CSCI 5352
Final Project Proposal: Predictive Toxicology Framework

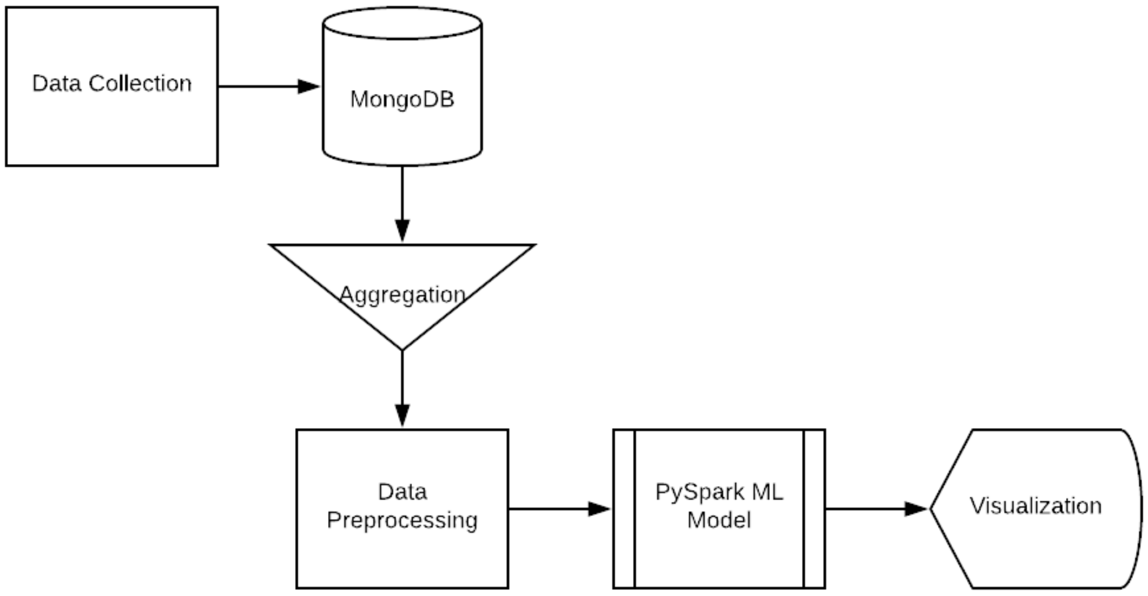
Students

- Ignacio Tripodi
- Harshini Priya Muthukrishnan
- Chu-Sheng Ku
- Chi Chen

Problem Description

The area of computational predictive toxicology has been increasingly expanding over the last few years due to recent toxicity assay requirements by regulatory agencies, especially in the European Union. The large volume of chemicals requiring a toxicity assessment and the possible different chemical endpoints (skin irritation, eye irritation, systemic toxicity, etc) doesn't scale to perform traditional assays, particularly animal models. This, in combination with a push towards replacing animal models due to humane reasons, makes computational predictions increasingly more useful as a pre-screening tool. Regulatory agencies are in a constant search for different computational models to parse the vast amounts of data available, and come up with classifiers to evaluate which untested chemicals are at a higher risk of posing a threat of eliciting a toxic response (or, which have a high likelihood of being completely safe at normal exposure levels).

Architecture



- Data collection: Download the dataset from NTP and CTD and store them to S3
- MongoDB: Use Python code to extract the data and save them to the database
- Aggregation: Use Spark Mongo connector to get the multiple tables from the database. Refer to Dataset portion of the proposal
- Data Preprocessing: Collect only data required for the classifier and clean it
- Machine learning model: Build a machine learning model on Spark. Refer to description below for details.
- Visualization: Visualize the analysis result to explain the useful knowledge found

Final Project Layout

The project will consist of two main components: data aggregation using cloud computing tools we've learned during the course of this class, and a machine learning framework to attempt prediction of untested chemicals.

Machine Learning

Once we have these merged data (and the rest of chemicals in CTD that are not included in the NTP datasets, to use as testing) we will train a classifier to determine whether the rest of CTD chemicals will pose a high or low risk of toxicity, for each of the three endpoints (dermal, oral, and inhalation), based on the genes and pathways they affect, how they affect them, and potentially additional properties of the chemical itself. In order to avoid complications with long training times, we will limit this task to simple yet appropriate machine learning classification algorithms, like random forests or AdaBoost. We will employ the SciKit Learn library for Python and test various classifiers to determine which display the best performance. The performance evaluation will be based on cross-validation using a random fraction of NTP's chemical list as testing, since we don't have a "ground truth" for the rest of chemicals. Our predictions can be shared with the scientific community for further evaluation.

Dataset

We will employ the full dataset of acute toxicity (dermal, oral, and inhalation) from the National Toxicology Program (NTP) which contains a list of compounds, their active chemical components, and various other metadata. The toxicity classification score will be used as the "label" to be predicted by our supervised learning setting. We will decide over the course of the project whether we will make this a multi-label classification problem, or if we define a threshold to determine whether a compound is "worth testing" or not (e.g., positive if it scores higher than 4 in OECD's scale which ranges from 1 to 6). The latter could also allow us to perform a linear regression instead. For each of these chemical compounds and their individual components, we will attempt to merge some information from the Comparative Toxicogenomics Database (CTD). Specifically, the chemical-gene and chemical-pathway relations. We can use the information extracted from CTD as features (which genes the chemical is known to interact with, and which biochemical pathway steps is known to participate in). This will be equivalent to a large "right join", where we will only consider those compounds or small molecules that have an entry in CTD. As a stretch goal, we could also obtain "fingerprints" from each of these chemicals, to also include as features. These are boolean or real-valued vectors that are generated from specialized libraries like RDkit, that reflect properties of the chemical (molecular weight, hydrophobicity, number of aromatic rings, etc). This first step could be performed using Hadoop or Spark.

- What format is it in - Structured csv

- Does the data need significant preprocessing? Yes, the dataset come from different sources and not compatible to each other. We have to aggregate, preprocess and clean them before we start to analyze them.
- Describe dataset: static
- How will the data be accessed? We could download the dataset from NTP and CTD website.
- Where will the data be stored and how? We will download the dataset and store them on S3 and collect the information to MongoDB by python scripting process.

Challenges

- The problem is hard to solve because we do not have many experts both master in computer science and biology.
- Most of us are lack of biological domain knowledge for finding the relations among all the dataset.
- We have to figure out which features are useful to build the machine learning model, and feature engineering is always a challenge for mining something useful.
- The chemical data may have a class imbalance, as the distribution of OECD toxicity codes may not be uniform.
- Data preprocessing and cleaning could be time-consuming since we need to integrate different datasets, which have incompatible formats.
- Integration of the different components in the architecture could also be a problem considering the huge size of the dataset.
- Unsure of how we want to show the results at the end. Hence visualization could be a challenge.

Timeline

- Data collection - 11/1
- Data aggregation architecture (stretch goal: incorporate RDkit features) - 11/8
- Machine learning architecture - 11/15
- Classifier model - 11/15
- Hyperparameter optimization & reclassification - 11/20
- Toxicity prediction of the rest of CTD chemicals - 11/22
- Visualization - 11/29

Responsibilities

- Ignacio Tripodi - Data Aggregation, ML Architecture, Optimization
- Chu-Sheng Ku - Classifier Model & ML Architecture
- Chi Chen - Classifier Model & Optimization

- Harshini Muthukrishnan - Toxicity Prediction & Visualization

Checkpoint 1

The scope of the project remains the same so far: to attempt classifying a chemical's toxicity based on the genes it's known to interact with, and the biochemical pathways it's known to affect.

Progress Update

Ignacio Tripodi

I completed the data collection and sanitation step. I've also identified the best ways to connect both large datasets (at least for the scope of a class project). For simplicity's sake and due to the time constraints, we will make some simplifications and assumptions from the collected datasets. We will use the CAS ID to link chemicals between the NTP and CTD datasets, as this is a universal unique identifier for a chemical compound. The CAS IDs are for the individual components of a mixture, so we will evaluate each of these separately. The most important columns of the NTP dataset will be the CAS number, the chemical name (reported as the "active ingredient"), the mixture name (sometimes a commercial product name, like "RoundUp"), and the two different toxicity classification scores from EPA (Environmental Protection Agency) and from OECD (Organization of Economic and Commercial Development, listed as "GHS" for their Global Harmonized System). These last two will serve as labels later on, for our machine learning task.

One challenge presented by the NTP dataset is that it includes different experiments from different labs that can potentially report different toxicity scores for the same chemical. For chemical components that have been evaluated by different labs and at different concentration levels, we will take a conservative approach and use the highest score reported in each case. We can also later try averaging the scores reported in each of the two categories and using the next rounded integer as the score for EPA and OECD classification, respectively. Chemicals without a given CAS number will be excluded from this analysis, as well as entries from NTP where there is no EPA or OECD toxicity score. Similarly, we will exclude from the analysis any chemicals in the CTD that do not have a CAS number associated to them.

The data from both NTP and CTD I curated has been used to populate different MongoDB collections. We want to keep the different toxicity endpoints separate (oral, dermal and inhalation) so that we can test each of these separately, since a chemical that has properties that make it cause an inhalation toxicant may not necessarily cause skin irritation on contact. The idea is, for each of the chemicals from NTP, to get a list of all the genes they are known to interact with, and what are the types of interactions on each case. Also we will link these chemicals to the pathways they are known to participate in (or disrupt). Using the genes and pathways as features for each chemical in the 3 NTP datasets, we will then proceed to guess a toxicity score for all other chemicals in CTD for both the EPA and OECD classification. A literature spot check will be perform for those chemicals being classified towards the end of the spectrum (especially those ranking as highly toxic according to machine learning results), to seek empirical validation. This will determine a loose metric of "success" for this task. A more comprehensive version of this study would recruit toxicologists that could help assess the computer-generated results to determine which are plausible.

I provided the data to be entered into MongoDB collections in various tab-delimited (*.tsv) files. The curated acute toxicity chemical datasets can be used as-is, and was pre-filtered for availability of CAS number and toxicity scores, and the CTD datasets just needed a few columns from what is on these tsv files.

I started writing the machine learning base code in Python that will be used to attempt classification once we can query MongoDB for a single chemical per row with all its associated proteins and pathways. We will begin with a simple approach of a one-hot encoding, then reduce dimensionality by using Autoencoder to a manageable feature set for classification. It will make biological sense to classify these chemicals separately for each category (oral toxicity, inhalation, etc), as the reasons resulting in a toxic outcome can be very dependent on the biochemical pathways triggered (e.g. something that is a skin irritant may not necessarily be harmful on the eyeballs).

Chu-Sheng Ku

- Wrote a Python script to populate the dataset from following tsv files to MongoDB
    - Chemical_Pathways.tsv 63.6MB
    - Chemical_Gene_IXNS.tsv 19.5MB

- Created an EC2 instance (t2.micro) to verify the data preparation process
    - Since we do not want to maintain a permanent storage fee and need to have the data stored in the database every time the server restarts, I verified this initial preparation process by creating MongoDB, provided by Harshini's script, and importing tsv files to the database.

Chi Chen

- Wrote python script to parse the raw tsv data file
    - Read tsv data into pandas dataframe
    - Filter the useless column
    - Drop the data without ID
    - Store the data back to tsv file

Harshini Muthukrishnan

- Set up mongodb server to accept data from Spark
- Wrote script to automate mongodb setup on an EC2 instance

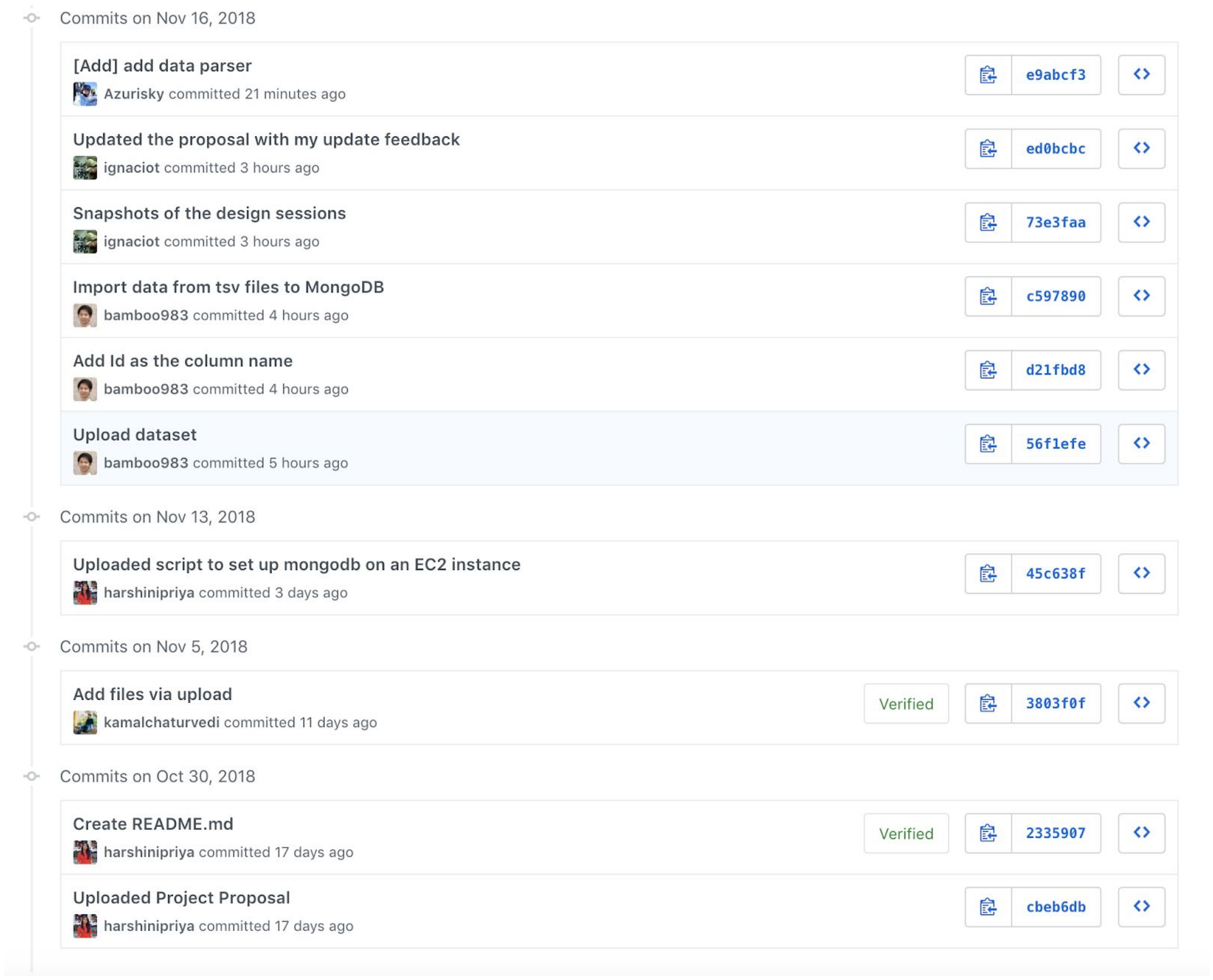
Costs

So far the costs have been nearly negligible, and related to the activation of an EC2 instance that will run MongoDB server.

Dataset Management

See Ignacio’s description above. No further data cleanup or incorporation is expected, and any additional filtering can be performed directly as part of the MongoDB query.

Snapshot of Github Commits



Commits on Nov 16, 2018

[Add] add data parser
Azurisky committed 21 minutes ago
e9abcf3

Updated the proposal with my update feedback
ignaciot committed 3 hours ago
ed0bcbc

Snapshots of the design sessions
ignaciot committed 3 hours ago
73e3faa

Import data from tsv files to MongoDB
bamboo983 committed 4 hours ago
c597890

Add Id as the column name
bamboo983 committed 4 hours ago
d21fbd8

Upload dataset
bamboo983 committed 5 hours ago
56f1efe

Commits on Nov 13, 2018

Uploaded script to set up mongodb on an EC2 instance
harshinipriya committed 3 days ago
45c638f

Commits on Nov 5, 2018

Add files via upload
kamalchaturvedi committed 11 days ago
Verified
3803f0f

Commits on Oct 30, 2018

Create README.md
harshinipriya committed 17 days ago
Verified
2335907

Uploaded Project Proposal
harshinipriya committed 17 days ago
cbeb6db