# FLUTE

*Release version 1.0*

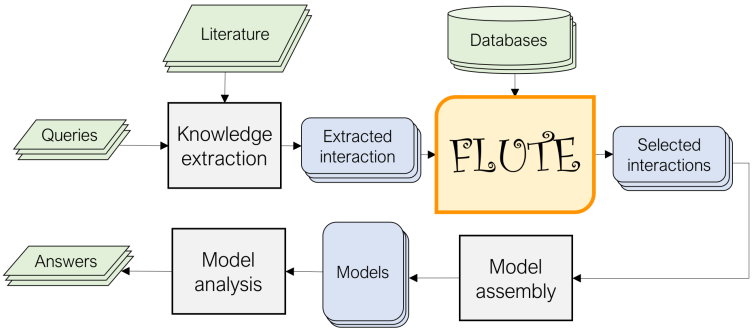
**Emilee Holtzapple**

**May 21, 2021**

# CONTENTS:

Understanding disease at the cellular level requires detailed knowledge of signaling networks. To aid in this task, many advances have been made in the field of natural language processing (NLP) to extract signaling events from biomedical literature.

However, even state-of-the-art NLP methods incorrectly interpret some signaling events described in the literature.




The FiLter for Understanding True Events (FLUTE) tool seeks to identify high-confidence signaling events from biomedical NLP output by comparing with existing biological databases. As such, FLUTE can reliably determine the confidence in the biomolecular events extracted by NLP methods and at the same time provide a speedup in event filtering by three orders of magnitude.

CHAPTER

ONE

# INSTALLATION INSTRUCTIONS

## 1.1 MySQL

1. Download the appropriate distribution of MySQL.
2. Restart your computer and add to path if necessary.
3. From the command line, access the MySQL environment by typing:

```
mysql -u root
```

If the first prompt fails, you may need to enter the password associated with your computer user account:

```
mysql -u root -p
```

3. You may choose to create a local username and password to keep your database private.
4. Install MySQL Python connector.

## 1.2 FLUTE database

1. Un-zip the FLUTE.sql file downloaded from BitBucket.
2. Log in to the MySQL environment using your username and password.
3. From there, create an empty database.
4. Log back out, and again from the command line:

```
mysql -u username -p database_name < FLUTE.sql
```

5. If you created a username and password, this will be your username in the above command, but do not enter your password above! Once you hit enter, it will prompt you for the password.
6. You can now run the "run_FLUTE.py" script, you will need to enter the database, host, username, etc. as an argument from the command line.

CHAPTER

TWO

# FLUTE USAGE

1. To filter interactions, run “run_FLUTE.py”. You must have Python3 installed.
2. **The script takes several parameters:**
    A. MySQL username
    B. MySQL password
    C. Host name - “localhost” for MacOSX, desktop name for Windows
    D. Database name (see step 3 from FLUTE DB installation instructions)
    E. Input filename
    F. Output filename for interactions
    G. Output filename for scores
3. Input files must have the following headers:

| Regulated-Name | Regulate-dID | Regulated-Type | Regulator-Name | Regula-torID | Regula-torType | Pa-perID |
|---|---|---|---|---|---|---|

4. Output files include list of reading interactions that pass filtration, and the filtration scores for those filtered interactions.

CHAPTER

# THREE

# RUN_FLUTE

This page describes the script that accesses the FLUTE database. The functions in this module ground element names and check against the FLUTE database.

## 3.1 Functions

run_FLUTE.**getRelatedPapers**(*db_user*, *db_pass*, *db_host*, *db_name*, *prot*)
This function retrieves related papers based on a protein name.

**db_user: str** Name of the MySQL user where the FLUTE DB is stored.

**db_pass: str** Password for the MySQL user where the FLUTE DB is stored.

**db_host: str** Host name for the local machine where the coopy of the FLUTE DB is stored.

**db_name: str** Name of the local copy of the FLUTE DB.

**prot: str** Input protein

Saves a file of related paper IDs

run_FLUTE.**getRelatedInts**(*db_user*, *db_pass*, *db_host*, *db_name*, *f*)
This function retrieves interactions from the same papers as

**db_user: str** Name of the MySQL user where the FLUTE DB is stored.

**db_pass: str** Password for the MySQL user where the FLUTE DB is stored.

**db_host: str** Host name for the local machine where the coopy of the FLUTE DB is stored.

**db_name: str** Name of the local copy of the FLUTE DB.

**f: str** File name of the list of papers

run_FLUTE.**getRecentPapers**(*f*)
This function should search the OA file and find all interactions occuring in papers less than X years old

**f: str** Input filename that contains list of interactions

None

run_FLUTE.**getDups**(*f*)
This function calculates the number of occurences of an interaction in a reading set.

**f: str** Filename of the list of interactions to be counted.

None

run_FLUTE.**getArgs**()

run_FLUTE.**convID**(*db_user*, *db_pass*, *db_host*, *db_name*, *X*)
This function uses the FLUTE DB to ground interactions

**db_user: str** Name of the MySQL user where the FLUTE DB is stored.

**db_pass: str** Password for the MySQL user where the FLUTE DB is stored.

**db_host: str** Host name for the local machine where the coopy of the FLUTE DB is stored.

**db_name: str** Name of the local copy of the FLUTE DB.

**X: numpy array** Array containing all grounded interactions

**X: numpy array** Grounded ineractions

run_FLUTE.**findInts**(*db_user*, *db_pass*, *db_host*, *db_name*, *ints*, *es*, *ts*, *ds*)
This function uses the FLUTE DB to filter interactions

**db_user: str** Name of the MySQL user where the FLUTE DB is stored.

**db_pass: str** Password for the MySQL user where the FLUTE DB is stored.

**db_host: str** Host name for the local machine where the coopy of the FLUTE DB is stored.

**db_name: str** Name of the local copy of the FLUTE DB.

**X: numpy array** Array containing all filtered interactions

**X: numpy array** Filtered interactions

run_FLUTE.**uniOnly**(*allInts*)
This function returns only proteins

**allInts: numpy array** All interactions from the input file

**rel_ints: numpy array** Protein-protein interactions only

run_FLUTE.**getChem**(*X*)
This function returns only protein-chemical interactions

**X: numpy array** All interactions from the input file

**X: numpy array** Protein-chemical interactions only

run_FLUTE.**getGo**(*a*)
This function returns only protein-biological process interactions

**a: numpy array** All interactions from the input file

**a: numpy array** Protein-biological process interactions only

## 3.2 Dependencies

**Python**:

pandas library

csv module

numpy library

MySQL Connector for Python3 library

argparse library re library

CHAPTER

FOUR

# LEGAL

Add here any information concerning usage, dowloads, and repurposing.

CHAPTER

FIVE

# LICENSING AND FUNDING

Supported by DARPA award..

CHAPTER

SIX

# INDICES AND TABLES

- genindex
- modindex
- search

# INDEX