make assignment

Xoel Mato Blanco

Using the make utility, create a makefile that generates index references for BWA mapper. Your makefile should take references in fasta format from a references folder. For each reference (ref_x.fasta or ref_x.fa) you should generate the corresponding ref_x.pac, ref_x.amb, ref_x.ann, ref_xbtw and ref_x.sa files. All these files should be in a folder named index/ref_x .

See http://bio-bwa.sourceforge.net/ for details about BWA index generation.

Add batch processing support in your previous makefile (i.e. generation of indexes should be done by submitting jobs to SLURM).

Your solution should include a brief report describing your solution as well as the corresponding make and submit files . All this information should be compressed into a single file named MAKE-.tar

Don't forget to add also your name at the brief report.

Structure

The workflow idea of my script starts defining the directories we're working with. Using wildcard calls we get the lists of fasta and fa files present in the references directories. We'll also need the basenames (i.e. no extension), so we use patern substitution on the previous lists. Then we establish which directories we'll use combining the index directory with the basenames for fasta and fa files. Then we combine the fasta and fa lists to have a single one for each type of variable. Then, using the names, we can create the output names using add sufix.
After defining all and clean as specific rules of this file for make to work, we start writing the rules.
First, all will check if the output files are present and will try to create them.The, according to their extension, the rule %.ann will be called. This rule will check for the expected index directories to exist. If it's not the case, it will run the rule $(TARGETDIRS) to create the directories. After that, the bwa index called is called. The input file will depend on the extension of the file we're dealing with. To do that, two if clauses were used. Finally, the clean rule is called to remove the indexing files and folders. # Non-sbatch version:

```
#!/bin/bash
# Defining some directories as variables:
INPUTDIR=./references/
INDEXDIR=./index/

# Getting existing files:
INPUTS_FA=$(wildcard $(INPUTDIR)*.fa)
INPUTS_FASTA=$(wildcard $(INPUTDIR)*.fasta)

# Extracting file base names:
NAMES_FA=$(patsubst $(INPUTDIR)%.fa, %, $(INPUTS_FA))
NAMES_FASTA=$(patsubst $(INPUTDIR)%.fasta, %, $(INPUTS_FASTA))

# Establishing final directories:
TARGETDIRS_FA=$(patsubst $(INPUTDIR)%.fa, $(INDEXDIR)%/, $(INPUTS_FA))
TARGETDIRS_FASTA=$(patsubst $(INPUTDIR)%.fasta, $(INDEXDIR)%/, $(INPUTS_FASTA))

# Combining both fasta and fa variables:
```

```
NAMES=$(NAMES_FA) $(NAMES_FASTA)
TARGETDIRS=$(TARGETDIRS_FA) $(TARGETDIRS_FASTA)

# Defining the expected output files:
TARGETNAMES=$(addsuffix .ann, $(NAMES))

# Exluding some keywords for make to work properly:
.PHONY: all clean

# Main target. Depends on TARGETNAMES, previously created as a list of the expected final
# files. The extension will then run the %.ann target.
all: $(TARGETNAMES)
    @echo $(TARGETNAMES)

# This target allows us to create the needed directories.
$(TARGETDIRS):
    @mkdir -p $@

# With this target we run the bwa index command to the indexing files. It depends on the
# index directories, which will run $(TARGETDIRS) rule:
%.ann: $(INDEXDIR)%/
    @echo ""
    @echo Indexing file: $*
    @echo ""
    # An if condition will check wheter we are indexing a .fasta or .fa file:
    # BWA calls were combined with -p to determine the output directory using make variables:
    @if test -f $(INPUTDIR)$*.fa; then bwa index -p $(INDEXDIR)$*/$* $(INPUTDIR)$*.fa; fi
    @if test -f $(INPUTDIR)$*.fasta; then bwa index -p $(INDEXDIR)$*/$* $(INPUTDIR)$*.fasta; fi

# Clean will delete all indexing files and folders.
clean:
    -rm -r index
```

sbatch version:

```
#!/bin/bash

# Defining some directories as variables:
INPUTDIR=./references/
INDEXDIR=./index/

# Getting existing files:
INPUTS_FA=$(wildcard $(INPUTDIR)*.fa)
INPUTS_FASTA=$(wildcard $(INPUTDIR)*.fasta)

# Extracting file base names:
NAMES_FA=$(patsubst $(INPUTDIR)%.fa, %, $(INPUTS_FA))
NAMES_FASTA=$(patsubst $(INPUTDIR)%.fasta, %, $(INPUTS_FASTA))

# Establishing final directories:
TARGETDIRS_FA=$(patsubst $(INPUTDIR)%.fa, $(INDEXDIR)%/, $(INPUTS_FA))
TARGETDIRS_FASTA=$(patsubst $(INPUTDIR)%.fasta, $(INDEXDIR)%/, $(INPUTS_FASTA))
```

```
# Combining both fasta and fa variables:
NAMES=$(NAMES_FA) $(NAMES_FASTA)
TARGETDIRS=$(TARGETDIRS_FA) $(TARGETDIRS_FASTA)

# Defining the expected output files:
TARGETNAMES=$(addsuffix .ann, $(NAMES))

# Exluding some keywords for make to work properly:
.PHONY: all clean

# Main target. Depends on TARGETNAMES, previously created as a list of the expected final
# files. The extension will then run the %.ann target.
all: $(TARGETNAMES)
    @echo $(TARGETNAMES)

# This target allows us to create the needed directories.
$(TARGETDIRS):
    @mkdir -p $@

# With this target we run the bwa index command to the indexing files. It depends on the
# index directories, which will run $(TARGETDIRS) rule:
%.ann: $(INDEXDIR)%/
    @echo ""
    @echo Indexing file: $*
    @echo ""
    # An if condition will check wheter we are indexing a .fasta or .fa file:
    # A sbatch job was submitted using commands for the required files and creating an output per submis
    # BWA calls were combined with -p to determine the output directory using make variables:
    @if test -f $(INPUTDIR)$*.fa; then sbatch --partition=research.q --output=$*.out --wrap="bwa index
    @if test -f $(INPUTDIR)$*.fasta; then sbatch --partition=research.q --output=$*.out --wrap="bwa inde

# Clean will delete all indexing files and folders.
clean:
    -rm -r index
```