Using the “dev” version

1. [Host] Login as “root”
2. [Host] Pull the development environment from DockerHub
   `# docker pull robinlin/de-identification:dev`
3. [Host] Pull project from Github
   `# git clone https://github.com/JiaMingLin/de-identification.git`
   Note: Suppose the project clone is under the path `/root/git/de-identification`
4. [Host] Launch the development environment
   `# export PROJECT_PATH=/root/git/de-identification`
   `# export MAPPING_PATH=/opt/de-identification`
   `# export IMAGE_NAME=robinlin/de-identification:dev`
   `# docker run -itd -v $PROJECT_PATH:$MAPPING_PATH -p 8080:8080 $IMAGE_NAME /bin/bash`
5. [Host] The development environment has been started.

Note: Under the development environment, the web-server is not activated. To start it, please follow these steps below

1. [Container] On the Path “`/opt/de-identification`”.
2. [Container] Starting web-server
   `# ./start.sh`

Known Issue: The editor “**vim**” was not installed in the container. Please install it manually.

1. [Container] Install editor “vim”.
   `# apt-get install vim -y`

Placing the Original Files

1. [Host] Put the original file in the directory
   `/root/git/de-identification/static/test/exp/`.
   (suppose your project folder path is `/root/git/de-identification`)

Modify the Experimental Configs

1. [Container] On the path “/opt/de-identification”
2. [Container] Using “vim” to modify experimental configurations.
   `# vim test/test_accuracy.py`
3. [test_accuracy.py] One should only modify the values of the parameters in `setUp`

```
def setUp(self):

        self.data_dir = os.path.join(c.TEST_FILE_PATH, 'exp')
        # epsilon.1
        self.eps1_levels = [3]

        # noises
        self.privacy_levels = [2,3]

        # k
        self.k_val = [10,50,250]

        # number of runs
        self.nrun = 1

        # test cases
        self.cases = [
                (
                        "data2",
                        []
                        #[["Age", "Income", "TRV"]]
                )
        ]

        # specified data domain
        self.specified_data_domain = True
```

4. [test_accuracy.py] Parameters explain
   [1] **eps1_levels:** Integer array type.
   The mapping parameters of randomized construct dependency graph.
   The corresponding values are as below

```
corr = {
        1:0.05,
        2:0.5,
        3:5,
        4:50,
        5:500
}
```

[2] **privacy_levels:** integer array type.
The mapping of Laplace noise parameters.
The corresponding values are as below.

```
corr = {
        1: 0.01,
        2: 0.1,
        3: 1,
        4: 10,
        5: 100
}
```

[3] **k_val:** Integer array type.
The k-aggregate parameter

[4] **nrun:** Integer.
The number of runs

[5] **cases:** Tuple array type.
The first parameter of tuple is file name, and the second is the user-customized relationships on dependency graph.

[6] **specified_data_domain:** boolean.
If one wants to using a specified data domain file, set to be “True”.

Experimental Tests

Be sure you have understand and go through the steps in **Placing the Original Files** and **Modify the Experimental Configs.**

1. [Host] Login as “root”.
2. [Host] Jumping into the container (development environment).
3. [Container] On the path “/opt/de-identification”.
4. [Container] Clean the environments and reset database
   `# ./clean.sh`
5. [Container] Run Experimental Tests
   `# python manage.py test -pattern="test_accuracy.py"`

6. [Host] The result would put in the directory
   /root/git/de-identification/mediate_data/task_1/