Bioinformatics, 2020, 1–9
doi: 10.1093/bioinformatics/btaa038
Advance Access Publication Date: 17 January 2020
Original Paper

Data and text mining

Secure multiparty computation for privacy-preserving drug discovery

Rong Ma[1,†], Yi Li[1,†], Chenxing Li[1,†], Fangping Wan[1], Hailin Hu [2], Wei Xu[1,*] and Jianyang Zeng[1,3,*]

[1]Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing 100084, China, [2]School of Medicine, Tsinghua University, Beijing 100084, China and [3]MOE Key Laboratory of Bioinformatics, Tsinghua University, Beijing 100084, China

*To whom correspondence should be addressed.
†The authors wish it to be known that, in their opinion, the last two authors should be regarded as Joint Last Authors.

Associate Editor: Jonathan Wren

Received on April 5, 2019; revised on January 8, 2020; editorial decision on January 13, 2020; accepted on January 15, 2020

Abstract

Motivation: Quantitative structure–activity relationship (QSAR) and drug–target interaction (DTI) prediction are both commonly used in drug discovery. Collaboration among pharmaceutical institutions can lead to better performance in both QSAR and DTI prediction. However, the drug-related data privacy and intellectual property issues have become a noticeable hindrance for inter-institutional collaboration in drug discovery.

Results: We have developed two novel algorithms under secure multiparty computation (MPC), including QSARMPC and DTIMPC, which enable pharmaceutical institutions to achieve high-quality collaboration to advance drug discovery without divulging private drug-related information. QSARMPC, a neural network model under MPC, displays good scalability and performance and is feasible for privacy-preserving collaboration on large-scale QSAR prediction. DTIMPC integrates drug-related heterogeneous network data and accurately predicts novel DTIs, while keeping the drug information confidential. Under several experimental settings that reflect the situations in real drug discovery scenarios, we have demonstrated that DTIMPC possesses significant performance improvement over the baseline methods, generates novel DTI predictions with supporting evidence from the literature and shows the feasible scalability to handle growing DTI data. All these results indicate that QSARMPC and DTIMPC can provide practically useful tools for advancing privacy-preserving drug discovery.

Availability and implementation: The source codes of QSARMPC and DTIMPC are available on the GitHub: https://github.com/rongma6/QSARMPC_DTIMPC.git.

Contact: weixu@mail.tsinghua.edu.cn or zengjy321@mail.tsinghua.edu.cn

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

In the early stage of drug discovery, identifying promising hits and optimizing various properties of lead compounds are two essential steps. Drug–target interaction (DTI) prediction based on chemical information of drugs, genomic information of proteins, known DTIs and other drug-related or protein-related information (Bleakley and Yamanishi, 2009; Luo *et al.*, 2017; Mei *et al.*, 2013; Xia *et al.*, 2009), has become a powerful way to identify promising compound hits. Optimizing various properties of compounds, such as absorption, distribution, metabolism and excretion (ADME), can be achieved by quantitative structure–activity relationship (QSAR) prediction, i.e. inferring the bioactivities from chemical structures.

Machine learning techniques have shown promising applications in drug discovery in the last few decades (Barrett and Langdon, 2006; Burbidge *et al.*, 2001; Gertrudes *et al.*, 2012; King *et al.*, 1992; Lavecchia, 2015; Murphy, 2011). For instance, the deep neural network has been successfully used for solving the QSAR regression problem (Ma *et al.*, 2015), and the DTINet algorithm (Luo *et al.*, 2017) has been proposed to predict novel DTIs from drug-related heterogeneous information and identify the new indications of old drugs. In general, larger datasets can help improve the performance of the machine learning-based approaches to solving such pharmaceutical research problems. However, more experiments (which thus cost more money and time) are generally required to obtain a larger dataset.

Public databases, such as ChEMBL (Gaulton *et al.*, 2012) and DrugBank (Knox *et al.*, 2010), can provide a large quantity of collected experimental data and other compound or drug-related information and thus make it easy for pharmaceutical companies and academic institutions to retrieve publically available data. However, these databases are also limited by the privacy concerns, in that

pharmaceutical organizations are generally reluctant to reveal their novel intellectual property information to public databases.

Recently, modern cryptographic techniques have started to be applied to drug discovery and other research fields in computational biology (Cho *et al.*, 2018; Hie *et al.*, 2018; Jagadeesh *et al.*, 2017). To achieve pharmaceutical collaboration without divulging private information, secure multiparty computation (MPC), which allows multiple participants to collaboratively perform computation on their secret data while protecting data from being leaked to others, has provided a suitable technique. In 2005, MPC was applied to perform a simple linear regression task on chemical data (Karr *et al.*, 2005). Unfortunately, linear regression is of limited use in practice. With the growing of computational capability and the further development of MPC (Tetko *et al.*, 2016), feasible MPC protocols for genomic diagnosis (Jagadeesh *et al.*, 2017), genome-wide association study (GWAS) (Cho *et al.*, 2018) and DTI prediction (Hie *et al.*, 2018) have been developed in the literature. However, widely applicable MPC protocols for different machine learning algorithms to solve various drug discovery problems are still underway.

Here, we introduce MPC to QSAR prediction for the *first* time and develop an MPC version of DTINet (Luo *et al.*, 2017) to achieve privacy-preserving DTI prediction. Computational experiments show that collaboration via our MPC protocols using protected private data delivers almost the same learning performance as public collaboration via the corresponding plaintext (i.e. all data are publically available) algorithms, and significantly outperforms the prediction strategy using private data owned by a single institution. All these results demonstrate the effectiveness and the great application potential of our MPC algorithms.

For QSAR prediction, we design QSARMPC, an MPC version of a two hidden layer neural network (Rumelhart *et al.*, 1986), which achieves collaborative QSAR prediction among different institutions without divulging chemical structures or corresponding bioactivities of compounds. In addition, the training time of QSARMPC increases linearly with the number of training instances and the dimension of features in the neural network, which implies that QSARMPC can be practically used even for large datasets. Different from Secure DTI (Hie *et al.*, 2018), a recently-developed neural network model under MPC for classification problems, QSARMPC is mainly designed for solving the regression problems and has demonstrated reasonably good performance on QSAR prediction under the MPC protocol.

For DTI prediction, we develop an MPC based version of our previously developed plaintext DTI prediction algorithm (Luo *et al.*, 2017), called DTIMPC, to predict novel DTIs from drug-related heterogeneous information. DTIMPC maintains the confidentiality of the drug-related information, which thus can encourage multiple institutions to collaborate for better DTI prediction. Our framework DTIMPC integrates a set of more relevant drug-related heterogeneous networks and surpasses both the plaintext DTI prediction algorithm with the original eight heterogeneous networks in our previous work (Luo *et al.*, 2017) and the state-of-the-art privacy-preserving DTI prediction algorithm Secure DTI (Hie *et al.*, 2018). Compared with Secure DTI (Hie *et al.*, 2018), DTIMPC achieves significantly higher AUPR score and is more suitable to perform DTI prediction under the MPC protocol by integrating drug-related heterogeneous information instead of focusing on large-scale homogeneous chemical–protein interaction data.

All these results indicate that QSARMPC and DTIMPC can provide practically powerful tools to perform privacy-preserving QSAR and DTI prediction efficiently and accurately. Moreover, our MPC based frameworks can be easily extended to other machine learning algorithms for solving various drug-related learning tasks, and thus can further advance privacy-preserving drug discovery.

2 Methods

2.1 Secure multiparty computation protocols

Consider the following MPC problem: suppose that there are n clients, denoted by $C_1, C_2, \ldots, C_n$, respectively, and each C_i holds private data D_i, $i = 1, 2, \ldots, n$. Denote P as public data. These n clients want to collaboratively calculate a function $f(D_1, D_2, \ldots, D_n, P)$, such that only the public data P and the results of the function $f(\cdot)$ are revealed.

Here, for efficiency, suppose that the MPC is in the client–server model. Each client can represent an institution, which owns private data. The number of clients can be arbitrary. We assume that there are four *semi-honest* servers (which are also called parties) here. The term semi-honest means that each party follows the designed protocol and will not send fake or false data to others but is curious about the private information and will mine sensitive information from the data as much as possible. We also make the assumption that any two of the four semi-honest parties do not collude with each other. We also assume that all communication channels are secure and the data transferred cannot be seen or modified by adversaries, which can be achieved through the Secure Sockets Layer (Li and Xu, 2019). Based on these assumptions, the four semi-honest parties know nothing about private information of individual clients. Also, each client obtains no information about the data from others, other than the information inferred from the training model or the predicted results.

In our framework, the MPC pipeline can be divided into three phases (Li and Xu, 2019) (Fig. 1). First, the private data are separated by each client locally into two parts, one shared with party S_1 and the other shared with party S_2. This operation is performed through a cryptographic technique, called 'secret sharing' (Shamir, 1979). In particular, we use the replicated 2-out-of-4 secret sharing scheme (Li and Xu, 2019) to carry out this task. Here, we use a simple example to illustrate this secret sharing concept. Suppose that an integer x is the private data in a client. Then this client picks a random integer r, and sends r and $x - r$ to parties S_1 and S_2, respectively. Here, r and $x - r$ are called the two secret shares of x. Because of the uniform randomness of r, parties S_1 and S_2 learn no information about x. After parties S_1 and S_2 receive the secret shares of x from a client, they negotiate another random integer r'. Then S_1 sends $r + r'$, r to parties S_a, S_b, respectively, and S_2 sends $x - r$, $x - r - r'$ to parties S_a, S_b, respectively. Parties S_a and S_b can learn no information about x either, but any two parties among S_1, S_2, S_a and S_b can collaborate to recover x. This procedure is called the *replicated 2-out-of-4 secret sharing* (Li and Xu, 2019) and implemented as the SecretSharing operation in our MPC protocol (Supplementary Table S3). We use $[x]$ to mean that x is secret and exists in the form of secret shares separately among S_1, S_2, S_a and S_b.

Second, we decompose the learning tasks into a sequence of basic operations, such as *addition*, *multiplication*, *comparison* and *division*, and some non-linear functions such as *sqrt* and *log*. More details about our task decomposition can be found in Sections 2.2 and 2.3 and Supplementary Notes 3 and 4. For each basic operation in format of $c = a$ op b, we call the same operation over secret shares $[c] = [a]$ op $[b]$ *private operation (PO)*. Each party should learn no information about input a and b when executing a PO. All the basic POs are implemented using the protocols available in PrivPy (Li and Xu, 2019), which require four parties. PrivPy claimed that the protocols designed on four parties can achieve more efficiency than those on three parties for many frequently used POs. The four parties complete the tasks by conducting a sequence of POs in order. Some POs can be executed in parallel to achieve better efficiency. Since each party cannot learn the input, intermediate values or final results from neither secret shares nor POs, the confidentiality of input is protected.

After completing all the POs, the four parties obtain secret shares of the final result y. Since any two secret shares can recover the final result, in the third phase, S_1 and S_2 send the secret shares of the final results to the corresponding clients. Then each client can add up these secret shares to obtain its final result. This procedure is formalized as the *Reveal* operation in our MPC protocol (Supplementary Table S3).

2.2 QSARMPC

In QSARMPC, each client C_i holds private data D_i, including local training data with chemical structure descriptors (Ma *et al.*, 2015)

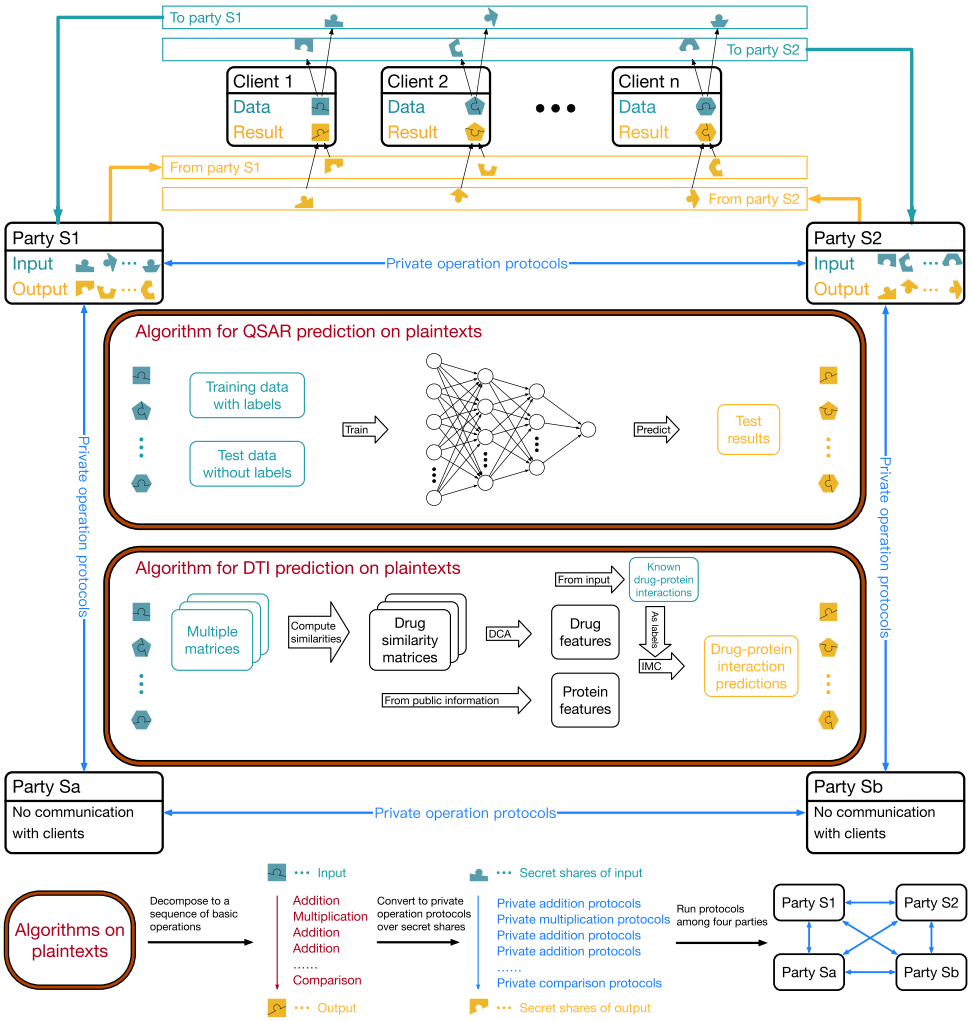

Fig. 1. The overview of QSARMPC and DTIMPC. For QSAR prediction, each client contains private training data with labels and its local test data without labels. For DTI prediction, each client contains the drug fingerprints in bit vectors, drug–disease association matrix and drug–protein interaction matrix for its private drugs. The protein feature matrix is pre-computed publically outside the MPC scheme using the public data of proteins. In the beginning, the client computes the secret shares of its private data and sends them to party S_1 and party S_2, separately, and then party S_1 and party S_2 initiate the replicated 2-out-of-4 secret sharing together with party S_a and party S_b (see the SecretSharing operation in Supplementary Table S3). The algorithm on plaintexts is decomposed to a sequence of basic operations, such as addition, multiplication and comparison. These basic operations have corresponding private operation protocols over secret shares, which are performed based on the replicated 2-out-of-4 secret sharing framework. The four parties complete the MPC algorithm by conducting a sequence of private operations in order. Note that the four parties only deal with secret shares and cannot know any information about the private data from the clients. Finally, party S_1 and party S_2 send the secret shares of the prediction results to the corresponding clients, and then each client adds up the secret shares and recovers the result for itself. DCA and IMC stand for DCA and IMC, respectively. More details can be found in the text

as features and the corresponding bioactivities as labels, and local test data with chemical structure descriptors without labels. QSARMPC trains a neural network under PO protocols by fully exploiting the training data from all the n clients to predict the QSAR activities for the test data, while only revealing the predicted activities to the corresponding clients (Fig. 1 and Supplementary Algorithm S14).

The three phases of QSARMPC are described below. In the beginning, the n clients execute secret sharing of their own private data to the MPC parties S_1 and S_2. Then parties S_1 and S_2 initiate the replicated 2-out-of-4 secret sharing together with parties S_a and S_b (Supplementary Algorithm S1 and lines 1–3 in Algorithm S14).

Then training (Supplementary Algorithm S9–S11) and prediction (Supplementary Algorithm S12) of the neural network model are performed by the four MPC parties S_1, S_2, S_a and S_b through executing a series of fundamental POs. The neural network contains two hidden layers (Supplementary Algorithm S13). The hyperparameters, including the learning rate, the number of neurons and the dropout rate (Srivastava *et al.*, 2014) for each hidden layer, are tuned by a random search procedure. Both hidden layers use the rectified linear unit (ReLU) (Nair and Hinton, 2010) as the activation function. The output layer only contains one neuron with a linear function. We use the mean squared error as the loss function (Supplementary Note). The training process is performed using backpropagation (Rumelhart *et al.*, 1986) in mini-batches with momentum (Sutskever *et al.*, 2013). The number of training epochs is set to 120 at most. In practice, early stopping is also used to remedy the potential overfitting issue (Caruana *et al.*, 2001). To avoid revealing at which epoch the early stopping criteria are met, a secret binary variable is used to help address this issue (Supplementary Note). In our experiments, we trained eight neural networks and averaged their predicted scores as the final results.

Finally, the obtained secret array $[Y_{testAll}]$ for the predicted test scores is split into secret submatrices $[\mathbf{Y}_{test}^{(1)}], [\mathbf{Y}_{test}^{(2)}], \ldots, [\mathbf{Y}_{test}^{(n)}]$ (Supplementary Algorithm S2 and line 11 in Supplementary Algorithm S14), and then the secret submatrix $[Y_{test}^{(k)}]$ is revealed to client C_k (Supplementary Table S3 and lines 12 and 13 in Algorithm S14). In this way, each client receives the predicted scores only for its test dataset.

2.3 DTIMPC

We extended our previous computational drug repositioning framework DTINet (Luo *et al.*, 2017) to predict novel DTIs from private pharmaceutical data under the MPC protocol. We call the extended MPC based version of DTINet DTIMPC (Fig. 1 and Supplementary Algorithm S28). In this problem, each client C_i (i.e. a pharmaceutical company) holds the following private information about the intellectual property protected drugs: the fingerprints of individual drugs (which are derived by RDKit (Landrum, 2013) locally in client C_i from the corresponding chemical structures), drug–disease associations and the known drug–protein interaction profiles. The public data P include the publically available target-related information, including protein–disease associations and pairwise protein–protein sequence similarity scores. After running the collaborative algorithm DTIMPC, each client C_i will obtain the predicted new DTIs only for its own drugs. More details about DTIMPC will be described below.

Except the public steps for computing the protein feature matrix (Luo *et al.*, 2017) (Supplementary Note), other parts of the DTIMPC algorithm follow the three-phase MPC pipeline. Initially, the n clients execute secret sharing of their private drug-related data (including known drug–protein interactions, drug fingerprints and drug–disease associations) to the MPC parties S_1 and S_2. Then parties S_1 and S_2 initiate the replicated 2-out-of-4 secret sharing together with parties S_a and S_b (Supplementary Algorithm S1 and lines 3–5 in Algorithm S28). Then the privacy-preserving similarity computation (Supplementary Algorithm S18 and S19), compact feature learning on the drug features (Supplementary Algorithm S24) and the inductive matrix completion (IMC) algorithm (Supplementary Algorithm S27) are executed by the four MPC parties S_1, S_2, S_a and S_b. We will give a brief explanation of these main functions in the below.

The drug similarity based on drug–disease associations is computed by a Jaccard similarity metric (Supplementary Eq. S3). The drug structure similarity between two drugs with the fingerprint vectors f_1 and f_2 of length q is calculated by the Dice similarity (Landrum, 2013) (Supplementary Eq. S4). The pair-wise similarity calculation can be paralleled by several matrix operations, which are benefited from the efficient optimization of private matrix operations (Supplementary Algorithm S18 and S19).

After the privacy-preserving drug similarity computation, we obtain two secret arrays, including drug similarity derived from drug–disease associations and drug structure similarity. Then we conduct the privacy-preserving diffusion component analysis (DCA) to find a secret array for drug features (Supplementary Algorithm S24), which contains two main steps as in the original DCA algorithm (Luo *et al.*, 2017; Tong *et al.*, 2006; Wang *et al.*, 2015). The first step is to run PoRWR for a privacy-preserving random walk with restart (RWR) (Supplementary Algorithm S20). After that, we obtain a secret array for the concatenated pairwise relevance score matrix. Next, we decompose this relevance score matrix. In DTIMPC, the matrix eigenvalue decomposition in the plaintext DCA is replaced by an iterative algorithm that can be easily extended to the MPC framework (Supplementary Algorithm S23). Here, we use the power method with Rayleigh quotient to find the most principal eigenvalue and its corresponding eigenvector (Supplementary Algorithm S21), which are then used to reduce the dimension of the current matrix to prepare for the next principal eigenvalue (Parlett, 1998).

Next, the privacy-preserving IMC operation takes the secret array for drug features, the public protein features and the secret array for known DTIs as input data and outputs the secret array $[I_{all}]$ as the predicted DTI scores (Supplementary Algorithm S27 and line 9 in Algorithm S28). In particular, we develop a privacy-preserving version of the conjugate gradient iterative optimization under square loss function (Natarajan and Dhillon, 2014; Yu *et al.*, 2014) to perform this task (Supplementary Algorithm S26).

Finally, the resulting secret array $[I_{all}]$ is split into secret submatrices $[I^{(1)}], [I^{(2)}], \ldots, [I^{(n)}]$ (Supplementary Algorithm S2 and line 10 in Algorithm S28), in which $[I^{(k)}]$ is revealed only to the corresponding client C_k (Supplementary Table S3 and lines 11 and 12 in Algorithm S28). After that, each client obtains the predicted DTI scores only for its drugs.

3 Results

3.1 Datasets

For QSAR prediction, we use the 15 datasets provided by the Kaggle competition (Ma *et al.*, 2015) to evaluate our approach. Each dataset is for a target or a type of ADME assay and is divided into training and test datasets. For DTI prediction, we use the drug-related heterogeneous dataset, which was processed in our previous work (Luo *et al.*, 2017), including the drug–protein interaction network obtained from DrugBank 3.0 (Knox *et al.*, 2010), the drug–disease association network and the protein–disease association network derived from the Comparative Toxicogenomics Database (Davis *et al.*, 2013), and the protein sequence similarity matrix computed according to the Smith–Waterman scores (Smith and Waterman, 1981). In addition, to incorporate the drug structure information, we generate the fingerprint of each drug by looking up the corresponding simplified molecular input line entry system (SMILES) in the DrugBank database (Knox *et al.*, 2010) and converting it into the Morgan fingerprint in the form of a 1024-bit vector using the RDKit Program (Landrum, 2013).

3.2 Comparison between public and MPC collaborations

We examined whether our MPC collaboration protocol using protected private data will cause the loss of the prediction accuracy when compared with the corresponding plaintext learning algorithms under public collaboration using all shared information. We first looked into the comparison between public collaboration and MPC collaboration of using QSARMPC for predicting QSARs. We used the squared Pearson correlation coefficient (R^2) as the criterion to evaluate the performance of different prediction strategies. When all the QSAR training data were publically available, the R^2 value on test data of the 15 datasets was 0.425 on average for random forest with the same hyperparameters as in Ma *et al.* (2015). QSARMPC achieved a better prediction performance, with an average R^2 of 0.446 over the 15 datasets, which was the same as the corresponding plaintext neural network (Supplementary Table S4).

With respect to the DTI prediction from drug-related heterogeneous information, we used both the area under the precision-recall curve (AUPR) and the area under the receiver operating characteristic curve (AUROC) to evaluate the DTI prediction performance. To simulate a realistic application scenario, we applied a 10-fold cross-validation on DTIs with an imbalanced distribution of positive and negative samples. In particular, we considered two test settings, one with 1:10 positive and negative samples, and the other with all samples. Here, MPC collaboration did not use the drug-side-effect associations or drug–drug interaction profiles, and the drug structure similarities were calculated based on fingerprints. We found that exploiting a larger feature space during the compact feature learning process could further improve the performance of DTINet (Luo *et al.*, 2017). We called the improved version of DTINet DTINet*. Furthermore, we found that integrating the five networks, including drug–protein interactions, protein sequence similarities, drug structure similarities, drug–disease associations and protein-disease associations, performed better than using other subsets of the eight networks in DTINet*. DTIMPC with these five heterogeneous networks yielded nearly the same prediction performance as that of public collaboration (i.e. DTINet* with the same five heterogeneous networks), and achieved a higher AUPR score than DTINet reported in our previous work (Luo *et al.*, 2017) and DTINet* with original data (Fig. 2 and Supplementary Fig. S1). To reduce the potential influence of similar drugs or homologous proteins on the performance, we also did the following additional experiments: (i) remove DTIs with similar drugs and proteins based on disease information (Jaccard similarity ≥ 0.6, Supplementary Fig. S2); (ii) remove DTIs with homologous proteins (sequence similarity ≥ 0.4, Supplementary Fig. S3); (iii) remove DTIs with similar drugs (structure similarity ≥ 0.6, Supplementary Fig. S4); and (iv) remove DTIs with both homologous proteins and similar drugs (Supplementary Fig. S5). In all these experiments, DTIMPC showed an improvement in AUPR scores, which implied that our five heterogeneous networks were

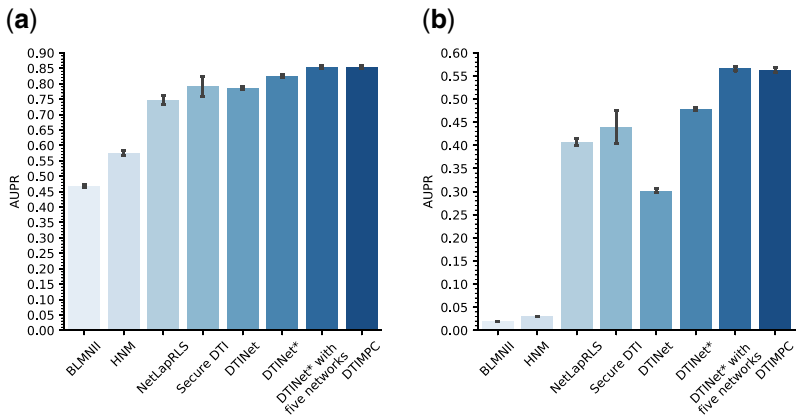

Fig. 2. Performance comparison according to AUPR between DTIMPC and other baseline methods. DTINet was the original DTINet algorithm proposed in our previous work (Luo *et al.*, 2017). DTINet* was an improved version of the DTINet algorithm by exploiting a larger feature space during the compact feature learning process. DTINet* with five networks (i.e. public collaboration) used the same five heterogeneous networks as in DTIMPC. The performance of Secure DTI was from the literature (Hie *et al.*, 2018). Among all the eight algorithms, only DTIMPC and Secure DTI are privacy-preserving. The experiments were performed in 10-fold cross-validation on all pairs of DTIs with (**a**) 1:10 positive and negative samples and (**b**) all samples. The results are shown as mean ± standard deviation of 10 trials

more relevant to DTI prediction and improved the performance, and our approximation in the logarithmic operation and the iterative algorithm for conducting the matrix eigenvalue decomposition under MPC did not degrade the performance. In all the above DTI prediction experiments, the hyperparameters were provided in Supplementary Table S5, which were found using a grid search procedure. All the above results demonstrated that under the QSARMPC and DTIMPC protocols, private pharmaceutical organizations or institutions could achieve high-quality collaboration and reach almost the same learning results as in the corresponding plaintext prediction algorithms on the publicly shared data.

3.3 Comparison between DTIMPC and other DTI prediction algorithms

We also compared the prediction performance between our DTIMPC and another four DTI prediction baseline methods, including three plaintext algorithms: BLMNII (Mei *et al.*, 2013), HNM (Wang *et al.*, 2014) and NetLapRLS (Xia *et al.*, 2010), and one privacy-preserving method Secure DTI (Hie *et al.*, 2018), through a 10-fold cross-validation procedure (Fig. 2 and Supplementary Fig. S1). The hyperparameters in these three plaintext baseline algorithms were tuned using the same grid search strategies as in our previous work (Wan *et al.*, 2019). The cross-validation settings on all pairs of DTIs for both 1:10 positive and negative samples and all samples were considered. Our DTIMPC achieved a significant improvement in AUPR on such skewed DTI data (Fig. 2). As shown in previous studies (Van Laarhoven *et al.*, 2011), AUPR is a more suitable metric than AUROC for the DTI prediction problem, because the DTIs with the higher AUPR scores predicted by an algorithm are more likely to be correct. These comparison results demonstrated the strong predictive power of our DTIMPC protocol.

3.4 Comparison between predictions using more data with MPC collaboration and using private data owned by a single institution

In an ideal secure collaboration scenario, samples from different pharmaceutical institutions are pooled together to form a more informative sample space and facilitate the training of better machine learning models. To mimic such an application setting, we randomly separated all available data into subsets and distributed them to individual institutions, which were then regarded as private data. For QSAR prediction, suppose that there are n_t instances in all available training data. Then a single institution owns a random subset of training data with $\lfloor x \cdot n_t \rfloor$ instances, where x stands for a fraction parameter within the range of (0, 1). We used the squared Pearson correlation coefficient (denoted by R^2) on the whole test dataset to evaluate the performance of different prediction strategies. For the case without MPC collaboration, every single institution ran a plaintext neural network, in which the hyperparameters were calibrated using the same random search strategy as in QSARMPC. We found that among all 15 datasets, when fully exploiting all available training data under the MPC protocol, QSARMPC outperformed the strategy of using only the private data from a single institution, especially for small x (Supplementary Fig. S6). This result demonstrated that the privacy-preserving collaboration by QSARMPC gained better performance than the prediction using only intra-institution private data, since QSARMPC can exploit more data to train the neural network in a secure manner.

For the DTI prediction or drug repositioning task, denote the total number of instances (drugs) by n_d. Suppose that a single institution owns a random subset of the currently available drug set with $\lfloor x \cdot n_d \rfloor$ instances, where x stands for a fraction parameter within (0, 1). When performing the drug repositioning task on the private data owned by a single institution (i.e. without MPC collaboration), the plaintext DTINet* algorithm (Luo *et al.*, 2017) on the same five heterogeneous networks as in DTIMPC was used to make a prediction. In this experiment, we ran 10-fold cross-validation on all pairs of DTIs with 1:10 positive and negative samples and all samples. In each fold, without MPC collaboration, every single institution only took samples related to its own drugs as its individual small training and test datasets. Here, the hyperparameters for the plaintext DTINet* using five heterogeneous networks within a single institution were calibrated using grid search separately for different x values. In the DTIMPC framework, the whole training data under the MPC protocol were used to train the model, while the same small test dataset from the single institution was used to assess the prediction performance. We looked into the average difference of the AUROC or AUPR scores over 10-folds for different x values between predictions using only private data within an institution and using all training data under MPC collaboration (Supplementary Fig. S7). Our comparison showed that when using all training data under MPC collaboration, DTIMPC can significantly outperform the plaintext DTINet* algorithm with five heterogeneous networks using only private data within a single institution, especially for smaller x values (Supplementary Fig. S7). All these results indicated that DTIMPC could fully take advantage of all existing drug data among different pharmaceutical organizations or institutions without divulging the private intellectual property information and thus provide a better choice to achieve all-win results.

3.5 Novel DTIs predicted by DTIMPC

We predicted novel DTIs by DTIMPC based on the training over all pairs of known DTIs. We selected those novel predictions whose

scores were significantly high among all the drugs and the proteins (using the three-sigma rule), and also excluded those easy predictions which were similar to any known DTIs in the training data (i.e. with drug structure similarity larger than 0.6 and protein sequence similarity larger than 0.4). Among the top 20 novel DTIs predicted by DTIMPC, many can be supported by the known evidence in the literature (Table 1). For instance, aripiprazole and ziprasidone are known to have high affinities for 5-hydroxytryptamine receptor 2B (HTR2B) (Shahid *et al.*, 2009) and our prediction result was consistent with this evidence. In addition, DTIMPC predicted that sorafenib can act on the vascular endothelial growth factor receptor 1 (FLT1), which plays an important role in the regulation of angiogenesis (UniProt Consortium, 2018). This prediction can be supported by the previous known evidence that sorafenib inhibits FLT1 (Kitagawa *et al.*, 2013). All these results indicated that the novel DTIs predicted by DTIMPC can provide useful clues for repositioning existing drugs and finding their new indications.

3.6 Scalability

Scalability has been an essential practical issue in the MPC protocol. Here, we examined the scalability of both QSARMPC and DTIMPC with respect to the sizes of training data. For QSARMPC, we tested the influence of both the number of training instances and the dimension of features on the training time (Fig. 3a and b). In our tests, when the number of training instances is no more than the batch size, the training process is conducted in a full-batch manner; otherwise, it is performed in a mini-batch scheme. To analyze the influence of the number of training instances on scalability, we examined the running time of one training epoch with respect to different numbers of training batches per epoch. For different dimensions of features, we investigated the running time of training a batch in one iteration. Our tests showed that the training time of QSARMPC displayed a linear trend with respect to both the number of training instances and the dimension of the encoded features in training data (Fig. 3a and b).

For DTIMPC, we simulated various datasets with different numbers of drugs and recorded the corresponding running time. Our test results showed that the running time of DTIMPC was almost linearly proportional to the number of drugs in the current scale tested, which thus demonstrated its feasible scalability in practice (Fig. 3c). We also investigated how the communication costs of the three main functions in DTIMPC scale with the number of drugs (Supplementary Fig. S8). The communication costs for the privacy-preserving pairwise drug–drug similarity computation and the privacy-preserving DCA both showed quadratic trends in the number of drugs. In contrast, the communication cost for the privacy-preserving IMC was linear to the number of drugs. On the other hand, in the current scale tested, the communication cost for IMC was higher than those for similarity computation and DCA, because of the large number of iterative updates in IMC. Our DTIMPC algorithm using the whole dataset processed in our previous work (Luo *et al.*, 2017) cost 1.51 h in a local area network (LAN) setting with 17.38 gigabytes communication at each of the four parties. The running time of the non-private methods, DTINet and DTINet*, was around 18 s and 34 s, respectively. These results showed that our DTIMPC algorithm protected the privacy of highly sensitive drug-related data at an acceptable extra time cost.

3.7 Hyperparameter calibration

We evaluated the robustness of the corresponding plaintext algorithms of QSARMPC and DTIMPC against different choices of hyperparameters to find out which hyperparameters need to be tuned in the MPC setting. In practice, the clients can test the performance of the corresponding plaintext algorithms of QSARMPC and DTIMPC using their local data and 10-fold cross-validation with different random seeds to determine which hyperparameters influence the accuracy little and thus can be preset, and which hyperparameters affect the accuracy a lot and need to be further tuned in the MPC setting. In consideration of the time complexity of the MPC algorithms, hyperparameter tuning under MPC should be performed using an independent held-out dataset, instead of 10-fold cross-validation with different random seeds. For the corresponding plaintext algorithm of QSARMPC, we tested performance in terms of the squared Pearson correlation coefficient with different choices of hyperparameters: the number of neurons for the two hidden layers (Supplementary Fig. S9a), the dropout rate for the two hidden layers (Supplementary Fig. S9b) and the learning rate (Supplementary Fig. S10). We observed that the corresponding plaintext algorithm of QSARMPC gained stable results with multiple choices of the number of neurons and the dropout rate for the hidden layers. More importantly, we observed that the training was underfitting within the maximum number of epochs when the learning rate was too small (Supplementary Fig. S10). Thus, the number of neurons and the dropout rate for the hidden layers can be preset, and all the clients can negotiate the preset values. A suitable range of the learning rate can be suggested by the clients using their local data and the optimal values can be finally determined using a grid search over only several values under MPC. Note that the time complexity of running QSARMPC with different learning rates is the same. Therefore, suppose that the running time of QSARMPC is s, then the time for hyperparameter tuning using a grid search with r values of learning rate would be sr.

For the corresponding plaintext algorithm of DTIMPC, we tested its performance in terms of both AUPR and AUROC with different choices of hyperparameters, including the restart probability of RWR p_r (Supplementary Fig. S11), the number of expansion terms of Taylor series in the logarithmic operation t_g (Supplementary Fig. S12), the number of power method iterations t_p (Supplementary Fig. S13), the dimensions of drug features f_d and protein features f_p (Supplementary Fig. S14), the number of IMC iterations t (Supplementary Fig. S15), the latent rank of IMC k (Supplementary Fig. S16), the regularization parameter of IMC λ (Supplementary Fig. S17) and the number of iterations in updating decomposed low-rank matrices in IMC t_c (Supplementary Fig. S18). Note that, because AUPR of unbalanced test datasets can better reflect the predictive power in the DTI prediction problems than AUROC (Van Laarhoven *et al.*, 2011), we mainly considered the performance in terms of AUPR on all pairs of DTIs with (i) 1:10 positive and negative samples and (ii) all samples. We observed that the corresponding plaintext algorithm of DTIMPC produced stable results over a wide range of hyperparameter settings. In practice, only coarsely tuning the dimensions of drug and protein features, the latent rank of IMC and the number of iterations in updating decomposed low-rank matrices is sufficient enough to achieve excellent prediction performance (Supplementary Figs S11–S18). The hyperparameters, except these four, can be preset, and all the clients can negotiate the preset values. Also, the corresponding plaintext

Table 1. Among the top 20 predictions by DTIMPC, eight novel DTIs have supporting evidence from the literature

Drug	Protein	Supporting literature
Sorafenib	FLT1	Sorafenib inhibits FLT1 (Kitagawa *et al.*, 2013)
Sorafenib	CSF1R	Sorafenib blocks CSF1R (Ullrich *et al.*, 2011)
Olanzapine and risperidone	HTR2B	Olanzapine and risperidone block HTR2B (Shahid *et al.*, 2009)
Risperidone	HTR7	Risperidone blocks HTR7 (Shahid *et al.*, 2009)
Aripiprazole and ziprasidone	HTR2B	Aripiprazole and ziprasidone show high affinities for HTR2B (Shahid *et al.*, 2009)
Haloperidol	DRD4	Haloperidol has high affinity for dopamine receptor subtype D_4 (Bymaster *et al.*, 1996)

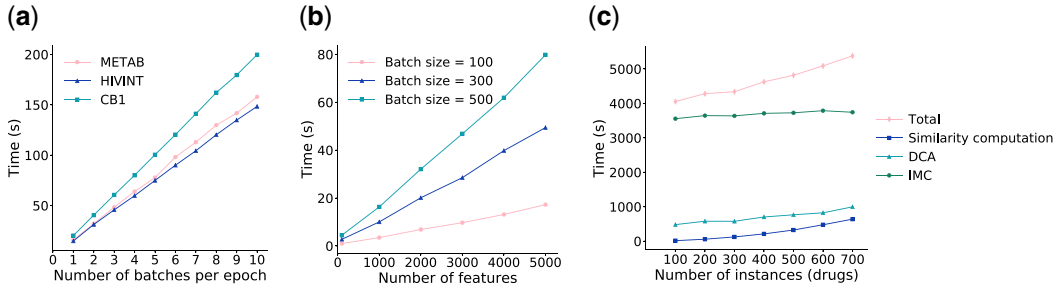

Fig. 3. Scalability of QSARMPC and DTIMPC. (**a**) Running time (in seconds) of training the neural network in QSARMPC in one epoch with different numbers of instances (in which the batch size was fixed as 128). The legend represents the different datasets. We used for three datasets as an example from the 15 datasets provided by the Kaggle competition (Ma *et al.*, 2015). The dimensions of input features were 4505, 4306 and 5877 for datasets METAB, HIVINT and CB1, respectively. (**b**) Running time (in seconds) of training a batch in QSARMPC in one iteration with different dimensions of features. We simulated datasets with different dimensions of features by subsampling the features of the original CB1 dataset. (**c**) Running time (in seconds) of DTIMPC with different numbers of instances (i.e. drugs). We used the simulated data for different numbers of drugs by subsampling the drugs from the original datasets processed in our previous work (Luo *et al.*, 2017). The running time of DTIMPC (with the legend 'Total') consists of the time for three steps, namely, the privacy-preserving pairwise drug–drug similarity computation based on the drug fingerprints and the drug–disease associations (with the legend 'Similarity computation'), the privacy-preserving DCA (with the legend 'DCA') and the privacy-preserving IMC (with the legend 'IMC'). The restart probability of the RWR p_r was fixed as 0.5, the number of expansion terms of Taylor series in the logarithmic operation was fixed as 100, the number of the power method iterations was fixed as 10, the dimension of drug features was fixed as 100, the dimension of protein features was fixed as 400, the number of the IMC iterations was fixed as 5, the latent rank of IMC was fixed as 50 and the number of iterations in updating the decomposed low-rank matrices in IMC was fixed as 200

algorithm of DTIMPC showed good stable performance when these four hyperparameters are large enough. Thus, presetting these four hyperparameters to reasonably large values is also acceptable. If the clients want to further perform hyperparameter tuning to verify whether these four hyperparameters are large enough, a grid search with several values under MPC is also acceptable. Suitable ranges of these four hyperparameters can be suggested by the clients using their local data in 10-fold cross-validation and the optimal values can be finally determined using a grid search over only several values through holdout validation under MPC. Hyperparameter tuning under MPC should be decomposed into four phases. In each phase, only one hyperparameter is tuned. Tuning one hyperparameter with r values would only take r additional times of running the DTIMPC algorithm under MPC.

4 Discussion

Our MPC algorithms achieve privacy-preserving computation based on the assumption that the parties are semi-honest. The clients can be malicious in terms of conspiracy since they are not involved in the four-party computation process. The clients only send the secret shares of their private data to the four parties and wait to receive the secret shares of the final results after the four parties completing the whole algorithms over four-party computation. We conceive that government agencies, cloud service platforms and research institutions can act as this role. Nevertheless, in real-world scenarios, it may be challenging to find such semi-honest parties. However, we can make our protocols achieve stronger security guarantees using Intel's Software Guard Extensions (SGX) (Schunter, 2016). Previous work has demonstrated a feasible solution for secure computation over biometric data based on SGX (Chen *et al.*, 2016). When our MPC protocols run in SGX, SGX can enforce all the four parties to follow the prescribed steps and prevent them from touching the intermediate results. Any party cannot be semi-honest or malicious unless it can break SGX. Even if one party is able to steal data from SGX, it can obtain no more information than secret shares. Therefore, SGX can reinforce security effectively.

On the one hand, entities such as pharmaceutical institutions generally aim to protect the confidentiality of drug-related intellectual property information. On the other hand, they expect to achieve better performance through cooperation than learning by their local data alone. Even for directly competing companies, the goal for achieving better performance makes collaboration more rational, because counterfeit data mixed by a malicious participant could also reduce the performance of the malicious participant itself. For QSAR prediction, a reasonable constraint for cooperation is that the size of the individual test dataset for each client has an upper limit. These upper limits can be determined by the size of the local training dataset owned by each client. A malicious client may add fake training data to obtain a higher upper limit of involved data or mislead other clients. But once the counterfeit data reduce the effectiveness of collaboration, the prediction scores for the malicious client's data would also lose the value of references. For DTI prediction, note that at the revealing phase of DTIMPC, each client can only recover the predicted DTI scores for its local drugs. If a client provides fake input data to mislead other clients, this client would also obtain bad results. Even though there is some client who aims to mislead competitors, honest clients can evaluate the performance of MPC collaboration by their results. For QSAR prediction, we could consider additional metrics (i.e. the squared Pearson correlation coefficient and the mean squared loss) on a reserved validation dataset and corresponding local training data during training to each client. In this way, honest clients can decide whether to trust the collaboration results. For DTI prediction, reliable clients can compare the results to the corresponding input known DTI profiles and have some idea of how the collaboration algorithm performs for ground-truth positives.

Federated learning algorithms are another kind of approaches for multiple participants to collaboratively perform learning while preventing their input data from being public. For instance, in privacy-preserving deep learning (Shokri and Shmatikov, 2015), a federated learning method, each participant trains a local neural network model by uploading/downloading a fraction of parameters to/from a global parameter center during the training process. However, the follow-up designed generative adversarial network (GAN) attack (Hitaj *et al.*, 2017) may act as a warning for using such collaborative learning for privacy-preserving purposes. In the GAN attack, a malicious participant can reconstruct the training data of a given label, by training a GAN simultaneously with the collaborative model, setting a specific misleading label to these fake data generated by the GAN in each epoch, and pretending that these counterfeit data with specific misleading labels are local training data. Note that our MPC algorithms naturally defend against this GAN attack, since the participants select their training data at once and cannot dynamically affect the learning process.

Which kind of results can be revealed to the participants is an important question to consider. For QSAR prediction, in consideration of the model inversion attacks (Fredrikson *et al.*, 2015), e.g. reconstruction attack, the trained model must not be revealed. In the model inversion attacks, the adversary designs the next data to be queried based on the prediction results of the previously queried data. Since our MPC algorithms limit each participant to perform a one-off selection of their local testing data and the size of testing data always has an upper limit, the predictions of dynamically designed test data are not available to the adversary. Thus, such

model inversion attacks do not damage the privacy-preserving property of our MPC algorithms. For DTI prediction, each participant only receives the predicted DTI scores for its local drugs. Revealing only the order of proteins with the top DTI scores instead of all prediction scores can further control information leakage. In this way, the participants cannot infer private data of others from the predicted DTI scores.

We suggest computing in plaintexts for insensitive information such as protein-related data, since it is unnecessary to protect the confidentiality of that information at the cost of more time and computing resources. For DTI prediction, we assume that the drug-related information is private, while the protein-related heterogeneous networks (not related to drugs) are public. This scenario is reasonable for real-world collaboration among pharmaceutical institutions because of the high cost to obtain drug-related data (especially the chemical structures of drugs) and the availability of protein-related information.

Our MPC protocols mainly use secret sharing schemes (Shamir, 1979) and combine garbled circuits (Yao, 1982) for efficient private comparison operations (Li and Xu, 2019). Because pure garbled circuit schemes as used in Jagadeesh *et al.* (2017) are not suitable for the DTINet algorithm (Luo *et al.*, 2017) or the neural network models, which are difficult to be implemented through low-depth parallel computation. Probably due to this reason, secure GWAS (Cho *et al.*, 2018) and Secure DTI (Hie *et al.*, 2018) also use secret sharing as primary schemes.

On the other hand, unlike secure GWAS (Cho *et al.*, 2018) or Secure DTI (Hie *et al.*, 2018) using three-party computation, our algorithms adopted PrivPy, a four-party computation framework (Li and Xu, 2019), because of the high efficiency of machine learning tasks under MPC using PrivPy, which was mainly benefited from the replicated 2-out-of-4 secret sharing, the corresponding well-designed fixed-point multiplication protocols and the elegant optimization for batch up and matrix multiplication (Li and Xu, 2019). Notably, the fixed-point multiplication in the four-party computation framework of PrivPy requires only one round communication without precomputation and each party sends only two messages, which is faster than ABY^3 (Mohassel and Rindal, 2018), the state-of-the-art three-party computation framework for arithmetics.

Compared with Secure DTI (Hie *et al.*, 2018), an MPC version of a neural network to perform DTI prediction or large-scale compound–protein interaction prediction, our DTIMPC inherits the same spirit of our original DTINet algorithm (Luo *et al.*, 2017), which can thus take full advantage of the drug-related heterogeneous networks to achieve privacy-preserving drug repositioning, and gain a much higher AUPR score than Secure DTI (Hie *et al.*, 2018) (Fig. 2). On the other hand, the STITCH dataset (Szklarczyk *et al.*, 2016) used in Secure DTI (Hie *et al.*, 2018) consists of 265 080 chemicals. Inherited from the plaintext DTINet algorithm (Luo *et al.*, 2017), our DTIMPC algorithm also includes a step to calculate pairwise drug–drug similarities based on heterogeneous drug-related networks and a step to perform DCA (Wang *et al.*, 2015) for drugs. Both steps require quadratic communication costs in the number of drugs (Supplementary Fig. S8) and thus are not feasible for datasets with drugs in such a large scale. In light of this, the current version of our DTIMPC algorithm is not suitable to run on the STITCH dataset. How to improve our algorithm to be feasible for such large datasets will be one future direction of our work.

The main difficulties in employing MPC in drug discovery can be summarized as follows: (i) to convince pharmaceutical institutions that collaboration can lead to better performance; (ii) to convince pharmaceutical institutions that collaboration can be completed without divulging their private and sensitive information; (iii) to implement the algorithms under MPC in an acceptable time complexity; and (iv) to be convenient for data scientists without cryptographic background to modify the source codes for algorithmic adjustment and improvement. Our work has made great efforts to resolve these difficulties: (i) We designed the single-institution experiments to mimic the predictions using only private data and showed that collaboration using MPC can greatly improve the prediction performance. (ii) We implemented the MPC algorithms using the four-party computation framework PrivPy (Li and Xu, 2019), and carefully designed which kinds of results need to be revealed to the clients. (iii) We fully took advantage of high computational efficiency embedded in PrivPy (Li and Xu, 2019). (iv) We fully exploited the friendly programming interface in PrivPy (Li and Xu, 2019) to design the privacy preserving algorithms for drug discovery. PrivPy (Li and Xu, 2019) provides a friendly Python front-end for developing a learning algorithm under MPC. It has automatic code rewriting to fit the algorithms more efficient under the MPC setting. Data scientists can write a machine learning algorithm under the MPC setting as easily as writing a plaintext version without much pain.

In this work, we convert our original DTINet algorithm and the neural network model to the MPC version and achieve privacy-preserving QSAR and DTI prediction. Based on the PrivPy framework (Li and Xu, 2019), other machine learning models can also be easily extended to a high-quality and efficient MPC version with minimal effort. Thus, our work provides a good example to demonstrate that secure MPC can be effectively used to advance privacy-preserving drug discovery.

Acknowledgements

The authors thank Brian Hie for providing the results of Secure DTI. The authors also thank Shuya Li for helpful discussions. The authors also thank Xin Liu and Xiaoyu Fan for their help in releasing the source code of QSARMPC and DTIMPC.

Funding

This work was supported in part by the National Natural Science Foundation of China [61872216, 61472205 and 81630103] and the Zhongguancun Haihua Institute for Frontier Information Technology.

Conflict of Interest: none declared.

References

Barrett,S. and Langdon,W. (2006) Advances in the application of machine learning techniques in drug discovery, design and development. In: Tiwari,A. *et al.* (eds), *Applications of Soft Computing*. Springer, Berlin and Heidelberg, pp. 99–110.

Bleakley,K. and Yamanishi,Y. (2009) Supervised prediction of drug-target interactions using bipartite local models. *Bioinformatics*, **25**, 2397–2403.

Burbidge,R. *et al.* (2001) Drug design by machine learning: support vector machines for pharmaceutical data analysis. *Comput. Chem.*, **26**, 5–14.

Bymaster,F.P. *et al.* (1996) Radioreceptor binding profile of the atypical antipsychotic olanzapine. *Neuropsychopharmacology*, **14**, 87–96.

Caruana,R. *et al.* (2001) Overfitting in neural nets: backpropagation, conjugate gradient, and early stopping. In Advances in Neural Information Processing Systems, pp. 402–408.

Chen,F. *et al.* (2016) Princess: privacy-protecting rare disease international network collaboration via encryption through software guard extensions. *Bioinformatics*, **33**, 871–878.

Cho,H. *et al.* (2018) Secure genome-wide association analysis using multiparty computation. *Nat. Biotechnol.*, **36**, 547–551.

Davis,A.P. *et al.* (2013) The comparative toxicogenomics database: update 2013. *Nucleic Acids Res.*, **41**, D1104–D1114.

Fredrikson,M. *et al.* (2015) Model inversion attacks that exploit confidence information and basic countermeasures. In *Acm Sigsac Conference on Computer & Communications Security*.

Gaulton,A. *et al.* (2012) ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res.*, **40**, D1100–D1107.

Gertrudes,J. *et al.* (2012) Machine learning techniques and drug design. *Curr. Med. Chem.*, **19**, 4289–4297.

Hie,B. *et al.* (2018) Realizing private and practical pharmacological collaboration. *Science*, **362**, 347–350.

Hitaj,B. *et al.* (2017) Deep models under the GAN: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ACM, pp. 603–618.

Jagadeesh,K.A. *et al*. (2017) Deriving genomic diagnoses without revealing patient genomes. *Science*, **357**, 692–695.

Karr,A.F. *et al*. (2005) Secure analysis of distributed chemical databases without data integration. *J. Comput. Aided Mol. Des*., **19**, 739–747.

King,R.D. *et al*. (1992) Drug design by machine learning: the use of inductive logic programming to model the structure-activity relationships of trimethoprim analogues binding to dihydrofolate reductase. *Proc. Natl. Acad. Sci. USA*, **89**, 11322–11326.

Kitagawa,D. *et al*. (2013) Activity-based kinase profiling of approved tyrosine kinase inhibitors. *Genes Cells*, **18**, 110–122.

Knox,C. *et al*. (2010) DrugBank 3.0: a comprehensive resource for 'omics' research on drugs. *Nucleic Acids Res*., **39(suppl_1)**, D1035–D1041.

Landrum,G. (2013) Rdkit documentation. *Release*, **1**, 1–79.

Lavecchia,A. (2015) Machine-learning approaches in drug discovery: methods and applications. *Drug Discov. Today*, **20**, 318–331.

Li,Y. and Xu,W. (2019) PrivPy: General and scalable privacy-preserving data mining. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ACM, pp. 1299–1307.

Luo,Y. *et al*. (2017) A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information. *Nat. Commun*., **8**, 573.

Ma,J. *et al*. (2015) Deep neural nets as a method for quantitative structure-activity relationships. *J. Chem. Inf. Model*., **55**, 263–274.

Mei,J.-P. *et al*. (2013) Drug-target interaction prediction by learning from local information and neighbors. *Bioinformatics*, **29**, 238–245.

Mohassel,P. and Rindal,P. (2018) ABY 3: a mixed protocol framework for machine learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ACM, pp. 35–52.

Murphy,R.F. (2011) An active role for machine learning in drug development. *Nat. Chem. Biol*., **7**, 327–330.

Nair,V. and Hinton,G.E. (2010) Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 807–814.

Natarajan,N. and Dhillon,I.S. (2014) Inductive matrix completion for predicting gene-disease associations. *Bioinformatics*, **30**, i60–i68.

Parlett,B.N. (1998) *The Symmetric Eigenvalue Problem*. Vol. **20**. SIAM, Philadelphia, PA.

Rumelhart,D.E. *et al*. (1986) Learning representations by back-propagating errors. *Nature*, **323**, 533–536.

Schunter,M. (2016) Intel software guard extensions: Introduction and open research challenges. In: *Proceedings of the 2016 ACM Workshop on Software Protection*, p. 1.

Shahid,M. *et al*. (2009) Asenapine: a novel psychopharmacologic agent with a unique human receptor signature. *J. Psychopharmacol*., **23**, 65–73.

Shamir,A. (1979) How to share a secret. *Commun. ACM*, **22**, 612–613.

Shokri,R. and Shmatikov,V. (2015) Privacy-preserving deep learning. In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1310–1321.

Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol*., **147**, 195–197.

Srivastava,N. *et al*. (2014) Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res*., **15**, 1929–1958.

Sutskever,I. *et al*. (2013) On the importance of initialization and momentum in deep learning. In *International Conference on Machine Learning*, pp. 1139–1147.

Szklarczyk,D. *et al*. (2016) STITCH 5: augmenting protein-chemical interaction networks with tissue and affinity data. *Nucleic Acids Res*., **44**, D380–D384.

Tetko,I.V. *et al*. (2016) BIGCHEM: challenges and opportunities for big data analysis in chemistry. *Mol. Inf*., **35**, 615–621.

Tong,H. *et al*. (2006) Fast random walk with restart and its applications. In *Data Mining, 2006. ICDM'06. Sixth International Conference*, IEEE, pp. 613–622.

Ullrich,K. *et al*. (2011) BAY 43-9006/Sorafenib blocks CSF1R activity and induces apoptosis in various classical Hodgkin lymphoma cell lines. *Br. J. Haematol*., **155**, 398–402.

UniProt Consortium. (2018) UniProt: the universal protein knowledgebase. *Nucleic Acids Res*., **46**, 2699.

Van Laarhoven,T. *et al*. (2011) Gaussian interaction profile kernels for predicting drug-target interaction. *Bioinformatics*, **27**, 3036–3043.

Wan,F. *et al*. (2019) NeoDTI: neural integration of neighbor information from a heterogeneous network for discovering new drug-target interactions. *Bioinformatics*, **35**, 104–111.

Wang,S. *et al*. (2015) Exploiting ontology graph for predicting sparsely annotated gene function. *Bioinformatics*, **31**, i357–i364.

Wang,W. *et al*. (2014) Drug repositioning by integrating target information through a heterogeneous network model. *Bioinformatics*, **30**, 2923–2930.

Xia,Z. *et al*. (2009) Semi-supervised drug-protein interaction prediction from heterogeneous spaces. In *The Third International Symposium on Optimization and Systems Biology*, Vol. **11**, Citeseer, pp. 123–131.

Xia,Z. *et al*. (2010) Semi-supervised drug-protein interaction prediction from heterogeneous biological spaces. *BMC Syst. Biol*., **4**, S6.

Yao,A.C. (1982) Protocols for secure computations. In *Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium*, IEEE, pp. 160–164.

Yu,H.-F. *et al*. (2014) Large-scale multi-label learning with missing labels. In *International Conference on Machine Learning*, pp. 593–601.