

ST-analyzer Tutorial

Ver. 0.1.0

Committee

Wonpil Im
Chair, University of Kansas

Jong Cheol Jeong
Developer, University of Kansas

Sunhwan Jo
University of Chicago

Yifei Qi
University of Kansas

Jeffery B. Klauda
University of Maryland

Lev Gorenstein
Purdue University

Min Sun Yeom
Korea Institute of Science and Technology Information

All rights reserved @ ImLab

This work is supported by NSF ABI



Table of Contents

Objective	1
Installation	2
Required modules & programs	2
<i>Python (*required)</i>	2
<i>Django (*required)</i>	4
<i>MDAnalysis (*required)</i>	5
<i>Pyhull (*optional)</i>	5
<i>ST-analyzer (*required)</i>	6
ST-analyzer GUI	9
Login	9
<i>Username & Password</i>	9
Workspace	10
<i>Menu bar</i>	10
<i>Button for minimizing window</i>	10
User accounts	11
<i>Create, delete, and edit account</i>	11
<i>Searching and ordering the lists</i>	12
<i>Logout 'admin' and login with new account</i>	12
ST-analyzer	12
<i>STEP1</i>	13
<i>Preparing job submission</i>	18
<i>STEP2</i>	27
Analysis modules	29
<i>System size</i>	30
<i>Helix tilt</i>	31
<i>Sterol tilt</i>	32
<i>RMSD</i>	34
<i>RMSF</i>	35
<i>Density Profiles</i>	36
<i>Order Parameters</i>	43
<i>Thickness</i>	44
<i>Average area per lipid</i>	47
Result Viewer	48
<i>Project Retrieval</i>	48
<i>Outputs</i>	49
<i>Page navigator</i>	50
<i>Data manager</i>	51
Selection Query	52
Selection keywords and usages	52

Objective

ST-analyzer is a standalone GUI toolset to perform various analyses of molecular dynamics simulation trajectories and provides a variety of analysis routines especially focused on membrane systems (e.g., lipid chain order parameter, lipid area, etc). Since trajectory files are generally too large to be uploaded to a remote server, ST-analyzer has been developed in cross-platform by installing it in a server where trajectories are located. Once ST-analyzer is installed, user's local machines with any types of existing OS can access ST-analyzer through HTTP. ST-analyzer is freely available through Github (<https://github.com/stanalyzer/ST-analyzer>).

Installation

Although ST-analyzer is developed in Python codes, to cover the wide range of analysis demands and to maintain the cross-platform characteristics, some external python modules and programs are required. For the users who only need to run particular modules, this documentation makes distinct 'required modules' from 'optional modules'.

Required modules & programs

Python (***required**)

Python v2.7 or above is required.

Python is available through <http://www.python.org/download/>

If you do not have privilege to install Python:

This issue is out of scope in this tutorial, but you may find more information as following:

Windows:

- Portable Python: <http://portablepython.com/>

OS X, Linux & Unix:

- **Install with virtualenv**
 - *Download python and install it your local directory*
 - mkdir ~/src
 - cd src
 - curl -O <http://www.python.org/ftp/python/2.7.5/Python-2.7.5.tgz>
 - tar -xvf Python-2.7.5.tgz
 - cd Python2.7.5
 - mkdir ~/localpython
 - ./configure --prefix=/path/for/your/localpython
 - make
 - make install
 - **Install virtualenv**
 - cd ~/src
 - curl -O <https://pypi.python.org/packages/source/v/virtualenv/virtualenv-1.9.1.tar.gz>
 - tar -xvf virtualenv-1.9.1.tar.gz
 - cd virtualenv-1.9.1
 - ~/localpython/bin/python setup.py install
 - **Create virtual environment**
 - mkdir ~/newenvs
 - cd ~/newenvs
 - ~/localpython/bin/virtualenv py2.7 --python /use absolute path /localpython/bin/python2.7

- **Activate virtual environment**
 - cd ~/newenvs/py2.7/bin
 - source ./activate
 - *Install any packages whatever you want*
- **Deactivate virtual environment (activate will show 'py2.7' sign followed by command prompt)**
 - (py2.7)\$deactivate
 - This part of article is referred from <http://jessenoller.com/2009/03/16/so-you-want-to-use-python-on-the-mac/>

Other information:

Using easy_install:

- <http://stackoverflow.com/questions/7465445/how-to-install-python-modules-without-root-access>
- <http://peak.telecommunity.com/DevCenter/EasyInstall#custom-installation-locations>
- <http://stackoverflow.com/questions/5506110/it-is-possible-to-install-another-version-of-python-to-virtualenv>

Django (*required)

ST-analyzer is optimized Django v1.4.1.

Django is available through <https://www.djangoproject.com/download/>

You can check Django version in python interactive mode as following:

```
>>>import django  
>>>django.get_version()          OR          >>>django.VERSION
```

Need to maintain different Django version?

<http://djangotricks.blogspot.com/2008/09/note-on-python-paths.html>

MDAnalysis (*required)

ST-analyzer is optimized MDAnalysis v0.7.6 and above.

MDAnalysis requires other modules; therefore, to make the installation simple, we encourage installing all-in-one package. Following list of packages has their own copyright, so please visit their websites and check the eligibility prior to the installation.

All-in-one package

- Anaconda (<http://continuum.io/downloads.html>) for Linux, Windows and Mac.
- Enthought Canopy (<https://www.enthought.com/products/canopy/>) for Linux, Windows and Mac.

Install MDAnalysis

- Install one of the all-in-one packages listed above
- Download and install MDAnalysis through <https://code.google.com/p/mdanalysis/>
- Details of installing MDAnalysis can be found in <https://code.google.com/p/mdanalysis/wiki/Install>
- We have been reported about problems of the installation. Most problems are caused by outdated version of Python and GNU C compiler (<http://gcc.gnu.org/>). If you have problems with install, please check the version of your GNU C compiler and Python and discuss with your system administrator
- For more questions about installation, please use discussion board at <https://code.google.com/p/mdanalysis/wiki/Install>
-

Pyhull (*optional)

Pyhull is Python wrapper to qhull (<http://www.qhull.org>) used for calculating ‘area-per-lipid’ in ST-analyzer. If calculating area-per-lipid is not necessary, users are not required installing this module.

Details of instruction for installing Pyhull can be found at <http://pythonhosted.org/pyhull/>

ST-analyzer (*required)

Download ST-analyzer from Git-hub: choose one of methods shown below

- Manual download: <https://github.com/stanalyzer/ST-analyzer>

Simulation Trajectory Analysis for Molecular Dynamics — Edit

44 commits 1 branch 0 releases 1 contributor

branch: master ST-analyzer / +

area per lipid and sterol tilt

stanalyzer authored 5 days ago latest commit 9bcefabb4a

File	Description	Last Commit
apps	area per lipid calculation	15 days ago
doc	Add density module	4 months ago
stanalyzer	area per lipid and sterol tilt	5 days ago
README.md	update gui	6 months ago
ST-analyzer.pdf	GUI update	5 months ago

README.md

ST-Analyzer

Simulation Trajectory Analysis for Molecular Dynamics

System diagram

Users ↔ St-Analyzer ↔ Cluster ↔ DB Server

CLICK Here!

SSH clone URL
git@github.com:st...
You can clone with HTTPS, SSH, Subversion, and other methods.

Clone in Desktop

Download ZIP

The screenshot shows the GitHub repository page for 'stanalyzer / ST-analyzer'. It displays basic repository statistics: 44 commits, 1 branch, 0 releases, and 1 contributor. Below this, a list of recent commits is shown. On the right side, there's a sidebar with links like 'Code', 'Issues', 'Pull Requests', 'Wiki', 'Pulse', 'Graphs', 'Network', and 'Settings'. Further down, there's an 'SSH clone URL' field and two buttons: 'Clone in Desktop' and 'Download ZIP'. A large red arrow points from the text 'CLICK Here!' to the 'Download ZIP' button. At the bottom of the main content area, there's a 'System diagram' illustrating the architecture: 'Users' interact with 'St-Analyzer', which in turn interacts with a 'Cluster' and a 'DB Server'. The 'Cluster' and 'DB Server' are shown with bidirectional arrows between them.

Figure 1 Manual download through Github

- Unzip the zip file: unzip ST-analyzer-master.zip

- Git clone (using commandline): git clone git@github.com:stanalyzer/ST-analyzer.git

```
[jcjeong@stanalyzer]$ git clone git@github.com:stanalyzer/ST-analyzer.git
Initialized empty Git repository in /Users/jcjeong/ST-analyzer/.git/
remote: Counting objects: 9326, done.
remote: Compressing objects: 100% (4882/4882), done.
remote: Total 9326 (delta 4290), reused 8764 (delta 3732)
Receiving objects: 100% (9326/9326), 129.10 MiB | 8493 KiB/s, done.
Resolving deltas: 100% (4290/4290), done.
Checking out files: 100% (2845/2845), done.
```

Figure 2. Using *git clone* to install ST-analyzer

Configuration

Let's assume ST-analyzer is stored into \$LOC=/home/your_account/ST-analyzer/st analyzer
At \$LOC, you can see following files and directories.

- *manage.py*: required to run Django server
- *stanalyzer.db*: database file used for ST-analyzer. ID: *admin*, Password: *12345*
- *gui*: directory containing 'models' and 'views'
- *media*: default directory storing the results
- *static*: directory storing APIs and background modules
- *stanalyzer*: directory containing system setup files
- *templates*: containing template files for ST-analyzer GUI
- *trajectory*: containing sample trajectory files

Checking DB consistency

At your system command line prompt, use followings:

- user@stanalyzer> cd /home/your_account/ST-analyzer/st analyzer
- user@stanalyzer> python manage.py syncdb

Run Django to launch ST-analyzer

- user@stanalyzer> cd /home/your_account/ST-analyzer/st analyzer
- user@stanalyzer> python manage.py runserver 8000
the number '8000' are used as a port number communicating with ST-analyzer. Thus the port number can be changed

Forwarding port

Use ssh configuration to forward port

- user@stanalyzer> cd ~/.ssh
- user@stanalyzer> vi config

Edit 'config' file as following:

Host *any_name*
HostName *your.server.com*
LocalForward 8000 127.0.0.1:8000

Use command

ssh -L 8000:localhost:8000 *userid@your.hostname.edu*

Connecting to ST-analyzer through your web-browser

- Open your terminal and connect server where ST-analyzer is installed by using 'Forwarding port' described above
- Connect ST-analyzer through http://127.0.0.1:8000
- You will see the ST-analyzer login.
- Initial account and password are 'admin' and '12345'

ST-analyzer GUI

GUI of ST-analyzer is designed upon three principles: *simple*, *neat*, and *useful*. From the intuitive interface, we found that many users did not face any problems while using ST-analyzer without having any information prior to their first attempts. Therefore, this section may not be necessary for most of users. However, this section intends to deliver more detailed and important information which can be helpful for both inexperienced and expertise to efficiently utilize ST-analyzer.

Login

Username & Password

ST-analyzer is designed for multiple user environments. This means that the contents or work environment belonging to an account is independent to each other. By using this concept, users can possess multiple IDs to store or work multiple tasks independently. The initial **user name** and **password** is 'admin' and '12345' respectively.

Notice: 'admin' account is just an initial account to create users' personal account. This means 'admin' account does not have any privilege such as accessing and modify other data.



Figure 3. Login menu

Workspace

Menu bar

Menu bar denoted as number (1) and (2) has two submenus ‘ST-analyzer’ and ‘About us’ containing help document and link for Im Lab website. The menu, ST-analyzer denoted as (1) in the figure 4 shows current user information and ‘Logout’ menu. To protect users data, user should logout to close current session by clicking this menu.

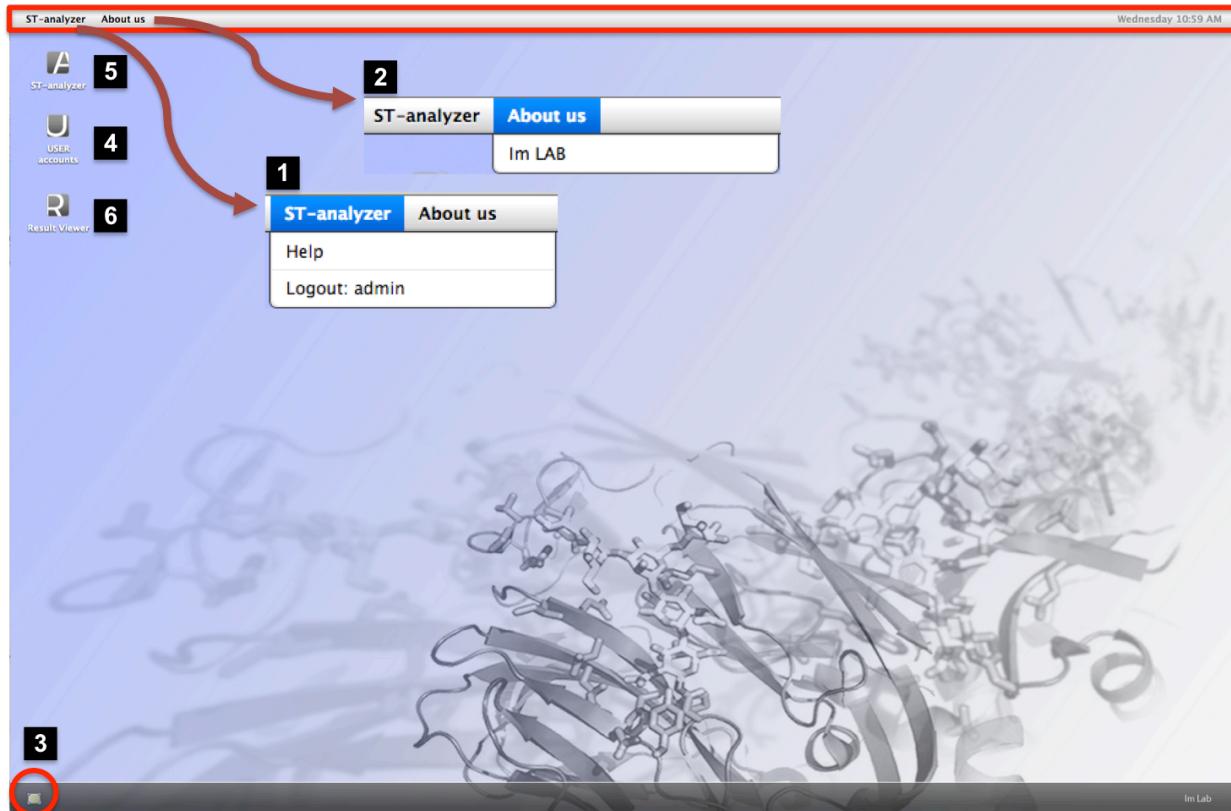


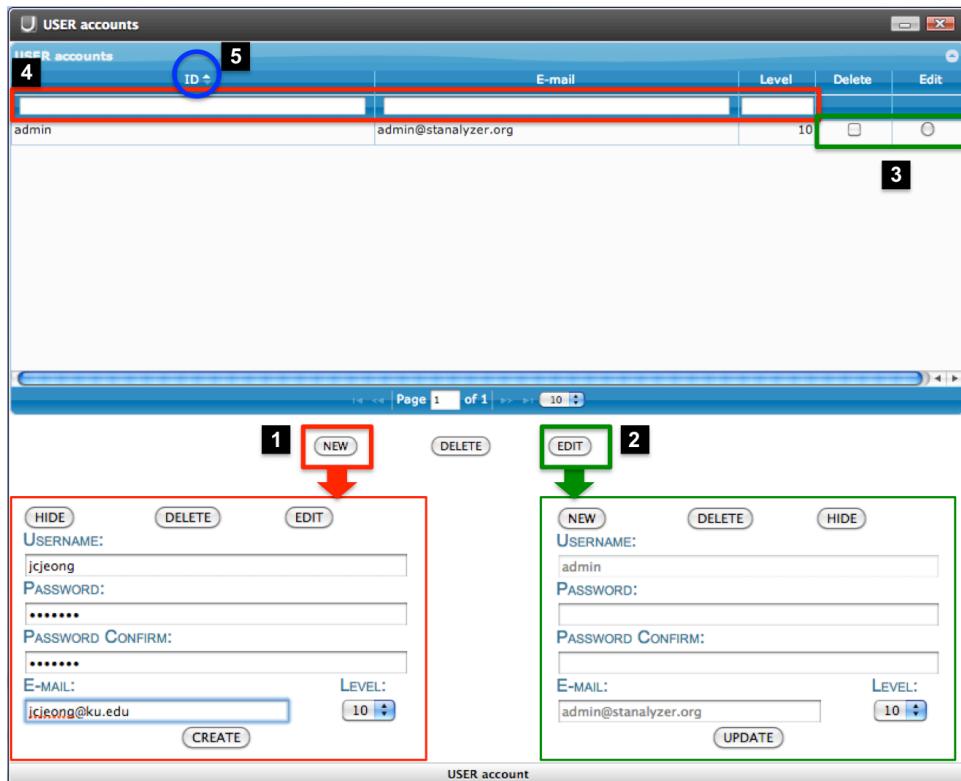
Figure 4. ST-analyzer workspace

Button for minimizing window

By clicking the button located at the bottom left of the workspace denoted as (3), you can hide all windows in the workspace.

User accounts

Once you logged in to ST-analyzer with ‘admin’ account, you have to create your own account so that your workspace can be protected from others.



Create, delete, and edit account

‘NEW’ button denoted as (1) in figure 5 account manager creates a new account. ‘Level’ is designed for hierarchical account management. Level 10 is the highest authority. The principles of hierarchical management is simple such that higher level can control lower level by means of deleting and editing accounts. Account cannot control other accounts having same level. Therefore, in order to delete or edit their own account, the user has to login to the system with the account expected to be controlled.

‘DELETE’ and ‘EDIT’ button denoted as (2) deletes and edits selected account. In order to execute these commands, the user has to choose the expected account prior to clicking these buttons by clicking the check or edit options denoted as (3).

Searching and ordering the lists

The list box contains simple searching and sorting functions. By typing any words matched with items in ID, e-mail, and level will be collected in the list by removing unmatched items in the list. To remove selections or see all lists, user needs to remove any character in the search box denoted as (4). By clicking ID column denoted as (5), the list will be sorted based on ID.

Logout ‘admin’ and login with new account

Once new account is created, please make sure to login with new account unless you want to keep using ‘admin’ account as your primary account. Please DO NOT USE admin as your personal account especially under multiple account environments.

ST-analyzer

By double clicking, ‘ST-analyzer’ icon in the workspace, users can see following window. This window is one of the most important GUI for preparing job submission of trajectory analysis. To submit the job user has to tell ST-analyzer about the system environment. The system environment is stored into database and can be recycled later.

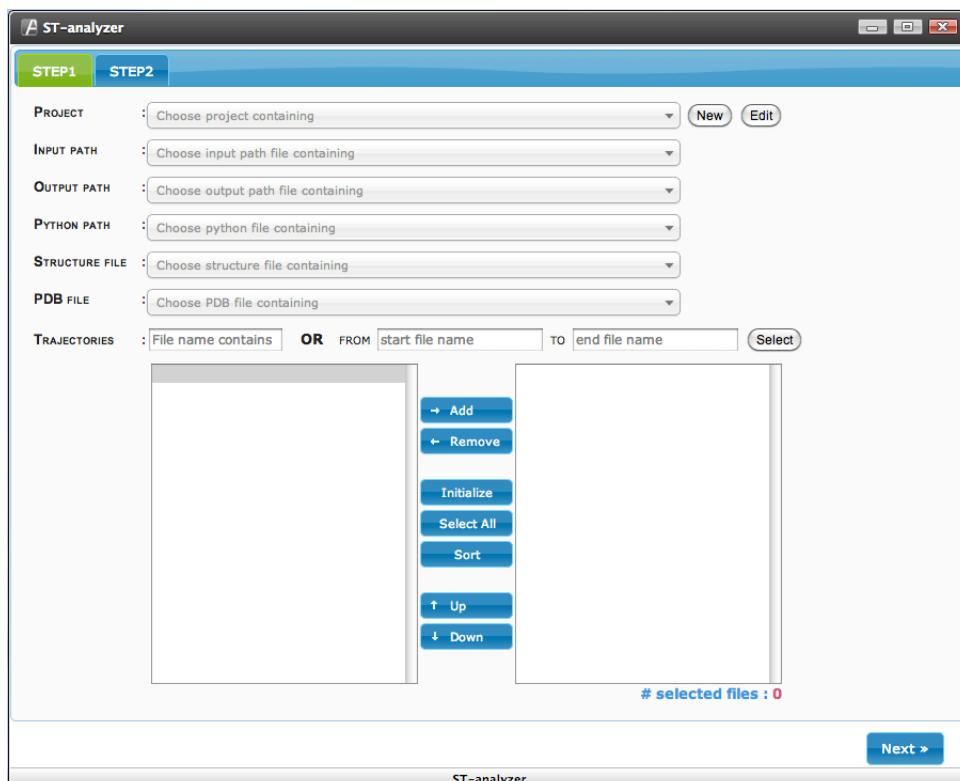


Figure 6. ST-analyzer initial window

STEP1

Step1 configures a project telling ST-analyzer information about system environment.

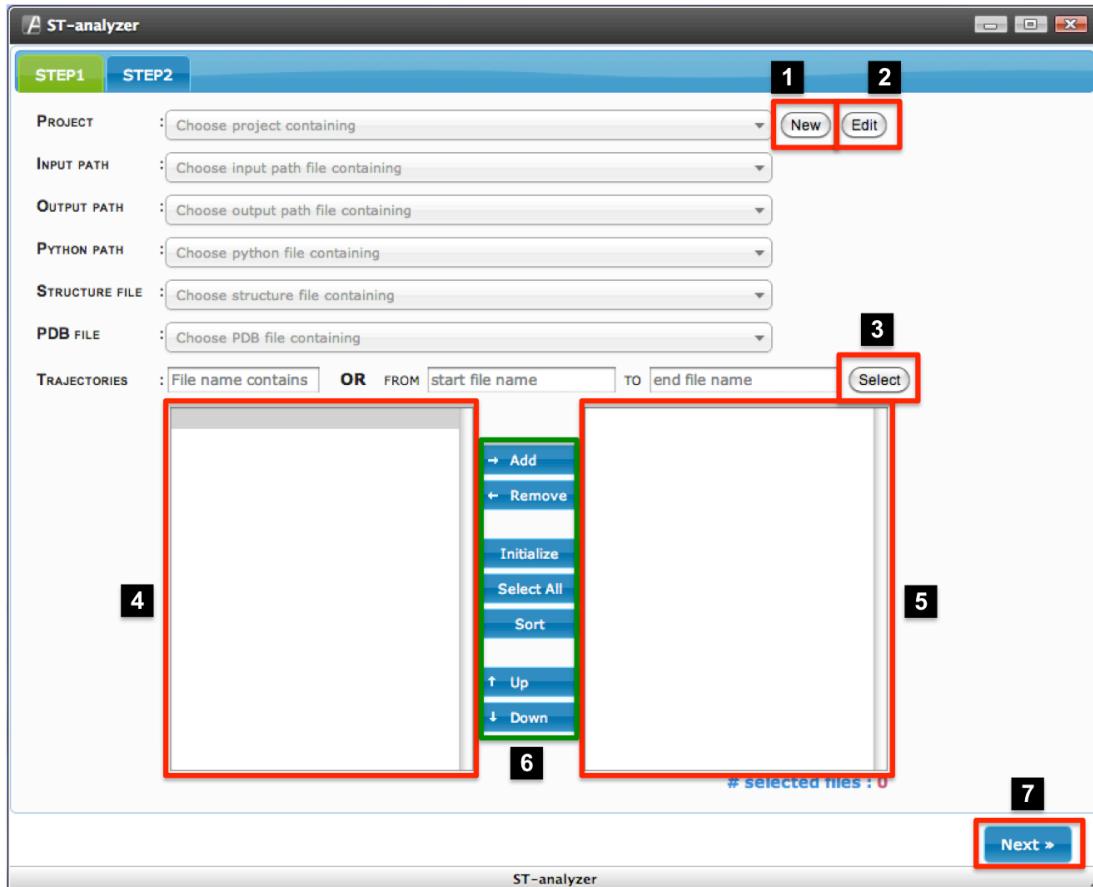


Figure 7 ST-analyzer at step 1: 1. Creating new project. 2. Editing the currently selected project. 3. Trajectory file selection (for detail, please read subsequent section, 'Trajectory file selection'). 4. List of existing files in current 'input path' 5. List of selected trajectory files. 6. Modification menus for selected trajectory files. 7. Button to move STEP 2.

New

Once clicking 'New' button in figure 7 (1), users can see project configuration window as following:

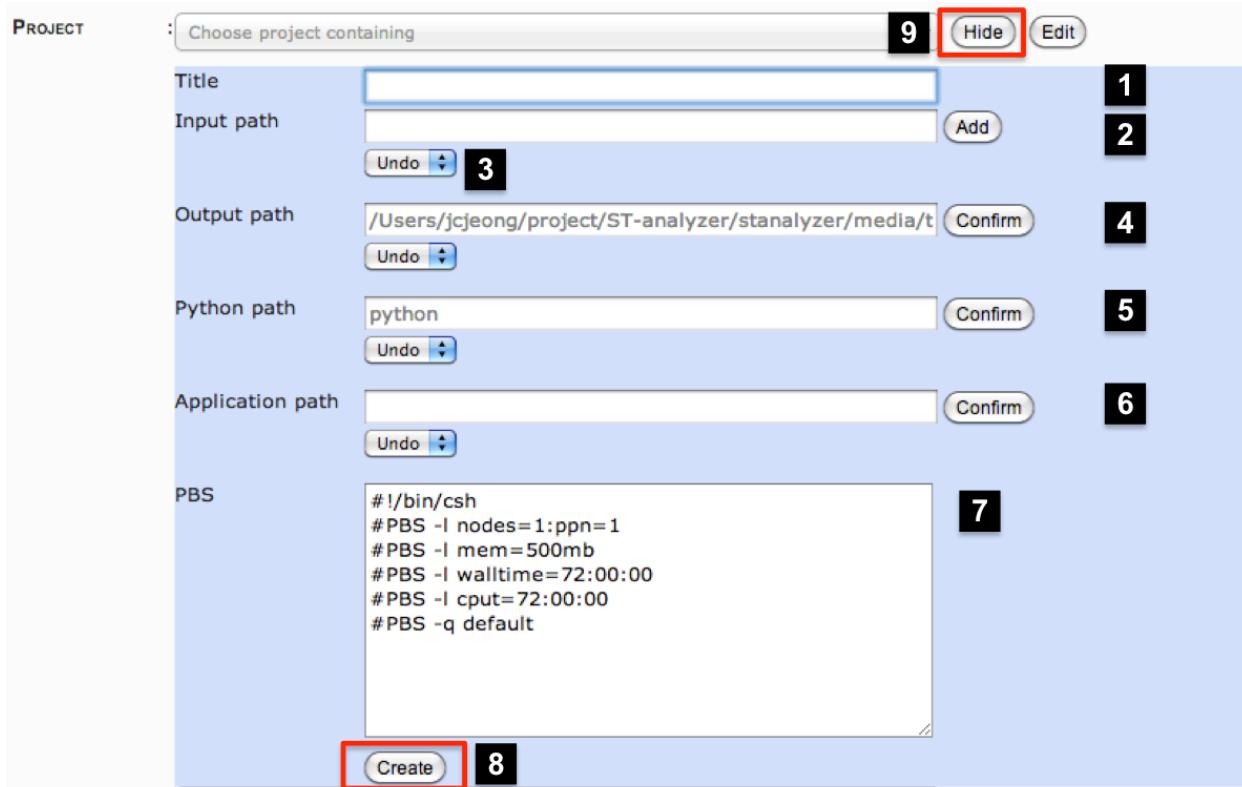


Figure 8. Step1: creating a new project

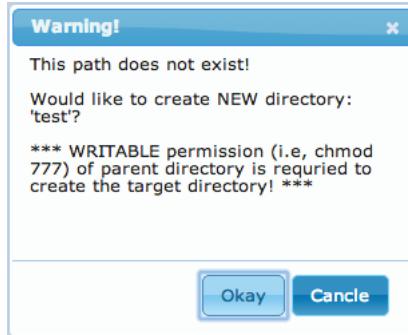
(1) Title: the name of project. (e.g. *My system: alpha helix*)

(2) Input path: the location of trajectory that requires absolute path in the system (e.g. /home/mysystem/protein). If the path does not exist, warning dialog box will appear.



(3) Undo drop box: recovering the delete items by simply choosing items in this drop box after entries are accidentally deleted.

(4) Output path: the location of output directory where the results are stored. Default path is given in the input box. If out path does not exist warning dialog box will appear. Be careful, if ST-analyzer is installed by not your account then output path must have write permission to the account that installed ST-analyzer.



(5) Python path: we have been reported that due to various reasons, system needs to maintain multiple versions of Python, to identify the right version. ST-analyzer requires specifying the Python having required modules. If the file does not exist or have executable permission, warning dialog will appear.



(6) Application path: this is designed for future usage, so at this moment users do not need to specify this option.

(7) PBS: if users want to submit the job to cluster machine then please give information about your PBS. Template of PBS is shown in the input box. Some information about PBS is described below

Table 1 PBS options

Definition of Important PBS Directives	
PBS Directive	Description
#PBS -l walltime=HH:MM:SS	This directive specifies the maximum walltime (real time, not CPU time) that a job should take. If this limit is exceeded, PBS will stop the job. Keeping this limit close to the actual expected time of a job can allow a job to start more quickly than if the maximum walltime is always requested.
#PBS -l pmem=SIZEgb	This directive specifies the maximum amount of physical memory used by any process in the job. For example, if the job would run four processes and each would use up to 2 GB (gigabytes) of memory, then the directive would read #PBS -l pmem=2gb. The default for this directive on Lion-XF and Lion-LSP is 1 GB (gigabyte) of memory. Other Lion clusters do not currently set a default.
#PBS -l nodes=N:ppn=M	This specifies the number of nodes (nodes=N) and the number of processors per node (ppn=M) that the job should use. PBS treats a processor core as a processor, so a system with eight cores per compute node can have ppn=8 as its maximum ppn request. Note that unless a job has some inherent parallelism of its own through something like MPI or OpenMP, requesting more than a single processor on a single node is usually wasteful and can impact the job start time.
#PBS -q queueName	This specifies what PBS queue a job should be submitted to. <i>This is only necessary if a user has access to a special queue.</i> This option can and should be omitted for jobs being submitted to a system's default queue.
#PBS -j oe	Normally when a command runs it prints its output to the screen. This output is often normal output and error output. This directive tells PBS to put both normal output and error output into the same output file.

(8) Create: store the information into database.

(9) Hide: hide project configuration window. Once a project configuration is done, the step 1 will reload trajectory data based on the information given during creating the project. The snapshot is shown below.

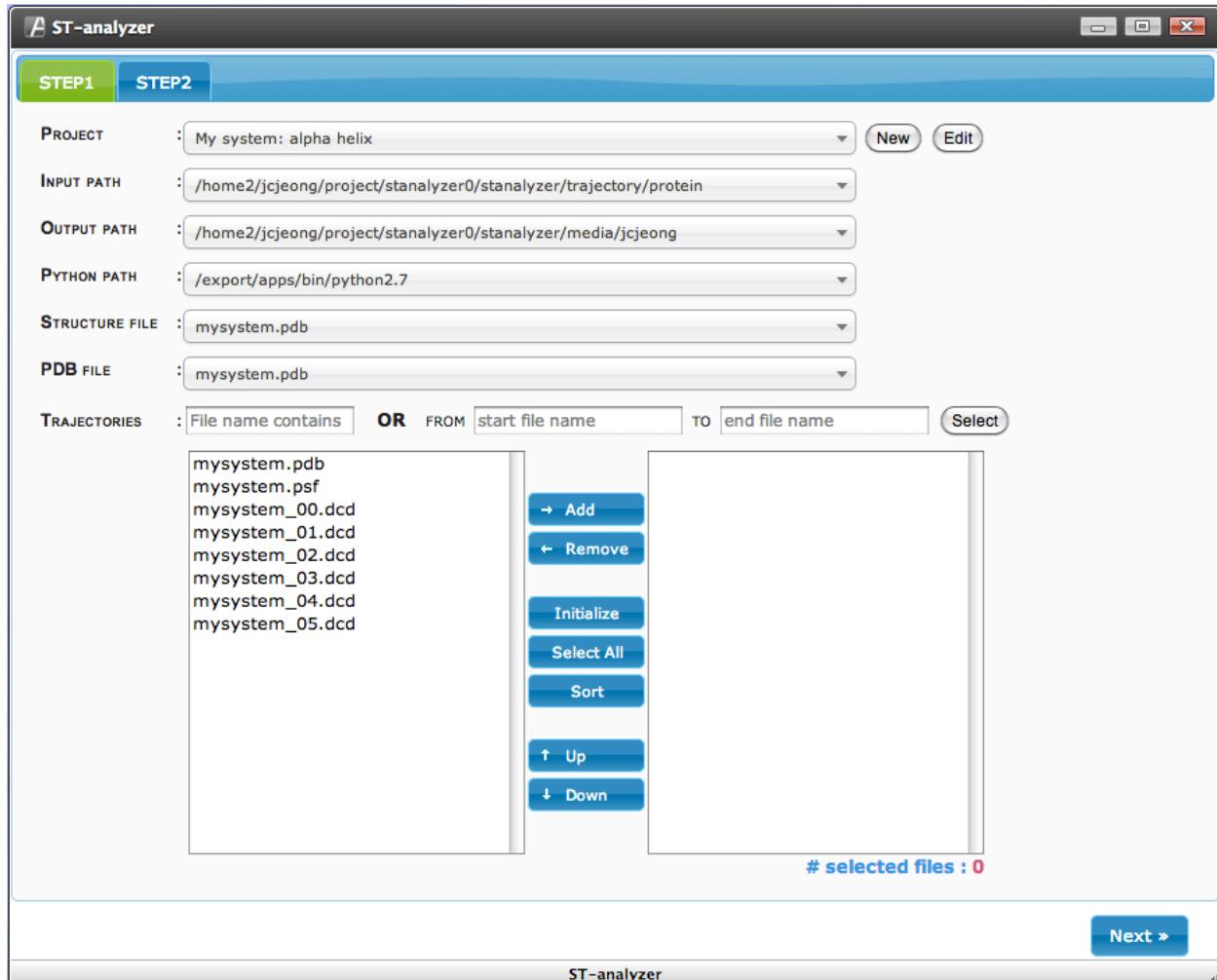


Figure 9 Step1 after creating a project

Edit

Once clicking 'EDIT' button in figure 7 (2), users can see the current project configuration window as following:

The screenshot shows a software interface for editing a project named 'My system: alpha helix'. The interface includes the following sections:

- Title:** My system: alpha helix
- Input path:** /home2/jcjeong/project/stanalyzer0/stanalyzer/trajectory/protein (with Add, Undo, and remove buttons)
- Output path:** /home2/jcjeong/project/stanalyzer0/stanalyzer/media/ (with Confirm, Undo, and remove buttons)
- Python path:** python (with Confirm, Undo, and remove buttons)
- Application path:** (empty field with Confirm and Undo buttons)
- PBS:** A text area containing a PBS script:

```
#!/bin/bash
#PBS -l nodes=1:ppn=1
#PBS -l mem=500mb
#PBS -l walltime=72:00:00
#PBS -l cput=72:00:00
#PBS -q default

export
LD_LIBRARY_PATH=/export/apps/compilers/gcc48/lib64
```

A red box highlights the 'Update' button at the bottom left of the PBS section.

Figure 10 Step1: Edit window

As shown in figure 10, all information defined in a project is displayed. To update current project, users can delete existing values or add new information. "Undo" drop box can help users to recover accidentally deleted items. As a final step of updating, users must click "Update" button to reflect current changes.

Preparing job submission

Project selection

Project contains all information about environment system; therefore, it is user's responsibility to use proper name to identify user-defined projects. Due the cases of storing multiple projects, ST-analyzer provides *quick search engine*, which shows items in the list containing user's search word and filtering out all other items.

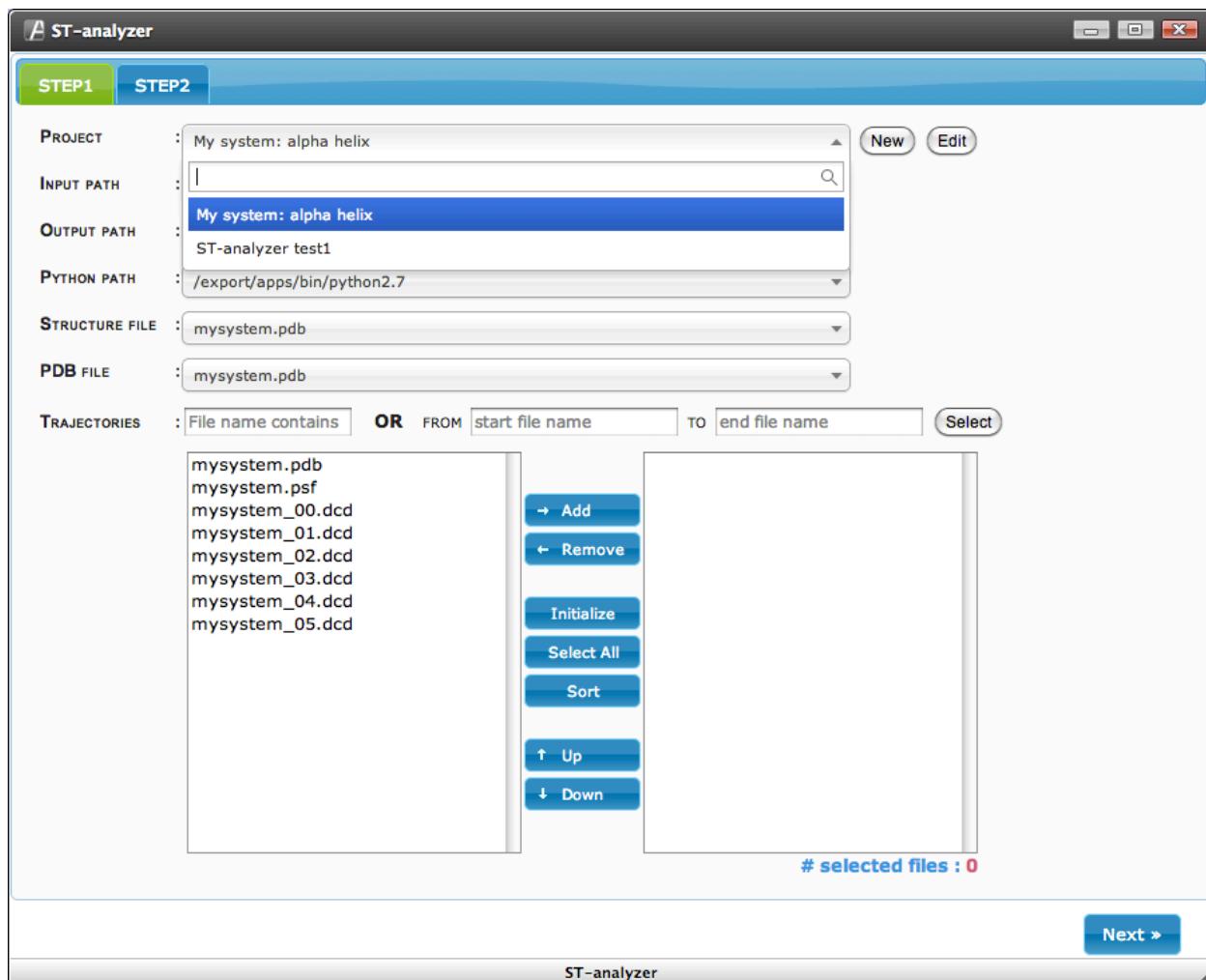


Figure 11 Project selection

Input path selection

Input path contains information about the locations of trajectory files. It is important that users MUST put all trajectories, PDB and structure files (e.g. 'psf' file: either CHARMM or NAMD format) in a same directory. Making easier to find targeted input path, quick search function is also provided.

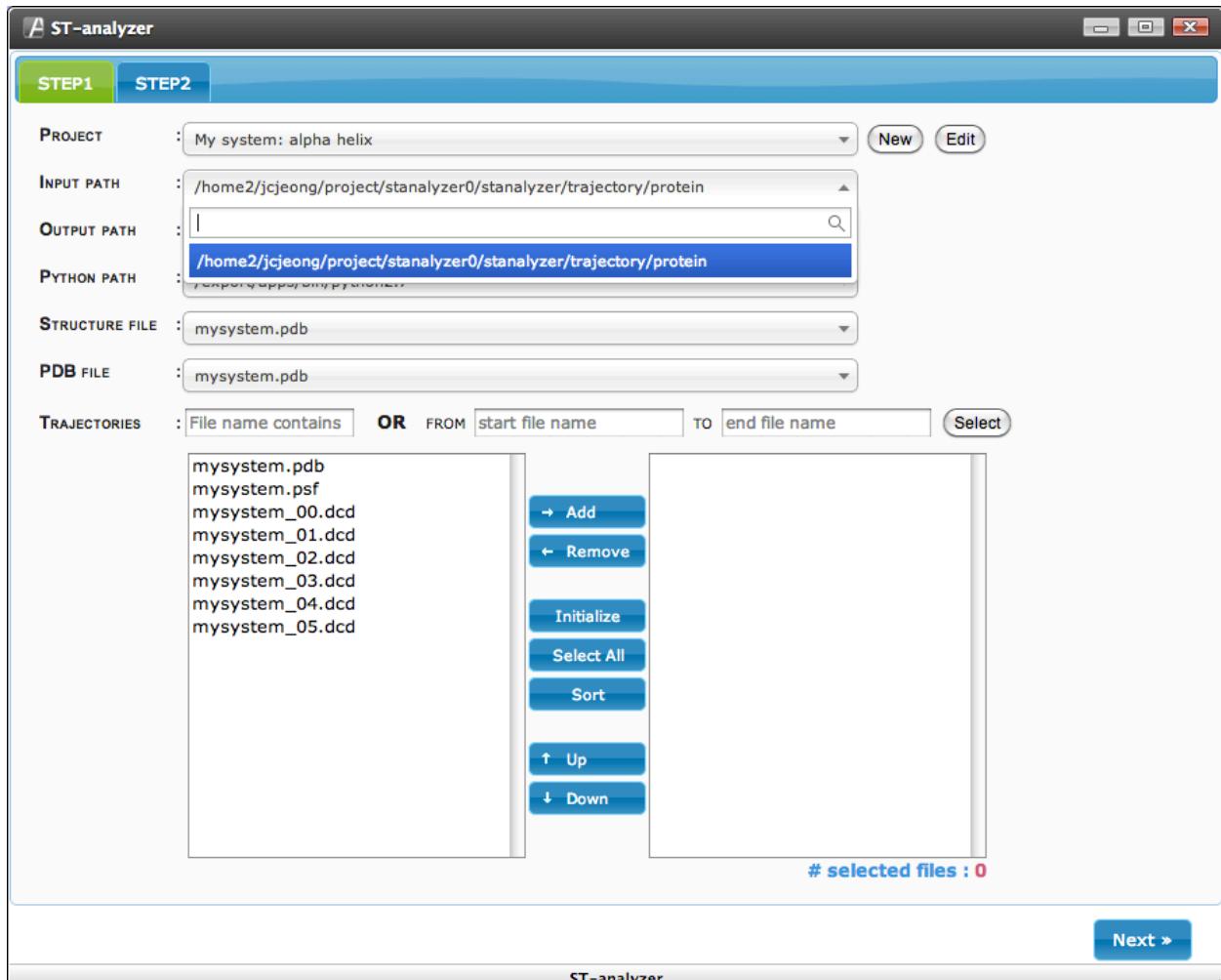


Figure 12 Input path selection

Output path selection

Output path contains information about the locations of output files resulted from ST-analyzer analysis modules.

The default value is located in “/ST-analyzer_HOME/media/user ID”, but the location can be varied upon users’ choice given during creation of project. With default path, ST-analyzer automatically creates subdirectory with random numbers at the inside of output path while submitting each job. The name of each subdirectory is defined as following format “Year+Month+date+hour+minutes+seconds+12 random characters”.

If user defined directory is used then ST-analyzer will directly write the output into the user-defined directory. Output path selection also provides *quick search* function.

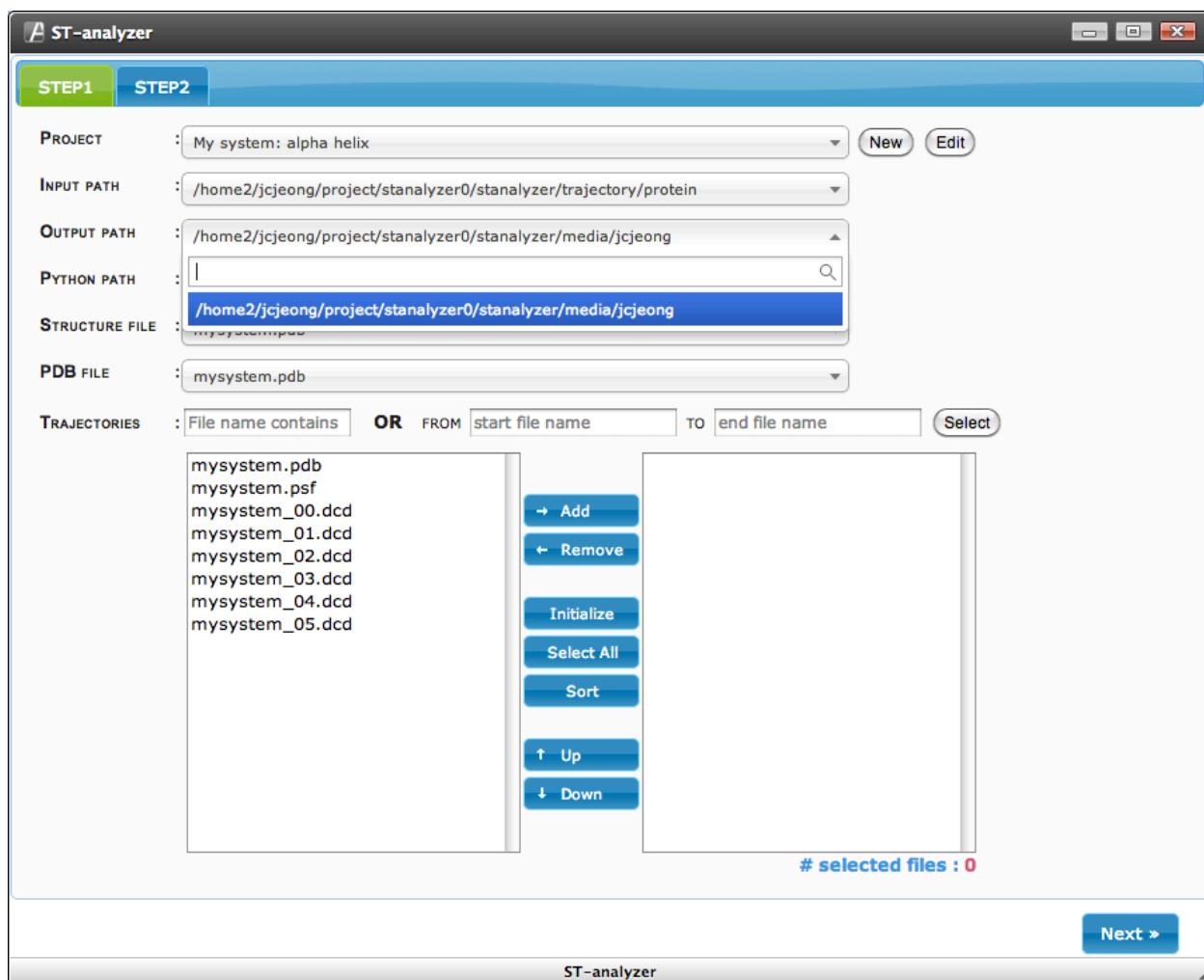


Figure 13 Output path selection

Python path selection

Some operating systems need to maintain multiple versions of python. Therefore, indicating proper python to ST-analyzer is crucial to run functional modules. An example using multiple versions of python is shown below, and in this case, python2.7 located in '/export/apps/bin/python2.7' is selected. Expected python can be found through quick search function if multiple python was defined while creating a project.

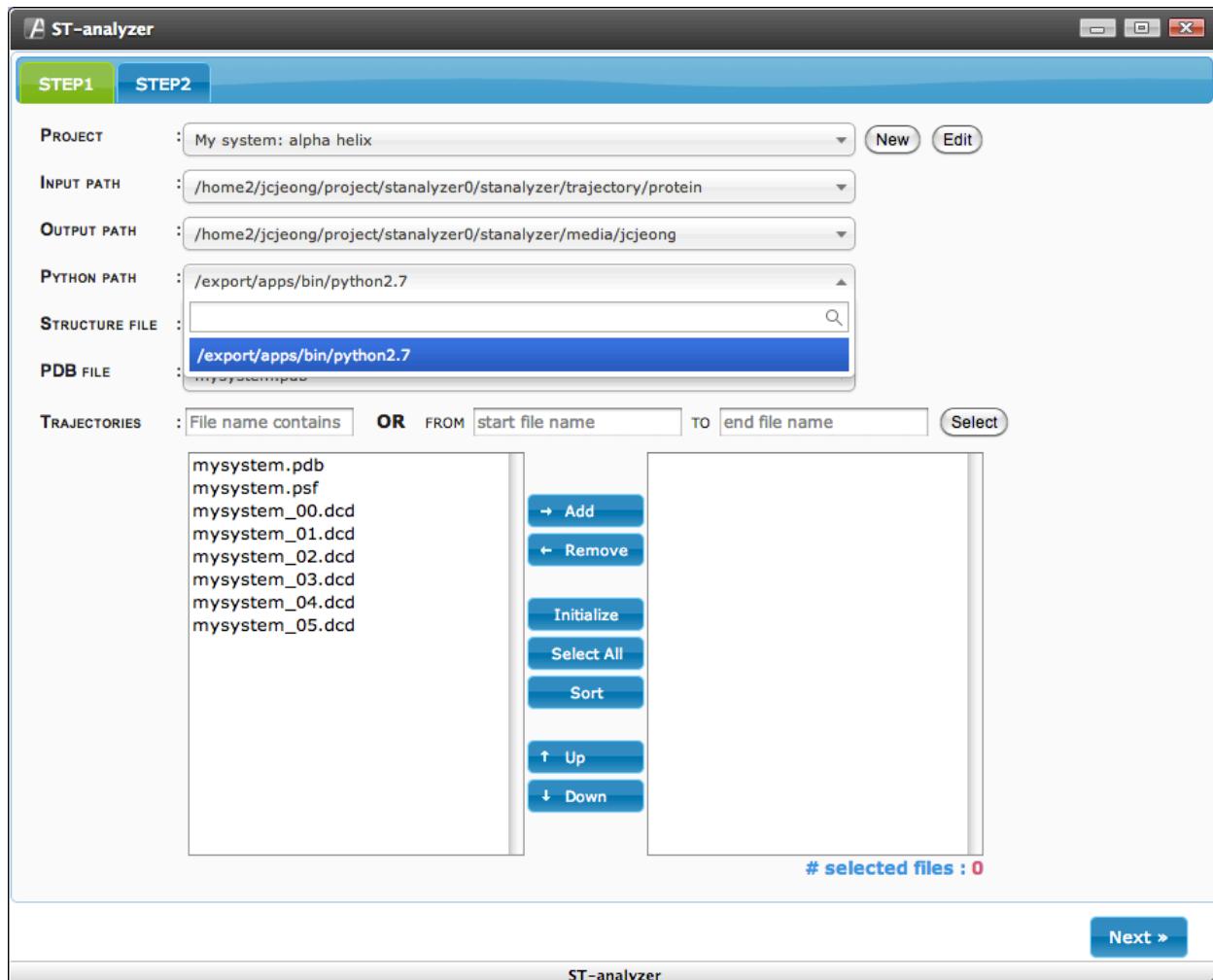


Figure 14 Python path selection

Structure file selection

Structure file contains all of the molecule-specific information needed to apply a particular force field to a molecular system such as a protein structure file (PSF) in CHARMM. Quick search will be useful to find structure file by typing a word contained in the name of structure file as shown below.

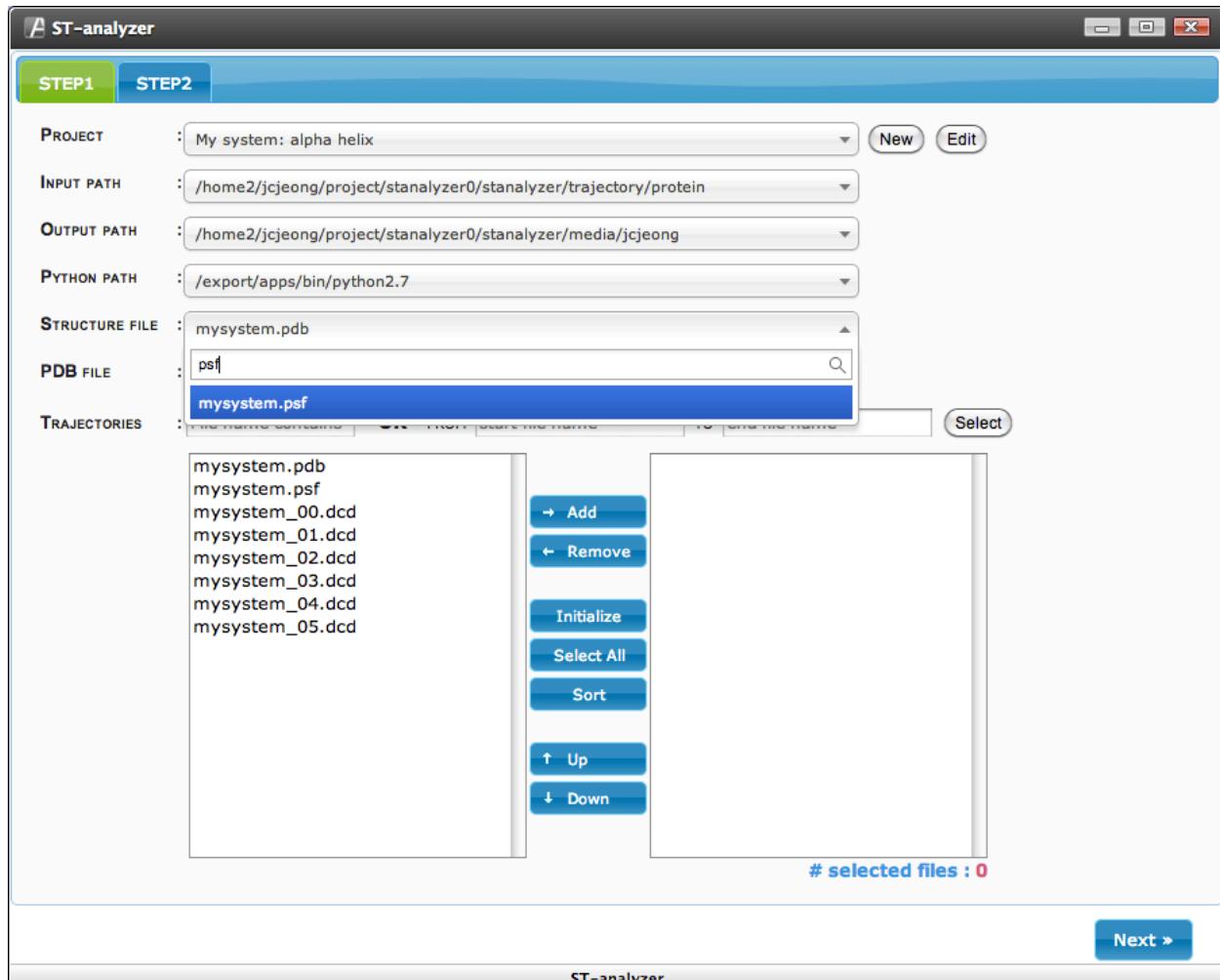


Figure 15 Structure file selection

PDB file selection

PDB file contains information about the structure of molecular system and can be easily found by using quick search function in ST-analyzer.

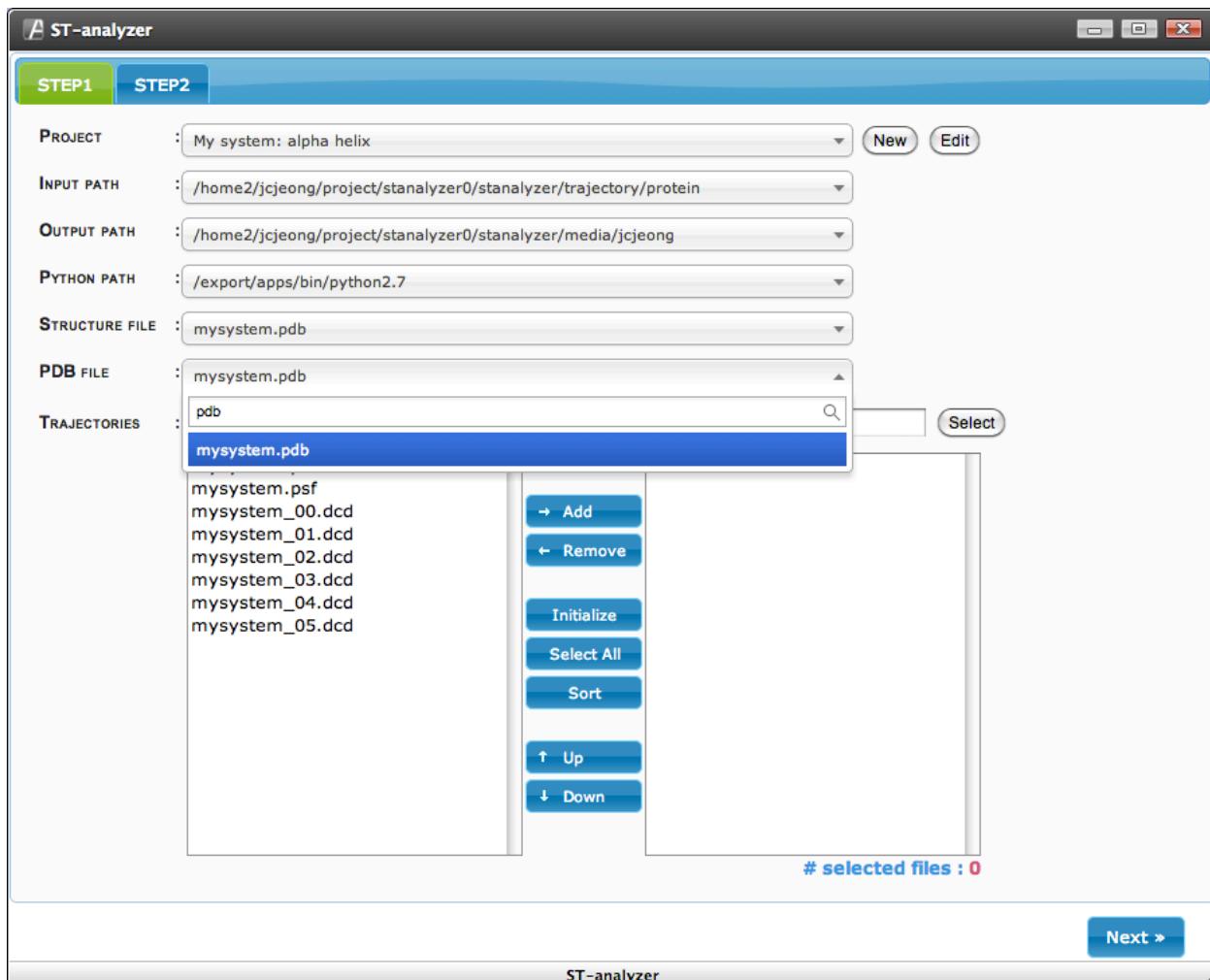


Figure 16 PDB file selection

Trajectory selection

Trajectory files are actual results obtained from molecular dynamics simulation. There are two ways to select trajectory files.

1) Using quick filter inputs:

- Type a word contained in names of trajectory files
- Select targeted files using mouse and keyboard (e.g. click start file with left mouse button and then click last file with 'shift' key + left mouse button)
- Click 'Add' button that will move the selected files into right list box which lists finally selected files used for analysis
- To recover the full list of files, please press 'ESC' button in your keyboard

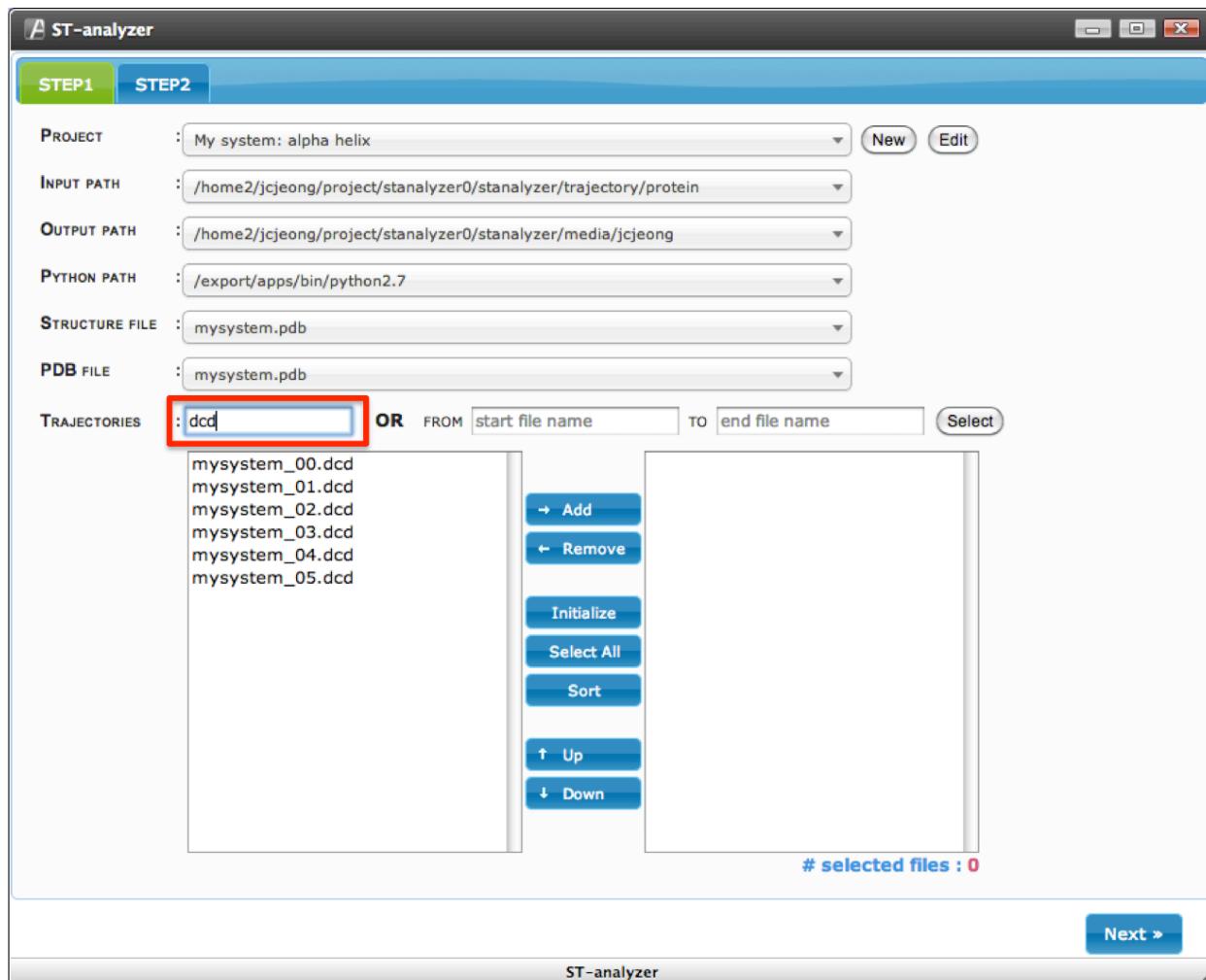


Figure 17 Trajectory file selection through a filter

2) Using quick selection filter:

- Type the name of first file in 'FROM' input box and last file name in 'TO' input box
- Click 'Select' button

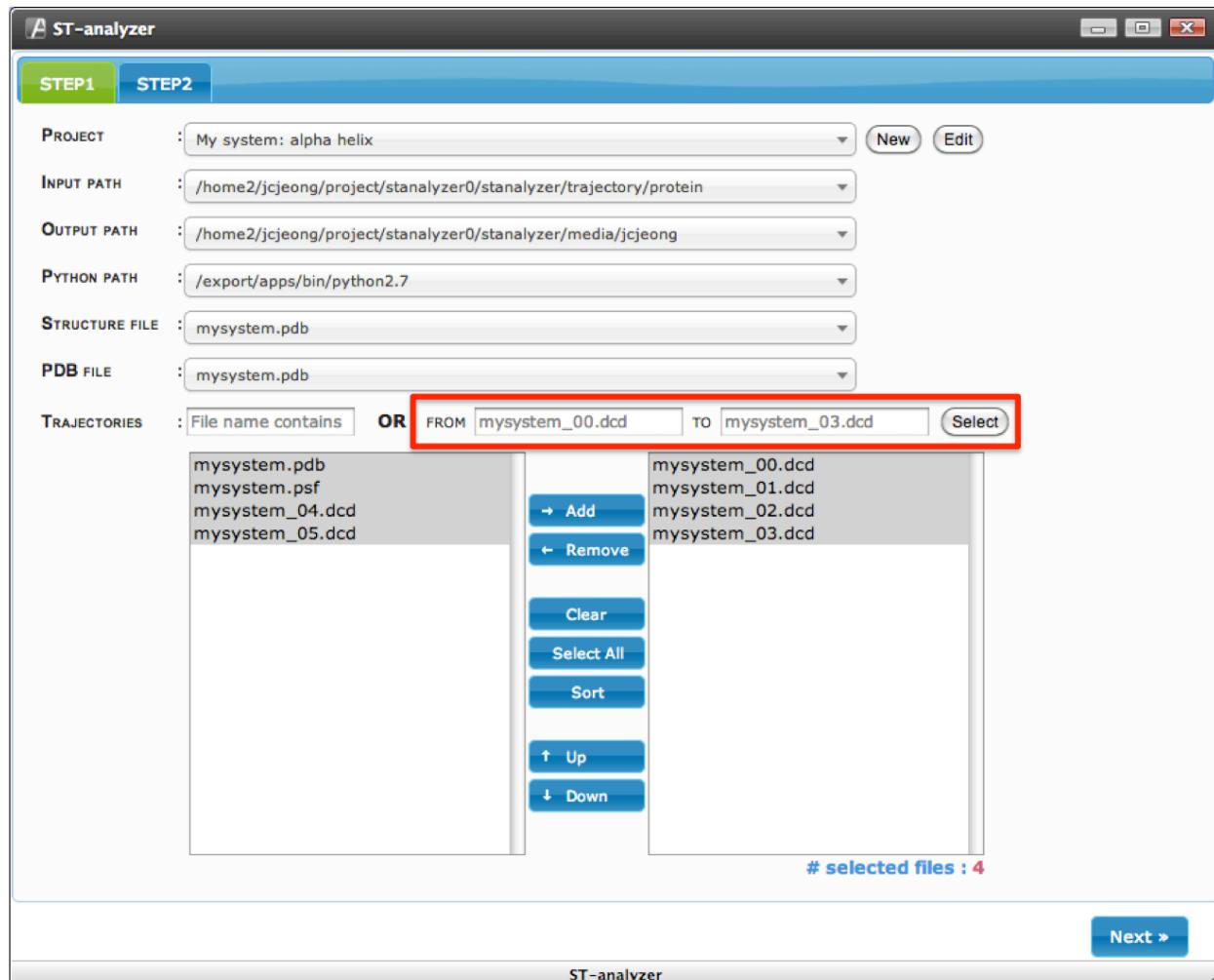


Figure 18 Trajectory file selection through file range inputs

Move to STEP 2

By clicking 'NEXT' button users can move forward to STEP 2 setting up the actual analysis modules.

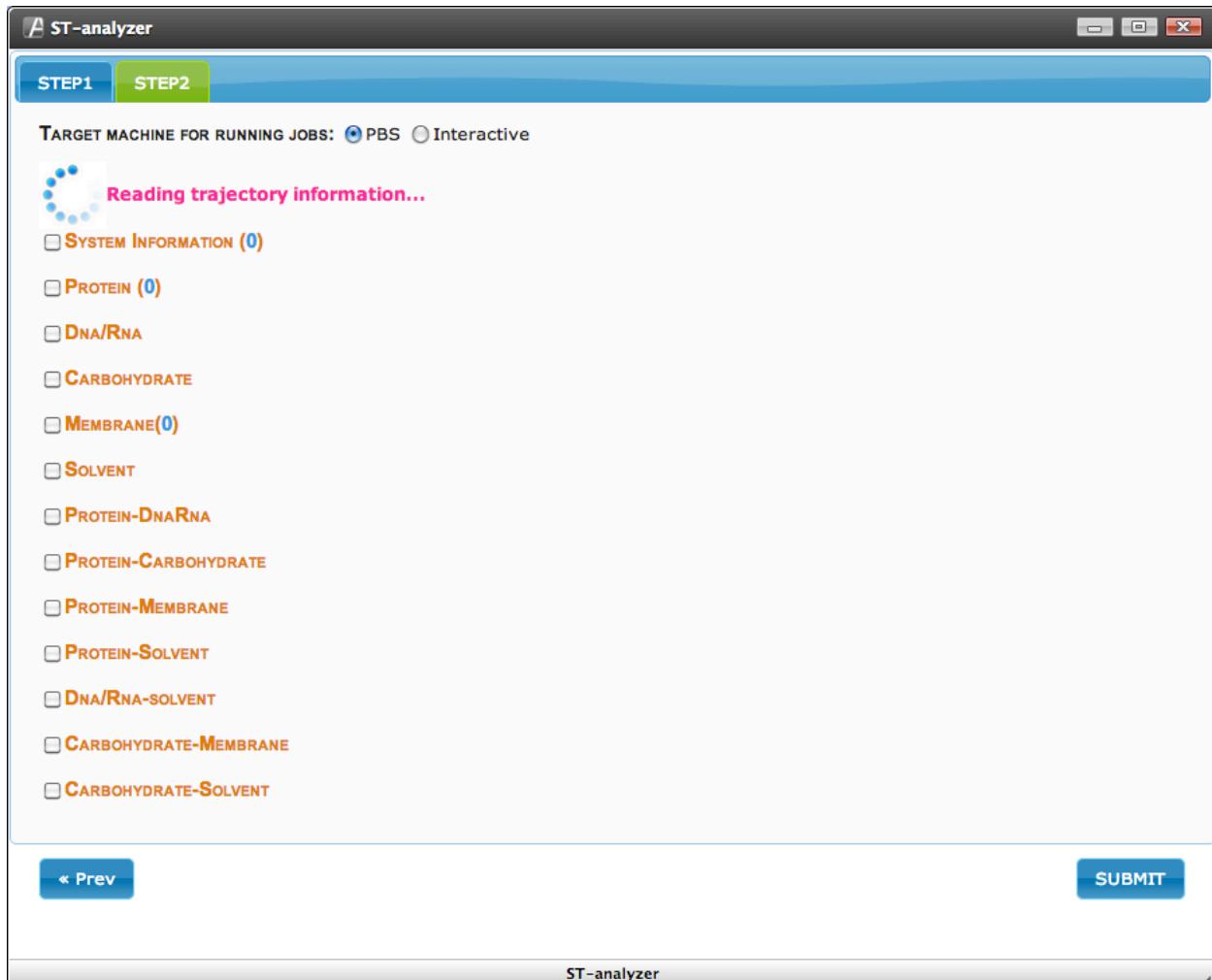


Figure 19 Reading trajectory information at Step 2

STEP2

Step 2 contains all available analysis modules and their configuration GUIs. Once moved to 'STEP2' window, summary of selected trajectory information will appear to make sure system recognize users inputs.

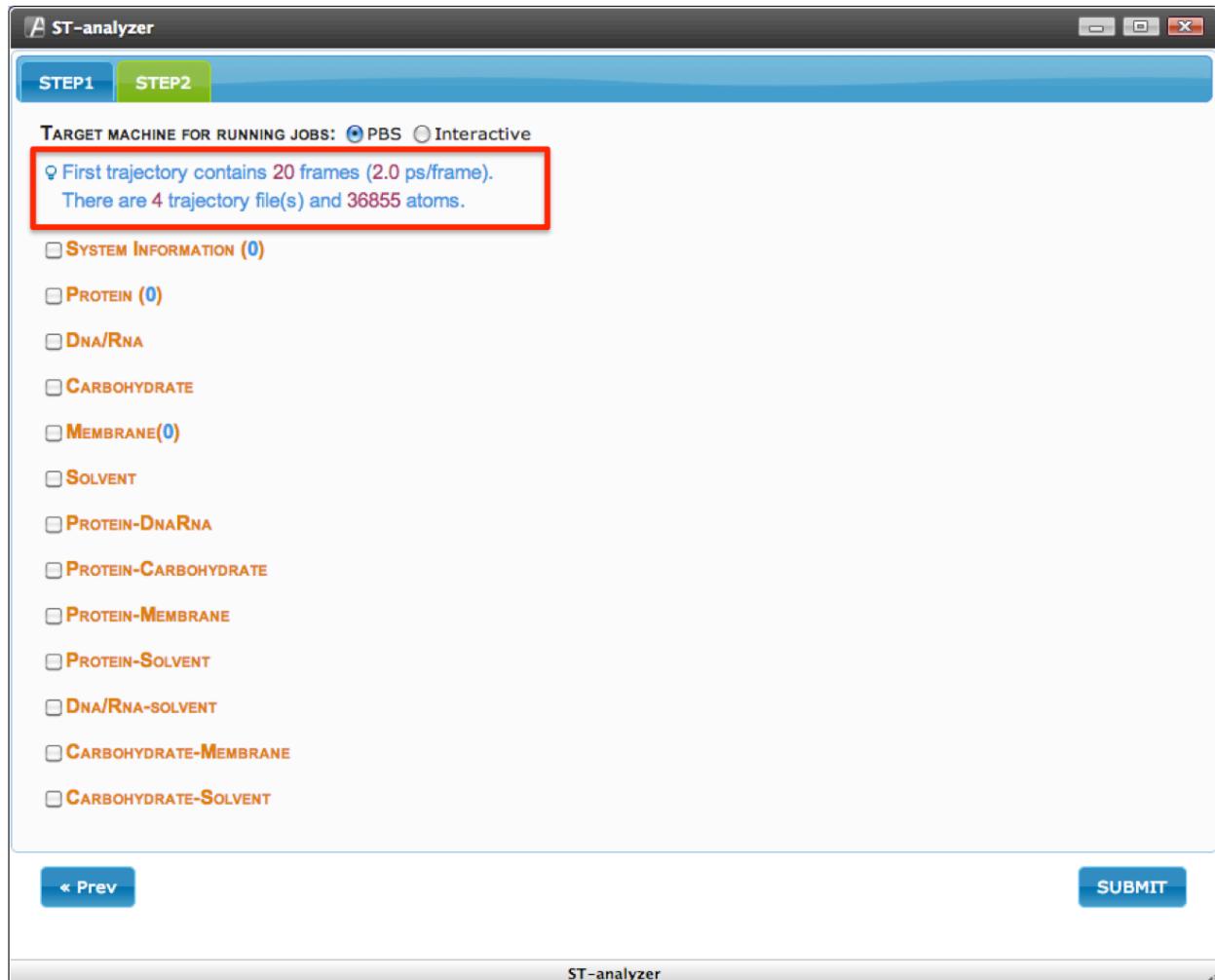


Figure 20 GUI at Step 2

Select machine type

ST-analyzer can run on two types of machine Cluster machines (PBS) or a server (Interactive). It is users responsibility to setup environment for running PBS or Interactive mode such that python and required modules must be working on expected system.

- PBS: processing analysis modules with cluster machines
- Interactive: processing analysis modules under a server where ST-analyzer is installed

The screenshot shows the ST-analyzer software interface. At the top, there are two tabs: 'STEP1' (highlighted in blue) and 'STEP2'. Below the tabs, the title 'TARGET MACHINE FOR RUNNING JOBS:' is followed by two radio buttons: 'PBS' (selected) and 'Interactive'. A red box highlights the 'PBS' radio button. Below this, a message states: 'First trajectory contains 20 frames (2.0 ps/frame). There are 4 trajectory file(s) and 36855 atoms.' A list of analysis categories follows, each preceded by an unchecked checkbox:

- SYSTEM INFORMATION (0)
- PROTEIN (0)
- DNA/RNA
- CARBOHYDRATE
- MEMBRANE(0)
- SOLVENT
- PROTEIN-DNA/RNA
- PROTEIN-CARBOHYDRATE
- PROTEIN-MEMBRANE
- PROTEIN-SOLVENT
- DNA/RNA-SOLVENT
- CARBOHYDRATE-MEMBRANE
- CARBOHYDRATE-SOLVENT

At the bottom left is a '« Prev' button, and at the bottom right is a 'SUBMIT' button. The window title bar says 'ST-analyzer'.

Figure 21 Choose target machine

Analysis modules

ST-analyzer comes with built in modules commonly used in MD analysis. The number of modules will keep increasing as planned but not limited in following lists.

To simplify the explanation, we now define ST-analyzer home directory as \$ST_HOME where ST-analyzer is installed

- System size
- Density profile
- Lipid deuterium order parameters
- Root-mean-squared
- Root-mean-squared fluctuation
- Lipid hydrophobic thickness
- Lipid surface area
- B-factors
- Residue-residue contacts
- Secondary structure
- Dihedral angles
- Distance between two atoms
- H-bond / salt-bridge profile
- Helix (β -hairpin) tilt, rotation, crossing angles, and distance
- Solid-state NMR properties (chemical shift, dipolar coupling constant)
- Channel pore size
- Membrane potential
- Water/ion movement
- Lipid lateral diffusion constant
- Lipid chain relaxation time
- Lipid rotation/wobble motions
- Sterol tilt angle
- Lateral ion density inside channel
- Residue-water/lipid contact information
- Solvent accessible surface area
- Lipid adaptation (through selection of "local" and "bulk" lipids)
- Substrate binding
- Pore hydration

System size

Analyzing changes of system size during simulations.

In order to run this example, users can use trajectories located in \$ST_HOME/trajectory/protein

- GUI location: \$ST_HOME/stanalyzer/templates/gui/systemInfo.html
- Module location: \$ST_HOME/static/analyzers/box.py

The image shows a screenshot of the ST-analyzer software interface. At the top, there is a checked checkbox labeled "System size". Below this, there are two input fields: "Frame Interval" with the value "1" and "Output File Name" with the value "box_size.dat".

Parameters

- Frame Interval: users can choose frame interval. For example, if time frame 2 is given then ST-analyzer will use every two frames (i.e. 2, 4, 6, 8, ...) to analyze the system instead of every frame.
- Output File Name: users can define the name of output file that contains text format of current module.

Helix tilt

Analyzing the tilt angle of helix.

In order to run this example, users can use trajectories located in \$ST_HOME/trajectory/protein

- GUI location: \$ST_HOME/stanalyzer/templates/gui/protein.html
- Module location: \$ST_HOME/static/analyzers/helix_tilt.py

Helix tilt

Frame Interval	Output File Name	Segments	Start	End	Add More
1	helix_tilt.dat	PROA	1-GLY	23-GLY	<input type="button" value="Add"/>
1	helix_tilt2.dat	PROA	6-SER	17-GLY	<input type="button" value="DELETE"/>
1	helix_tilt3.dat	PROA	24-GLY	46-ALA	<input type="button" value="DELETE"/>

Parameters

- Frame Interval: users can choose frame interval. For example, if time frame 2 is given then ST-analyzer will use every two frames (i.e. 2, 4, 6, 8, ...) to analyze the system instead of every frame.
- Output File Name: users can define the name of output file that contains text format of current module.
- Segments: choose segment ID that contains helices. NOTE: To select start and end residue, the segments has to be defined first.
- Start: start residue – drop box contains a pair of residue ID and three-letter residue name. By changing the segment ID, residue ID and names will be automatically updated.
- End: end residue – drop box contains a pair of residue ID and three-letter residue name. By changing the segment ID, residue ID and names will be automatically updated.
- Add: by pressing 'Add' button, users can define multiple angle positions.

Sterol tilt

Analyzing the tilt angle of sterols.

In order to run this example, users can use trajectories located in \$ST_HOME/trajectory/lipids

Ring tilt

- GUI location: \$ST_HOME/stalyzer/templates/gui/protein.html
- Module location: \$ST_HOME/static/analyzers/sterol_tilt_ring.py



Parameters

- Frame Interval: users can choose frame interval. For example, if time frame 2 is given then ST-analyzer will use every two frames (i.e. 2, 4, 6, 8, ...) to analyze the system instead of every frame.
- Output File Name: users can define the name of output file that contains text format of current module.
- Segments: choose segment ID that contains sterol – this system contains cholesterol (CHL1) at MEMB segments.
- Selection Query: this enables calculating different angles given by users. In this example C3 and C17 atom are used for defining ring tilt. For more information about using query please refer the '**Selection Query**' section in this document. **NOTE:** the given query is shown as an example. This means it is users' responsibility to modify the query according to their system.
- Verify: this button shows the summary of query results as shown below.

Selected Atoms							
#Total Number of selected types:80							
# Selected unique types:							
CRL1							
# Total Number of selected residues: 40							
# Selected unique residues:							
CHL1							
# Total Number of selected atoms: 80							
# Selected unique atoms:							
C3 C17							
#SDG_ID	RES_ID	RES_NAME	NAME	TYPE	X-axis	Y-axis	Z-axis
MEMB	1	CHL1	C3	CRL1	0.10800002801	-2.71499991417	
MEMB	1	CHL1	C17	CRL1	-0.925000011921	-2.46600008011	
MEMB	2	CHL1	C3	CRL1	14.2650003433	-3.09400010109	
MEMB	2	CHL1	C17	CRL1	14.5670003891	-3.90000009537	
MEMB	3	CHL1	C3	CRL1	18.4249992371	-22.4950008392	
MEMB	3	CHL1	C17	CRL1	18.5960006714	-24.0480003357	
MEMB	4	CHL1	C3	CRL1	-4.23000001907	-19.7310009003	
MEMB	4	CHL1	C17	CRL1	-5.87300014496	-18.9370002747	
MEMB	5	CHL1	C3	CRL1	-25.1529998779	-19.5809993744	
MEMB	5	CHL1	C17	CRL1	-24.2000007629	-20.3540000916	
MEMB	6	CHL1	C3	CRL1	-14.390999794	-1.60899996758	
MEMB	6	CHL1	C17	CRL1	-14.5480003357	-1.76900005341	
MEMB	7	CHL1	C3	CRL1	-5.81599998474	14.9230003357	
MEMB	7	CHL1	C17	CRL1	-7.6859998703	16.1289997101	
MEMB	8	CHL1	C3	CRL1	6.69999980927	2.34400010109	
MEMB	8	CHL1	C17	CRL1	6.44299983978	-0.1000000149	
MEMB	9	CHL1	C3	CRL1	-17.6369991302	-17.283000946	
MEMB	9	CHL1	C17	CRL1	-19.0709991455	-16.9689998627	
MEMB	10	CHL1	C3	CRL1	1.6529999714	-16.1299991608	
MEMB	10	CHL1	C17	CRL1	1.1289999485	-16.5620002747	
MEMB	11	CHL1	C3	CRL1	22.6840000153	15.218000412	
MEMB	11	CHL1	C17	CRL1	22.6469993591	14.1969995499	

Tail tilt

- GUI location: \$ST_HOME/stanalyzer/templates/gui/protein.html
- Module location: \$ST_HOME/static/analyzers/sterol_tilt_tail.py

Tail tilt

Frame Interval	Output File Name	Segments	Residue
1	sterol_tail_tilt.i	MEMB	CHL1
Selection Query			
name C17 or name C25			Verify

Parameters

- Frame Interval: users can choose frame interval. For example, if time frame 2 is given then ST-analyzer will use every two frames (i.e. 2, 4, 6, 8, ...) to analyze the system instead of every frame.
- Output File Name: users can define the name of output file that contains text format of current module.
- Segments: choose segment ID that contains sterol – this system contains cholesterol (CHL1) at MEMB segments.
- Selection Query: this enables calculating different angles given by users. In this example C17 and C25 atom are used for defining tail tilt. For more information about using query please refer the ‘Selection Query’ section in this document. **NOTE: the given query is shown as an example. This means it is users’ responsibility to modify the query according to their system.**
- Verify: this button shows the summary of query results as shown below.

Selected Atoms							
#Total Number of selected types:80							
# Selected unique types:							
CRL1 CTL1							
# Total Number of selected residues: 40							
# Selected unique residues:							
CHL1							
# Total Number of selected atoms: 80							
# Selected unique atoms:							
C17 C25							
#SEG_ID	RES_ID	RES_NAME	NAME	TYPE	X-axis	Y-axis	Z-axis
MEMB	1	CHL1	C17	CRL1	-0.925000011921	-2.46600008011	
MEMB	1	CHL1	C25	CRL1	-2.03600001335	-2.3759992752	
MEMB	2	CHL1	C17	CRL1	14.5670003891	-3.90000009537	
MEMB	2	CHL1	C25	CRL1	15.2419996262	-3.39599990845	
MEMB	3	CHL1	C17	CRL1	18.5960006714	-24.0480003357	
MEMB	3	CHL1	C25	CRL1	17.1159992218	-24.3040008545	
MEMB	4	CHL1	C17	CRL1	-5.87300014496	-18.9370002747	
MEMB	4	CHL1	C25	CRL1	-7.10200023651	-19.1830005646	
MEMB	5	CHL1	C17	CRL1	-24.2000007629	-20.3540000916	
MEMB	5	CHL1	C25	CRL1	-22.9969997406	-21.5240001678	
MEMB	6	CHL1	C17	CRL1	-14.5480003357	-1.76900005341	
MEMB	6	CHL1	C25	CRL1	-15.2650003433	-1.9479995422	
MEMB	7	CHL1	C17	CRL1	-7.6859998703	16.1289997101	
MEMB	7	CHL1	C25	CRL1	-10.1160001755	16.1800003052	
MEMB	8	CHL1	C17	CRL1	6.44299983978	-0.10000000149	
MEMB	8	CHL1	C25	CRL1	8.15100002289	-0.73600001431	
MEMB	9	CHL1	C17	CRL1	-19.0709991455	-16.9689998627	
MEMB	9	CHL1	C25	CRL1	-18.5729999542	-16.4160003662	
MEMB	10	CHL1	C17	CRL1	1.1289999485	-16.5620002747	
MEMB	10	CHL1	C25	CRL1	1.96000003815	-17.5900001526	
MEMB	11	CHL1	C17	CRL1	22.6469993591	14.1969995499	
MEMB	11	CHL1	C25	CRL1	22.0599994659	13.8120002747	

RMSD

Analyzing Root-mean-square deviation.

$$RMSD = \sqrt{\frac{1}{N} \sum_{i=1}^N \delta_i^2}$$

Here, δ is the distance between N pairs of atoms (e.g. C α)

In the example, C α atoms in PROA segment are used for aligning structures and then RMSD among C β atoms is calculated. In order to run this example, users can use trajectories located in \$ST_HOME/trajectory/protein

- GUI location: \$ST_HOME/stanalyzer/templates/gui/protein.html
- Module location: \$ST_HOME/static/analyzers/rmsd.py

RMSD

Frame Interval Output File Name
1 rmsd_output.dat

Atom selection for alignment:
segid PROA and name CA

Atom selection for RMSD calculation:
segid PROA and name CB

Parameters

- Frame Interval: users can choose frame interval. For example, if time frame 2 is given then ST-analyzer will use every two frames (i.e. 2, 4, 6, 8, ...) to analyze the system instead of every frame.
- Output File Name: users can define the name of output file that contains text format of current module.
- Atom selection for alignment: selecting atoms for aligning the structure.
- Atom selection for RMSD calculation: selecting expected atoms to see their RMSD based on the alignment made by alignment atoms selected above.
- Verify: this button shows the summary of query results as shown below.

Selected Atoms							
# Total Number of selected types:46							
# Selected unique types:							
CT1 CT2							
# Total Number of selected residues: 46							
# Selected unique residues:							
GLN ILE SER GLU MET							
LYS THR ALA GLY ASP							
LEU ARG GLU TYR							
# Total Number of selected atoms: 46							
# Selected unique atoms:							
CA							
#SEQ_ID	RES_ID	RES_NAME	NAME	TYPE	X-axis	Y-axis	Z-axis
PROA 1	GLY	CA	CT2	22.5249996185	-12.12699855		
PROA 2	VAL	CA	CT1	20.6529998779	-10.1219997406		
PROA 3	ILE	CA	CT1	22.482000351	-6.77299976349		
PROA 4	ASP	CA	CT1	25.593000412	-5.96600008011		
PROA 5	THR	CA	CT1	25.9290008545	-2.43400001526		
PROA 6	SER	CA	CT1	29.1370007477	-2.43400001526		
PROA 7	ALA	CA	CT1	29.1229991913	-3.32899999619		
PROA 8	VAL	CA	CT1	26.843000412	-0.31099999851		
PROA 9	GLU	CA	CT1	29.2269992828	1.99000000954		
PROA 10	SER	CA	CT1	32.1370010376	0.799000024796		
PROA 11	ALA	CA	CT1	29.908000946	1.54600000381		
PROA 12	ILE	CA	CT1	29.0160007477	5.09299993515		
PROA 13	THR	CA	CT1	32.600000709	5.869000018311		
PROA 14	ASP	CA	CT1	34.61199917	4.45699995		
PROA 15	GLY	CA	CT2	32.1469993591	3.29099988977		
PROA 16	GLN	CA	CT1	31.0580005646	6.85599994659		
PROA 17	GLY	CA	CT2	32.2340011597	7.23099994659		
PROA 18	ASP	CA	CT1	30.0809993744	8.80900001526		
PROA 19	MET	CA	CT1	31.7709999084	6.52899980545		
PROA 20	LYS	CA	CT1	29.7689990997	3.32899999619		

RMSF

Analyzing Root-mean-square deviation.

$$RMSF = \sqrt{\frac{1}{T} \sum_{t_j=1}^T (x_i(t_j) - \bar{x}_i)^2}$$

Here, T is the time (i.e. number of frames), $x_i(t_j)$ is the position of atom i at time t_j , and \bar{x}_i is time averaged position of the atom i .

In the example below, C α atoms in PROA segment are used for aligning structures and then RMSF of all atoms in segment PROA is calculated. In order to run this example, users can use trajectories located in \$ST_HOME/trajectory/protein

- GUI location: \$ST_HOME/stanalyzer/templates/gui/protein.html
- Module location: \$ST_HOME/static/analyzers/rmsf.py

The screenshot shows the RMSF analysis configuration window. It includes fields for Frame Interval (set to 1), Output File Name (set to rmsf_ouput.dat), and two Verify buttons. There are also sections for Atom selection for alignment and Atom selection for RMSF calculation, both containing specific residue and atom definitions.

Parameters

- Frame Interval: users can choose frame interval. For example, if time frame 2 is given then ST-analyzer will use every two frames (i.e. 2, 4, 6, 8, ...) to analyze the system instead of every frame.
- Output File Name: users can define the name of output file that contains text format of current module.
- RMSF based on residue (*default = based on individual atom): output of RMSF can be made of either residue or individual atom.
- Atom selection for alignment: selecting atoms for aligning the structure.
- Atom selection for RMSF calculation: selecting expected atoms to see their RMSF based on the alignment made by alignment atoms selected above.
- Verify: this button creates the summary of query results unless the query contains syntax or logical errors.

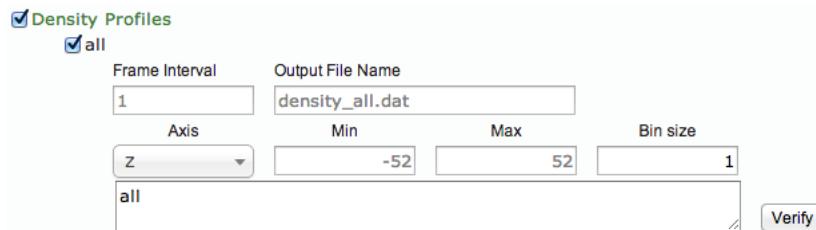
Density Profiles

Analyzing the density of atoms based on the given axis and bin size.

all

This module calculates density of all atoms in system. In order to run this example, users can use trajectories located in \$ST_HOME/trajectory/lipids

- GUI location: \$ST_HOME/stanalyzer/templates/gui/membrane.html
- Module location: \$ST_HOME/static/analyzers/density_all.py



Parameters

- Frame Interval: users can choose frame interval. For example, if time frame 2 is given then ST-analyzer will use every two frames (i.e. 2, 4, 6, 8, ...) to analyze the system instead of every frame.
- Output File Name: users can define the name of output file that contains text format of current module.
- Axis: the basis axis to calculate density.
- Min: the lowest coordinate of expecting range. **NOTE:** this value only affects the range of bin.
- Max: the highest coordinate of expecting range. **NOTE:** this value only affects the range of bin.
- Bin size: individual bin size to analyze density.
- Users can select atoms involved in density calculation. In this case *all* atoms are involved.

Lipid head

This module calculates density of atoms corresponding to lipid head group in system. In order to run this example, users can use trajectories located in \$ST_HOME/trajectory/lipids

- GUI location: \$ST_HOME/stanalyzer/templates/gui/membrane.html
- Module location: \$ST_HOME/static/analyzers/density_lpH.py

lipid head (atom selection for CHARMM phosphatidylcholine head group)

Frame Interval	Output File Name		
1	density_lipid_head.dat		
Axis	Min	Max	Bin size
Z	-52	52	1
segid MEMB and (name P or name N or (name C1* and not name C1) or name O1*)			
<input type="button" value="Verify"/>			

Parameters

- Frame Interval: users can choose frame interval. For example, if time frame 2 is given then ST-analyzer will use every two frames (i.e. 2, 4, 6, 8, ...) to analyze the system instead of every frame.
- Output File Name: users can define the name of output file that contains text format of current module.
- Axis: the basis axis to calculate density.
- Min: the lowest coordinate of expecting range. **NOTE:** this value only affects the **range of bin**.
- Max: the highest coordinate of expecting range. **NOTE:** this value only affects the **range of bin**.
- Bin size: individual bin size to analyze density.
- Users can select atoms involved in density calculation. To select atoms in head group, this example uses “segid MEMB and (name P or name N or (name C1* and not name C1) or name O1*)”. The query selects all carbon atoms of which the name starts with C1 or O1 except C1, and the atom name itself is P or N in the MEMB.
- Verify: this button creates the summary of query results unless the query contains syntax or logical errors.

Lipid tail

This module calculates density of atoms corresponding to lipid tail group in system. In order to run this example, users can use trajectories located in \$ST_HOME/trajectory/lipids

- GUI location: \$ST_HOME/stanalyzer/templates/gui/membrane.html
- Module location: \$ST_HOME/static/analyzers/density_lpT.py

lipid tail (atom selection for CHARMM lipid tails)

Frame Interval	Output File Name		
1	density_lipid_tail.dat		
Axis	Min	Max	Bin size
Z	-52	52	1
segid MEMB and (name C2* or name C3*) and not (name C21 or name C31)			
<input type="button" value="Verify"/>			

Parameters

- Frame Interval: users can choose frame interval. For example, if time frame 2 is given then ST-analyzer will use every two frames (i.e. 2, 4, 6, 8,...) to analyze the system instead of every frame.
- Output File Name: users can define the name of output file that contains text format of current module.
- Axis: the basis axis to calculate density.
- Min: the lowest coordinate of expecting range. **NOTE:** this value only affects the **range of bin**.
- Max: the highest coordinate of expecting range. **NOTE:** this value only affects the **range of bin**.
- Bin size: individual bin size to analyze density.
- Users can select atoms involved in density calculation. To select atoms in tail group, this example uses “*segid MEMB and (name C2* or name C3*) and not (name C21 or name C31)*”. The query selects all carbon atoms in MEMB segment of which the name starts with C2 or C3 except C2 and C3 themselves.
- Verify: this button creates the summary of query results unless the query contains syntax or logical errors.

Water

This module calculates density of water atoms in system. In order to run this example, users can use trajectories located in \$ST_HOME/trajectory/lipids

- GUI location: \$ST_HOME/stanalyzer/templates/gui/membrane.html
- Module location: \$ST_HOME/static/analyzers/density_water.py

The screenshot shows a configuration dialog for the 'water' module. At the top left is a checked checkbox labeled 'water'. Below it are several input fields and controls:

- 'Frame Interval': A dropdown menu showing '1'.
- 'Output File Name': An input field containing 'density_water.dat'.
- 'Axis': A dropdown menu showing 'z'.
- 'Min': An input field containing '-52'.
- 'Max': An input field containing '52'.
- 'Bin size': An input field containing '1'.
- A large text input area at the bottom contains the query 'segid TIP3 and name OH2'.
- A 'Verify' button is located to the right of the text input area.

Parameters

- Frame Interval: users can choose frame interval. For example, if time frame 2 is given then ST-analyzer will use every two frames (i.e. 2, 4, 6, 8,...) to analyze the system instead of every frame.
- Output File Name: users can define the name of output file that contains text format of current module.
- Axis: the basis axis to calculate density.
- Min: the lowest coordinate of expecting range. **NOTE:** this value only affects the **range of bin**.
- Max: the highest coordinate of expecting range. **NOTE:** this value only affects the **range of bin**.
- Bin size: individual bin size to analyze density.
- Users can select atoms involved in density calculation. To select water atoms, this example uses "*segid TIP3 and name OH2*". The query selects all OH2 atoms in TIP3 segment.
- Verify: this button creates the summary of query results unless the query contains syntax or logical errors.

Water dipole

This module calculates water dipole in system. In order to run this example, users can use trajectories located in \$ST_HOME/trajectory/lipids

- GUI location: \$ST_HOME/stanalyzer/templates/gui/membrane.html
- Module location: \$ST_HOME/static/analyzers/density_waterdp.py

The screenshot shows a configuration dialog for the 'water dipole' module. At the top left is a checked checkbox labeled 'water dipole'. Below it are several input fields and dropdown menus:

- 'Frame Interval': A dropdown menu set to '1'.
- 'Output File Name': An input field containing 'density_water_dipole.dat'.
- 'Axis': A dropdown menu set to 'z'.
- 'Min': An input field containing '-52'.
- 'Max': An input field containing '52'.
- 'Bin size': An input field containing '1'.
- 'segid TIP3': An input field containing 'segid TIP3'.

A 'Verify' button is located at the bottom right of the dialog.

Parameters

- Frame Interval: users can choose frame interval. For example, if time frame 2 is given then ST-analyzer will use every two frames (i.e. 2, 4, 6, 8,...) to analyze the system instead of every frame.
- Output File Name: users can define the name of output file that contains text format of current module.
- Axis: the basis axis to calculate distribution.
- Min: the lowest coordinate of expecting range. **NOTE:** this value only affects the **range of bin**.
- Max: the highest coordinate of expecting range. **NOTE:** this value only affects the **range of bin**.
- Bin size: individual bin size to analyze density.
- Users can select atoms involved in density calculation. To select water atoms, this example uses "segid TIP3". The query selects all atoms in TIP3 segment that consists of H1, H2, and OH2.
- Verify: this button creates the summary of query results unless the query contains syntax or logical errors.

Vector selection

This module calculates the distribution of angles (i.e. $\cos\theta$) between the basis axis and selected atoms. In order to run this example, users can use trajectories located in \$ST_HOME/trajectory/lipids

- GUI location: \$ST_HOME/stanalyzer/templates/gui/membrane.html
- Module location: \$ST_HOME/static/analyzers/density_vector.py

The screenshot shows a configuration window titled "vector selection". It includes the following fields:

- Frame Interval: 1
- Output File Name: density_vector.dat
- Axis: Z
- Min: -52
- Max: 52
- Bin size: 1
- Query: segid MEMB and name P

A "Verify" button is located at the bottom right.

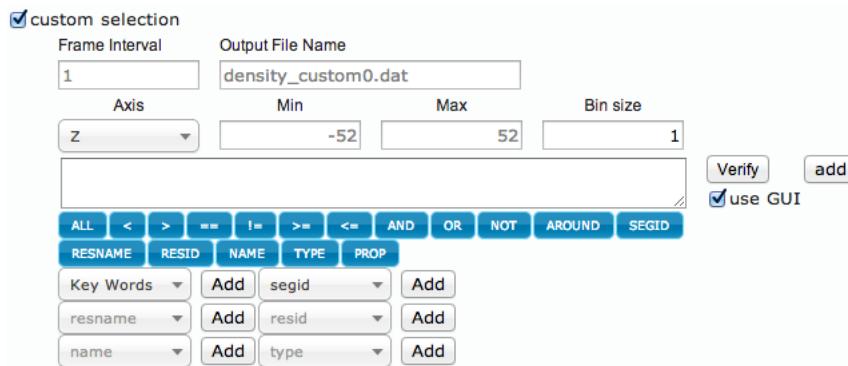
Parameters

- Frame Interval: users can choose frame interval. For example, if time frame 2 is given then ST-analyzer will use every two frames (i.e. 2, 4, 6, 8,...) to analyze the system instead of every frame.
- Output File Name: users can define the name of output file that contains text format of current module.
- Axis: the basis axis to calculate distribution.
- Min: the lowest value of expecting range. **NOTE:** this value only affects the range of bin.
- Max: the highest value of expecting range. **NOTE:** this value only affects the range of bin.
- Bin size: individual bin size to analyze density.
- Users can select atoms involved in vector calculation. This example uses "*segid MEMB and name P*". The query selects all phosphate atoms in MEMB segment.
- Verify: this button creates the summary of query results unless the query contains syntax or logical errors.

Custom selection

This module calculates density based on user-defined atoms. In order to run this example, users can use trajectories located in \$ST_HOME/trajectory/lipids

- GUI location: \$ST_HOME/stalyzer/templates/gui/membrane.html
- Module location: \$ST_HOME/static/analyzers/density_custom.py



Parameters

- Frame Interval: users can choose frame interval. For example, if time frame 2 is given then ST-analyzer will use every two frames (i.e. 2, 4, 6, 8,...) to analyze the system instead of every frame.
- Output File Name: users can define the name of output file that contains text format of current module.
- Axis: the basis axis to calculate distribution.
- Min: the lowest value of expecting range. **NOTE:** this value only affects the range of bin.
- Max: the highest value of expecting range. **NOTE:** this value only affects the range of bin.
- Bin size: individual bin size to analyze density.
- Users can select any atoms in system to analyze the density of them.
- Verify: this button creates the summary of query results unless the query contains syntax or logical errors.
- Add: users can select multiple atoms to make separate outputs.
- Use GUI: by checking this option users can write query with simple GUI interface. For the details about query, please refer '**Selection Query**' section.

Order Parameters

Calculating order parameters of selected atoms.

CHARMM format

This module calculates order parameters of selected atoms. Currently CHARMM format is only available. In order to run this example, users can use trajectories located in \$ST_HOME/trajectory/lipids

- GUI location: \$ST_HOME/stanalyzer/templates/gui/membrane.html
- Module location: \$ST_HOME/static/analyzers/ordpara_charmm.py

The screenshot shows a user interface for calculating order parameters. At the top, there are two checked checkboxes: 'Order Parameters' and 'CHARMM format'. Below these are several input fields and dropdown menus:

- A dropdown menu for 'segid' set to 'MEMB'.
- A dropdown menu for 'resname' set to 'DOPC'.
- A text input field for 'Selection query' containing the text "name C2* and not (name C2 or name C21)".
- A dropdown menu for 'Frame Interval' set to '1'.
- An input field for 'Output File Name' containing "ordpara_charmm_0.dat".
- A dropdown menu for 'Axis' set to 'z'.
- Two buttons at the bottom right: 'Verify' and 'Add'.

Parameters

- segid: choosing segment ID – MEMB is selected.
- resname: choosing residue name – DOPC is selected.
- Selection query: specifying atoms – one of DOPC tail is selected by “*name C2* and not (name C2 or name C21)*”.
- Frame Interval: users can choose frame interval. For example, if time frame 2 is given then ST-analyzer will use every two frames (i.e. 2, 4, 6, 8,...) to analyze the system instead of every frame.
- Output File Name: users can define the name of output file that contains text format of current module.
- Axis: the basis axis to get bilayer normal.
- Verify: this button creates the summary of query results unless the query contains syntax or logical errors.
- Add: users can select multiple atoms to make separate outputs.

Thickness

Calculating thickness of membrane.

Using Phosphate

This module calculates thickness based on phosphate atoms. In order to run this example, users can use trajectories located in \$ST_HOME/trajectory/lipids

- GUI location: \$ST_HOME/stanalyzer/templates/gui/membrane.html
- Module location: \$ST_HOME/static/analyzers/thickness_phosphate.py

The screenshot shows a configuration window for the 'Thickness' module. At the top, there are two checked checkboxes: 'Thickness' and 'Using Phosphate'. Below these are four input fields: 'Query' containing 'name P', 'Frame Interval' containing '1', 'Output File Name' containing 'thickness_phosphate.dat', and an unlabeled empty input field.

Parameters

- Query: selecting phosphate atom.
- Frame Interval: users can choose frame interval. For example, if time frame 2 is given then ST-analyzer will use every two frames (i.e. 2, 4, 6, 8,...) to analyze the system instead of every frame.
- Output File Name: users can define the name of output file that contains text format of current module.

Using Carbon

This module calculates thickness based on carbon atoms. In order to run this example, users can use trajectories located in \$ST_HOME/trajectory/lipids

- GUI location: \$ST_HOME/stalyzer/templates/gui/membrane.html
- Module location: \$ST_HOME/static/analyzers/thickness_carbon.py

Using Carbon

Query	name C22 or name C32
Frame Interval	Output File Name
1	thickness_carbon.dat

Parameters

- Query: selecting carbon atoms. Users must redefine atoms corresponding to user's system.
- Frame Interval: users can choose frame interval. For example, if time frame 2 is given then ST-analyzer will use every two frames (i.e. 2, 4, 6, 8,...) to analyze the system instead of every frame.
- Output File Name: users can define the name of output file that contains text format of current module.

Using custom

This module calculates thickness based on user-defined atoms. In order to run this example, users can use trajectories located in \$ST_HOME/trajectory/lipids

- GUI location: \$ST_HOME/stalyzer/templates/gui/membrane.html
- Module location: \$ST_HOME/static/analyzers/thickness_custom.py

The screenshot shows a configuration dialog for the 'Using custom group' module. It includes dropdown menus for 'segid' (MEMB) and 'resname' (DOPG), a condition field 'name C2* and not (name C2 or na)', a 'Frame Interval' input (1), an 'Output File Name' input ('thickness_custom_0.dat'), and two buttons 'Verify' and 'Add'.

Parameters

- segid: choosing segment ID – MEMB is selected.
- resname: choosing residue name – DOPG is selected.
- Query: selecting atoms. Users can select atoms based on their own purpose.
- Frame Interval: users can choose frame interval. For example, if time frame 2 is given then ST-analyzer will use every two frames (i.e. 2, 4, 6, 8,...) to analyze the system instead of every frame.
- Output File Name: users can define the name of output file that contains text format of current module.
- Add: users can select multiple atoms to make separate outputs.

Average area per lipid

Calculating area of individual lipids and area of lipid surface.

Using Voronoi diagram

This module calculates area per lipid by utilizing Voronoi diagram calculated by Pyhull. In order to run this example, Python should have Pyhull module (<http://pythonhosted.org/pyhull/>). Trajectories used in this example is located in \$ST_HOME/trajectory/lipids

- GUI location: \$ST_HOME/stanalyzer/templates/gui/membrane.html
- Module location: \$ST_HOME/static/analyzers/lipid_per_area_voro.py

Average area per lipid

System Size x= y=

Using Voronoi diagram

Query

Verify

Frame Interval Output File Name

Parameters

- System size: the information is obtained from initial PDB file.
- Query: selecting atoms to define surface area – The given system contains five different residues CHL1, POPC, POPI, PIPI13, and DOPC, so the query must specify atoms to define surface area. In order to define the surface area this example uses a query “*segid MEMB and ((resname CHL1 and name O3) or (resname DOPC and (name C2 or name C21 or name C31)) or (resname POPC and (name C2 or name C21 or name C31)) or (resname POPI and (name C2 or name C21 or name C31)) or (resname PIPI13 and (name C2 or name C21 or name C31)))*” such that Cholesterol uses an atom (i.e. O3) and all other residues including DOPC, POPC, POPI, and PIPI13 use 3 atoms (i.e. C2, C21, and C31) to define the surface.
- Frame Interval: users can choose frame interval. For example, if time frame 2 is given then ST-analyzer will use every two frames (i.e. 2, 4, 6, 8,...) to analyze the system instead of every frame.
- Output File Name: users can define the name of output file that contains text format of current module.

Result Viewer

One of unique characteristics in ST-analyzer is data retrieval function utilized by SQLite3. This section introduces ST-analyzer data retrieval GUI, 'Result Viewer' that consists of simple search engine, graphic viewer, data download, and data management tools.

Project Retrieval

Once *Result Viewer* is launched, users can see the list of existing projects. For data retrieval, *Result Viewer* provides several tools to navigate stored data as following:

- 1) Quick search engine: this tool lists any data that contains user typed words in ID or Project.
- 2) Sorting: by clicking the triangles at ID column, the viewer sorts the data either ascending or descending order.
- 3) Collapsing lists: by clicking the triangle, the data lists associated with a project are collapsed.
- 4) Expanding list: by clicking the triangle, the row is expanded to show associated data lists

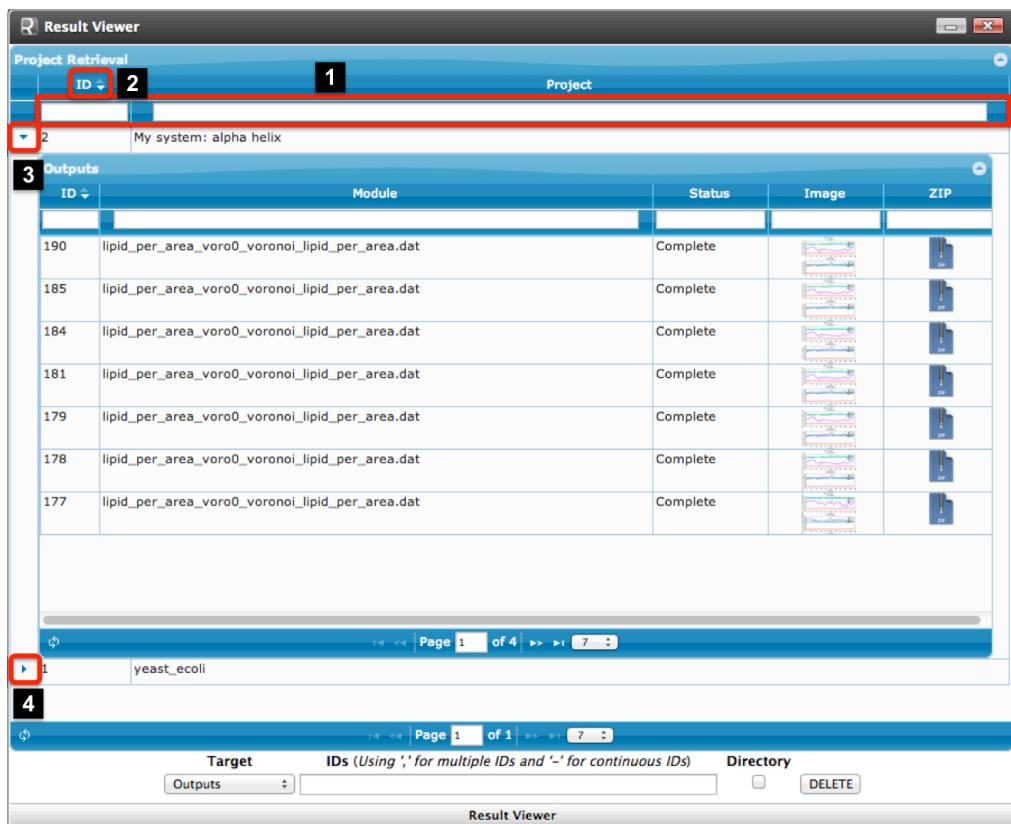


Figure 22. Result Viewer: project retrieval

Outputs

Users can see the results of submitted jobs by expanding each project as explained above. To retrieve actual outputs, ST-analyzer provides two types of data: graph viewer and file download. Output window also supports quick search tool.

- 1) Quick search engine: retrieving data containing the words corresponding to each column
- 2) Image viewer: by clicking thumbnails, users can see corresponding image in a popup window.
- 3) File download: some modules results in multiple outputs, but Output GUI only show an image at a time; therefore downloading entire datasets are necessary. The files contain the raw data of plotting graph, information about defining variables of applied modules and more.

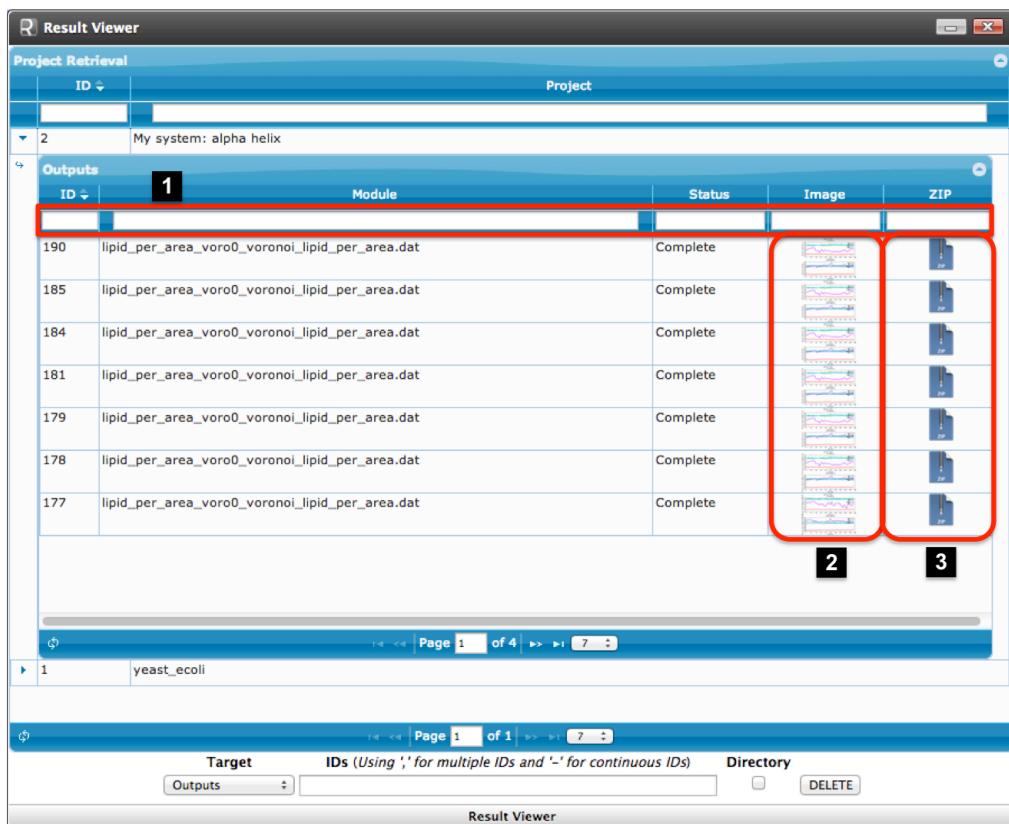


Figure 23. Result Viewer: Outputs

Page navigator

The *Outputs* and *Project Retrieval* can contain many records that cannot be displayed in a small window. To make organized view, ST-analyzer provides page-based lists.

- 1) Refresh button: this button will refresh the lists
- 2) Page navigator: providing easy navigation between pages. The navigator also changes the number of records at each page.

The screenshot shows the 'Result Viewer' application interface. At the top, there's a 'Project Retrieval' section with a table header 'ID' and a row for 'My system: alpha helix'. Below it is an 'Outputs' section with a table header 'ID' and columns for 'Module', 'Status', 'Image', and 'ZIP'. There are seven rows of data, all labeled 'Complete' and showing small thumbnail images and ZIP file icons. At the bottom of the screen, there's a navigation bar with two large black numbers '1' and '2'. Between them is a red-bordered box containing a refresh icon, the text 'Page 1 of 4', and a dropdown menu set to '7'. Below this is another navigation bar with similar controls, labeled 'Page 1 of 1'. At the very bottom, there are buttons for 'Target', 'IDs (Using ',' for multiple IDs and '-' for continuous IDs)', 'Directory', 'DELETE', and 'Result Viewer'.

Figure 24. Result Viewer: Page navigator

Data manager

ST-analyzer provides simple data managing tool that can delete data and jobs in queue.

- 1) Target: select target group to delete items by using corresponding IDs
 - a. Project Retrieval: by deleting a project, all associated outputs are also deleted
 - b. Outputs: deleting output records
 - c. Queue: delete jobs in queue with “*job id*”. **NOTE: deleting outputs will also deletes jobs in queue if the submitted job is in queue.**
- 2) Delete records
 - a. IDs: either project or output IDs are used to identify targeted records. For the consecutive IDs, users can use short hand description. For example to delete Outputs from ID 111 to ID 115, users can simply type 111-115, which is equivalent to writing 111, 112, 113, 114, 115.
 - b. Directory: by checking this option, users can delete both records in database and physical data in output directory.

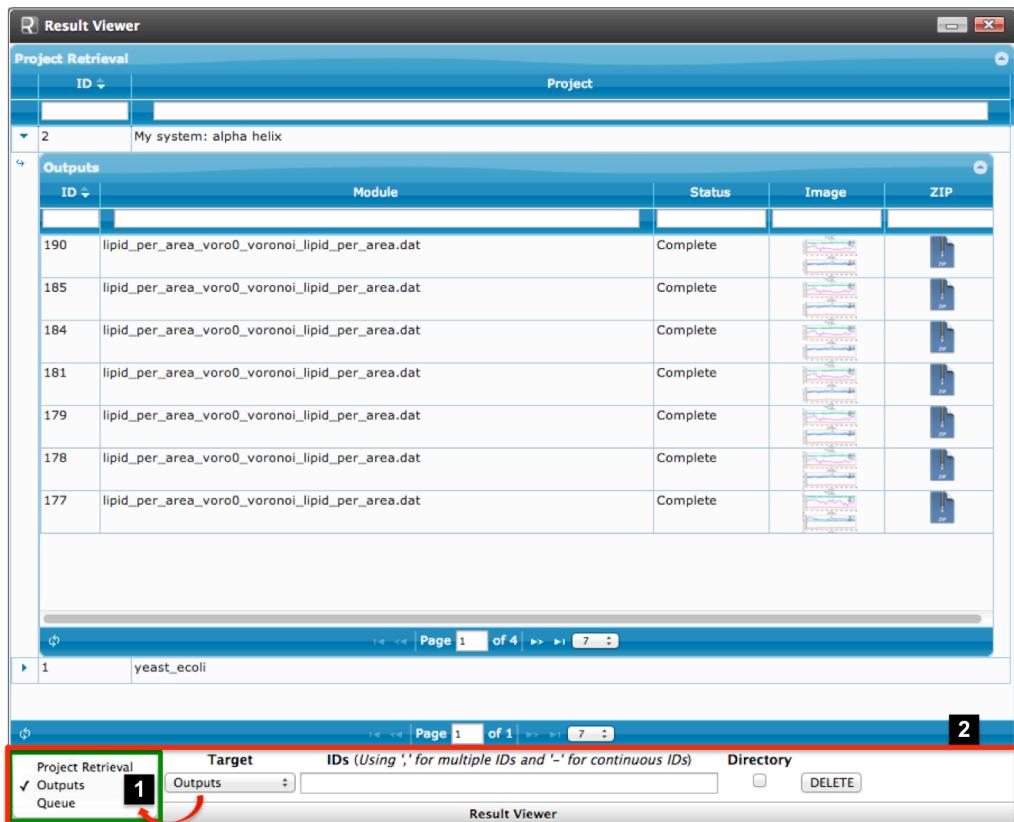


Figure 25. Result Viewer: Data manager

Selection Query

Although ST-analyzer is designed for automatically analyzing simulation trajectories, there are some technical issues making ST-analyzer fully automatic. One of the reasons could be users' diverse demands and interests on the scope of analysis and various biological systems. To maximize the validity of predefined modules in ST-analyzer, selection queries used in MDAnalysis module (<https://code.google.com/p/mdanalysis/>) is incorporated. This section introduces syntax of selection queries with some example queries used in ST-analyzer.

Selection keywords and usages

NOTICE:

- 1) Keywords are case sensitive
- 2) Selections are parsed left to right and parentheses can be used for grouping.
- 3) Pattern only allows wild card ‘*’
- 4) ‘<>’ indicates required user inputs
- 5) ‘[]’ indicates optional user inputs
- 6) ‘|’ indicates alternative choices

- * : wild card used for pattern matching.
 - C1* will retrieve C11, C12, C13, ...
- **segid <segment name>**: select by segid
 - segid DOPC – select all atoms in segment name DOPC
- **resid <residue-number [: range]>**: select by residue ID
 - resid 134 – select residue having ID = 134
 - resid 134:150 – select residues from residue ID 134 to 150
- **resname <residue name>**: select by residue name
 - resname LYS – select Lysine
- **name <atom name>**: select by atom name as given in the topology. Often this is force field dependent
 - name CA – select C α atom
- **type <atom type>**: select by atom type; this is either a string or a number and depends on the force field; it is read from the topology file (e.g. the CHARMM PSF file contains numeric atom types)
 - type OHL – select atoms associated with OHL type

- **atom <segment name> <residue id> <atom name>**: select specified single atom
 - atom DOPC 1 C2 – select C2 atom associated with first residue of DOPC
 - **not <selections>**: select all atoms except atoms associated with selections
 - not name C1 – select all atoms except C1 atom
 - **and | or**: combine two selections according to the rules of Boolean algebra
 - segid MEMB and (not name C1 or not name O1*) – select all atoms from MEMB segment except C1 atom and atom names start with O1.
 - **around <distance> <selections>**: selects all atoms a certain cutoff away from another selection
 - around 3.5 protein – select all atoms that are within 3.5Å from the protein and not belonging to protein
 - **point <x> <y> <z> <distance>**: select all atoms located within certain distance from a point defined with x, y, and z coordinates
 - point 5.0 5.0 5.0 3.5 – select all atoms within 3.5Å of the point located at (5.0, 5.0, 5.0)
 - **prop [abs] <x|y|z> <<|=|=|=|=|=|=>> <value>**: select atoms based on position
 - prop abs z <= 5.0 – select all atoms coordinates of z axis within -5.0 <= z <= 5.0

For details and up-to-dated selection queries, please visit:

http://mdanalysis.googlecode.com/git/package/doc/html/documentation_pages/selections.html