Requirement

- Python

  Useful links
  - It is possible to install another version of Python to Virtualenv?
  - Custom Installation Locations
  - So you want to use python on the Mac?

- Django

  Useful links
  - Django Tricks

- MDanalysis

  Useful links
  - Enthought Python Distribution (EPD)

Install

1. Download ST-Analyzer from Git-hub
   - Manual download: https://github.com/jcjeong/Membrane-Analyzer
   - Git clone (using commandline): git clone git@github.com:jcjeong/Membrane-Analyzer.git

2. Configuration

   Let's assume ST-Analyzer is stored into /home/your_account/Membrane-Analyzer/stanalyzer

   i. You now can see files including 'manage.py' 'stanalyzer.db' and directories including 'gui', 'stanalyzer', 'templates', and 'trajectory'
   - ***manage.py***: executable python file to run Django server
   - ***stanalyzer.db***: Database file to store information (e.g. account, project, jobs and etc). Initial account and password are 'admin' and '12345' respectively.
   - ***gui***: most important part of ST-analyzer including 'models' and 'views'
   - ***media***: default directory to store the output of Jobs
   - ***static***: directory storing APIs and functions used in ST-Analyzer
   - ***stanalyzer***: directory containing system setup files
   - ***templates***: containing template files
   - ***trajectory***: containing sample trajectory files - "It is okay to remove this directory!"

   ii. Check out DB consistency

   ```
   1) you@system> cd /home/your_account/Membrane-Analyzer/stanalyzer
   ```

2) you@system> python manage.py syncdb

iii. Run Django to launch ST-Analyzer
1) you@system> cd /home/your_account/Membrane-Analyzer/stanalyzer
2) you@system> python manage.py runserver [8000] # the number corresponding to the port that you want to communicate with ST-Analyzer

iv. Open your browser and connect to ST-Analyzer through http://127.0.0.1:8000/
- You will see the ST-Analyzer login.
- Initial account and password are 'admin' and '12345'

Known Issues

- Firewall
  - Due to firewall, certain system environment may not allow to communicate between client and server through Django. To go around this problem, you can use ssh trick by forwarding the port to local port
    ```
    you@system> cd ~/.ssh
    you@ysstem> vi config
        Edit 'config' file as following:
        Host any_name
        HostName your.server.com
        LocalForward 8000 127.0.0.1:8000
    ```
  - If you need more information about ssh tricks please visit here

- Running Django in background: this will not work behind firewall due to the requirement of forwarding local port
  - you@system> cd /your_home/stanalyzer
  - you@system> python manage.py runserver & # run django background
  - you@system> jobs # checkout job number in same console
  - you@system> disown -h job_number # you now are safe with logout
  - you@system> kill -9 pid # shutdown Django