iDASH Privacy & Security Workshop 2019 - Secure Genome Analysis Competition
Track 1: Distributed Gene-Drug Interaction Data Sharing based on Blockchain and Smart Contracts

Readme for Data and Skeleton Code

Data Format

The training dataset contains 4 files, each with 10,000 rows using the following format (separated by the tab character):

Gene-Name
Variant-Number
Drug-Name
Outcome
Suspected-Gene-Outcome-Relation
Serious-Side-Effect

The Gene-Name and Drug-Name is extracted from Clinical Pharmacogenetics Implementation Consortium (CPIC, https://cpicpgx.org/genes-drugs/). The Suspected-Gene-Outcome-Relation and Serious-Side-Effect are binary values. The Variant-Number is an integer value between 1 and 99. There are 3 types of outcomes:

IMPROVED, UNCHANGED, DETERIORATED

The testing data will be in the same format, and may contain a different number of records (but in a similar scale).

Note for Skeleton Code

We have provided a skeleton contract named GeneDrugRepo.sol which contains empty functions which must be completed. All testing will be through interaction with these functions. Participants are not limited to only using these functions, and may create other functions as well. The only restriction in participant's design is that each of the initial functions must maintain the provided function names, input, and output formats.

Deployment of participant's contract will be handled by the testers. The submission should contain only uncompiled solidity code. We will compile it ourselves using solc 5.8. For further information on the specifics of the code requirements, please refer to the comments in GeneDrugRepo.sol. For initial development we recommend to begin by using Remix (remix.ethereum.org).

Additional Rules of Solution

The solution should be designed in such a way as to be optimized for reading and storing live changes to a data sharing system, instead of compressing and handling a million entries at once. Therefore, the solution may be tested by sending batches of lines (e.g. 1000 lines, or even 1 line) at a time. To simulate a real-world data sharing solution, any sort of buffering method (which may lead to a loss of data) is not allowed.

Also, the solution should be able to store the data on one node of the blockchain network and retrieve on any other node, without any other method of communication between machines. The solution should be symmetric (i.e. identical software for each node), and no other computing/storage resources will be available, only the virtual machines provided for testing. Furthermore, the solution should be standalone, such that upon installation it should be able to read and write to an existing blockchain without any external setup or configuration, no calls to outside resources.