bioinformatics Centre, Department of Biology, University of Copenhagen, Copenhagen, Denmark. [4]Faculty of Industrial Engineering and Technology Management, Holon Institute of Technology, Holon, Israel. [5]Faculty of Digital Technologies in Medicine, Holon Institute of Technology, Holon, Israel. [6]HITS gGmbH, Heidelberg, Germany. [7]Technical University of Munich, Campus Straubing for Biotechnology and Sustainability, Bioinformatics, Straubing, Germany. [8]Weihenstephan-Triesdorf University of Applied Sciences, Straubing, Germany. [9]Technical University of Munich, Department of Informatics, Garching, Germany. [10]Institute of Computational Biology, Helmholtz Zentrum München, Munich, Germany. [11]Comprehensive Pneumology Center (CPC)/Institute of Lung Biology and Disease (ILBD), Helmholtz Zentrum München, Member of the German Center for Lung Research (DZL), Munich, Germany. [12]TUM School of Life Sciences, Technical University of Munich, Freising, Germany. [13]Division Data Science in Biomedicine, Peter L. Reichertz Institute for Medical Informatics, Technical University of Braunschweig and Hannover Medical School, Braunschweig, Germany. [14]Braunschweig Integrated Centre of Systems Biology (BRICS), Braunschweig, Germany. [15]LipiTUM, Chair of Experimental Bioinformatics, TUM School of Life Sciences, Technical University of Munich, Freising, Germany. [16]Chair of Methods in Medical Informatics, Department of Computer Science, University of Tübingen, Tübingen, Germany. [17]Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany. [18]Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark. [19]Department of Biochemistry and Molecular Biology, University of Southern Denmark, Odense, Denmark. [20]Biocenter, Institute of Bioinformatics, Medical University of Innsbruck, Innsbruck, Austria. [21]Department of Radiation Oncology, MAASTRO Clinic, Maastricht, The Netherlands. [22]School of Oncology, Maastricht University, Maastricht, The Netherlands. [23]BIO3-Systems Genetics, GIGA-R Medical Genomics, University of Liège, Liège, Belgium. [24]BIO3-Systems Medicine, Department of Human Genetics, KU Leuven, Leuven, Belgium. [25]Bioinformatics Core, Experimental Research Center, Hospital de Clínicas de Porto Alegre, Porto Alegre, Rio Grande do Sul, Brazil. [26]Postgraduate Program in Genetics and Molecular Biology, Federal University of Rio Grande do Sul, Porto Alegre, Brazil. [27]Instituto de Física Interdisciplinar y Sistemas Complejos IFISC (CSIC-UIB), Campus UIB, Palma de Mallorca, Spain. [28]Department Artificial Intelligence in Biomedical Engineering (AIBE), Friedrich-Alexander University Erlangen-Nürnberg (FAU), Erlangen, Germany. [29]These authors jointly supervised this work: Jan Baumbach, David B. Blumenthal.

✉*e-mail: jan.baumbach@uni-hamburg.de; david.b.blumenthal@fau.de*

Published online: 25 August 2021
https://doi.org/10.1038/s41592-021-01241-0

References

1. Bepler, T. et al. *Nat. Methods* **16**, 1153–1160 (2019).
2. Demichev, V., Messner, C. B., Vernardis, S. I., Lilley, K. S. & Ralser, M. *Nat. Methods* **17**, 41–44 (2020).
3. Amodio, M. et al. *Nat. Methods* **16**, 1139–1145 (2019).
4. Hyland, S. L. et al. *Nat. Med.* **26**, 364–373 (2020).
5. Liu, X. et al. *Lancet Digit. Health* **1**, e271–e297 (2019).
6. Baker, M. *Nature* **533**, 452–454 (2016).
7. Ioannidis, J. P. A. *PLoS Med.* **13**, e1002049 (2016).
8. Gundersen, O. E. & Kjensmo, S. In McIlraith, S. A. & Weinberger, K. Q. (eds.) *AAAI 2018*, 1644–1651 (AAAI Press, 2018).
9. Hutson, M. *Science* **359**, 725–726 (2018).
10. Davenport, T. & Kalakota, R. *Future Healthc. J.* **6**, 94–98 (2019).
11. Stupple, A., Singerman, D. & Celi, L. A. *NPJ Digit. Med.* **2**, 2 (2019).
12. Haibe-Kains, B. et al. *Nature* **586**, E14–E16 (2020).
13. Mateen, B. A., Liley, J., Denniston, A. K., Holmes, C. C. & Vollmer, S. *J. Nat. Mach. Intell.* **2**, 554–556 (2020).
14. Luo, W. et al. *J. Med. Internet Res.* **18**, e323 (2016).
15. Gottesman, O. et al. *Nat. Med.* **25**, 16–18 (2019).
16. Celi, L. A., Citi, L., Ghassemi, M. & Pollard, T. J. *PLoS One* **14**, e0210232 (2019).
17. Collins, G. S. & Moons, K. G. M. *Lancet* **393**, 1577–1579 (2019).
18. Norgeot, B. et al. *Nat. Med.* **26**, 1320–1324 (2020).
19. Liu, X., Cruz Rivera, S., Moher, D., Calvert, M. J. & Denniston, A. K. *Nat. Med.* **26**, 1364–1374 (2020).
20. Cruz Rivera, S., Liu, X., Chan, A. W., Denniston, A. K. & Calvert, M. *J. Nat. Med.* **26**, 1351–1363 (2020).
21. Kakarmath, S. et al. *NPJ Digit. Med.* **3**, 134 (2020).
22. Goh, W. W. B., Wang, W. & Wong, L. *Trends Biotechnol.* **35**, 498–507 (2017).
23. Schölz, C. et al. *Nat. Methods* **12**, 1003–1004 (2015).
24. Semmes, O. *J. Clin. Chem.* **51**, 1571–1572 (2005).
25. Altman, N. & Krzywinski, M. *Nat. Methods* **15**, 399–400 (2018).
26. Indyk, P. & Motwani, R. In *STOC 1998*, 604–613 (ACM, 1998); https://doi.org/10.1145/276698.276876
27. van Rijn, J. N. & Hutter, F. In Guo, Y. & Farooq, F. (eds.) *KDD 2018*, 2367–2376 (ACM, 2018).
28. Probst, P., Boulesteix, A.-L. & Bischl, B. *J. Mach. Learn. Res.* **20**, 53.1–53.32 (2019).
29. Skelly, A. C., Dettori, J. R. & Brodt, E. D. *Evid. Based Spine Care J.* **3**, 9–12 (2012).
30. Avsec, Ž. et al. *Nat. Biotechnol.* **37**, 592–600 (2019).
31. Kraker, P., Leony, D. & Reinhardt, W. *Int. J. Technol. Enhanc. Learn.* **3**, 643–654 (2011).
32. Vicente-Saez, R., Gustafsson, R. & Van den Brande, L. *Technol. Forecast. Soc. Change* **156**, 120037 (2020).

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No 826078 (R.R., J.M., N.K.W., J.B.) and No 777111 (T.K., J.B.). M.Z. received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant no. 851255). K.V.S. acknowledges funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant TranSYS (grant no. 860895). This publication reflects only the authors' view, and the European Commission is not responsible for any use that may be made of the information it contains. J.B., T.K., M.L. and Z.L. were supported by the German Federal Ministry of Education and Research (BMBF) within the e:Med framework (J.B., T.K., M.L. and Z.L.: grant no. 01ZX1908A; J.B.: grant no. 01ZX1910D). J.B. and O.Z. were supported by the BMBF within the CLINSPECT-M framework (grant no. 031L0214A). A.B.'s research is supported by a grant from Ariel University and Holon Institute of Technology, Israel: Applications of Artificial Intelligence for Enhancing the Efficiency of Vaccination Programs (grant no. RA19000000649). M.Z. acknowledges the Spanish State Research Agency, through the Severo Ochoa and María de Maeztu Program for Centers and Units of Excellence in R&D (grant no. MDM-2017-0711). M.V.d.F. and G.C.V.S. thank the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) for the fellowships provided (M.V.d.F.: process no. 140002/2018-9; G.C.V.S.: process no. 148615/2018-0). M.V.d.F. and G.C.V.S. also thank FIPE/HCPA for financial support. Contributions by J.K.P. and O.L. were funded by the Bavarian State Ministry of Science and the Arts within the framework coordinated by the Bavarian Research Institute for Digital Transformation (bidt; J.K.P.: Junior Research Group LipiTUM; O.L.: Doctoral Fellow). N.P. was supported by the DFG Cluster of Excellence Machine Learning – New Perspectives for Science (EXC 2064/1, project no. 390727645), by the BMBF (Tübingen AI Center, FKZ: 01IS18039A) and by the BMBF within the Medical Informatics Initiative (DIFUTURE, reference no. 01ZZ1804D).

Author contributions

J.M., T.K., O.L., M.L., J.K.P., R.R., N.K.W., J.B. and D.B.B. conceived AIMe. All authors designed the AIMe2021 standard in a collaborative document editing effort. J.M. and D.B.B. coordinated this effort. J.M. implemented the AIMe registry. D.B.B. wrote the initial draft of the manuscript. All authors critically reviewed and revised the manuscript.

Competing interests

G.S. receives consulting fees from Pieris Pharmaceuticals GmbH.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41592-021-01241-0.

Peer review information: *Nature Methods* thanks the anonymous reviewers for their contribution to the peer review of this work.

Eidesstattliche Versicherung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Hamburg, den ____________________________

____

Julian Oskar Matschinske