

DTarray_pro installation and basic usage

Aaron Maurais

24 September 2018

Contents

1	Introduction	1
2	Installation	1
2.1	Download and unpack DTarray_pro archive file from GitHub .	2
2.2	Build DTarray_pro executable	2
2.3	Adding a shortcut for DTarray_pro (optional)	3
3	Usage	5

1 Introduction

DTarray_pro extracts Uniprot ID numbers, molecular weights, and spectral counts from `.dtafilter` files stored in the working directory. Protein data is combined into one dataset and written to the working directory as a tab delimited text file (.tsv). This document will describe how to install the latest version of DTarray_pro step by step. Some experience using a unix shell is assumed.

2 Installation

DTarray_pro is hosted at GitHub, which is a free hosting service for distributed version control in software development. The latest stable version of DTarray_pro will be posted at https://github.com/ajmaurais/DTarray_pro/releases

2.1 Download and unpack DTarray_pro archive file from GitHub

- Navigate to the [releases](#) tab on the DTarray_pro GitHub page.
- The files for the latest release should be at the top of the page.
- Download the file: **Source code (tar.gz)** for the latest release, to your computer.
- DTarray_pro expects to be installed in `~/local`. The program needs data stored in text files in `~/local/DTarray_pro-1.7.4/db` for some features to work. First make the directory `~/local` on your `pleiades` account if it doesn't already exist.
- Transfer the source code archive (should be named something like `DTarray_pro-1.7.4.tar`) to your `pleiades` account using your FTP client of choice.
- The source code archive has to be unpacked before you can access it. To unpack the `.tar` type the following commands in your terminal.

```
$ cd ~/local
$ tar -xvf DTarray_pro-1.7.4.tar
```

- As a result, a new directory should be created in `/local` named `DTarray_pro-1.7.4`
- Once you have unpacked the archive, you no longer need the `.tar` file and can delete it if you wish.

2.2 Build DTarray_pro executable

- Before you can use DTarray_pro, you have to build the executable from source. Fortunately DTarray_pro is configured to work with a build automation tool called `make` so the process should be straightforward.
- To build DTarray_pro run the following commands in your terminal.

```
$ cd ~/local/DTarray_pro-1.7.4/
$ ./configure
$ make
```

- After you have run `make`, there should be several new files in the `DTarray_pro-1.7.4` directory. If everything worked, the executable file should be located at `DTarray_pro-1.7.4/bin/DTarray`

2.3 Adding a shortcut for `DTarray_pro` (optional)

To run `DTarray_pro` you have to navigate on your terminal to a folder which contains `DTASelect-filter` files then type the full path to the executable file relative from the directory you are currently in. Its possible to install a program system wide so you don't have to type the path every time, but without administrative privileges, its a bit complicated. A workaround is to create a shortcut or alias to the executable file. This section will explain how to add an alias for `DTarray_pro` to your shell profile on `pleiades`

- To add an alias for `DTarray_pro`, you will have to edit your shell profile, which is a file stored in your home directory named `.tcshrc`.
- To edit your shell profile, you will use a command line text editor called `nano`. To open `.tcshrc` in `nano`, type:

```
$ cd
$ nano .tcshrc
```

- After starting `nano`, your terminal window should look something like this:

```

GNU nano 2.0.9 File: .tcshrc Modified
setenv MODULE_VERSION 3.2.9
source /usr/public/Modules/3.2.9/init/tcsh

#python enviroment vars
setenv PATH ~/bin:$PATH
setenv PATH ~/local/python/Python-2.7.11/;$PATH
setenv PYTHONPATH ~/local/python/Python-2.7.11
setenv PATH ~/local/bin:$PATH

module load base torque sequest
module load python/2.7.10
module load gcc/4.8.4

limit coredumpsizes 0
# Set window name and prompt
set prompt="%{033}0;%m:%c\007%)[%n@%m %c]$ "
set correct-command
set autologout=180

umask 27
alias back 'set back=$old; set old=$cwd; cd $back; unset back; dirs'
alias h history
alias gunzip 'gzip -d'
alias home 'cd ~/'
alias ls "ls -Cr"

[ Read 56 lines ]
Get Help WriteOut Read File Prev Page Cut Text Cur Pos
Exit Justify Where Is Next Page UnCut Text To Spell

```

- Scroll to the bottom of the file and add the line:
alias DTarray "~/local/DTarray_pro-1.7.4/bin/DTarray"

```

GNU nano 2.0.9      File: .tcshrc      Modified
alias cim_combine_byProtein "cimage_combine by_protein output_rt_10_sn_2.5.to_excel.txt dta"
alias scanExtract "bash ~/scripts/precursorScanExtractor/precursorScanExtractor.sh"
alias expandSupInfo "~/scripts/expandSupInfo/bin/expandSupInfo"
alias ls "ls --color=auto"
alias cimsetup "bash ~/scripts/cimage_scripts/cimage_setup.sh"
alias cimpbs "bash ~/scripts/cimage_scripts/generate_cim_pbs.sh"
alias purgeCimage "bash ~/scripts/cimage_scripts/purgeCimage.sh"
alias fdhl "foreach d (heavy/ light/)"
alias fshl "foreach s (heavy/ light/)"
alias cssetup "bash ~/scripts/cimage_scripts/sequet_cimage_setup.sh"
alias rpath "python /home/mauraisa/scripts/py_realpath/src/main.py"
alias qstatHist "python /home/mauraisa/scripts/qstatHist.py"
alias annotateMS2 "~/scripts/ms2_annotator/bin/annotate_ms2"
alias nlutil "python /home/mauraisa/scripts/py_nlutil/src/py_nlutil.py"
alias makeMS2 "Rscript ~/scripts/ms2_annotator/rscrips/main.R"
alias parseCimage "python ~/scripts/parseCimage/src/parseCimage.py"
alias qdel_all "python /home/mauraisa/local/qdel_all.py"
alias seqStat "python /home/mauraisa/local/seqStat.py"
alias getResidueNumbers "python /home/mauraisa/local/getResidueNumbers/src/getResidueNumbers.py"

alias DTarray "~/local/DTarray_pro-1.7.4/bin/DTarray"

```

- To save and exit the file, hit `^+ X`. A dialog should show up at the bottom which says:

```
Save modified buffer (ANSWERING "No" WILL DESTROY
CHANGES) ?
```

- Hit `y`
- Next a dialog should show up at the bottom which says

```
File name to write: .tcshrc
```

- Hit `enter` to exit.
- Finally you have to tell the computer to reload your shell profile after you have modified it with the command:

```
$ source .tcshrc
```

- You can test your alias by typing the following command from your home directory:

```
$ DTarray --version
```

- If the alias is recognized by the computer, it should display something like:

```
DTarray_pro 1.7  
Last git commit: Sat Sep 22 20:28:17 2018  
git revision: 04  
d30f4fd7790abfca60197d85cefc9d2a877cfc
```

3 Usage