Automated Docking Workflow

This repository contains a set of Python scripts and configuration files designed to automate the docking process using AutoDock Vina. The workflow includes preparing input files, running multiple docking simulations, analyzing the results, and selecting the most promising poses.

Table of Contents

Overview

This automated docking workflow streamlines the process of running multiple AutoDock Vina simulations and analyzing the resulting poses. It consists of several Python scripts that handle different aspects of the docking procedure, including:

- Running multiple Vina docking calculations
- Extracting and sorting poses based on energy
- Calculating RMSD values to identify unique poses
- Merging and refining selected poses for further analysis

Requirements

To use this workflow, you will need the following software and libraries installed:

- Python 3.x
- AutoDock Vina
- numpy
- glob
- os
- shutil
- subprocess
- time

Installation

1. Clone this repository to your local machine:

```
git clone <repository_url>

cd <repository_name>
```

2. Ensure that all required Python libraries are installed. You can use pip to install them:

```
pip install numpy
```

Make sure AutoDock Vina is installed and accessible in your system's PATH.

Usage

1. Prepare your ligand and receptor files in .pdbqt format.
2. Create a directory for your docking experiment.
3. Place the ligand .pdbqt, receptor .pdbqt, all the scripts (Automated_Docking_VINA.py, docking_calculus.py, add_name.py, analysis_docking_general_auto.py, analysis_docking_particular_auto.py, and sort_poses.py) and the config.txt file into this directory.
4. Modify the config.txt file and the scripts as needed to match your specific docking parameters (see Configuration).
5. Run the Automated_Docking_VINA.py script using sbatch:

```
sbatch Automated_Docking_VINA.py
```

File Descriptions

- Automated_Docking_VINA.py: Main script that orchestrates the entire docking workflow. It launches docking_calculus.py, copies necessary scripts, executes pose analysis, and sorts the final poses.
- docking_calculus.py: Script that runs AutoDock Vina multiple times (default is 50) using the parameters specified in config.txt. It creates a folder named "receptor_ligand" to store the output files from each run.
- add_name.py: Script that adds the filename to each MODEL line in the .pdbqt files, which is useful for tracking the origin of each pose.
- analysis_docking_general_auto.py: Script that filters poses based on RMSD values and merges the selected poses into a single merged.pdbqt file.
- analysis_docking_particular_auto.py: Script used for further analysis of the merged poses, generating specific output files for final selection.
- sort_poses.py: Script that sorts the poses based on their energy scores and creates a final directory containing the selected poses.
- config.txt: Configuration file for AutoDock Vina, containing parameters such as receptor and ligand coordinates, search box dimensions, and other docking parameters.

Configuration

config.txt

This file contains the configuration parameters for AutoDock Vina. You should modify this file to specify the following:

- receptor: Path to the receptor .pdbqt file.
- ligand: Path to the ligand .pdbqt file.
- center_x, center_y, center_z: Coordinates of the center of the search box.
- size_x, size_y, size_z: Dimensions of the search box.
- num_modes: The number of poses to generate.

docking_calculus.py

- folder: Name of the folder where the docking results will be saved (default is "receptor_ligand").
- The number of docking runs (default is 50) can be modified by changing the range in the for loop:

python

```
for i in range(1, 51):
```

analysis_docking_general_auto.py

- pose_inicial: The first pose to be analyzed (default is 1).
- pose_final: The last pose to be analyzed (default is 10).
  These parameters determine the range of poses that will be considered for RMSD analysis and merging.
- RMSD threshold: the criteria for RMSD can be changed in the line:

python

```
 if RMSD_final[i][2]<2:
```

Adaptable parameters

File	Parameter	Description
config.txt	ligand, receptor names	Names of the ligand and receptor .pdbqt files.
config.txt	box parameters	Coordinates and dimensions of the search box.
config.txt	num_modes	Number of poses to generate for each docking run.
docking_calculus.py	folder name	Name of the folder where docking results are stored.

File	Parameter	Description
docking_calculus.py	number of docking runs	Number of times AutoDock Vina is executed.
analysis_docking_general_auto.py	pose_inicial, pose_final	Range of poses to be considered for RMSD analysis.
analysis_docking_general_auto.py	RMSD threshold	Maximum RMSD value allowed for considering poses as similar (default is 2).

Procedure

1. **Preparation**:
    - Create a folder for the docking run.
    - Place all necessary scripts (.py files), the config.txt file, and the ligand and receptor .pdbqt files in this folder.
2. **Docking Calculation**:
    - Run the Automated_Docking_VINA.py script. This will:
        - Launch docking_calculus.py to perform multiple Vina docking runs.
        - Create a subfolder (e.g., "receptor_ligand") containing the results of each docking run.
3. **Analysis**:
    - The Automated_Docking_VINA.py script will then:
        - Copy the necessary analysis scripts into the subfolder.
        - Execute add_name.py to add filenames to the MODEL lines in the .pdbqt files.
        - Run analysis_docking_general_auto.py to select poses based on energy and RMSD, merging them into merged.pdbqt.
4. **Final Selection**:
    - The script creates a final_selection_1 folder.
    - analysis_docking_particular_auto.py is executed for further analysis.
5. **Sorting and Final Output**:
    - An ultimate_selection folder is created, and the selected .pdbqt files are moved into it.
    - sort_poses.py is executed to sort the poses based on their energy scores, and the final results are placed in the resultado_energia folder.
6. **Analysis in Discovery Studio**:

  - Download the files from the ultimate_selection folder and analyze them using Discovery Studio or your preferred molecular visualization software.

Authors

- Cristina Garcia Iriepa
- Lorenzo Gramolini (github.com/lorenzo-gram)
- Richard Lopez Corbalan (github.com/richardloopez)

Citation

If you use this code in your research, please cite the following:

- Garcia-Iriepa, C
- Gramolini, L
- Lopez-Corbalan, R