

# PARGT User's Guide for Windows

## (Version 1.0)

PARGT Documentation (last updated: December 17, 2019)

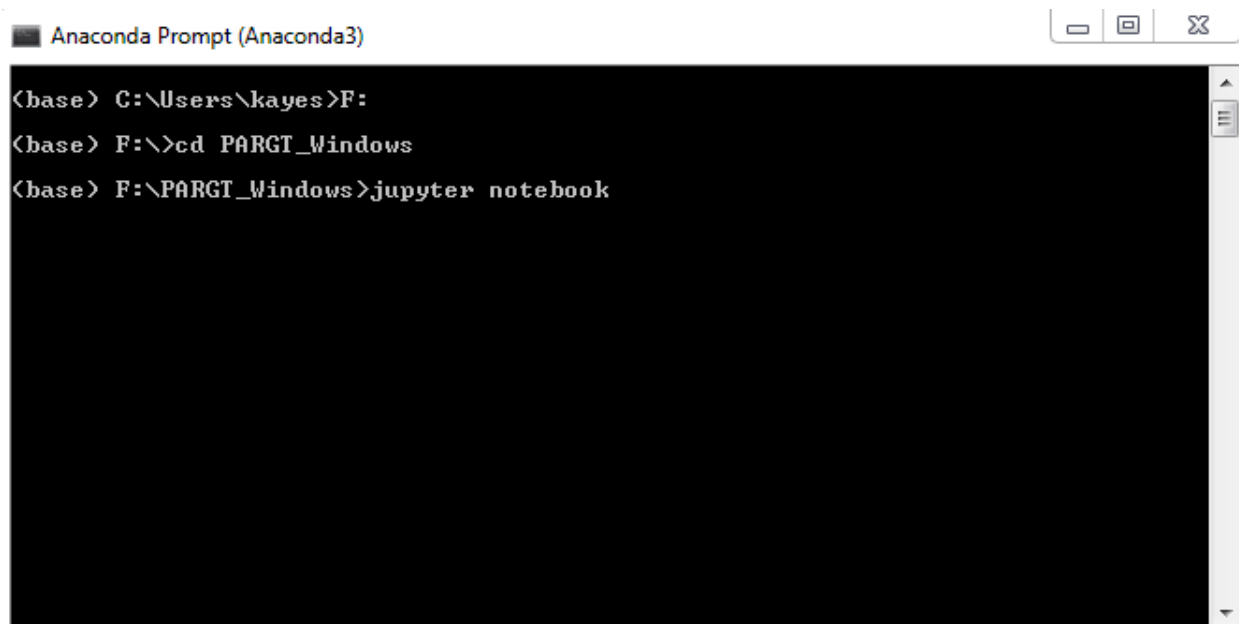
School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA, USA

For more information, please contact Abu Sayed Chowdhury (Email: [abu.chowdhury@wsu.edu](mailto:abu.chowdhury@wsu.edu),  
[abu.chowdhury2005@gmail.com](mailto:abu.chowdhury2005@gmail.com))

**PARGT** is a standalone software package developed to predict antimicrobial resistance proteins in bacteria. It is written mainly in python3 though some modules are implemented in R and called from main python script.

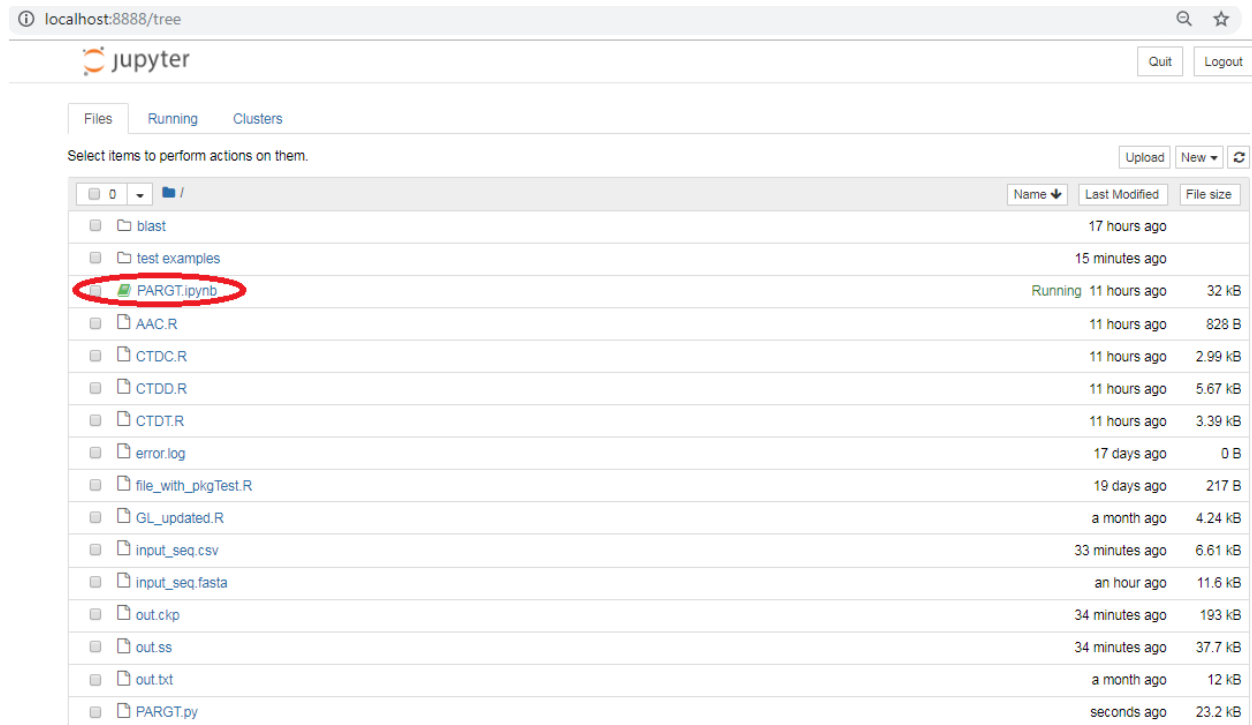
1. To install and use PARGT on Windows:

- Download “PARGT\_Windows.zip” from Github link- <https://github.com/abu034004/PARGT> . Unzip it and save it in a drive.
- Download “Materials\_PARGT\_Windows.zip” from <https://drive.google.com/file/d/1pUFyEKdh2DjuzMB4j0mK28DQM8RzH3i4/view?usp=sharing> . Unzip it and save the unzipped files in the “PARGT\_Windows” folder. Please note that this zip file contains legacy BLAST and PSIPRED that also can be downloaded from <ftp://ftp.ncbi.nlm.nih.gov/blast/executables/legacy.NOTSUPPORTED/2.2.26/> and <https://drive.google.com/file/d/1TMiXngVS0HkWorsZWBHiaJiexphaiwyH/view?usp=sharing> , respectively if different versions of BLAST and PSIPRED needed to install in your machine.
- To install “Jupyter Notebook”, visit the link- <https://www.anaconda.com/download/> and install anaconda for python 3. To install it from command prompt, please visit the link- <https://jupyter.org/install> .
- Edit the fasta file “input\_seq.fasta” to include sequences you want to test. Note that there are some default example test sequences in this file, and you can also use example test sequences available in the folder “test examples”.
- Now, to start Jupyter Notebook, open “Anaconda Prompt (Anaconda 3)”. Then write command to go the the directory where “PARGT\_Windows” folder is saved and then type command- “Jupyter notebook”. For example, if the folder “PARGT\_Windows” is saved in “F:\PARGT\_Windows” location then the run the commands given below.



```
Anaconda Prompt (Anaconda3)
(base) C:\Users\kayes>F:
(base) F:\>cd PARGT_Windows
(base) F:\PARGT_Windows>jupyter notebook
```

Then after couple of seconds, a new page for Jupyter Notebook will appear in your default browser. A screenshot is given below.



localhost:8888/tree

jupyter

Quit Logout

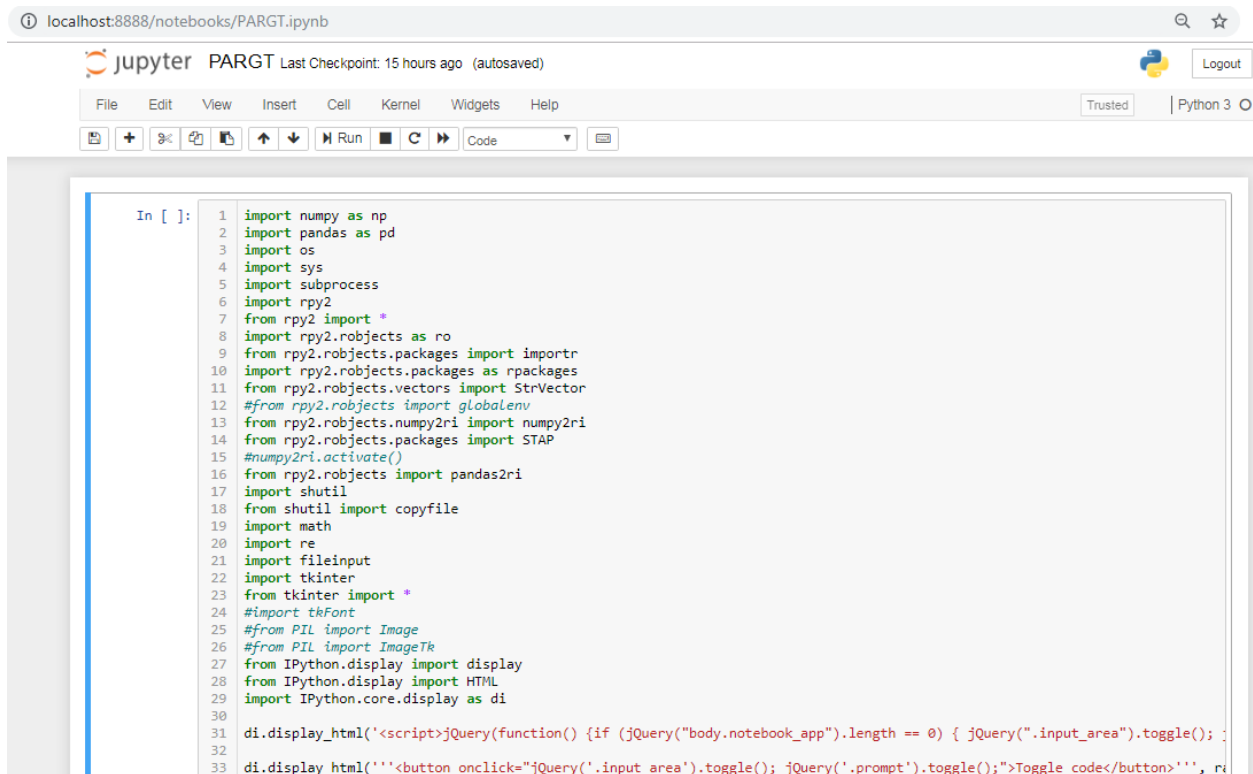
Files Running Clusters

Select items to perform actions on them.

Upload New

Name	Last Modified	File size
blast	17 hours ago	
test examples	15 minutes ago	
PARGT.ipynb	Running 11 hours ago	32 kB
AAC.R	11 hours ago	828 B
CTDC.R	11 hours ago	2.99 kB
CTDD.R	11 hours ago	5.67 kB
CTDT.R	11 hours ago	3.39 kB
error.log	17 days ago	0 B
file_with_pkgTest.R	19 days ago	217 B
GL_updated.R	a month ago	4.24 kB
input_seq.csv	33 minutes ago	6.61 kB
input_seq.fasta	an hour ago	11.6 kB
out.ckp	34 minutes ago	193 kB
out.ss	34 minutes ago	37.7 kB
out.txt	a month ago	12 kB
PARGT.py	seconds ago	23.2 kB

Now, click on PARGT.ipynb (marked by red circle in the above figure). Then it will be open up in a new browser. A screenshot is given below. Please note that PARGT.py script is given if a user wants to use it instead of notebook (.ipynb) version.

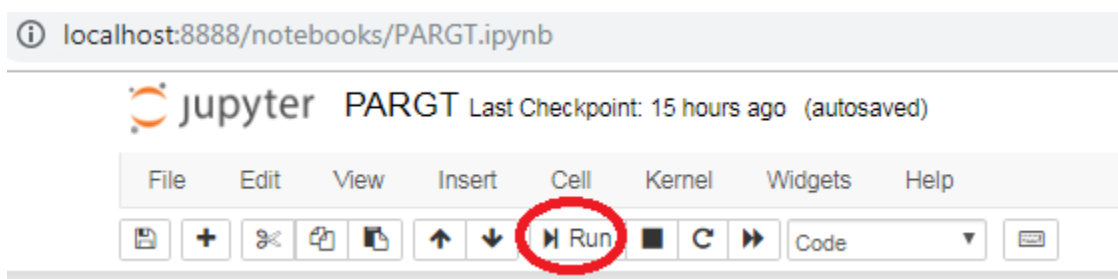


The screenshot shows a Jupyter Notebook interface in a web browser. The address bar shows 'localhost:8888/notebooks/PARGT.ipynb'. The notebook title is 'PARGT' with a subtitle 'Last Checkpoint: 15 hours ago (autosaved)'. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations and execution. A code cell is active, containing the following Python code:

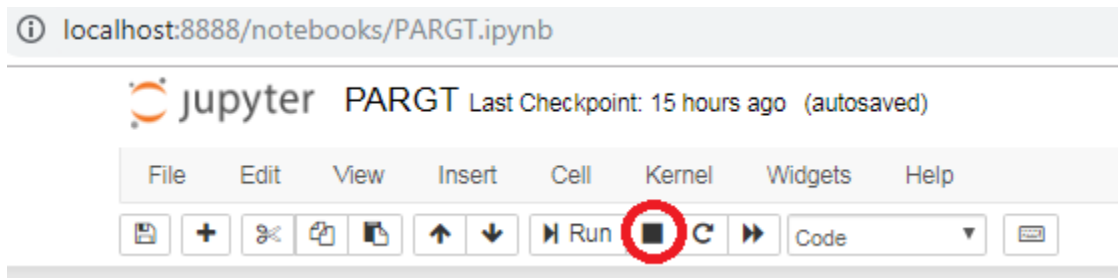
```
In [ ]: 1 import numpy as np
2 import pandas as pd
3 import os
4 import sys
5 import subprocess
6 import rpy2
7 from rpy2 import *
8 import rpy2.robjects as ro
9 from rpy2.robjects.packages import importr
10 import rpy2.robjects.packages as rpackages
11 from rpy2.robjects.vectors import StrVector
12 #from rpy2.robjects import globalenv
13 from rpy2.robjects.numpy2ri import numpy2ri
14 from rpy2.robjects.packages import import STAP
15 #numpy2ri.activate()
16 from rpy2.robjects import pandas2ri
17 import shutil
18 from shutil import copyfile
19 import math
20 import re
21 import fileinput
22 import tkinter
23 from tkinter import *
24 #import tkFont
25 #from PIL import Image
26 #from PIL import ImageTk
27 from IPython.display import display
28 from IPython.display import HTML
29 import IPython.core.display as di
30
31 di.display_html('<script>jQuery(function() {if (jQuery("body.notebook_app").length == 0) { jQuery(".input_area").toggle(); ;
32
33 di.display_html('<<button onclick="jQuery('.input_area').toggle(); jQuery('.prompt').toggle();">Toggle code</button>''', r
```

Before running the script, please install rpy2 package using command “conda install -c r rpy2” from the Anaconda command prompt. Also, please install tzlocal package using command “conda install -c conda-forge tzlocal” if needed. For details, please visit <https://anaconda.org/r/rpy2> and <https://anaconda.org/conda-forge/tzlocal> for rpy2 and tzlocal packages, respectively. If your system cannot locate e1071 package, please install e1071 package using command “conda install -c r r-e1071”. For details, please visit <https://anaconda.org/r/r-e1071> for e1071 packages. All of the remaining packages should come with Jupyter Notebook by default. In the worst case, if any package is missing, please use ‘conda install’ command to install the required packages. Also, users can consider ‘pip install’ command to install necessary packages as an alternative option.

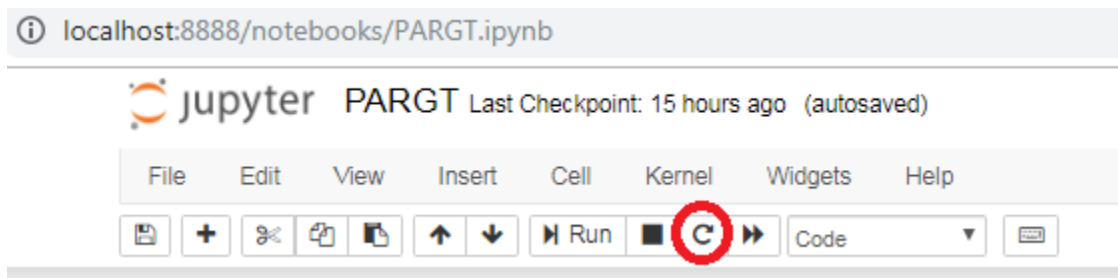
Now, click inside a code cell and then click on “Run” button (red marked in the following screenshot).



If you want to stop running the script, please click on the “Interrupt the kernel” (i.e., stop button) (red marked in the following figure).



If you want to restart the kernel after facing an error, please click on “restart the kernel” button (red marked in the following figure).



After running the script, a GUI should appear as below.



Please click on the option menu to see all options available in the tool (please see below the figure).



Brief description of the options is given below.

- (a) **Predict aac/bla/dfr/bac/van resistance sequences:** This option is for predicting acetyltransferase (*aac*), beta-lactamase (*bla*), dihydrofolate reductase (*dfr*), bacitracin (*bac*) and vancomycin (*van*) antimicrobial resistance (AMR) proteins from the input fasta sequences in the "input\_seq.fasta" file.
- (b) **Include new aac/bla/dfr/bac/van resistance sequences:** These options are for a user who wants to include new known AMR sequences to the original training data comes with this tool. Again the sequences that a user wants to add need to be given in the "input\_seq.fasta" file.
- (c) **Include new aac/bla/dfr/bac/van non-resistance sequences:** These options are for a user who wants to include new known non-AMR sequences to the training data that comes with this tool. Again the sequences that a user wants to add need to be given in the "input\_seq.fasta" file.

(d) **Restore training sets:** This option is to reset all training datasets back to the original version came with this tool.

After successful execution, the tool will show a message- “The operation is successful!” and the predicted output will be available in the “predicted\_resistance\_sequences.fasta” file located in the “PARGT\_Windows” folder.

If you find our tool useful, please cite our following papers.

**Citations:**

1. Chowdhury, A.S., Call, D.R. and Broschat, S.L., 2020. PARGT: A Software Tool for Predicting Antimicrobial Resistance in Bacteria. Scientific Reports.
2. Chowdhury, A.S., Call, D.R. and Broschat, S.L., 2019. Antimicrobial Resistance Prediction for Gram-Negative Bacteria via Game Theory-Based Feature Evaluation. Scientific Reports.