Paacman Description and Installation

I. Introduction

In order to analyze interesting amino acid composition trends between subgroups of the *E. coli* ribosome (e.g., 30S, 50S, and the accessory/translation factors), a script was developed to quickly analyze an entire set of individual FASTA files. This Python script is called "Protein amino acid composition analysis," or Paacman. Paacman is most similar to the "ProtParam" tool that is available on the ExPASy bioinformatics resource portal.[1] Paacman is able to analyze multiple protein FASTA files at once to show the amino acid composition of an entire set of proteins. Paacman also lists all possible di-amino acid sequences, as well as highlights di-amino acid sequences important in chemical protein synthesis, such as potential ligation junctions, aspartimide-prone sequences, and pseudoproline sites.

A simple workflow for Paacman is shown in Figure 1. Paacman first counts the number of each amino acid within the FASTA files located in the user's folder of interest. The script then generates an Excel output file listing the number of individual amino acids, as well as total amino acid numbers, for the entire list of proteins within the folder (sheet 1 in Figure 1). In addition, Paacman generates a heat map for amino acid composition for the set of proteins. This heat map is found within the output Excel file.

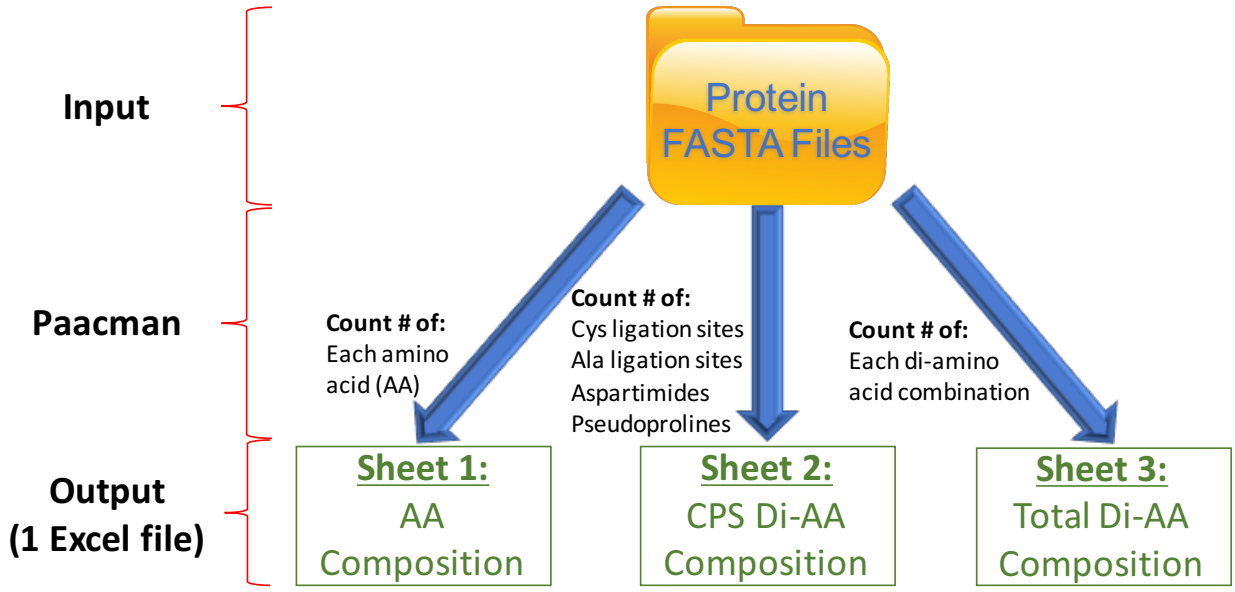

Figure 1: Workflow for running Paacman on a group of FASTA files.

Paacman also performs a di-amino acid composition analysis on the user's FASTA files. The script currently performs 2 types of di-amino acid searches: a search for di-amino acid sequences important to complete protein synthesis (sheet 2 in Figure 1), and a total di-amino acid search (e.g., all 400 possible di-amino acid sequences; sheet 3 in Figure 1). Potential cysteine ligation sites (XC), alanine ligation sites (XA), aspartimide-prone sites (DX), and serine/threonine pseudoprolines (S/T)X are specifically highlighted in the "CPS Di-AA Composition" search. However, the script simply lists the number of these sites present in each protein of interest;

Paacman does not currently remove di-amino acid sequences that are not compatible with total chemical synthesis strategies. The script lists the results for each search within the same Excel output file as the amino acid composition, but each result is in a separate sheet.

II. Installing Paacman

A. Mac OS X

The installation processes are done using the Terminal application, which can be found in the Utilities section of the Applications folder.

Mac OS X comes with Python 2.7 (http://www.python.org) pre-installed. Type "python" in the terminal to confirm the presence of Python version 2.7.x (the value of x is not important). Otherwise, download and install Python version 2.7.x (https://www.python.org/downloads). Paacman will not work with Python 3.0 or higher. After checking the version of Python, you can exit Python by entering "exit()" into the terminal.

1. Prepare a script directory

1.1: On your desktop, prepare a new folder. This folder will be your "script directory," which will store Paacman and allow the script to be called within Terminal.

1.2: To run Paacman in any location on your Mac, your operating system needs to know where the script is located. Your "bash profile" tells the operating system where to look when commands are entered into Terminal, so you will need to export your script directory's path into your bash profile.
- Enter "cd" into the Terminal window.
- Enter "nano .bash_profile".
- Export the path of your script folder in your bash profile. This can be done by typing into a blank line "export PATH=$PATH:" followed by dragging your script folder into the Terminal window. This will paste your folder's path into the line. (example: export PATH=$PATH:/Users/nszabo/Desktop/scripts)
- Press "ctrl" + "o", followed by "enter", to write out and save your modified bash profile. Press "ctrl" + "x" to exit the bash profile.
- Restart Terminal to have the changes take effect.

1.3: Any scripts that you place into this script directory can now be called within Terminal by simply typing the entire name of the script into the Terminal window.

2. Download "Paacman.py" and place into the script directory

2.1: Go to the following Github repository to find the source code for Paacman: https://github.com/kay-lab/Paacman

2.2: Click the "Clone or download" button. Choose the "Download ZIP" option to download "Paacman.py" and all the accompanying files to your computer.

2.3 Place the "Paacman.py" script into your script directory. The accompanying files do not need to be stored in the script directory, as these just contain information about what Paacman does, as well as describe the "MIT" license.

3. Install the openpyxl Python library

3.1: Enter "cd" into the Terminal window.

3.2: Enter "sudo easy_install openpyxl". You will then be prompted to enter a password, which is the one linked to your Mac user account.
- If your account does not have administrator privileges, then this command will not install openpyxl. If you know the username and password of an account with administrative privileges, you will need to log in to your Mac with this account to install openpyxl.

B. Windows

The following instructions describe how to install Cygwin, a collection of tools that provides functionality similar to Linux on Windows. Since Linux and Mac OS are similar to each other, Cygwin is similar in feeling to the Mac OS's Terminal application.

While this tutorial was written and validated by using Cygwin to run Paacman, you should not need to install Cygwin to run Paacman. Alternatively, you can install Python 2.7, install the "openpyxl" package into Python 2.7's package library, and download "Paacman.py" from the following Github repository: https://github.com/kay-lab/Paacman. In order to run Paacman through this method, you will need to make sure that you place the "Paacman.py" script into the folder containing your FASTA text files (see section III below), and then run Paacman through the IDLE shell that comes with Python 2.7.

1. Download and install Cygwin

1.1: Download and run the Cygwin installation executable file, which can be found on Cygwin's website: https://cygwin.com/install.html. You will need to install either the 32- or 64-bit version of the program, depending on the bit version of your Windows operating system.

1.2: Open the installation executable file. Follow the on-screen Cygwin installation instructions, as well as the following instructions, to correctly install Cygwin with the necessary packages to run Paacman:
- When the installation executable asks you to choose a download source, choose "install from Internet".
- For the "Select Root Install Directory" page, you can place Cygwin anywhere that you wish, as well as make it available for all users or just for your user account. Leaving the default root directory will work.
- For the "Select Local Package Directory" page, you can select any location to store the installation files. The default location suggested by the executable will work fine.
- For the "Choose a download site" page, you can select any site to install Cygwin from. When preparing this tutorial, the "http://cygwin.mirror.constant.com" site was selected and worked.

- When the “Select packages” page appears, type “Python” into the Search box. You will see a category called “Python” appear in the list of packages. Click on the word “Default” listed to the right of the Python package, and the word will change to “Install”.
- Next, search for “nano” in the search box. Next to the word “All” that appears in the results, click on the word “Default” to change it to “Install”. Nano is a text editor that will help with setting up your script directory later.
- Click the “Next” button in the bottom right corner of the page.
- Install the suggested dependency files. Ignore any warning messages telling you to install packages from Win32, as these are not important for Paacman functionality.
- Allow Cygwin to install on your computer. This can take a long time (1-2 hours). If you need to pause the installation, you can close the installation executable and simply restart the executable at a later time. All of your selections will still remain when you restart, so you just need to click “Next” until the installation picks up where it left off.
- Once Cygwin is finished installing, you may see a few post-installation errors. These can be ignored, as they do not affect the functionality of Paacman. Finish the executable installation instructions, and Cygwin will be ready to use!

2. Check for the correct version of Python in Cygwin.

2.1: Launch Cygwin and enter “python” into the command line. This will launch Python, and you will see the version next to the word “Python”. Make sure that version 2.7.x (the value of x does not matter) is what launches, as Paacman is written for Python 2.7. If a later version of Python launches, you will need to go back to the Cygwin installation executable to uninstall this version of Python and make sure that 2.7.x gets installed.

2.2: Enter “exit()” into the command line to close Python after you have checked the version.

3. Prepare a script directory

3.1: On your desktop, prepare a new folder. This folder will be your “script directory,” which will store Paacman and allow the script to be called within Cygwin.

3.2: To run Paacman in any location on your computer, your operating system needs to know where the script is located. Your “bashrc” tells the operating system where to look when commands are entered into Cygwin, so you will need to export your script directory’s path into your bashrc.
- Enter “cd” into the Cygwin command line.
- Enter “nano –w ~/.bashrc” into the command line.
- Scroll down to the bottom of the file that appears in the Cygwin window. The cursor needs to appear in the first blank line at the end of the file (a line that does not start with “#”).
- Export the path of your script folder into bashrc. This can be done by typing into the blank line “export PATH=$PATH:” followed by dragging your script folder into the terminal window. This will paste the folder’s path into the Cygwin window. (example: export PATH=$PATH:/Users/nszabo/Desktop/scripts)
- Press “ctrl” + “o”, followed by “enter”, to write out and save your modified bashrc. Press “ctrl” + “x” to exit the bashrc.

- Restart Cygwin to have the changes take effect.

3.3: Any scripts that you place into this script directory can now be called within Cygwin by simply typing the entire name of the script into the Cygwin window.

4. Download "Paacman.py" and place into the script directory

4.1: Go to the following Github repository to find the source code for Paacman: https://github.com/kay-lab/Paacman

4.2: Click the "Clone or download" button. Choose the "Download ZIP" option to download "Paacman.py" and all the accompanying files to your computer.

4.3: Place the "Paacman.py" script into your script directory. The accompanying files do not need to be stored in the script directory, as these just contain information about what Paacman does, as well as describe the "MIT" license.

5. Install the openpyxl Python library

5.1: Enter "cd" into the Cygwin command line.

5.2: Enter "easy_install-2.7 openpyxl" into the Cygwin command line. This will install the openpyxl Python package into your Python package library.

III. Using Paacman

A. Download separate FASTA files for each protein of interest. Save each FASTA file as separate .txt files, and place all of the .txt files into one folder on your computer.
   - NOTE: You may name the file with whatever you wish, but the end of the file must have the extension ".txt" to work with Paacman. In addition, if you place "fasta" or ".fasta" right before the ".txt" extension (e.g., Insulin fasta.txt or Insulin.fasta.txt), then Paacman will trim the word "fasta" out of the file name when writing the protein name into the output Excel file. However, if "fasta" is written in uppercase, then Paacman will not trim it out of your protein name. The script still works to analyze amino acid compositions of your protein, but the name of your protein will have FASTA written at the end in the Excel file (e.g., Insulin FASTA or Insulin.Fasta).

B. Using the Terminal (Mac) or Cygwin (Windows) application, change the working directory to the folder containing the FASTA .txt files:
   - Type "cd " into the Terminal or Cygwin window (note the space following cd), followed by dragging your folder into the Terminal/Cygwin window. This pastes the path of your folder into the command line.
   - Press "enter".

C. Launch Paacman by simply entering "Paacman.py" into the Terminal/Cygwin window.

D. When Paacman is finished, you will see an output Excel document within your folder. The title of the file will be "AA Analysis for <FolderName>". This Excel file shows the single amino acid composition analysis in the first sheet, the important di-amino acid sequences for complete protein synthesis in the second sheet, and the total di-amino acid sequence analysis in the third sheet.

References

1. Wilkins MR, Gasteiger E, Bairoch A, et al. Protein identification and analysis tools in the ExPASy server. *Methods Mol Biol.* 1999;112:531-552.