predictSGRNA Package (Version 1.0.1)
Usage Tips

Pei Fen Kuan

May 20, 2017

1 Citing predictSGRNA

If you have used `predictSGRNA` in your work, please cite the package using the following:

P.F. Kuan, S. Powers, S. He, K. Li, X. Zhao and B. Huang (2017). "A systematic evaluation of nucleotide properties for CRISPR sgRNA design." BMC Bioinformatics, 18:297, DOI: 10.1186/s12859-017-1697-6

2 Introduction

This is an example of using `predictSGRNA` package in `R`. `predictSGRNA` package predicts sgRNA efficiency from position dependent dinucleotide model of Kuan et al. (2017). This vignette aims to demonstrate the usage of `predictSGRNA` through an example data.

3 Description and Usage

The main function in `predictSGRNA` package is `predictSGRNA` which predicts the efficiency of sgRNA target sequences.

```
predictSGRNA(seq.vec,outfile='output')
```

4 Arguments

4.1 seq.vec

seq.vec is a vector of sequences. Each sequence must of length 40 bp which consists of 5' flanking, sgRNA target sequence (including PAM (NGG)) and 3' flanking sequences. The PAM (NGG) sequence must be at positions 31, 32 and 33 of the 40 bp sequence.

4.2 outfile

Specify the name of the output file where the results will be saved.

5 Value

The results will be stored in `outfile.csv`, which consists of 3 columns. Column 1 is the sequence. Column 2 is the predicted value (either "Efficient" or "NotEfficient") based on predicted probability cutoff 0.5. Column 3 is the predicted probability of being "Efficient". The higher the probability, the more likely the sgRNA will be efficient.
NOTE: The predicted class (Column 2 of the output) is based on using probability cutoff 0.5 (from Column 3). User may choose different cutoffs for different levels of stringency.

6 Example

You may download the package from
`http://www.ams.sunysb.edu/~pfkuan/predictSGRNA/predictSGRNA_1.0.tar.gz`

To install the package

```
> install.packages('predictSGRNA_1.0.tar.gz',type='source')
```

To load the package

```
> library(predictSGRNA)
```

An example data with 20 sequences is stored in `exampleData`. The first few lines of `exampleData` is given below:

```
> data(exampleData)

> head(exampleData)
[1] GACACCTCGGCAGAAGCGAGACTTTGGGTACGGCTTGTTC
[2] GGGCTCTTCCACTACTTACCAGGGACACCTCGGCAGAAGC
[3] GACAGCTGCTCATATTCATCTGACACCATGTGGCCACAAA
[4] GACGAACAAAGAATCCTGCCTTACCTTCAGAGGACAGCTG
[5] GCTGCTTTTCTTCTCCCTACCTAGCCCTGGAGGCTGCCCG
[6] CCAGGGACACCTCGGCAGAAGCGAGACTTTGGGTACGGCT
20 Levels: AAAGGCAAAAGTGGATGAGTTTCCGCTTTGTGGCCACATG ...

> length(exampleData)
[1] 20
```

Alternatively, you may also download an example file from
`http://www.ams.sunysb.edu/~pfkuan/predictSGRNA/SampleData.txt`

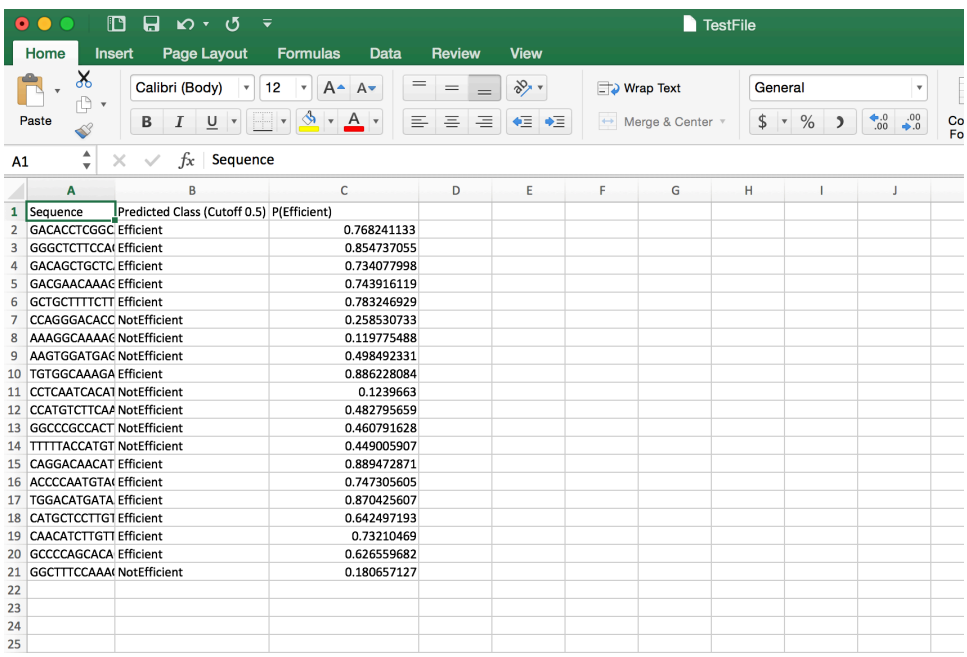
To load this input sequence file in R

```
> exampleData <- read.delim('SampleData.txt',header=FALSE)$V1
```

To run `predictSGRNA`:

```
> predictSGRNA(exampleData,'TestFile')
Make sure PAM (NGG) sequence is at position 31,32,33 of each sequence
Number of input sequence:  20
Number of input sequence with L=40:  20
DONE! Results saved in file  TestFile.csv
```

The results are stored in `TestFile.csv`. A snapshot of `TestFile.csv` is shown in Figure 1. NOTE: The predicted class (Column 2 of the output) is based on using probability cutoff 0.5 (from Column 3). User may choose different cutoffs for different levels of stringency.



Sequence	Predicted Class (Cutoff 0.5)	P(Efficient)
GACACCTCGGC	Efficient	0.768241133
GGGCTCTTCCA	Efficient	0.854737055
GACAGCTGCTC	Efficient	0.734077998
GACGAACAAAC	Efficient	0.743916119
GCTGCTTTTCTT	Efficient	0.783246929
CCAGGGACACC	NotEfficient	0.258530733
AAAGGCAAAAC	NotEfficient	0.119775488
AAGTGGATGAC	NotEfficient	0.498492331
TGTGGCAAAGA	Efficient	0.886228084
CCTCAATCACA	NotEfficient	0.1239663
CCATGTCTTCA	NotEfficient	0.482795659
GGCCCGCCACT	NotEfficient	0.460791628
TTTTTACCATGT	NotEfficient	0.449005907
CAGGACAACAT	Efficient	0.889472871
ACCCCAATGTA	Efficient	0.747305605
TGGACATGATA	Efficient	0.870425607
CATGCTCCTTGT	Efficient	0.642497193
CAACATCTTGTT	Efficient	0.73210469
GCCCCAGCACA	Efficient	0.626559682
GGCTTTCCAAA	NotEfficient	0.180657127

Figure 1: Snapshot of output file `TestFile.csv`.