Algorithms and Applications for MicroRNA Research - Sheet 4

Max Jakob, Malte Groß

June 14, 2018

Exercise 1

From *e!nsembl* we downloaded the 3' UTR sequences of humans, by selecting the dataset 'Human genes (GRCH38.p12)' and than filtered this by selecting sequences 3'UTR in the Attributes section within *BioMart*. We than used *wget* on the server download this by the following XML query (obtained by clicking on the XML button)

```
wget -O human_utr3s.fa -bqc '
http://www.ensembl.org/biomart/martservice?query=
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Query>
<Query  virtualSchemaName = "default" formatter = "FASTA"
header = "0" uniqueRows = "0" count = "" datasetConfigVersion = "0.6" >
    <Dataset name = "hsapiens_gene_ensembl" interface = "default" >
        <Attribute name = "ensembl_gene_id" />
        <Attribute name = "ensembl_transcript_id" />
        <Attribute name = "3utr" />
    </Dataset>
</Query>
```

Since there are *FASTA* headers without known sequences these were removed by two *grep* terminal commands.

```
grep -B1 -A1 "Sequence unavailable" human_utr3s.fa > "unkown_seqs.fa"
```

which filters for the headers with unkown sequence and

```
grep -Fvxf unkown_seqs_utr3s.fa human_utr3s.fa > human_utr3s_clean.fa
```

which removes all of those missing sequences within the original file.

An *bowtie*-index was than created by using

```
bowtie-build human_utr3s.fa human_utr3s.fa &> log.txt &
```

Next we downloaded the current mirbase release from the projects's ftpWebsite.
We again used *wget* on the server to download mature.fa

```
wget ftp://mirbase.org/pub/mirbase/CURRENT/mature.fa.gz
```

this file was than filtered for human miRNAs by using

```
zcat mature.fa.gz | grep "Homo sapiens" -A1 > mirbase_human_mature.fa
```

two be comaptible with *bowtie* Uracils where replaced by Thymines using

```
sed '/^[^>]/ y/uU/tT/'  mirbase_human_mature.fa > T_mirbase_human_mature.fa
```

Last we created a the 5-8 mers using a small python script

```
import sys
import os.path

# substrings mirnas starting at position 2
def extractMiRNAs(writer, inputFile, i):
    inputFile = open(inputFile, "r")
    for line in inputFile:
        if line[0] == ">":
            writer.write(line)
        else:
            writer.write(line[1:i+1] + "\n")
    inputFile.close()


if __name__ == '__main__':


    # create mature-miRNA-substrings of length 5-8
    for i in range(5, 9):
        fileCut = open( os.path.join(sys.argv[1][0:-3]+"_"+str(i)+".fa"), "w")
        extractMiRNAs(fileCut, sys.argv[1], i)
        fileCut.close()
    print("Done")
```

Aligning the sequences

1. Using the whole sequence

   ```
   bowtie --nofw -f -v 2 human_utr3s.fa T_mirbase_human_mature.fa \
   1> whole_mirnas.txt 2> errors.txt
   ```

2. 5-mers

```
bowtie --nofw -f -v 1 human_utr3s.fa T_mirbase_human_mature_5.fa \
1> 5_mirnas.txt 2> errors.txt
```

3. 6-mers

```
bowtie --nofw -f -v 1 human_utr3s.fa T_mirbase_human_mature_6.fa \
1> 6_mirnas.txt 2> errors.txt
```

4. 7-mers

```
bowtie --nofw -f -v 0 human_utr3s.fa T_mirbase_human_mature_7.fa \
1> 7_mirnas.txt 2> errors.txt
```

5. 8-mers

```
bowtie --nofw -f -v 0 human_utr3s.fa T_mirbase_human_mature_8.fa \
1> 8_mirnas.txt 2> errors.txt
```

where

- --nofw no forward strand alginment
- -v maximal mismatch count
- 1> output
- 2> errors