University of Glasgow

User Manual

ScaleHD: Automated HD microsatellite genotyping

alastair.maxwell@glasgow.ac.uk

May 17, 2016

Contents

1 Requirements

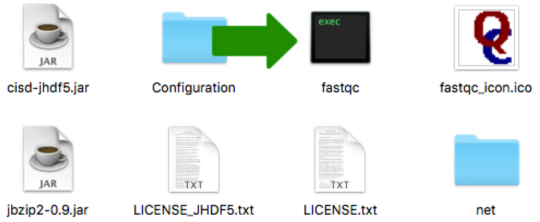
Hello! Here's how to install/use the ScaleHD pipeline. ScaleHD provides the ability to trim your sequence reads, align them, and genotype (if you are working on Huntington Disease – other diseases are planned to be supported in the future). If you're looking for demultiplexing, you want to use Machete before using ScaleHD. Machete is available at `https://github.com/helloabunai/Machete`.
In order for ScaleHD to function, third-party binaries/programs are required on your system. This will guide you through how to install each of the required packages, before explaining how to install and use ScaleHD. In this manual, I will assume you are using a OS X based computer, with at least version 10.9 installed, and are at least mildly comfortable with the command line. Let's go!

1.1 FastQC

FastQC is utilised in the Quality Control stage of ScaleHD, and produces quality reports similar to those which you will be used to seeing generated in CLC Genomics. If you select the Quality Control stage, FastQC will be run automatically on every file you wish ScaleHD to process. First, let's download FastQC. The package is hosted by it's developer *Babraham Bioinformatics*. It is located at `http://www.bioinformatics.babraham.ac.uk/projects/download.html#fastqc`. On this page, you want to download the package found under the title:

FastQC 0.11.5 (Win/Linux zip file).

The version number may change, but yes, even if you are currently working on a Mac, you want the Win/Linux zip file. Once downloaded, extract the zip. Within the zip, locate the file fastqc:



This is the FastQC binary which will allow us to analyse our data's quality. In order for ScaleHD to interface with this binary, it needs to be added to something called our *system path*. This is a user-created 'index' which tells your system where non-system binaries are, and where to run them from. Basically, it will allow us to run FastQC from anywhere in our command-line environment – very desirable for

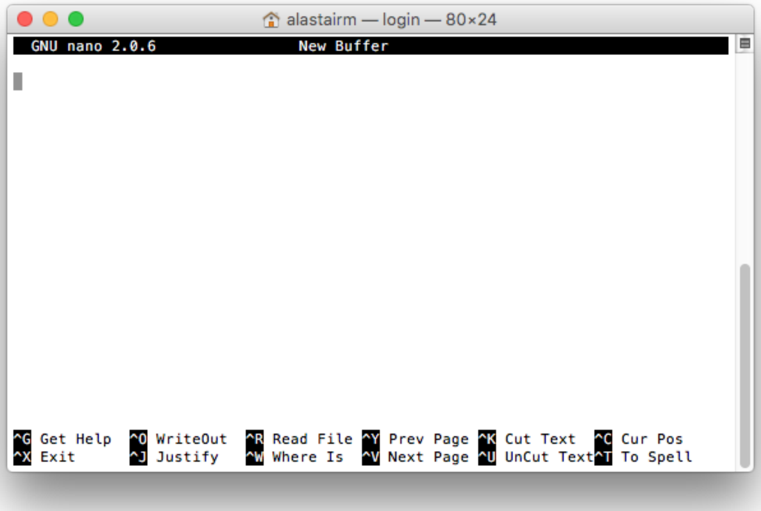
ScaleHD! In order to add FastQC to our system path, two things must be carried out:

- Creating a *Builds* directory

In order to keep things tidy and user-friendly, it is recommended to create a folder for all third party binaries to be located within. Personally, I place my built binaries within /Users/<username>/Documents/Builds/, but you are free to choose your own folder. However, if you are uncomfortable with command line installation instructions, it might be easier to follow this guide verbatim. From this point onwards, I will assume you have made a folder in ˜/Documents/Builds. Now, copy the *entire* extracted FastQC folder (binary and all other files) we located earlier, into ˜/Documents/Builds/FastQC/.

- Adding FastQC to your system path

This will allow your system to be able to identify the FastQC binary from anywhere in the 'environment', i.e. regardless of where you are working from, it has the location of the specified binary. In order to do this, we need to make a text file. However, the standard text editor on OS X doesn't allow for plain file editing, and usually forces you into a Rich Text Format, which we don't want. So, we will use a command line text editor (don't worry, it is simple).
First, open the terminal. If you can't find it, press Command + Space to bring up Spotlight Search, and type 'terminal'. Once in the app, you should see the command line. Type nano .bash_profile and hit enter:

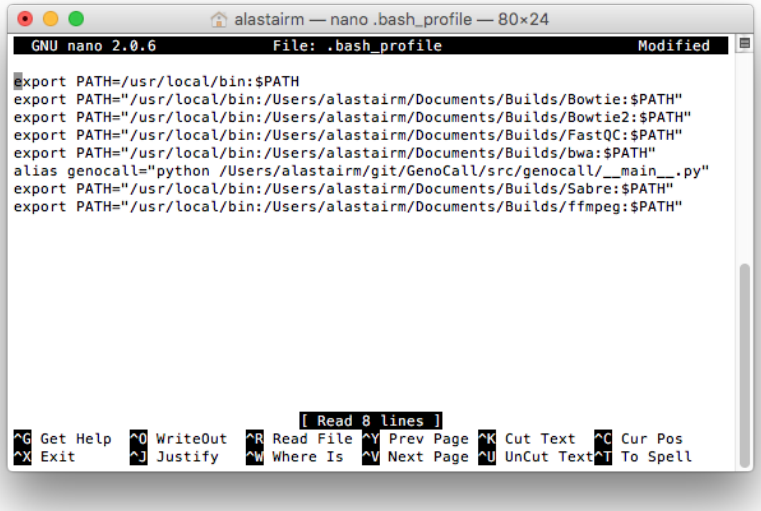


```
  GNU nano 2.0.6                  New Buffer



^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

from here, add the text:

- export PATH=/usr/local/bin:$PATH
- export PATH=/usr/local/bin:/Path/To/Your/Folder:$PATH"

replacing /path/to/your/folder with the actual location of FastQC (in our example, /Users/my_username/Documents/Builds/FastQC. An example .bash_profile file can be seen below. After you are done, press the button pair Control + X. At the bottom of the terminal, a message will appear asking if you wish to "save modified buffer", to which you would press Y. Then, nano will ask you what you wish to call the file. We need the file to be called .bash_profile. Type that in (if it is not already present), and hit enter to save. You're done!



```
  GNU nano 2.0.6              File: .bash_profile                     Modified

export PATH=/usr/local/bin:$PATH
export PATH="/usr/local/bin:/Users/alastairm/Documents/Builds/Bowtie:$PATH"
export PATH="/usr/local/bin:/Users/alastairm/Documents/Builds/Bowtie2:$PATH"
export PATH="/usr/local/bin:/Users/alastairm/Documents/Builds/FastQC:$PATH"
export PATH="/usr/local/bin:/Users/alastairm/Documents/Builds/bwa:$PATH"
alias genocall="python /Users/alastairm/git/GenoCall/src/genocall/__main__.py"
export PATH="/usr/local/bin:/Users/alastairm/Documents/Builds/Sabre:$PATH"
export PATH="/usr/local/bin:/Users/alastairm/Documents/Builds/ffmpeg:$PATH"









                                [ Read 8 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

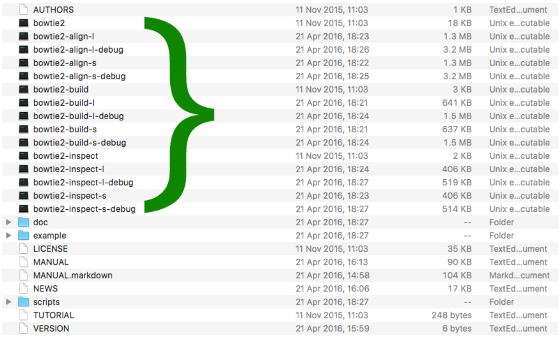
FastQC is now "installed" and will be visible to ScaleHD when required. In order to double check your success, close your current terminal, open a new terminal, and type which fastqc. A response should tell you where the system thinks your FastQC binary is located!

1.2 Cutadapt

Cutadapt is also utilised in the Quality Control stage of ScaleHD, and is used for trimming sequences. The easiest way to install Cutadapt is via PIP. If you do not have PIP, we will install that first. In a terminal, type the commands "sudo easy_install -U setuptools" and "sudo easy_install pip". This will ask you for your password, but for security reasons it will not display anything as you type. This is normal! Once PIP is installed, cutadapt can be installed by typing the command "pip install –upgrade cutadapt" (note: –upgrade is two dashes, not one). This simple program takes care of all system path linking for you, so the same stages as was required for FastQC are not required for Cutadapt!

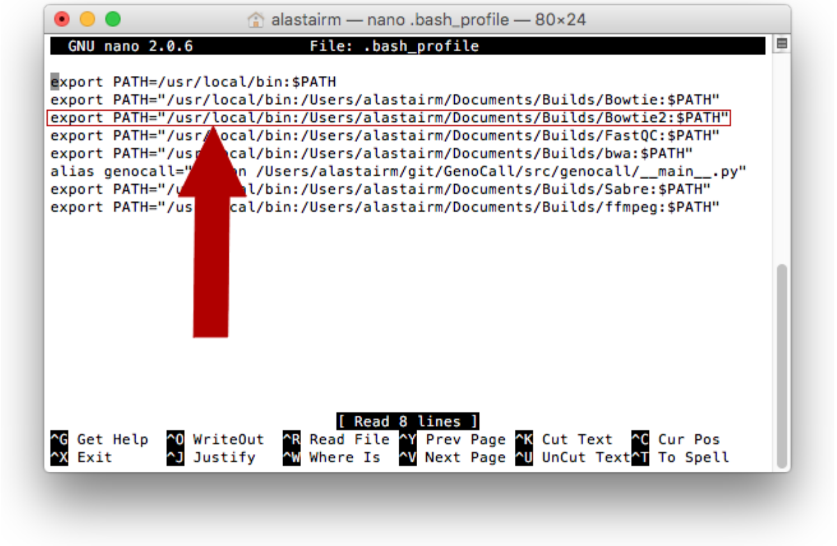
1.3 Bowtie2

Bowtie2 is the aligner which is used within ScaleHD to align sequence reads for further analysis. This might change in future versions of ScaleHD, but for now we will stick with Bowtie2. First, head to `https://sourceforge.net/projects/bowtie-bio/files/bowtie2/2.2.9/` and download the bowtie2-2.2.9-macos-x86_64.zip. As with all software, the current version number within this manual may be out of date by the time you are reading this. Once downloaded, you will be presented with the following files:



AUTHORS	11 Nov 2015, 11:03	1 KB	TextEd...ument
bowtie2	11 Nov 2015, 11:03	18 KB	Unix e...cutable
bowtie2-align-l	21 Apr 2016, 18:23	1.3 MB	Unix e...cutable
bowtie2-align-l-debug	21 Apr 2016, 18:26	3.2 MB	Unix e...cutable
bowtie2-align-s	21 Apr 2016, 18:22	1.3 MB	Unix e...cutable
bowtie2-align-s-debug	21 Apr 2016, 18:25	3.2 MB	Unix e...cutable
bowtie2-build	11 Nov 2015, 11:03	3 KB	Unix e...cutable
bowtie2-build-l	21 Apr 2016, 18:21	641 KB	Unix e...cutable
bowtie2-build-l-debug	21 Apr 2016, 18:24	1.5 MB	Unix e...cutable
bowtie2-build-s	21 Apr 2016, 18:21	637 KB	Unix e...cutable
bowtie2-build-s-debug	21 Apr 2016, 18:24	1.5 MB	Unix e...cutable
bowtie2-inspect	11 Nov 2015, 11:03	2 KB	Unix e...cutable
bowtie2-inspect-l	21 Apr 2016, 18:24	406 KB	Unix e...cutable
bowtie2-inspect-l-debug	21 Apr 2016, 18:27	519 KB	Unix e...cutable
bowtie2-inspect-s	21 Apr 2016, 18:23	406 KB	Unix e...cutable
bowtie2-inspect-s-debug	21 Apr 2016, 18:27	514 KB	Unix e...cutable
doc	21 Apr 2016, 18:27	--	Folder
example	21 Apr 2016, 18:27	--	Folder
LICENSE	11 Nov 2015, 11:03	35 KB	TextEd...ument
MANUAL	21 Apr 2016, 16:13	90 KB	TextEd...ument
MANUAL.markdown	21 Apr 2016, 14:58	104 KB	Markd...cument
NEWS	21 Apr 2016, 16:06	17 KB	TextEd...ument
scripts	21 Apr 2016, 18:27	--	Folder
TUTORIAL	11 Nov 2015, 11:03	248 bytes	TextEd...ument
VERSION	21 Apr 2016, 15:59	6 bytes	TextEd...ument

From here, we want to take all the binaries (bowtie2, bowtie2-align... etc) and move them to another folder, this time in ˜/Documents/Builds/Bowtie2. Once the binaries are all located within this folder, we need to edit our .bash_profile file. Assuming that you have copied all the required binaries into the aforementioned Bowtie2 folder in your Builds directory, we will need to add that Bowtie2 folder to our system path. The procedure for this is the same as in previous stages, but with a slightly different variable. In a terminal, open our bash profile with nano .bash_profile. Add the required information on a new line, following the latest entry in your file. Again, save the file by pressing CTRL+X, then Y, and then Enter.



```
  GNU nano 2.0.6              File: .bash_profile

export PATH=/usr/local/bin:$PATH
export PATH="/usr/local/bin:/Users/alastairm/Documents/Builds/Bowtie:$PATH"
export PATH="/usr/local/bin:/Users/alastairm/Documents/Builds/Bowtie2:$PATH"
export PATH="/usr/ ocal/bin:/Users/alastairm/Documents/Builds/FastQC:$PATH"
export PATH="/us    cal/bin:/Users/alastairm/Documents/Builds/bwa:$PATH"
alias genocall="     n /Users/alastairm/git/GenoCall/src/genocall/__main__.py"
export PATH="/u       l/bin:/Users/alastairm/Documents/Builds/Sabre:$PATH"
export PATH="/us    cal/bin:/Users/alastairm/Documents/Builds/ffmpeg:$PATH"

                                  [ Read 8 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

With this, bowtie is now installed on our path and can be used by ScaleHD when required.

1.4 Samtools

1.4.1 Xcode developer tools

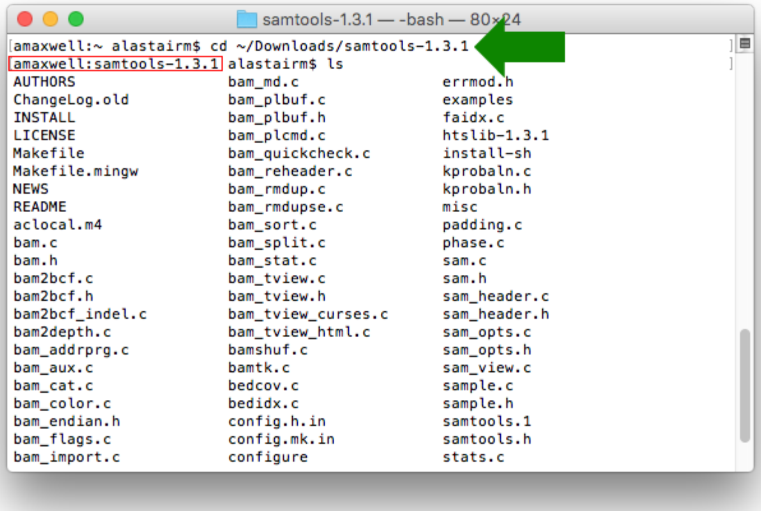
Before we begin installing samtools, you may need to follow this stage first. For whatever nonsensical reason, Apple do not include a C compiler with OS X by default. In order to check if you have a compiler installed already, open a terminal and type the command WHICH GCC. This should output a path, telling you where gcc is installed, if at all. If the output is blank, we must install XCode Command Line Tools in order to acquire a C compiler – required to compile samtools.
Open a terminal, and type the command: XCODE-SELECT –INSTALL (note: – install has two dashes). This will produce a pop-up, with the options: *Get Xcode, Now Now* and *Install*. Press *Install*. This will download and install the required tools, without any further input from the end-user. Once this is complete, you can check that the required compiler has been installed, by typing WHICH GCC. This command should notify you of a path that GCC has been installed to.

1.4.2 Installing Samtools

The final package required for manual installation in samtools. This is used in the Alignment/Genotyping stages, and extracts information about the sequence assembly created by aligning input reads via Bowtie2.

Samtools requires us to build the binary executable from the source, which we will need to download. To do so, head on over to the Samtools sourceforge page at `https://sourceforge.net/projects/samtools/files/samtools/`, and grab the latest version (1.3.1 at the time of writing). Within this folder, will be the source code for Samtools. We will again require a terminal in order to carry out this installation process.

Extract the downloaded zip file, and open a terminal session. We need to change the current directory that this terminal is 'within', which we do with the command cd. For example, if my extracted folder was in ˜/Downloads/samtools-1.3.1/, i would type cd ˜/Downloads/samtools-1.3.1 (highlighted by a green arrow, in the following figure). You can check you are within the correct directory by looking at the left of the terminal line – it should say your username followed by the current directory you are 'within' (highlighted in red, in the following figure). Another way to ensure you are in the right place is by typing the command ls, which lists all files in the current working directory, and will look like this:

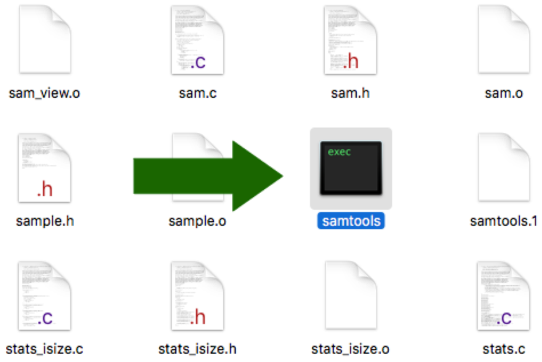


```
amaxwell:~ alastairm$ cd ~/Downloads/samtools-1.3.1
amaxwell:samtools-1.3.1 alastairm$ ls
AUTHORS                 bam_md.c                errmod.h
ChangeLog.old           bam_plbuf.c             examples
INSTALL                 bam_plbuf.h             faidx.c
LICENSE                 bam_plcmd.c             htslib-1.3.1
Makefile                bam_quickcheck.c        install-sh
Makefile.mingw          bam_reheader.c          kprobaln.c
NEWS                    bam_rmdup.c             kprobaln.h
README                  bam_rmdupse.c           misc
aclocal.m4              bam_sort.c              padding.c
bam.c                   bam_split.c             phase.c
bam.h                   bam_stat.c              sam.c
bam2bcf.c               bam_tview.c             sam.h
bam2bcf.h               bam_tview.h             sam_header.c
bam2bcf_indel.c         bam_tview_curses.c      sam_header.h
bam2depth.c             bam_tview_html.c        sam_opts.c
bam_addrprg.c           bamshuf.c               sam_opts.h
bam_aux.c               bamtk.c                 sam_view.c
bam_cat.c               bedcov.c                sample.c
bam_color.c             bedidx.c                sample.h
bam_endian.h            config.h.in             samtools.1
bam_flags.c             config.mk.in            samtools.h
bam_import.c            configure               stats.c
```

Now that we're in the right place, we can compile the application. While in the extracted samtools folder, enter the following three commands into your terminal, separately:

```
./configure
make
make install
```

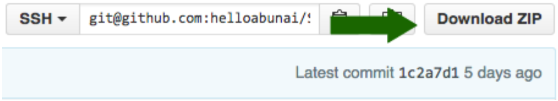
Each command will output various things to your terminal, until complete. Once the final command has finished, a SAMTOOLS binary file will be located within your extracted folder. **NOTE**: the command MAKE INSTALL may complain about not having permission to run (Error 71). In order to fix this, run the command SUDO MAKE INSTALL instead, entering your admin password in order to execute.



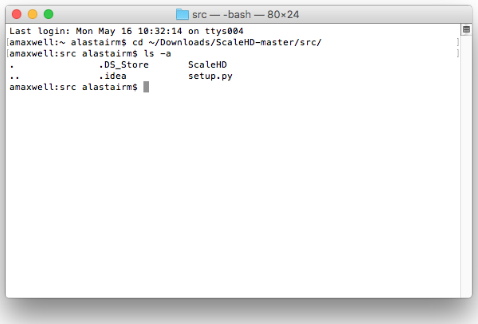
Due to us manually compiling the software ourselves, the samtools binary by default is installed to our system path automatically – to ensure this is the case, type WHICH SAMTOOLS into a terminal. As a result of this, we don't need to copy these files anywhere, or edit our .bash_profile file. This means that all the thid party required packages for ScaleHD are now installed on your system, and we can install the actual pipeline!

1.5 Installing ScaleHD

First, we need to download the source for ScaleHD from GitHub, a website specialising in online software version control. The package is located at: `https://github.com/helloabunai/ScaleHD`. On this webpage, we want to press the following button:


SSH ▾
git@github.com:helloabunai/!
Download ZIP
Latest commit 1c2a7d1 5 days ago

Once downloaded, extract the zip folder. Open a terminal, and navigate to where you downloaded the folder. Your current working directory in this terminal should be something like: ~/Downloads/ScaleHD-master/src/. In this folder, should be a file called SETUP.PY and a folder called ScaleHD:

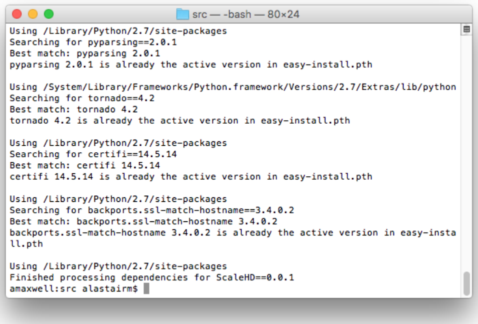


```
Last login: Mon May 16 10:32:14 on ttys004
amaxwell:~ alastairm$ cd ~/Downloads/ScaleHD-master/src/
amaxwell:src alastairm$ ls -a
.               .DS_Store       ScaleHD
..              .idea           setup.py
amaxwell:src alastairm$ 
```

Now we can install ScaleHD. Before running the setup.py script, we need to ensure setuptools is up-to-date on your system. Here, I assume you are running on OS X. We previously ran these commands when installing Cutadapt, but they are specified here again for posterity: "sudo easy_install -U setuptools" and "sudo easy_install pip". This installs a python package manager for automatically installing software from the Pypi database.

However, one package does not install properly on OSX when downloaded via setuptools; we need to do it manually. Type the command: pip install matplotlib.

Now that setuptools is up-to-date, we can run the command python setup.py install, to install ScaleHD. If you get error messages about permissions from this command, you need to run it with sudo, granting the terminal session administrator rights. This process will install all python package dependencies that ScaleHD requires, automatically; this procedure will produce a large amount of compiling output to the terminal, until complete:

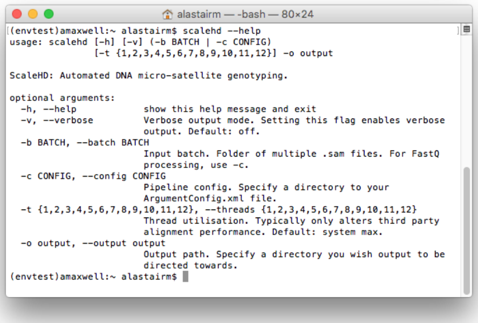


```
Using /Library/Python/2.7/site-packages
Searching for pyparsing==2.0.1
Best match: pyparsing 2.0.1
pyparsing 2.0.1 is already the active version in easy-install.pth

Using /System/Library/Frameworks/Python.framework/Versions/2.7/Extras/lib/python
Searching for tornado==4.2
Best match: tornado 4.2
tornado 4.2 is already the active version in easy-install.pth

Using /Library/Python/2.7/site-packages
Searching for certifi==14.5.14
Best match: certifi 14.5.14
certifi 14.5.14 is already the active version in easy-install.pth

Using /Library/Python/2.7/site-packages
Searching for backports.ssl-match-hostname==3.4.0.2
Best match: backports.ssl-match-hostname 3.4.0.2
backports.ssl-match-hostname 3.4.0.2 is already the active version in easy-insta
ll.pth

Using /Library/Python/2.7/site-packages
Finished processing dependencies for ScaleHD==0.0.1
amaxwell:src alastairm$ 
```

We're done. Now we can use the software.

2 Usage

Now that we're installed, this section of the manual will describe how to use ScaleHD and it's various functions. ScaleHD is ran via the command line, by typing the command: SCALEHD. This is the root of the command, which has various 'flags' or 'options', allowing the user to tailor their functionality. To see all available options, type: SCALEHD –HELP.



```
(envtest)amaxwell:~ alastairm$ scalehd --help
usage: scalehd [-h] [-v] (-b BATCH | -c CONFIG)
               [-t {1,2,3,4,5,6,7,8,9,10,11,12}] -o output

ScaleHD: Automated DNA micro-satellite genotyping.

optional arguments:
  -h, --help            show this help message and exit
  -v, --verbose         Verbose output mode. Setting this flag enables verbose
                        output. Default: off.
  -b BATCH, --batch BATCH
                        Input batch. Folder of multiple .sam files. For FastQ
                        processing, use -c.
  -c CONFIG, --config CONFIG
                        Pipeline config. Specify a directory to your
                        ArgumentConfig.xml file.
  -t {1,2,3,4,5,6,7,8,9,10,11,12}, --threads {1,2,3,4,5,6,7,8,9,10,11,12}
                        Thread utilisation. Typically only alters third party
                        alignment performance. Default: system max.
  -o output, --output output
                        Output path. Specify a directory you wish output to be
                        directed towards.
(envtest)amaxwell:~ alastairm$ 
```

In detail, the options are:

-h Displays the help message.

-v A flag to enable verbose terminal output. Without this flag, the command line will not show any messages until all processing is complete.

-b One of two input modes. This mode takes in a folder of sequence assemblies in .SAM format, and only genotypes them.

-c The second of two input modes. This mode takes in a XML configuration file, which has further options. This is the 'full pipeline' mode – these options are described in a further section of this manual.

-t An optional flag for specifying the number of processor threads to use. By default, it uses the maximum available on the system. Mainly affects sequence alignment performance.

-o Pipeline output. Specifies where you wish all output for this run to be directed towards.

An example command for each of the two input modes would look something like the following:

```
scalehd -v -b ~/Path/To/SAMData/ -o ~/Output/Path/
scalehd -v -c ~/Path/To/XMLConfig.xml -o ~/Output/Path/
```

2.1 XML Configuration

This subsection will explain how the XML input works for the full pipeline (-c) mode. An example XML input file can be found at `https://github.com/helloabunai/ScaleHD/blob/master/src/ScaleHD/config/ArgumentConfig.xml`. For the sake of simplicity, it will be easier for you to download the example XML file and edit it to your liking. Now let's look at the first section, the CONFIG section.

2.1.1 General configuration

Listing 1: ArgumentConfig.XML Example

```xml
<config data_dir="/PathToData/" forward_reference="../forward.fa" reverse_reference="../reverse.fa">
...
</config>
```

Here, we specify our references and data. The element DATA_DIR should be changed to the system path to your input data folder. Your input folder **MUST** have files which end in _R1.FASTQ and _R2.FASTQ (or .fq/.fq.gz). There must also be an equal number of R1 (forward) and R2 (reverse) files; ScaleHD takes every 2 files (alphabetically; R1+R2) and assumes a sample pair – name your files appropriately!

Next, the elements FORWARD/REVERSE_REFERENCE point to a reference file, that must be in the fasta/fa/fas format.

2.1.2 Instance Flags

The next element within the XML file is the INSTANCE_FLAGS element.

Listing 2: ArgumentConfig.XML Example

```xml
<instance_flags quality_control="True" sequence_alignment="False" genotype_prediction="False"/>
```

This is straight-forward; specifiy True or False for the stages you wish ScaleHD to run on your input data. However, there are stipulations about the order of flags you specify. Since the data in this mode of ScaleHD is unaligned FastQ files, we cannot only specify GENOTYPE_PREDICTION as true; SEQUENCE_ALIGNMENT must also be true if you wish to genotype.

2.1.3 Trim Flags

Next, is TRIM_FLAGS.

Listing 3: ArgumentConfig.XML Example

```
<trim_flags trim_data="True" trim_type="Adapter" quality_threshold
    ="5" adapter_flag="-a" adapter="sequence_here"/>
```

TRIM_DATA enables trimming of FastQ reads if set to True. The stage is skipped if set to False. All other flags are input for Cutadapt. Trim type specifies which type of trimming you would like to execute: Adapter, Quality, or Both – No other keywords are considered valid. If you specified Quality, then an integer value is required in the QUALITY_THRESHOLD parameter. This dictates the minimum quality cutoff that reads which fall below will be trimmed.

If you specified Adapter, then both ADAPTER and ADAPTER_FLAG must be filled out. ADAPTER is the sequence you wish Cutadapt to look for, and remove. The Adapter flag relates to the position of the adapter, as described here: `http://cutadapt.readthedocs.io/en/stable/guide.html#removing-adapters`. Valid options for ADAPTER_FLAG are: -a, -g, -a$, -g^ and -b.

2.1.4 Alignment Flags

Here we look at the input for alignment flags, the section of the XML document that acts as an interface for Bowtie2 input.

Listing 4: ArgumentConfig.XML Example

```
<alignment_flags extension_threshold="15" seed_size="2"
    align_mismatch="0" substr_length="22" substr_interval_start="1"
     substr_interval_end="1.15" read_gap_open="" read_gap_extend=""
     ref_gap_open="" ref_gap_extend="" min_mismatch_pen=""
    max_mismatch_pen=""/>
```

Below is a table which equates the XML input to it's original Bowtie2 argument.

-D Extension_threshold: Give up extending the alignment after <int>failed extends in a row.

-R Seed_size: For reads with repetitive seeds, try <int>sets of seeds.

-N Align_mismatch: Max number of mismatches in seed alignment (0 or 1).

-L Substr_length: Length of seed substring (3 <x <32).

-i Substr_interval_start/end: Interval between seed substrings with relation to read length.

–rdg Read_gap_open/extend: Gap opening/extension penalty score (current read).

–rfg Ref_gap_open/extend: Gap opening/extension penalty score (reference).

–mp Min/Max_mismatch_pen: Mismatch score.

2.1.5 Prediction Flags

Finally, the flags to tailor the genotyping stage of ScaleHD. As this is a work in progress, altering these flags does not offer much in the way of functionality, and are subject to change:

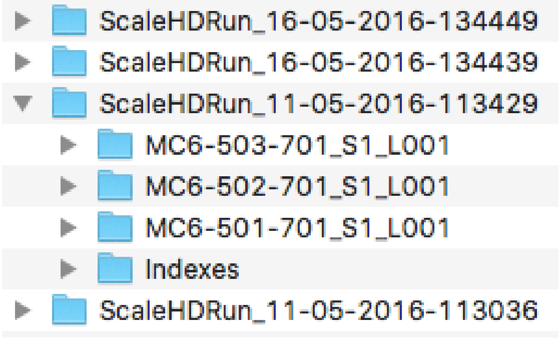
Listing 5: ArgumentConfig.XML Example

```
<prediction_flags probablity_estimate="True" max_iteration="-1"/>
```

Now that you've appropriately filled out your XML document, you can pass it to ScaleHD with the -c flag. Any errors in your XML document will be detected by the application; ScaleHD will not proceed until all input is error free. Assuming all files were named properly, and your xml document is error free, ScaleHD should process without errors.

3 Output

This section will discuss the folder heirarchy of ScaleHD's output. In the directory you specified with the -o flag in the command line, will be a folder for each 'run' of ScaleHD. Within each run's folder, will be a folder for every sample pair that ScaleHD processed:



```
▶ ScaleHDRun_16-05-2016-134449
▶ ScaleHDRun_16-05-2016-134439
▼ ScaleHDRun_11-05-2016-113429
    ▶ MC6-503-701_S1_L001
    ▶ MC6-502-701_S1_L001
    ▶ MC6-501-701_S1_L001
    ▶ Indexes
▶ ScaleHDRun_11-05-2016-113036
```

For each run, your reference files are indexed and stored under the Indexes subfolder. Depending on which stages you have selected ScaleHD to run, a folder for that stage will be created in the sample-pair subfolder. For example, if you chose to run all three stages of Quality Control, Sequence Alignment and Genotype Prediction, you would find SeqQC, Align and Predict folders within.

Within the SeqQC folder, if present, is the output of sequence trimming – your raw input reads are untouched, but trimmed versions of those reads are saved within this folder. Also present is the output of FastQC analysis on your input data, providing a visual report on sequence quality, similar to that of which you are used to generating with CLC Genomics.

Within the Align folder, if present, is the output of sequence alignment – A folder for each forward/reverse read file within the current sample pair contains the sam/bam alignment output, an alignment performance metrics report and generic alignment report from Bowtie2, and a samtools-scraped reference repeat distribution CSV file.

Within the Predict folder, if present, is the output of genotype prediction – as this is still a work in progress, output is not finalised for this stage and can be safely disregarded.

A summary for each sample pair is added to a "master" run-summary, which is located in the run's output folder. Basic reports are included here, with the more in-depth reporting remaining within each sample-pair's specific subfolder. When more polish is added to ScaleHD, distribution graphs and other typical manual tasks the DGM group carries out for analysis will be automated and included in the output process.

With that, hopefully you can get started with ScaleHD without too many issues.

4 Contact

I can be reached at alastair.maxwell@glasgow.ac.uk.