The DNAshapeR package

(version 2.0)

Tsu-Pei Chiu and Federico Comoglio
Molecular and Computational Biology Program, Departments of Biological Sciences
University of Southern California, Los Angeles, USA
Department of Biosystems Science and Engineering
ETH Zürich, Basel, Switzerland
tsupeich@usc.edu
federico.comoglio@bsse.ethz.ch

October 21, 2015

Abstract

DNAshapeR predicts DNA shape features in an ultra-fast, high-throughput manner from genomic sequencing data. The package takes either nucleotide sequence or genomic intervals as input, and generates various graphical representations for further analysis. DNAshapeR further encodes DNA sequence and shape features for statistical learning applications by concatenating feature matrices with user-defined combinations of k-mer and DNA shape features that can be readily used as input for machine learning algorithms.

Contents

1 Importing DNA sequences into the R session

The input data for the analysis can be nucleotide sequence in FASTA file format. This can be done by giving customer defined sequence file or retrieving fasta files from defining GRanges and BSgenome.

1.1 Get sequence from own fasta file

```
library(DNAshapeR)
filename <- system.file("extdata", "CGRsample.fa", package = "DNAshapeR")
pred <- getShape(filename)

## Reading the input sequence......
## Reading the input sequence......
## Reading the input sequence......
## Reading the input sequence......

## Parsing files......
## Record length:  2000
## Record length:  1999
## Record length:  2000
## Record length:  1999
## Done
```

1.2 Get sequence from defined GRanges and BSgenome

Before getting the data, you need to use and install some packages

```
source("http://bioconductor.org/biocLite.R")
library(GenomicRanges)
library(BSgenome)
biocLite("BSgenome.Scerevisiae.UCSC.sacCer3")
library(BSgenome.Scerevisiae.UCSC.sacCer3)
```

Then you can get the fasta file and run prediction

```
gr <- GRanges(seqnames = c("chrI"),
              strand = c("+","-","+"),
              ranges = IRanges(start = c(100,200,300), width = 100))
getFasta(gr, Scerevisiae, width = 100, filename = "tmp.fa")
filename <- "tmp.fa"
pred <- getShape(filename)
```

1.3 Get sequence from public domain projects

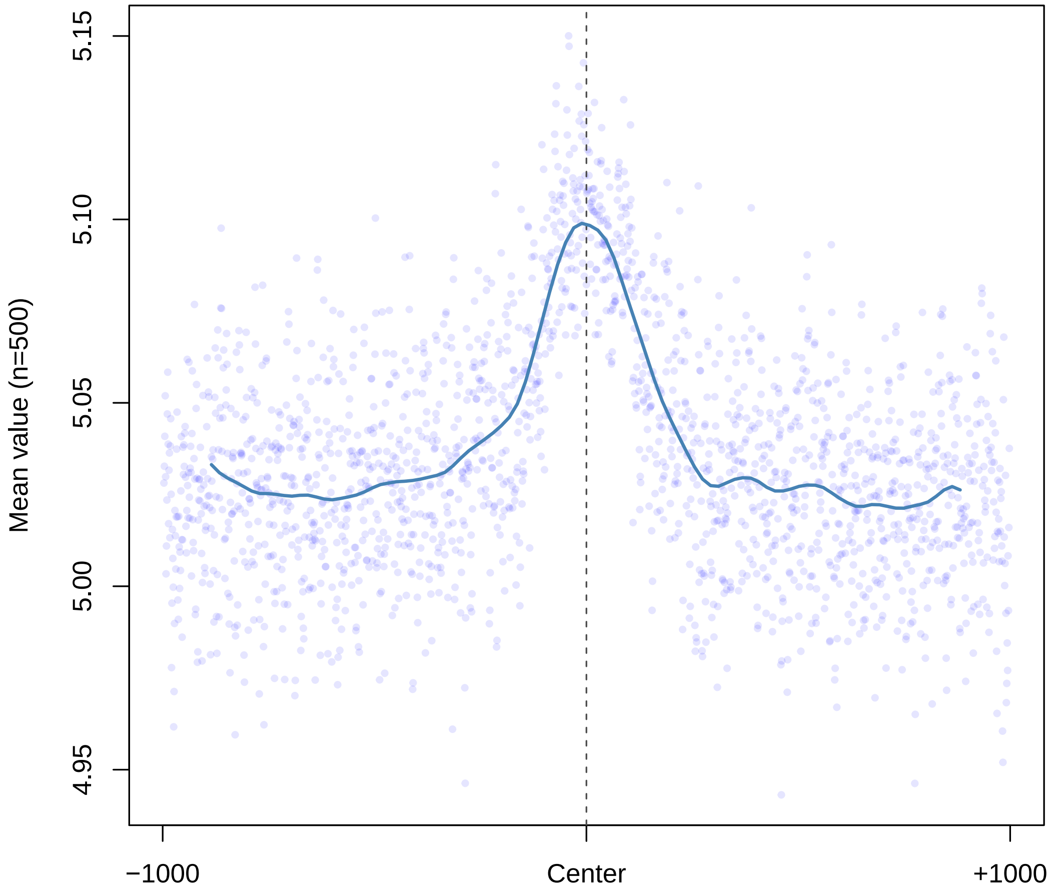
```
biocLite("AnnotationHub")
library(AnnotationHub)
ah <- AnnotationHub()
ah <- subset(ah, species=="Homo sapiens")
ah <- query(ah, c("H3K4me3", "Gm12878", "Roadmap"))

getFasta(ah[[1]], Hsapiens, width=150, filename = "tmp.fa")
```

```
filename <- "tmp.fa"
pred <- getShape(filename)
```

2 Visulize the prediction result

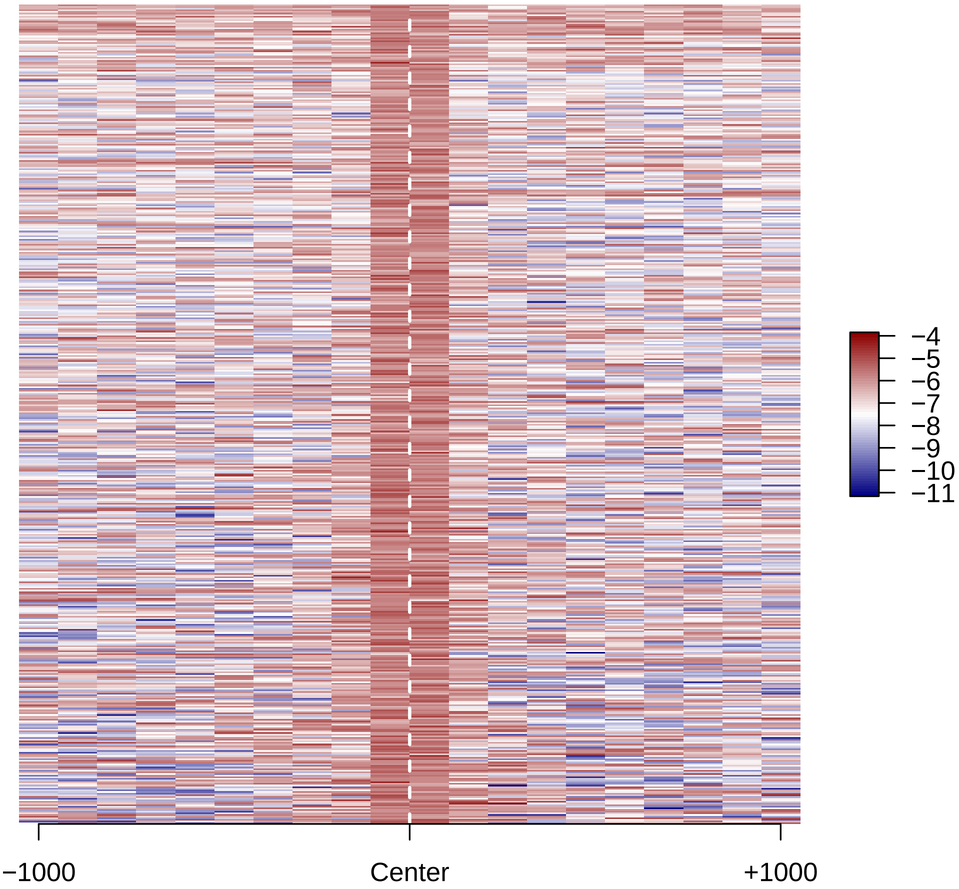
2.1 Plot

```
plotShape(pred$MGW)
```



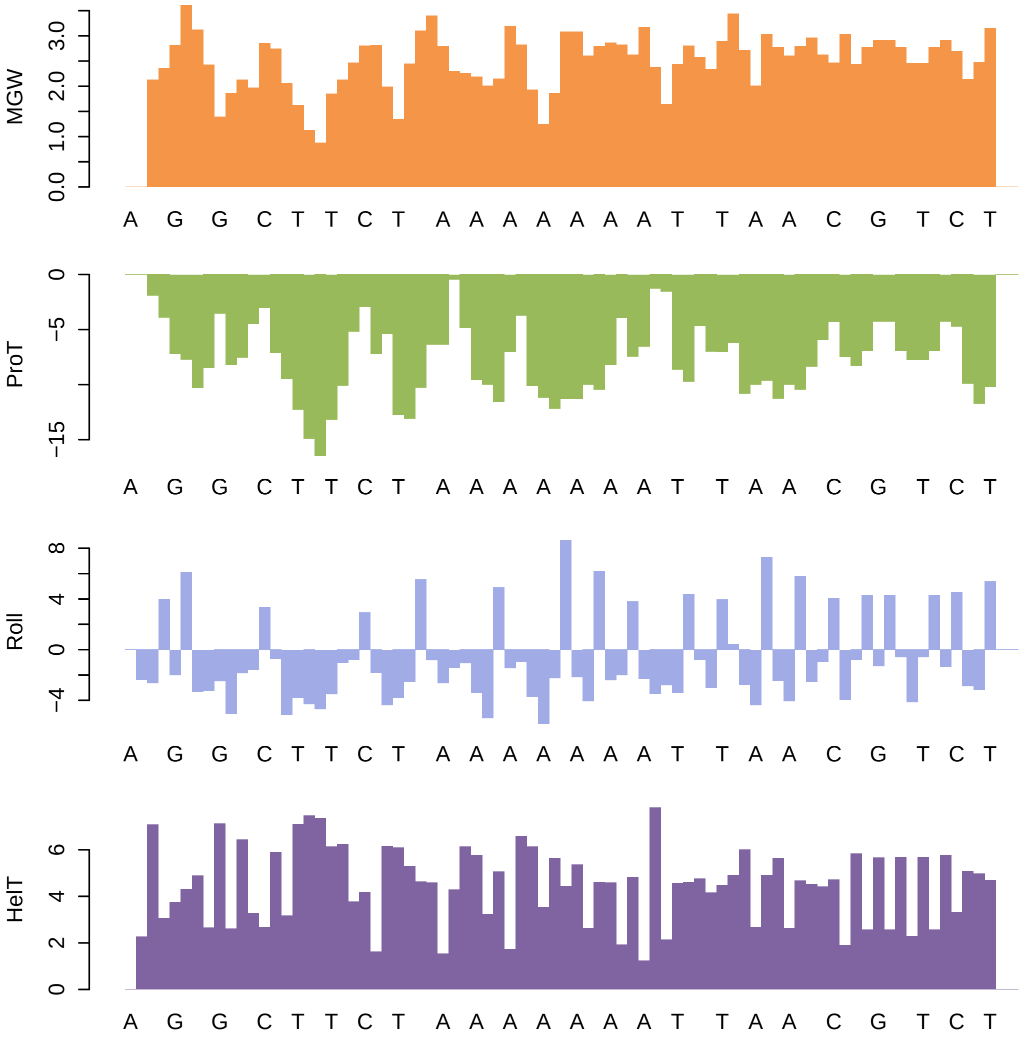
2.2 Heat maps

```
library(fields)

## Loading required package:  spam
## Loading required package:  grid
## Spam version 1.2-1 (2015-09-30) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g.  'help( chol.spam)'.
##
## Attaching package:  'spam'
##
## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve
##
## Loading required package:  maps
##
##  # ATTENTION: maps v3.0 has an updated 'world' map.        #
##  # Many country borders and names have changed since 1990.  #
##  # Type '?world' or 'news(package="maps")'.  See README_v3.  #

heatShape(pred$ProT, 20)
```



2.3 Genome Browser Tracks

```
filename2 <- system.file("extdata", "SingleSeqsample.fa", package = "DNAshapeR")
pred2 <- getShape(filename2)

## Reading the input sequence......
## Reading the input sequence......
## Reading the input sequence......
## Reading the input sequence......

## Parsing files......
## Record length:  80
## Record length:  79
## Record length:  80
## Record length:  79
## Done

trackShape(filename2, pred2)
```



3 Feature Encoding

```
library(Biostrings)
featureNames <- c("1-mer","1-shape")
featureVector <- encodeSeqShape(filename, pred, featureNames)
```

3.1 Usecase of Machine Learning Application

```
filename3 <- system.file("extdata", "SELEXsample.s", package = "DNAshapeR")
experimentalData <- read.table(filename3)
df <- data.frame(affinity=experimentalData $V1, featureVector)

library(caret)
trainControl <- trainControl(method = "cv", number = 10, savePredictions = TRUE)
model <- train (affinity˜ ., data = df, trControl=trainControl, method="lm", preProcess=NULL)
summary(model)
```

4 Session Info

```
sessionInfo()

## R version 3.2.2 (2015-08-14)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 7 x64 (build 7601) Service Pack 1
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] fields_8.3-5     maps_3.0.0-2     spam_1.2-1       DNAshapeR_0.99.1
## [5] knitr_1.11
##
## loaded via a namespace (and not attached):
## [1] magrittr_1.5  formatR_1.2.1 tools_3.2.2   Rcpp_0.12.1   stringi_0.5-5
## [6] highr_0.5.1   stringr_1.0.0 evaluate_0.8
```

References

[1] T. Zhou, L. Yang, Y. Lu, I. Dror, A.C. Dantas Machado, T. Ghane, R. Di Felice, and R. Rohs (2013) DNAshape: a method for the high-throughput prediction of DNA structural features on a genome-wide scale. *Nucleic Acids Res* **41**: W56-62