SCARF INSTALLATION

Ted Toal <twtoal@ucdavis.edu>

This describes how to install all elements needed to run the SCARF software. It assumes installation is on a Mac OSX system. For Linux or Windows systems, you can attempt installation using these instructions as a guide. Please take detailed notes and email them to me, especially if you are successful! I will incorporate them into these instructions.

If not done already, download SCARF files:

1. Browse to https://github.com/BradyLab/SCARF
2. At bottom of right column on screen, click "Download ZIP" and choose a place to put it on your computer.
3. Unzip the zip file on your computer.
4. Rename the unzipped folder from "SCARF-master" to just "SCARF".
5. If you are a GitHub user and wish to use the SCARF GitHub repository directly, instead of the above steps, you can fork the SCARF repository and work on your own copy. The "work" branch is where I do work. The "master" branch is where the releases reside, each tagged with a version number. You can create a branch whose contents are the same as a tagged version with the command:

   ```
   git checkout -b my_V1.0_branch v1.0"
   ```

You can then do the installation process using those files.

I will try to monitor for pull requests if you make useful changes.

To install SCARF:

1. **Consider whether the computer is capable enough**

   Before starting the installation, consider whether the computer on which the software is to be installed has enough memory for the problem at hand. I have been running SCARF using 16GB of RAM, and I'd recommend no less than that amount. For larger genomes, you may need more. My computer has a 2.4 GHz processor, which has been adequate.

2. **Work from the command line**

   Most work here is done from the command line, by opening the Terminal application. You should be familiar with some basic command line commands such as "cd", "ls", "cp", "rm", "mv", "sudo", "less" or "more", "man", and "mkdir". If you do not know how to use the command line or don't know these basic commands, take time to learn a bit. A suggested tutorial (on the long side, but spend as little or as much time as you need to feel you can get started here) from UC Davis is at:

   http://korflab.ucdavis.edu/Unix_and_Perl/current.html

3. **Know how to use a plain text editor and have one available**

You must have a plain text editor you know how to use. If nothing else, the Mac "TextEdit" program will work (use Plain Text format). The free open-source TextWrangler program is strongly recommended, and its $50 more capable version called BBEdit is totally worth the money for anyone who regularly edits text files. TextWrangler is available from the Apple App Store (Applications, App Store) or from:

http://www.barebones.com/products/textwrangler

4. **Copy allParameters.template file to new file allParameters.mytemplate**

   A variety of applications must be installed. Before installing them, it is helpful to prepare a text file for editing. There is a text file in the SCARF main directory that contains parameter settings for running SCARF: * allParameters.template is a template file containing sample settings for running SCARF on your genome data.

   The file includes settings of paths where applications have been installed. Although you may make many copies of the file for running SCARF using different genomes or parameters, all will have the same application paths in them. To make your own template file containing your own application paths, copy the file to a new filename, replacing ".template" with ".mytemplate". You will edit the new file and add application paths to it as you install applications. You can do the copying in Mac's Finder or from the command line:

   ```
   cd ~/Documents/SCARF    (or whatever is appropriate for your system)
   cp allParameters.template allParameters.mytemplate
   ```

5. **Open the allParameters.mytemplate file in a plain text editor**

   Open the new allParameters.mytemplate file created above in your plain text editor for editing. If you are in a hurry, you don't need to read anything in the file, but can simply search for "# ", which are comment lines marking items that may need to be changed. Each "#" comment says whether it must be changed, might need to be changed, or probably will never need to be changed, etc. Many of these items typically will never need to be changed, so the actual number of changes that need to be made is smaller than it might first appear. It is recommended that rather than hurrying, you take time to read through the file, as the comments explain the purpose of each parameter, and you will need to know this information, at least for key parameters, in order to run SCARF properly.

6. **Set WD to your SCARF directory in the allParameters.mytemplate file**

   Search for "# " in the allParameters.mytemplate file and locate the one that is followed by a line setting the value of the "WD" parameter. It looks like:

   ```
   WD := $(BRADYLAB)/Genomes/kmers/SCARF
   ```

   Change the assigned value to the path of your SCARF directory (where you unzipped the SCARF files). For example, maybe it would look like this:

   ```
   WD := /Users/johndoe/Documents/SCARF
   ```

7. **Install make and g++ (Xcode in Mac OSX)**

   SCARF makes use of a utility called "make", and also, some of the applications used by SCARF are distributed as source code that must be compiled and built into a runnable application on the

user's computer, which requires a C `compiler (g` utility). On Mac OSX, the Apple Developer Toolkit named Xcode provides these utilities, and it is available free from the Apple App Store (Applications, App Store). Xcode is a good thing to have installed anyway. If you don't have it installed already, run the App Store application, search for "Xcode", and double-click the "Install" button to install it. Installation takes quite a long time, during which it appears nothing is happening. When the installation is finished, you can verify that it was installed successfully by finding the Xcode application icon in Applications and running it. It may then display a box requesting your computer administrator password so it can install additional components. Then, close the Xcode application and go to the command line and enter the following command, which checks to see if the command line tools such as "make" and "g++" are installed, and if not, installs them:

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

To verify they are installed, you can enter this command:

```
g++
```

and you should see the error message "clang: error: no input files".

8. **Install Jellyfish and set its path**

Jellyfish is a free open-source bioinformatics application that searches FASTA sequence files for k-mers of a specified size and writes them to a file. SCARF uses Jellyfish to extract unique (occurring once) k-mers from the genome sequences being used. You can find the Jellyfish at:

```
http://www.genome.umd.edu/jellyfish.html
```

I chose the "latest source and binaries" link, then downloaded the .tar.gz file. I double-clicked this file in Finder, in the Downloads folder, and it unpacked to produce a jellyfish folder. I moved this folder to a directory I made named "src" under my user root directory:

```
cd ~
pwd
mkdir src
cp Downloads/jellyfish-2.2.3 src
```

This version of SCARF was tested with Jellyfish version 2.2.3. Newer versions should work as well.

Now the jellyfish program must be compiled and built into an application, and installed on your computer. I used these commands, which worked without error:

```
cd ~/src/jellyfish-2.2.3
./configure
make
sudo make install
```

The "sudo" command prompts for a password, and I entered my computer's administrator password. When the above commands are finished, I verified that Jellyfish was installed and that I could run it with these commands:

```
which jellyfish
jellyfish --version
```

Finally, the two .mytemplate files must have the path to Jellyfish included in them. Search the files for "# " and assign the path to Jellyfish, which was shown when you gave the "which jellyfish" command above, to the parameter "PATH_JELLYFISH". The path will probably already be correct because Jellyfish usually gets installed in a standard location.

```
PATH_JELLYFISH := /usr/local/bin/jellyfish
```

Also, set the value JELLYFISH_HASH_SIZE, which follows, to something that seems appropriate for your computer and its memory. Read the comments for each parameter to learn more about it. If you don't know how much memory your Mac computer has, choose Apple Icon, About This Mac, and look for "Memory". The value shown may work fine, but if you are working with k-mer sizes or genome sizes that produce lots more than 24 million k-mers, you may need to increase the size (and have sufficient computer memory).

```
JELLYFISH_HASH_SIZE := 80M
```

9. **Install Perl if necessary and set its path**

   Perl is a programming language used by SCARF. Using it requires a Perl interpreter application on your computer. The Mac OSX system comes with a Perl interpreter already installed, and this should be sufficient. This version of SCARF was tested with Perl version 5.16.0, although later versions, and earlier V5 versions, will probably be fine. You can find out if you have Perl installed, where it is located, and what its version is with this command:

   ```
   which perl
   perl --version
   ```

   If you do not have Perl installed, look for it here:

   ```
   https://www.perl.org/get.html
   ```

   After installing it, re-run the "which perl" command to find the path to it.

   The two .mytemplate files must have the path to Perl included in them. Assign the path, which was shown with the "which perl" command, to the parameter "PATH_PERL". For example, maybe your path will be:

   ```
   PATH_PERL := /usr/local/bin/perl
   ```

10. **Install R and set its path**

11. **Install Primer3 and set its path**

12. **Install ePCR and set its path**

13. **Build findMers**

14. **Test trashing and choose deletion method**

15. **Copy primer3settings.default.txt and edit Primer3 settings**

16. **Copy allParameters.test.template file to new file allParameters.test and add the changes made to allParameters.mytemplate**

File allParameters.test.mytemplate is more-or-less a copy of the allParameters.template file, modified for testing SCARF. Use one of two methods to modify it to incorporate the same changes you just made to allParameters.mytemplate:

Method 1 (straightforward)

Copy allParameters.test.template to allParameters.test, using either Finder or the command line:

```
cp allParameters.test.template allParameters.test
```

Edit the new allParameters.test file with your plain text editor and put **the same** parameter changes into it as you just finished doing with allParameters.mytemplate.

Method 2 (easiest and more fun)

There is another way to do this rather than using your text editor: merging the changes. There is a marvelous file comparison and merging tool called "FileMerge" that comes with Xcode. To run it, start Xcode, then on the menu choose Xcode, Open Developer Tool, FileMerge. When it opens up, you may want to find its icon on the dock and set it to stay put in the dock, then you can close Xcode and in the future get to it directly from the dock. When you run FileMerge, it prompts for two or three or four file names. To see an example of use, enter the first two, "left" and "right", setting "left" to allParameters.mytemplate and "right" to allParameters.test, then click "Compare". You will see a comparison of the two files, with the differences clearly shown. If you wanted to incorporate changes from one of these files into the other (we don’t), you can do this easily by using the up/down arrow keys to go through the differences one by one, and use the left/right arrow keys to select whether you want the left or right side file text in the output, and you can also click in the box on the bottom that shows the merged text and edit it; when finished you can save the merged text to a new file or overwrite one of the two compared files, using File, Save Merge. However, FileMerge actually makes the job at hand easier than that. Do another File, Compare Files with FileMerge, and set the four file text boxes to the following files:

1. Left: allParameters.test.template file path

2. Right: allParameters.mytemplate file path

3. Ancestor: allParameters.template file path

4. Merge: allParameters.test file path

Now when you click "Compare", the majority of the arrows in the center of the FileMerge screen will point left, indicating that the essential changes in allParameters.test.template will be retained. However, every line you changed in allParameters.mytemplate should have an arrow pointing to the right, to your changed line (because the Right file differs from the Ancestor file), indicating that your changes will be incorporated. Go through to make sure all your changes have right arrows, and all the other differences have arrows pointing to the left. Then choose File, Save Merge, and the allParameters.test file will be overwritten with a new version containing your new parameter settings (application paths, mainly). Load allParameters.test into your text editor and quickly browse it to make sure it looks correct.

17. **Run SCARF using the test parameters in allParameters.test and check for success**

That completes the installation of SCARF.

To run SCARF to generate markers after installation:

- Find file RUN.pdf or RUN.html in the SCARF folder on your computer and open either one and follow the instructions.