Prerequisite Packages

```
# prerequisites
!pip3 install cloudmesh-installer
!pip3 install cloudmesh-common
!pip3 install kaggle
```

```
  Downloading https://files.pythonhosted.org/packages/87/89/479dc97e18549e21354893e4e
Requirement already satisfied: py>=1.5.0 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: atomicwrites>=1.0 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: attrs>=17.4.0 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: pluggy<0.8,>=0.5 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: more-itertools>=4.0.0 in /usr/local/lib/python3.7/dist
Requirement already satisfied: six>=1.10.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: pip>=6.0.0 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/
Requirement already satisfied: click>=7 in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: typing-extensions>=3.6.4; python_version < "3.8" in /u
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (f
Building wheels for collected packages: python-hostlist, ordered-set
  Building wheel for python-hostlist (setup.py) ... done
  Created wheel for python-hostlist: filename=python_hostlist-1.21-cp37-none-any.whl
  Stored in directory: /root/.cache/pip/wheels/0b/5b/55/ddcf52288f0b10f4564ca1b253159
  Building wheel for ordered-set (setup.py) ... done
  Created wheel for ordered-set: filename=ordered_set-4.0.2-py2.py3-none-any.whl size
  Stored in directory: /root/.cache/pip/wheels/e1/c6/9b/651d8a21d59b51a75ab9c070838f9
Successfully built python-hostlist ordered-set
ERROR: pytest-cov 2.11.1 has requirement coverage>=5.2.1, but you'll have coverage 3.
ERROR: pytest-cov 2.11.1 has requirement pytest>=4.6, but you'll have pytest 3.6.4 wh
Installing collected packages: bump2version, oyaml, simplejson, python-hostlist, colo
Successfully installed bump2version-1.0.0 cloudmesh-common-4.3.66 cloudmesh-installer
Requirement already satisfied: cloudmesh-common in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: simplejson in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: tabulate in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: pyfiglet in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from c
Requirement already satisfied: oyaml in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: colorama in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from c
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: python-hostlist in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: psutil in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: humanize in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: pathlib in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: pyyaml in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-pa
```

```
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: kaggle in /usr/local/lib/python3.7/dist-packages (1.5.
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: urllib3 in /usr/local/lib/python3.7/dist-packages (frc

Requirement already satisfied: python-dateutil in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages (frc
Requirement already satisfied: python-slugify in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from k
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages
```

Sign Up for Kaggle Account and Generate API Token

1. Sign Up for Kaggle account at https://www.kaggle.com.
2. In Kaggle 'Profile'->'Account', generate api token by clicking 'Create New API Token'.

Upload Token into Colab

3. Upload json file, with Kaggle api token and username, into Colab by running code below. Select downloaded 'kaggle.json' file when prompted and click 'Uplaod'.

NB third-party cookies should be enabled for upload to work.

```
from google.colab import files

# upload json file with api token and username
files.upload()

# create directory for token in Colab
!mkdir ~/.kaggle

# move file to directory
!mv kaggle.json ~/.kaggle/

# modify permissions on directory
!chmod 600 ~/.kaggle/kaggle.json
```

Choose Files kaggle.json

- **kaggle.json**(application/json) - 67 bytes, last modified: 5/3/2021 - 100% done

```
Saving kaggle.json to kaggle.json
```

Download Merck Molecular Activity Challenge Dataset

```
# download dataset
!kaggle competitions download -c MerckActivity

# make train and test directories for raw data
!mkdir -p /content/raw/train /content/raw/test

# unzip train and test datasets into respective folders; only overwrite exiting files if extr
!unzip /content/TrainingSet.zip -d /content/raw/train

!unzip /content/TestSet.zip -d /content/raw/test

# paths to raw data
path_to_raw_train_data = '/content/raw/train/TrainingSet/'
path_to_raw_test_data = '/content/raw/test/TestSet/'
```

```
Warning: Looks like you're using an outdated API Version, please consider updating (ser
TestSet.7z: Skipping, found more recently modified local copy (use --force to force dow
Rsquared.R: Skipping, found more recently modified local copy (use --force to force dow
ntree20_benchmark.R: Skipping, found more recently modified local copy (use --force to
ntree20_basicBenchmark.csv.zip: Skipping, found more recently modified local copy (use
TrainingSet.7z: Skipping, found more recently modified local copy (use --force to force
TrainingSet.zip: Skipping, found more recently modified local copy (use --force to forc
TestSet.zip: Skipping, found more recently modified local copy (use --force to force do
Archive:  /content/TrainingSet.zip
   creating: /content/raw/train/TrainingSet/
  inflating: /content/raw/train/TrainingSet/ACT10_competition_training.csv
  inflating: /content/raw/train/TrainingSet/ACT11_competition_training.csv
  inflating: /content/raw/train/TrainingSet/ACT12_competition_training.csv
  inflating: /content/raw/train/TrainingSet/ACT13_competition_training.csv
  inflating: /content/raw/train/TrainingSet/ACT14_competition_training.csv
  inflating: /content/raw/train/TrainingSet/ACT15_competition_training.csv
  inflating: /content/raw/train/TrainingSet/ACT1_competition_training.csv
  inflating: /content/raw/train/TrainingSet/ACT2_competition_training.csv
  inflating: /content/raw/train/TrainingSet/ACT3_competition_training.csv
  inflating: /content/raw/train/TrainingSet/ACT4_competition_training.csv
  inflating: /content/raw/train/TrainingSet/ACT5_competition_training.csv
  inflating: /content/raw/train/TrainingSet/ACT6_competition_training.csv
  inflating: /content/raw/train/TrainingSet/ACT7_competition_training.csv
  inflating: /content/raw/train/TrainingSet/ACT8_competition_training.csv
  inflating: /content/raw/train/TrainingSet/ACT9_competition_training.csv
Archive:  /content/TestSet.zip
   creating: /content/raw/test/TestSet/
  inflating: /content/raw/test/TestSet/ACT10_competition_test.csv
  inflating: /content/raw/test/TestSet/ACT11_competition_test.csv
  inflating: /content/raw/test/TestSet/ACT12_competition_test.csv
  inflating: /content/raw/test/TestSet/ACT13_competition_test.csv
  inflating: /content/raw/test/TestSet/ACT14_competition_test.csv
  inflating: /content/raw/test/TestSet/ACT15_competition_test.csv
  inflating: /content/raw/test/TestSet/ACT1_competition_test.csv
  inflating: /content/raw/test/TestSet/ACT2_competition_test.csv
  inflating: /content/raw/test/TestSet/ACT3_competition_test.csv
  inflating: /content/raw/test/TestSet/ACT4_competition_test.csv
  inflating: /content/raw/test/TestSet/ACT5_competition_test.csv
  inflating: /content/raw/test/TestSet/ACT6_competition_test.csv
```

```
  inflating: /content/raw/test/TestSet/ACT7_competition_test.csv
  inflating: /content/raw/test/TestSet/ACT8_competition_test.csv
  inflating: /content/raw/test/TestSet/ACT9_competition_test.csv
```

Preprocess Dataset

```
import pandas as pd
from cloudmesh.common.StopWatch import StopWatch

# make train and test directories for preprocessed data
!mkdir -p /content/preprocessed/train /content/preprocessed/test

# paths to processed data
path_to_preprocessed_train_data = '/content/preprocessed/train/'
path_to_preprocessed_test_data = '/content/preprocessed/test/'

# cycle through 15 data sets preprocessing them for learning
StopWatch.start("preprocessing")
dataset_file_no = 1

while dataset_file_no <= 15:

    dataset_train_file_name = 'ACT' + str(dataset_file_no) + '_competition_training.csv'
    dataset_test_file_name = 'ACT' + str(dataset_file_no) + '_competition_test.csv'

    train_filename = path_to_raw_train_data + dataset_train_file_name
    test_filename =  path_to_raw_test_data + dataset_test_file_name

    train_filename_processed = path_to_preprocessed_train_data + dataset_train_file_name
    test_filename_processed = path_to_preprocessed_test_data + dataset_test_file_name

    print ('Preprocessing dataset ', 'ACT' + str(dataset_file_no))

    train = pd.read_csv(train_filename)
    test = pd.read_csv(test_filename)

    print (len(train.columns.values))
    print (len(test.columns.values))

    train_inx_set = set(train.columns.values)
    test_inx_set = set(test.columns.values)

    # remove molecule label and columns that are not common to both training and test sets
    train_inx = [inx for inx in train.columns.values if inx in set.intersection(train_inx_set
    test_inx = [inx for inx in test.columns.values if inx in set.intersection(train_inx_set,

    train_inx.insert(0,'Act')
    train_inx.remove('MOLECULE')
```

```
        test_inx.remove('MOLECULE')

        #print (train_inx)
        #print (test_inx)

        train = train[train_inx]
        test = test[test_inx]

        #print (train.shape)
        #print (test.shape)

        # save data to csv
        train.to_csv(train_filename_processed, index=False)
        test.to_csv(test_filename_processed, index=False)

        print(train.head(5))
        print ('Preprocessing dataset ', 'ACT' + str(dataset_file_no), ' complete')

        dataset_file_no += 1
    StopWatch.stop("preprocessing")

    StopWatch.benchmark()
```

```
5555
      Act  D_6  D_37  D_38  D_39  ...  D_10963  D_10995  D_11012  D_11027  D_11030
0  8.6182    0     0     0     0  ...        0        0        0        0        0
1  9.8521    0     0     0     0  ...        0        0        0        0        0
2  8.3264    0     0     0     0  ...        0        0        0        0        0
3  8.2581    0     0     0     0  ...        0        0        0        0        0
4  7.3552    0     0     0     0  ...        0        0        0        0        0

[5 rows x 5553 columns]
Preprocessing dataset  ACT15  complete

+--------------------+-----------------------------------------------------------------
| Attribute          | Value
|--------------------+-----------------------------------------------------------------
| BUG_REPORT_URL     | "https://bugs.launchpad.net/ubuntu/"
| DISTRIB_CODENAME   | bionic
| DISTRIB_DESCRIPTION | "Ubuntu 18.04.5 LTS"
| DISTRIB_ID         | Ubuntu
| DISTRIB_RELEASE    | 18.04
| HOME_URL           | "https://www.ubuntu.com/"
| ID                 | ubuntu
| ID_LIKE            | debian
| NAME               | "Ubuntu"
| PRETTY_NAME        | "Ubuntu 18.04.5 LTS"
| PRIVACY_POLICY_URL | "https://www.ubuntu.com/legal/terms-and-policies/privacy-poli
| SUPPORT_URL        | "https://help.ubuntu.com/"
| UBUNTU_CODENAME    | bionic
| VERSION            | "18.04.5 LTS (Bionic Beaver)"
| VERSION_CODENAME   | bionic
| VERSION_ID         | "18.04"
| cpu_count          | 4
| mem.active         | 1.0 GiB
```

```
| mem.available       | 24.4 GiB
| mem.free            | 938.3 MiB
| mem.inactive        | 22.8 GiB
| mem.percent         | 4.3 %
| mem.total           | 25.5 GiB
| mem.used            | 10.9 GiB
| platform.version    | #1 SMP Thu Jul 23 08:00:38 PDT 2020
| python              | 3.7.10 (default, May  3 2021, 02:48:31)
|                     | [GCC 7.5.0]
| python.pip          | 19.3.1
| python.version      | 3.7.10
| sys.platform        | linux
| uname.machine       | x86_64
| uname.node          | 7e0207301e13
| uname.processor     | x86_64
| uname.release       | 4.19.112+
| uname.system        | Linux
| uname.version       | #1 SMP Thu Jul 23 08:00:38 PDT 2020
| user                | collab
+---------------------+-------------------------------------------------------------------

+---------------+----------+---------+---------+---------------------+-------+-------
| Name          | Status   |    Time |     Sum | Start               | tag   | Node
|---------------+----------+---------+---------+---------------------+-------+-------
| preprocessing | ok       | 536.097 | 536.097 | 2021-05-07 17:08:11 |       | 7e0207
+---------------+----------+---------+---------+---------------------+-------+-------
```

Predicting Molecular Activity

```
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function
import numpy as np
import pandas as pd
import keras.backend as K
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout, InputLayer, Concatenate
from keras.utils import to_categorical, plot_model
from keras.utils.vis_utils import model_to_dot
from keras.optimizers import Adam
from cloudmesh.common.StopWatch import StopWatch

# make outputs directory
!mkdir -p /content/outputs/

path_to_train_data = '/content/preprocessed/train/'
path_to_test_data = '/content/preprocessed/test/'
path_to_outputs = '/content/outputs/'

# define parameters
```

```
    # define parameters
    hidden_units = 512
    dropout = 0.45
    BATCH_SIZE = 128
    feature_dim = 128
    opti = Adam(lr=0.0001, beta_1=0.5)

    # define fully connected network/MLP
    def fcn_model(input_shape=(feature_dim,)):
        model = Sequential()
        model.add(Activation('relu'))
        model.add(Dropout(dropout))
        model.add(Dense(hidden_units))
        model.add(Activation('relu'))
        model.add(Dropout(dropout))
        model.add(Dense(hidden_units))
        model.add(Activation('relu'))
        model.add(Dropout(dropout))
        model.add(Dense(hidden_units))
        model.add(Activation('relu'))
        model.add(Dropout(0.10))
        model.add(Dense(num_labels))
        model.add(Activation('softmax'))
        model.build(input_shape)
        model.summary()
        model.compile(loss='mean_squared_error', optimizer=opti, metrics=[Rsqured])
        print("\nTraining on dataset number", act_ds," of 15:\n")
        model.fit(x_train, y_train, epochs=15, batch_size=BATCH_SIZE)
        loss, R2 = model.evaluate(x_test,y_test,batch_size=BATCH_SIZE)
        print("\nCorrelation coefficient:", R2)
        return model

    # define correlation coefficient (R^2) formula
    def Rsqured(x,y):

        # approach adopted from RuwanT
        # URL: https://github.com/RuwanT/merck/blob/master/main.py

        x = K.batch_flatten(x)
        y = K.batch_flatten(y)

        avx = K.mean(x)
        avy = K.mean(y)

        num = K.sum((x-avx) * (y-avy))
        num = num * num

        denom = K.sum((x-avx)*(x-avx)) * K.sum((y-avy)*(y-avy))

        return num/denom

    # iterate through 15 distinct datasets of high throughput screening (HTS) assays
```

```
    StopWatch.start("train-evaluate-predict")
    act_ds = 1
    while act_ds <= 15:
        print("\nReading from dataset", act_ds,"of 15:\n")
        data_train_main = pd.read_csv(path_to_train_data + 'ACT' + str(act_ds) + '_competition_tr
        data_ac = pd.read_csv(path_to_test_data + 'ACT' + str(act_ds) + '_competition_test.csv')

        # split each of the datasets into set for training (80%), set for testing/evluation (10%)
        # set for use with validating prediction (10%)
        data_train = data_train_main.sample(frac = 0.8)
        data_test = data_train_main.drop(data_train.index)
        data_prediction = data_test.sample(frac = 0.5)
        data_test = data_test.drop(data_prediction.index)

        activity_inx = data_train.columns.get_loc('Act')
        feature_dim = data_train.shape[1] - (activity_inx+1)
        #print("no. of feature columns:", feature_dim)

        # identify molecular activity labels
        y_train = data_train['Act']
        #print("shape of y_train:", y_train.shape)
        num_labels = len(np.unique(y_train))
        #print("no. of unique labels:", num_labels)
        y_test = data_test['Act']

        # identify and filter for feature-set/molecular-substructure frequencies
        train_set_inx = set(data_train.columns.values)
        test_set_inx = set(data_ac.columns.values)

        train_inx = [inx for inx in data_train.columns.values if inx in set.intersection(train_se
        test_inx = [inx for inx in data_test.columns.values if inx in set.intersection(train_set_
        predict_inx = [inx for inx in data_prediction.columns.values if inx in set.intersection(t

        data_train = data_train[train_inx]
        data_test = data_test[test_inx]
        data_prediction = data_prediction[predict_inx]
        #print(data_train.head(5))

        # format feature-set input data
        x_train = data_train[0:]
        x_train = np.asarray(x_train).astype('float32')
        x_test = data_test[0:]
        x_test = np.asarray(x_test).astype('float32')
        x_predict = data_prediction[0:]
        x_predict = np.asarray(x_predict).astype('float32')

        input_size = x_train.shape[0]


        # call fully connected network for learning and evaluation. Predict biological activity f
        y_predict = fcn_model(input_shape=(input_size,feature_dim)).predict(x_predict,batch_size=
```

```
    # display inputs and outputs for the prediction step and write results to text file

    prediction_file = open(path_to_outputs +"molecular_activity_prediction.txt","a")

    for i in range(len(x_predict)):
      print("Substructure/feature frequencies=%s, Predicted biological activity=%s" % (x_pred
      prediction_file.write(str(x_predict[i])+","+str(np.mean(y_predict[i]))+"\n")
    act_ds +=1
    prediction_file.close()
StopWatch.stop("train-evaluate-predict")

StopWatch.benchmark()
```

```
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
 Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
```

```
Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
Substructure/feature frequencies=[0. 0. 0. ... 0. 0. 0.], Predicted biological activi
```

✓ 7m 47s completed at 2:43 PM