21 May 2018

vsearch

`vsearch` is a program we'll use for three steps in the 16S rRNA analysis pipeline

- pair the reads in FASTQ files
- simplify the data set by removing duplicate sequences
- run a clustering algorithm to group similar sequences

In this document:

- instructions for installing `vsearch`
- example of how to use it to pair reads

Getting Organized

The first step is figuring out where you want to put the software

Suggestion:

- create a folder where you will put software you download
- examples:

  `~/Applications`

  `~/Applications/Bioinformatics`

  `~/Classes/Bi410/downloads`

Later in the installation process we'll see how to add the new download to your path

Create the folder, start a terminal session, cd to the folder

GitHub Page

The project home page is on GitHub:

```
https://github.com/torognes/vsearch
```

About GitHub

GitHub was created to allow software developers to share open source code

Software projects each have their own **repository**

- anyone who wants to contribute can “clone” a repository
- they make changes, upload it to their own repository, send a note to the original developer
- the project leader reviews the changes, and if they are accepted, the new code is merged with the existing project

URLs at GitHub include the developer’s user name and the project name

Navigating a GitHub Site

The first thing you'll see is a directory listing

- subdirectories with names like `man` (for "manual", for the program's man pages) and `src` (for "source", the source code for the project)
- open source license and other files

As end users we can skip all this — scroll down to the "readme" section

GitHub encourages a standard layout for this part of the page as well

- a project overview [read this]
- examples of how to use the program [skim for now]
- instructions for downloading and installing [normally read this, but see below]

Installation Instructions

GitHub was designed for developers who know how to download the sources and run a compiler to build the application

Many projects (especially bioinformatics applications) have **pre-compiled binaries** for end users

- these are what we want
- (usually) simple to download one file, save it somewhere on our system, ready to run

Binary Packages for `vsearch`

Find the section of the readme document that describes the "binary distribution" (somewhere under the general installation instructions)

Windows users: copy and paste the URL into your browser; it should download and unzip the file

Mac users: GitHub tells you to use a command named `wget`. Unfortunately, that's not a standard Unix command, you'll probably see "wget: command not found"

Instead, you can download the file from our server (type this on one line):

```
$ curl pages.uoregon.edu/conery/Bi410/
     vsearch-2.7.1-macos.tar > vsearch.tar
```

Then:

```
$ tar xvf vsearch.tar
```

Both Mac and Windows: you should now have a folder with a name that starts `vsearch-2.7.1`

If it's not already there move the new folder to the place you created previously (e.g. `~/Applications`)

HomeBrew [Optional for Mac Users]

If you want to set up your computer so you can download and compile other open source software install a popular package manager named HomeBrew

- it's like Anaconda, but it downloads source files

(1) Install Apple's developer tools: start a terminal window, type

```
$ xcode-select --install
```

(2) Install HomeBrew (`https://brew.sh`)

(3) Use HomeBrew to install `wget`:

```
$ brew install wget
```

(4) Type the two instructions shown on GitHub

- use `wget` to download the tar file
- type the `tar` command to "unzip" the package

Test the Download

If you look inside the folder you just downloaded you should see files named `LICENSE.txt` and `README.md` and a folder named `bin`

- bin stands for “binary”
- it’s a common name for a folder that contains compiled C and C++ programs

cd to the `bin` directory and type this command to make sure `vsearch` is installed and ready to use:

```
$ vsearch --help
```

If you see their friendly help message (instead of “command not found” or something similar) you’re ready to go.

Add `vsearch` to Your Path

The last step is to update your shell configuration so it puts `vsearch` in your path

- you could always cd to the `bin` directory each time you want to run `vsearch`
- but then the path to your input data and the folders where you want to put the output will be more complicated

Instead most people update their `.bash_profile` to include the directories of software they install

Add a line like this to the end of your `.bash_profile`:

```
PATH=[path to vsearch bin]:$PATH
```

Example: I put `vsearch` in `Classes/410` on my laptop, so I have

```
PATH=/Users/conery/Classes/410/vsearch-2.7.1-macos-x86_64/bin:$PATH
```

> *Note: your text editor might not see files with names that start with a period*
>
> *If you don't see .bash_profile rename it temporarily:*
>
> `$ mv .bash_profile bash_profile`
>
> *After you edit the file change the name back:*
>
> `$ mv bash_profile .bash_profile`

Start a New Terminal Window

The current terminal session doesn't know about the new directory in your path

- the changes won't take effect until the next time the shell reads `.bash_profile`

Either:

- quit the Terminal application and restart it
- open a new Terminal window (you can have multiple windows open at once)
- type this shell command while in your home directory:

```
$ source .bash_profile
```

Test Again

Now you should be able to type

```
$ vsearch –help
```

from your home directory

Optional: Link to `vsearch`

The instructions on the `vsearch` project page tell you to make a "symlink" to `vsearch`

Using a symlink is a little more complicated but worth the effort if you're going to be installing more software in the future

- you won't have to keep adding new folders to your path
- you can have multiple versions of the same application and easily switch back and forth between versions

Create a Common Folder for All Applications

Pick a location for your links. One idea is to make your own `bin` directory, *i.e.* in your home directory type

```
$ mkdir bin
```

Add The New Folder to Your Path

Instead of adding `vsearch` to your path put this line at the end of your `.bash_profile`:

```
PATH=~/bin:$PATH
```

Now the new directory (which is still empty) will be in your path the next time you start a terminal window.

Create the Symlink

Each time you add a new application to your system `cd` to your `bin` directory and type the `ln` ("link") command

The general format is

```
$ ln -s [path to application] .
```

The "`-s`" means "make a symbolic link"

Example: since I put `vsearch` in my Classes folder on my laptop I would type this:

```
$ ln -s ~/Classes/410/vsearch-2.7.1-macos-x86_64/bin/
vsearch .
```

Running `vsearch`

To run the program just type `vsearch`

Each time we run it, we need to supply a command line option that tells the program which operation to perform

- use additional arguments that depend on the operation

Example: use `--fastq_mergepairs` to tell it to pair up reads from two different FASTQ files:

```
$ vsearch --fastq_mergepairs A_R1.fastq
     --reverse A_R2.fastq --fastaout A.fasta
```

Another example: `--cluster_smallmem` tells `vsearch` to run a clustering algorithm; other options include the similarity cutoff (sequences more similar than this are put in the same cluster)

```
$ vsearch --cluster_smallmem clusters/seeds.fasta
      --uc clusters/clusters.txt --id 0.97
```