

# バイオインフォマティクス人材育成カリ キュラム(次世代シーケンサ)速習 コース

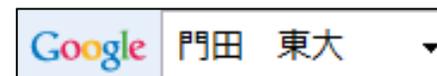
## 3. データ解析基礎 | 3-4. R Bioconductor I

東京大学・大学院農学生命科学研究科  
アグリバイオインフォマティクス教育研究ユニット

門田幸二(かどた こうじ)

kadota@iu.a.u-tokyo.ac.jp

<http://www.iu.a.u-tokyo.ac.jp/~kadota/>



# Contents

- 3-4. R Bioconductor I、2014/09/09 10:30-14:45、中級、実習
  - Tips: setwd関数を利用した効率的な作業ディレクトリの変更
  - データの型1: translate関数の入力情報(AAStringSet)
  - Tips: オブジェクトの消去
  - データの型2: 翻訳配列取得のコードの中身を解説
  - バージョン情報把握とバージョンアップ
  - バージョン違いの影響
    - 過去から現在: BiostringsパッケージのreadDNAStrngSet関数
    - 現在から未来: BSgenome.Hsapiens.UCSC.hg19パッケージでプロモーター配列取得
  - Bioconductor概観



- イントロ | 一般 | [任意の長さの可能な全ての塩基配列を作成](#) (last modified 2013/06/14)
- イントロ | 一般 | [任意の位置の塩基を置換](#) (last modified 2013/09/12)
- イントロ | 一般 | [指定した範囲の配列を取得](#) (last modified 2014/03/08)
- イントロ | 一般 | [指定したID\(染色体やdescription\)の配列を取得](#) (last modified 2014/03/10)
- イントロ | 一般 | [翻訳配列\(translate\)を取得](#) (last modified 2013/06/14)
- イントロ | 一般 | [相補鎖\(complement\)を取得](#) (last modified 2013/06/14)
- イントロ | 一般 | [逆相補鎖\(reverse complement\)を取得](#) (last modified 2013/06/14)

Rに慣れてくると、Gui画面上での煩雑な「ファイル - ディレクトリの変更」作業を効率的に行いたくなります。

## イントロ | 一般 | [翻訳配列\(translate\)を取得](#) **NEW**

塩基配列を読み込んでアミノ酸配列に翻訳するやり方を示します。  
「ファイル」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

### 1. FASTA形式ファイル(sample1.fasta)の

```
in_f <- "sample1.fasta"
out_f <- "hoge1.fasta"

#必要なパッケージをロード
library(Biostrings)

#入力ファイルの読み込み
fasta <- readDNAStringSet(in_f)

#本番
hoge <- translate(fasta)
names(hoge) <- names(fasta)
fasta <- hoge

#ファイルに保存
writeXStringSet(fasta, file=out_f)
```

```
rcode_translate.txt x
##### ↓
### 翻訳配列取得 ↓
##### ↓
in_f <- "sample1.fasta" #入力ファイル名を指定してin_fに格納 ↓
out_f <- "hoge1.fasta" #出力ファイル名を指定してout_fに格納 ↓
↓
#必要なパッケージをロード ↓
library(Biostrings) #パッケージの読み込み ↓
↓
#入力ファイルの読み込み ↓
fasta <- readDNAStringSet(in_f, format="fasta") #in_fで指定したファイルの読み込み ↓
↓
#本番 ↓
hoge <- translate(fasta) #fastaをアミノ酸配列に翻訳した結果をhogeに格納 ↓
names(hoge) <- names(fasta) #現状では翻訳した結果のオブジェクトhogeに格納 ↓
fasta <- hoge #hogeの中身をfastaに格納 ↓
fasta #確認してるだけです ↓
↓
#ファイルに保存 ↓
writeXStringSet(fasta, file=out_f, format="fasta", width=50) #fastaの中身を指定 ↓
```

# 作業ディレクトリの変更

「Windows(C:)」となっている場合もあるが、気にしない

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ

R コードのソースを読み込み...  
新しいスクリプト  
スクリプトを開く...  
ファイルの表示...  
作業スペースの読み込み...  
作業スペースの保存...  
履歴の読み込み...  
履歴の保存...  
ディレクトリの変更... ①  
印刷...  
ファイルを保存...  
終了

作業ディレクトリの変更  
C:\

ローカル ディスク (C:) ②

空き領域: 280 GB  
合計サイズ: 453 GB

フォルダー(F): ローカル ディスク (C:)

新しいフォルダーの作成(N) OK キャンセル

④はヒトそれぞれ

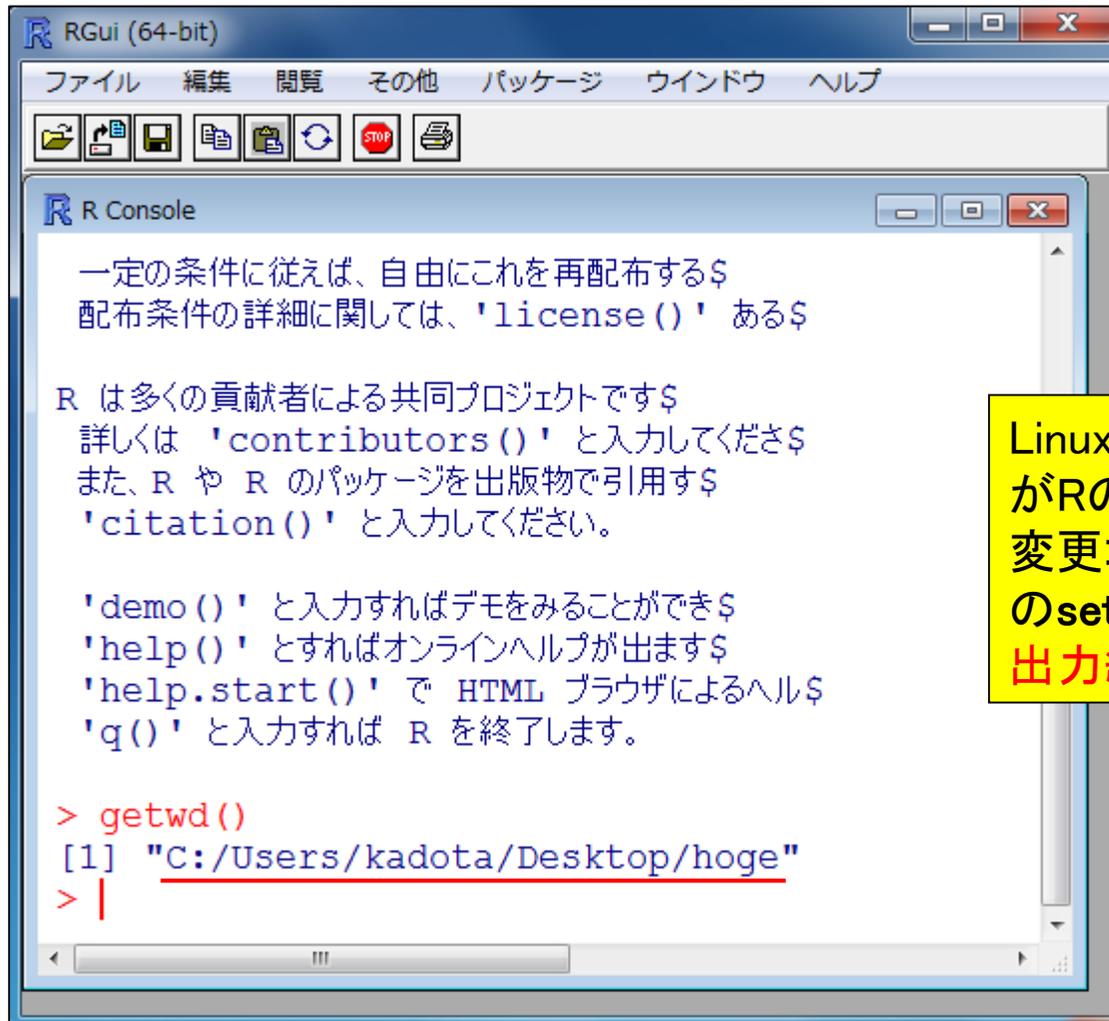
作業ディレクトリの変更  
C:\Users\kadota\Desktop\hoge

Users ③  
Default  
kadota ④  
AppData  
Dropbox  
Roaming  
アドレス帳  
お気に入り  
ダウンロード  
デスクトップ ⑤  
hoge ⑥

フォルダー(F): hoge

新しいフォルダーの作成(N) OK ⑦ キャンセル

# 作業ディレクトリの変更



The screenshot shows the RGui (64-bit) window with a menu bar (ファイル, 編集, 閲覧, その他, パッケージ, ウィンドウ, ヘルプ) and a toolbar. The R Console window is open, displaying the following text:

```
一定の条件に従えば、自由にこれを再配布する$  
配布条件の詳細に関しては、'license()' がある$  
  
R は多くの貢献者による共同プロジェクトです$  
詳しくは 'contributors()' と入力してくださ$  
また、R や R のパッケージを出版物で引用す$  
'citation()' と入力してください。  
  
'demo()' と入力すればデモをみることができ$  
'help()' とすればオンラインヘルプが出ます$  
'help.start()' で HTML ブラウザによるヘル$  
'q()' と入力すれば R を終了します。  
  
> getwd()  
[1] "C:/Users/kadota/Desktop/hoge"  
> |
```

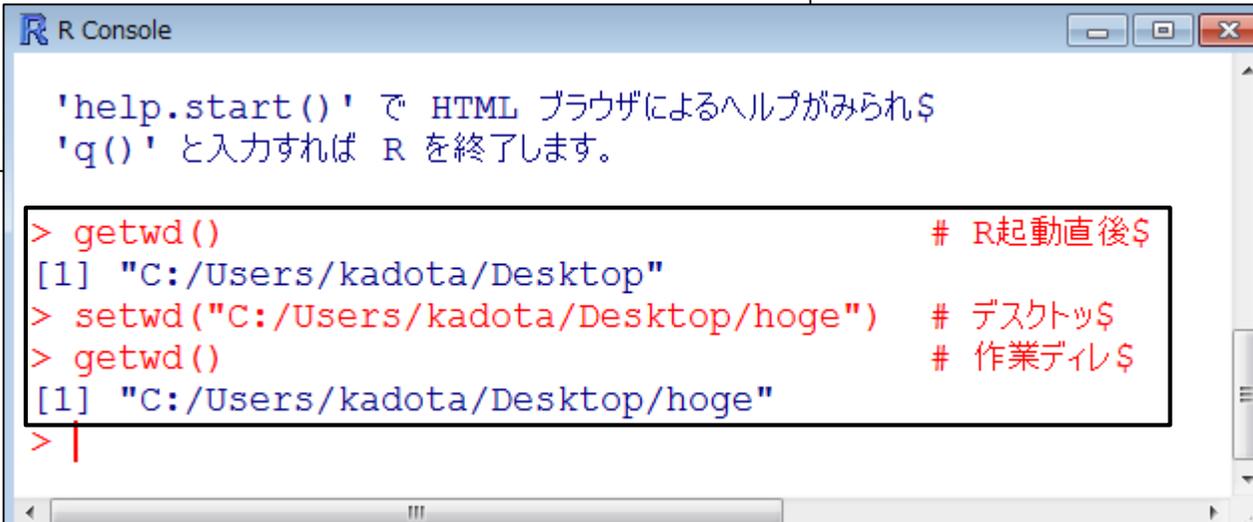
Linuxのpwdコマンドに相当するの  
がRのgetwd()。Linuxのディレクトリ  
変更コマンドcdに相当するのがR  
のsetwd()。重要なのは、getwd()の  
出力結果を把握しておくことです。

# デスクトップのhogeと決めるなら...

ファイル名: rcode\_20140909.txt

Rを再起動してコピー。setwd関数実行によって、作業ディレクトリの変更がコピーで完了していることが分かる。

```
#####↓  
### Tips (作業ディレクトリの変更)↓  
### Windows PCでユーザがkadotaかiuの場合↓  
#####↓  
getwd() # R起動直後の作業ディレクトリを表示↓  
setwd("C:/Users/kadota/Desktop/hoge") # デスクトップ上のhogeフォルダにしたい場合↓  
getwd() # 作業ディレクトリが変更できているか確認↓  
↓  
getwd() # R起動直後の作業ディレクトリを表示↓  
setwd("C:/Users/iu/Desktop/hoge") # デスクトップ上のhogeフォルダにしたい場合↓  
getwd() # 作業ディレクトリが変更できているか確認↓  
↓  
setwd("C:/Users/kadota/Desktop")  
setwd("C:/Users/kadota/Documents")  
↓
```



```
R Console  
# 'help.start()' で HTML ブラウザによるヘルプがみられ$  
# 'q()' と入力すれば R を終了します。  
  
> getwd() # R起動直後$  
[1] "C:/Users/kadota/Desktop"  
> setwd("C:/Users/kadota/Desktop/hoge") # デスクトップ$  
> getwd() # 作業ディレ$  
[1] "C:/Users/kadota/Desktop/hoge"  
> |
```

# デスクトップのhogeと決めるなら...

ファイル名: rcode\_20140909.txt

アグリバイオ貸与PCはユーザ名がiu  
なのでエラーが出ないと思います。

```
#####↓  
### Tips (作業ディレクトリの変更)↓  
### Windows PCでユーザがkadotaかiuの場合↓  
#####↓  
getwd() # R起動直後の作業ディレクトリを表示↓  
setwd("C:/Users/kadota/Desktop/hoge") # デスクトップ上のhogeフォルダにしたい場合↓  
getwd() # 作業ディレクトリが変更できているか確認↓  
↓  
getwd() # R起動直後の作業ディレクトリを表示↓  
setwd("C:/Users/iu/Desktop/hoge") # デスクトップ上のhogeフォルダにしたい場合↓  
getwd() # 作業ディレクトリが変更できているか確認↓  
↓  
setwd("C:/Users/kadota/Desktop")  
setwd("C:/Users/kadota/Documents")  
↓
```



```
R Console  
# 'help.start()' で HTML ブラウザによるヘルプがみら$  
# 'q()' と入力すれば R を終了します。  
  
> getwd() # R起動直$  
[1] "C:/Users/kadota/Desktop"  
> setwd("C:/Users/iu/Desktop/hoge") # デスクト$  
以下にエラー setwd("C:/Users/iu/Desktop/hoge") :  
作業ディレクトリを変更できません  
> getwd() # 作業ディ$  
[1] "C:/Users/kadota/Desktop"  
> |
```

# 自分で解析していた頃のコード例

イントロ | 一般 | [翻訳配列\(translate\)を取得](#) **NEW**

塩基配列を読み込んでアミノ酸配列に翻訳するや  
「ファイル」-「ディレクトリの変更」で解析したいファ

## 1. FASTA形式ファイル(sample1.fasta)の場合:

```

in_f <- "sample1.fasta"
out_f <- "hoge1.fasta"

#必要なパッケージをロード
library(Biostrings)

#入力ファイルの読み込み
fasta <- readDNAStringSet(in_f, form

#本番
hoge <- translate(fasta)
names(hoge) <- names(fasta)
fasta <- hoge
fasta

#ファイルに保存
writeXStringSet(fasta, file=out_f, f

```

ファイル名: rcode\_translate2.txt

```

#####↓
### 作業ディレクトリの変更↓
#####↓
setwd("C:/Users/kadota/Desktop/hoge") # デスクトップ上のhogeフォルダにしたい場合↓
#setwd("C:/Users/kadota/Desktop") # 「デスクトップ」にしたい場合↓
#setwd("C:/Users/kadota/Documents") # 「マイ ドキュメント」にしたい場合↓
↓
#####↓
### 翻訳配列取得↓
#####↓
in_f <- "sample1.fasta" #入力ファイル名を指定してin_fに格納↓
out_f <- "hoge1.fasta" #出力ファイル名を指定してout_fに格納↓
↓
#必要なパッケージをロード↓
library(Biostrings)
↓
#入力ファイルの読み込み↓
fasta <- readDNAStringSet(in_f, format="fasta")#in_fで指定したファイルの読み込み↓
↓
#本番↓
hoge <- translate(fasta) #fastaをアミノ酸配列に翻訳した結果をhogeに格納↓
names(hoge) <- names(fasta) #現状では翻訳した結果のオブジェクトhogeのdescri
fasta <- hoge #hogeの中身をfastaに格納↓
fasta #確認してるだけです↓
↓
#ファイルに保存↓
writeXStringSet(fasta, file=out_f, format="fasta", width=50)#fastaの中身を指定したファ
↓

```

**作業ディレクトリ変更を含んだ一連のコードを用意しておき、Rを起動直後にコピー**

# 自分で解析していた頃のコード例

ファイル名: rcode\_translate2.txt

```
##### ↓
### 作業ディレクトリの変更 ↓
##### ↓
setwd("C:/Users/kadota/Desktop/hoge") ← # デスクトップ上のhogeフォルダにしたい場合 ↓
#setwd("C:/Users/kadota/Desktop")      # 「デスクトップ」にしたい場合 ↓
#setwd("C:/Users/kadota/Documents")     # 「マイドキュメント」にしたい場合 ↓
↓
##### ↓
### 翻訳配列取得 ↓
##### ↓
in_f <- "sample1.fasta"                 # 入力ファイル名を指定してin_fに格納 ↓
out_f <- "hoge1.fasta"                  # 出力ファイル名を指定してout_fに格納 ↓
↓
#必要なパッケージをロード ↓
library(Biostrings)                    # パッケージの
↓
#入力ファイルの読み込み ↓
fasta <- readDNASTringSet(in_f, format="fasta") # in_
↓
#本番 ↓
hoge <- translate(fasta)                # fastaをアミ
names(hoge) <- names(fasta)            # 現状では翻訳
fasta <- hoge                           # hogeの中身を
fasta                                    # 確認してるた
↓
#ファイルに保存 ↓
writeXStringSet(fasta, file=out_f, format="fasta",
↓
```

いくつかのテンプレートとして使う可能性のあるものを予め用意しておき、実際に使うもののみ#を外して利用していました。

```
R Console
> hoge <- translate(fasta)                # fastaをアミ
> names(hoge) <- names(fasta)            # 現状では$
> fasta <- hoge                           # hogeの中$
> fasta                                    # 確認して$
  A AASTringSet instance of length 1
    width seq          names
[1]     4 SDGL          kadota
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta"$
> |
```

塩基配列を読み込んでアミノ酸配列に翻訳するやり方を示します。

「ファイル」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

## 1. FASTA形式ファイル(sample1.fasta)の場合:

```

in_f <- "sample1.fasta"      #入力ファイル名を指定してin_fに格納
out_f <- "hoge1.fasta"      #出力ファイル名を指定してout_fに格納

#必要なパッケージをロード
library(Biostrings)        #パッケージの読み込み

#入力ファイルの読み込み
fasta <- readDNASTringSet(in_f, format="fasta")#in_fで指定したファイルの読み込み

#本番
hoge <- translate(fasta)    #fastaをアミノ酸配列に翻訳した結果をhogeに格納
names(hoge) <- names(fasta) #現状では翻訳した結果のオブジェクトhogeのdescript
fasta <- hoge              #hogeの中身をfastaに格納
fasta                    #確認してるだけです
    
```

#ファイルに保存  
writeXStringSet

入力: 塩基配列ファイル(sample1.fasta)

```

sample1.fasta - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V)
ヘルプ(H)
>kadota
AGTGACGGTCTT
    
```

出力: アミノ酸配列ファイル(hoge1.fasta)

```

hoge1.fasta - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V)
ヘルプ(H)
>kadota
SDGL
    
```

# Contents

- 3-4. R Bioconductor I、2014/09/09 10:30-14:45、中級、実習
  - Tips: setwd関数を利用した効率的な作業ディレクトリの変更
  - データの型1: translate関数の入力情報(AAStringSet)
  - Tips: オブジェクトの消去
  - データの型2: 翻訳配列取得のコードの中身を解説
  - バージョン情報把握とバージョンアップ
  - バージョン違いの影響
    - 過去から現在: BiostringsパッケージのreadDNAStrngSet関数
    - 現在から未来: BSgenome.Hsapiens.UCSC.hg19パッケージでプロモーター配列取得
  - Bioconductor概観



# データの型1

ファイル名: rcode\_translate2.txt

```
##### ↓
### 作業ディレクトリの変更 ↓
##### ↓
setwd("C:/Users/kadota/Desktop/hoge") # デスクトップ上のhogeフォルダにした
#setwd("C:/Users/kadota/Desktop") # 「デスクトップ」にしたい場合 ↓
#setwd("C:/Users/kadota/Documents") # 「マイドキュメント」にしたい場合
↓
##### ↓
### 翻訳配列取得 ↓
##### ↓
in_f <- "sample1.fasta" # 入力ファイル
out_f <- "hoge1.fasta" # 出力ファイル
↓
#必要なパッケージをロード ↓
library(Biostrings) # パッケージの
↓
#入力ファイルの読み込み ↓
fasta <- readDNASTringSet(in_f, format="fasta") # in_
↓
#本番 ↓
hoge <- translate(fasta) # fastaをアミ
names(hoge) <- names(fasta) # 現状では翻訳
fasta <- hoge # hogeの中身を
fasta # 確認してるた
↓
#ファイルに保存 ↓
writeXStringSet(fasta, file=out_f, format="fasta",
↓
```

左記のコード実行後のfastaオブジェクトにはアミノ酸配列情報が格納されていることが分かる。AAStringSetのAAはAmino Acids (アミノ酸)の略。AAStringSetクラスオブジェクトとかAAStringSetクラスなどと呼ばれるが、実用上はAAStringSetというもののまたは型という程度の理解でよい。

```
R Console
> fasta
A AAStringSet instance of length 1
width seq names
[1] 4 SDGL kadota
>
> translate(fasta)
以下にエラー (function (classes, fdef, mtable) :
unable to find an inherited method for function $
> hoge <- translate(fasta)
以下にエラー (function (classes, fdef, mtable) :
unable to find an inherited method for function $
> hoge
A AAStringSet instance of length 1
width seq names
[1] 4 SDGL kadota
> |
```

# データの型1

ファイル名: rcode\_translate2.txt

```
##### ↓
### 作業ディレクトリの変更 ↓
##### ↓
setwd("C:/Users/kadota/Desktop/hoge") # デスクトップ上のhogeフォルダにした
#setwd("C:/Users/kadota/Desktop") # 「デスクトップ」にしたい場合 ↓
#setwd("C:/Users/kadota/Documents") # 「マイドキュメント」にしたい場合 ↓
↓
##### ↓
### 翻訳配列取得 ↓
##### ↓
in_f <- "sample1.fasta" # 入力ファイル
out_f <- "hoge1.fasta" # 出力ファイル
↓
#必要なパッケージをロード ↓
library(Biostrings) # パッケージの
↓
#入力ファイルの読み込み ↓
fasta <- readDNASTringSet(in_f, format="fasta") # in_
↓
#本番 ↓
hoge <- translate(fasta) # fastaをアミ
names(hoge) <- names(fasta) # 現状では翻訳
fasta <- hoge # hogeの中身を
fasta # 確認してるた
↓
#ファイルに保存 ↓
writeXStringSet(fasta, file=out_f, format="fasta",
↓
```

fastaオブジェクトを入力としてtranslate関数を実行してもエラーが出る。この理由は明確。translate関数は塩基配列を入力としてアミノ酸配列を出力するものであり、アミノ酸配列情報からなるfastaオブジェクトを入力とするのは想定外だから。

```
R Console
> fasta
A AAStringSet instance of length 1
width seq names
[1] 4 SDGL kadota
>
> translate(fasta)
以下にエラー (function (classes, fdef, mtable) :
unable to find an inherited method for function $
> hoge <- translate(fasta)
以下にエラー (function (classes, fdef, mtable) :
unable to find an inherited method for function $
> hoge
A AAStringSet instance of length 1
width seq names
[1] 4 SDGL kadota
> |
```

# データの型1

ファイル名: rcode\_translate2.txt

```
##### ↓
### 作業ディレクトリの変更↓
##### ↓
setwd("C:/Users/kadota/Desktop/hoge") # デスクトップ上のhogeフォルダにしたい場合↓
#setwd("C:/Users/kadota/Desktop") # 「デスクトップ」にしたい場合↓
#setwd("C:/Users/kadota/Documents") # 「マイドキュメント」にしたい場合↓
↓
##### ↓
### 翻訳配列取得↓
##### ↓
in_f <- "sample1.fasta" #入力ファイル
out_f <- "hoge1.fasta" #出力ファイル
↓
#必要なパッケージをロード↓
library(Biostrings) #パッケージの
↓
#入力ファイルの読み込み↓
fasta <- readDNASTringSet(in_f, format="fasta")#in_
↓
#本番↓
hoge <- translate(fasta) #fastaをアミ
names(hoge) <- names(fasta) #現状では翻訳
fasta <- hoge #hogeの中身を
fasta #確認してるた
↓
#ファイルに保存↓
writeXStringSet(fasta, file=out_f, format="fasta",
↓
```

塩基配列情報からなるfastaオブジェクトを入力としたときには正常に動作していた。

当然ながら、アミノ酸配列情報からなるfastaオブジェクトを入力としている以上、それと全く同じコードに変更してもエラーが出る。

```
R Console
> fasta
A AAStringSet instance of length 1
width seq names
[1] 4 SDGL kadota
>
> translate(fasta)
以下にエラー (function (classes, fdef, mtable) :
unable to find an inherited method for function $
> hoge <- translate(fasta)
以下にエラー (function (classes, fdef, mtable) :
unable to find an inherited method for function $
> hoge
A AAStringSet instance of length 1
width seq names
[1] 4 SDGL kadota
> |
```

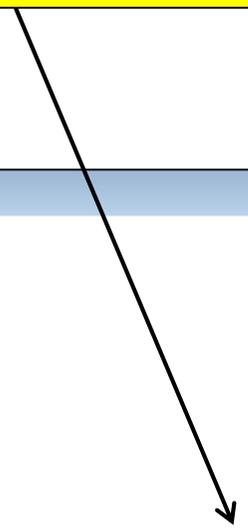
# データの型1

ファイル名: rcode\_translate2.txt

エラーメッセージの全貌。translate関数の基本動作が分かっている状態で眺め、「AAStringSetというアミノ酸配列情報を入力としているからダメ」と言われているのだろう、という対策方針を立てるべし

```
##### ↓
### 作業ディレクトリの変更 ↓
##### ↓
setwd("C:/Users/kadota/Desktop/hoge") # デスクトップ上のhogeフォルダにしたい場合 ↓
#setwd("C:/Users/kadota/Desktop") # 「デスクトップ」にしたい場合 ↓
#setwd("C:/Users/kadota/Documents") # 「マイドキュメント」にしたい場合 ↓
↓
##### ↓
```

```
R Console
> fasta
  A AAStringSet instance of length 1
    width seq          names
[1]      4 SDGL        kadota
>
> translate(fasta)
以下にエラー (function (classes, fdef, mtable) :
unable to find an inherited method for function 'translate' for signature "AAStringSet"
> hoge <- translate(fasta)
以下にエラー (function (classes, fdef, mtable) :
unable to find an inherited method for function 'translate' for signature "AAStringSet"
> hoge
  A AAStringSet instance of length 1
    width seq          names
[1]      4 SDGL        kadota
> |
```



# エラーなのに正しい結果が得られている?!

ファイル名: rcode\_translate2.txt

```
##### ↓
### 作業ディレクトリの変更 ↓
##### ↓
setwd("C:/Users/kadota/Desktop/hoge") # デスクトップ上のhogeフォルダ
#setwd("C:/Users/kadota/Desktop") # 「デスクトップ」にした
#setwd("C:/Users/kadota/Documents") # 「マイドキュメント」に
↓
##### ↓
### 翻訳配列取得 ↓
##### ↓
in_f <- "sample1.fasta" # 入力ファイル
out_f <- "hoge1.fasta" # 出力ファイル
↓
#必要なパッケージをロード ↓
library(Biostrings) # パッケージの読み込み
↓
#入力ファイルの読み込み ↓
fasta <- readDNASTringSet(in_f, format="fasta") # 読み込み
↓
#本番 ↓
hoge <- translate(fasta) ① # fastaをアミノ酸配列に変換
names(hoge) <- names(fasta) # 現状では翻訳されたアミノ酸配列
fasta <- hoge # hogeの中身をfastaに代入
fasta # 確認してる
↓
#ファイルに保存 ↓
writeXStringSet(fasta, file=out_f, format="fasta",
↓
```

③のhogeは、②の実行結果として得られたhogeではなく、①の実行結果として得られたhogeです。エラーが出ているのになぜかうまくいっている、というわけではありません。単純に以前作成した同名オブジェクトの中身が残っていたというだけ。

```
R Console
> fasta
A AAStringSet instance of length 1
width seq names
[1] 4 SDGL kadota
>
> translate(fasta)
以下にエラー (function (classes, fdef, mtable) :
unable to find an inherited method for function $)
> hoge <- translate(fasta) ②
以下にエラー (function (classes, fdef, mtable) :
unable to find an inherited method for function $)
> hoge ③
A AAStringSet instance of length 1
width seq names
[1] 4 SDGL kadota
> |
```

# Contents

- 3-4. R Bioconductor I、2014/09/09 10:30-14:45、中級、実習
  - Tips: setwd関数を利用した効率的な作業ディレクトリの変更
  - データの型1: translate関数の入力情報(AAStringSet)
  - Tips: オブジェクトの消去
  - データの型2: 翻訳配列取得のコードの中身を解説
  - バージョン情報把握とバージョンアップ
  - バージョン違いの影響
    - 過去から現在: BiostringsパッケージのreadDNAStrngSet関数
    - 現在から未来: BSgenome.Hsapiens.UCSC.hg19パッケージでプロモーター配列取得
  - Bioconductor概観



# オブジェクトの消去はrm関数を利用

ファイル名: rcode\_translate2.txt

```
##### ↓
### 作業ディレクトリの変更↓
##### ↓
setwd("C:/Users/kadota/Desktop/hoge") # デスクトップ上のhogeフォルダに移動
#setwd("C:/Users/kadota/Desktop") # 「デスクトップ」にした場合
#setwd("C:/Users/kadota/Documents") # 「マイドキュメント」にしたい場合↓
↓
##### ↓
### 翻訳配列取得↓
##### ↓
in_f <- "sample1.fasta" #入力ファイル名
out_f <- "hoge1.fasta" #出力ファイル名
↓
#必要なパッケージをロード↓
library(Biostrings) #パッケージの読み込み
↓
#入力ファイルの読み込み↓
fasta <- readDNASTringSet(in_f, format="fasta")#in_fの中身を読み込み
↓
#本番↓
hoge <- translate(fasta) #fastaをアミノ酸配列に変換
names(hoge) <- names(fasta) #現状では翻訳結果のオブジェクト名がfastaのまま
fasta <- hoge #hogeの中身をfastaに代入
fasta #確認してるか
↓
#ファイルに保存↓
writeXStringSet(fasta, file=out_f, format="fasta", width=50)#fastaの中身を指定したファイルに保存
```

以前作成したhogeオブジェクトの中身を消去すべく、rm関数を用いてhogeオブジェクトを消去。消去後に、再びtranslate関数を実行した結果をhogeに格納しようとしてエラーが出ることを再確認。

```
R Console
> rm(hoge)
> hoge
エラー: オブジェクト 'hoge' がありません
> hoge <- translate(fasta)
以下にエラー (function (classes, fdef, mtable) :
unable to find an inherited method for function $
> hoge
エラー: オブジェクト 'hoge' がありません
> ls()
[1] "fasta" "in_f" "out_f"
> objects()
[1] "fasta" "in_f" "out_f"
> |
```

# オブジェクトの消去はrm関数を利用

ファイル名: rcode\_translate2.txt

```
##### ↓
### 作業ディレクトリの変更↓
##### ↓
setwd("C:/Users/kadota/Desktop/hoge") # デスクトップ上のhogeフォルダに移動
#setwd("C:/Users/kadota/Desktop") # 「デスクトップ」にした
#setwd("C:/Users/kadota/Documents") # 「マイドキュメント」に移動
↓
##### ↓
### 翻訳配列取得↓
##### ↓
in_f <- "sample1.fasta" #入力ファイル
out_f <- "hoge1.fasta" #出力ファイル
↓
#必要なパッケージをロード↓
library(Biostrings) #パッケージの読み込み
↓
#入力ファイルの読み込み↓
fasta <- readDNASTringSet(in_f, format="fasta")#in_fの中身をfastaに読み込む
↓
#本番↓
hoge <- translate(fasta) #fastaをアミノ酸配列に変換
names(hoge) <- names(fasta) #現状では翻訳結果のオブジェクト名がfastaと同じ
fasta <- hoge #hogeの中身をfastaに代入
fasta #確認してるか
↓
#ファイルに保存↓
writeXStringSet(fasta, file=out_f, format="fasta", width=50)#fastaの中身を指定したファイルに保存
↓
```

translate関数実行結果をhogeに格納しようとしてエラーが出た場合は、本来hogeが生成されないことを確認しているだけです。ls関数やobjects関数は、現在作成されているオブジェクトをリストアップ。確かにhogeは存在しないことが分かる。

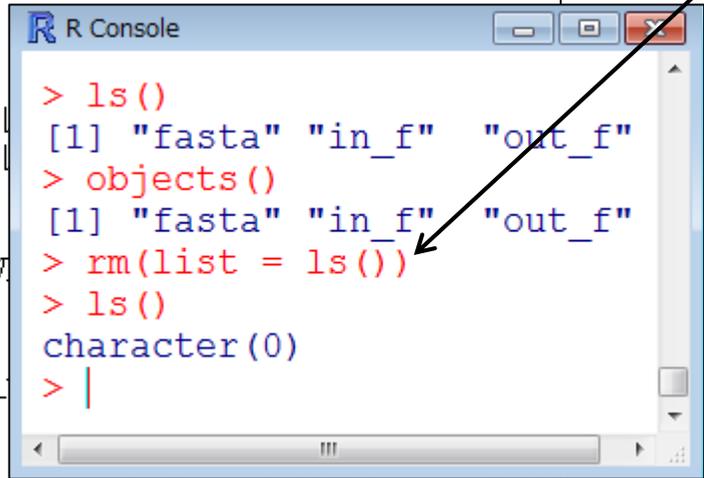
```
R Console
> rm(hoge)
> hoge
エラー: オブジェクト 'hoge' がありません
> hoge <- translate(fasta)
以下にエラー (function (classes, fdef, mtable) :
unable to find an inherited method for function $
> hoge
エラー: オブジェクト 'hoge' がありません
> ls()
[1] "fasta" "in_f" "out_f"
> objects()
[1] "fasta" "in_f" "out_f"
> |
```

# オブジェクトの消去はrm関数を利用

ファイル名: rcode\_translate2.txt

```
##### ↓
### 作業ディレクトリの変更 ↓
##### ↓
setwd("C:/Users/kadota/Desktop/hoge") # デスクトップ上のhogeフォルダに移動
#setwd("C:/Users/kadota/Desktop") # 「デスクトップ」にした
#setwd("C:/Users/kadota/Documents") # 「マイドキュメント」に移動
↓
##### ↓
### 翻訳配列取得 ↓
##### ↓
in_f <- "sample1.fasta" # 入力ファイル
out_f <- "hoge1.fasta" # 出力ファイル
↓
#必要なパッケージをロード ↓
library(Biostrings) # パッケージの読み込み
↓
#入力ファイルの読み込み ↓
fasta <- readDNASTringSet(in_f, format="fasta") # in_fの中身をfastaに格納
↓
#本番 ↓
hoge <- translate(fasta) # fastaをアミノ酸配列に翻訳した結果をhogeに格納
names(hoge) <- names(fasta) # 現状では翻訳した結果のオブジェクトhogeのdescription
fasta <- hoge # hogeの中身をfastaに格納
fasta # 確認してるだけです ↓
↓
#ファイルに保存 ↓
writeXStringSet(fasta, file=out_f, format="fasta", width=50) # fastaの中身を指定したファイルに保存
↓
```

現在作成されている全オブジェクトを全消去するやり方はrm(list = ls())。消去後に再びls関数を用いて利用可能なオブジェクトを表示させようとしても何もないという結果が得られる。character(0)は何もないという意味です。



#fastaをアミノ酸配列に翻訳した結果をhogeに格納  
#現状では翻訳した結果のオブジェクトhogeのdescription  
#hogeの中身をfastaに格納  
#確認してるだけです ↓

## ファイル名: rcode\_translate3.txt

```
##### ↓
### 作業ディレクトリの変更 ↓
##### ↓
setwd("C:/Users/kadota/Desktop/hoge") # デスクトップ上のhogeフォルダにしたい場合
#setwd("C:/Users/kadota/Desktop") # 「デスクトップ」にしたい場合 ↓
#setwd("C:/Users/kadota/Documents") # 「マイドキュメント」にしたい場合 ↓
↓
##### ↓
### オブジェクトの消去 ↓
##### ↓
rm(list = ls()) ←
↓
##### ↓
### 翻訳配列取得 ↓
##### ↓
in_f <- "sample1.fasta" # 入力ファイル名を指定してin_fに格納 ↓
out_f <- "hoge1.fasta" # 出力ファイル名を指定してout_fに格納 ↓
↓
# 必要なパッケージをロード ↓
library(Biostrings) # パッケージの読み込み ↓
↓
# 入力ファイルの読み込み ↓
fasta <- readDNASTringSet(in_f, format="fasta") # in_fで指定したファイルの読み込み ↓
↓
# 本番 ↓
hoge <- translate(fasta) # fastaをアミノ酸配列に翻訳した結果をhogeに格納 ↓
names(hoge) <- names(fasta) # 現状では翻訳した結果のオブジェクトhogeのディメンションをhogeに格納 ↓
fasta <- hoge # hogeの中身をfastaに格納 ↓
fasta # 確認してるだけです ↓
↓
# ファイルに保存 ↓
writeXStringSet(fasta, file=out_f, format="fasta", width=50) # fastaの中身を指定したファイルに保存 ↓
↓
```

**初級者: 変なことが起こったらRを一旦終了してから再起動**  
**中級者: Rの再起動はやらず、オブジェクトの全消去で対応**

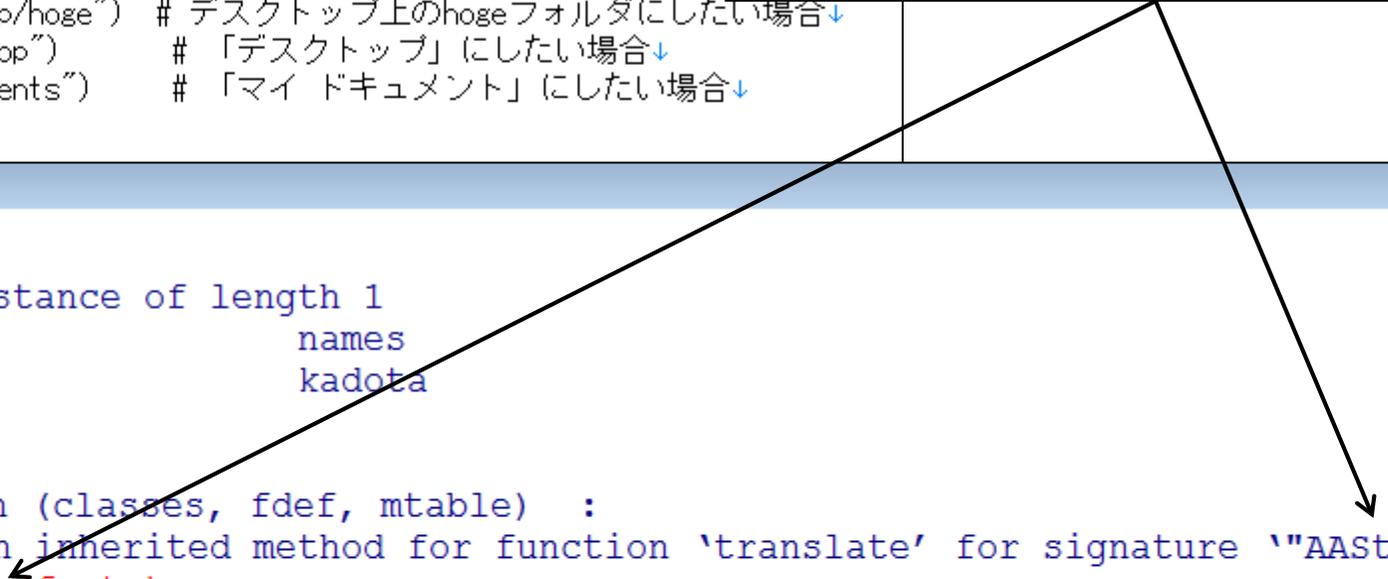
# データの型

ファイル名: rcode\_translate2.txt

「?translate」または「help(translate)」でも translate関数が受け付けるものの中に AAStringSetが含まれないと予想。入力として与える fastaオブジェクトやオプションのことを引数(Arguments)といいます。

```
##### ↓
### 作業ディレクトリの変更 ↓
##### ↓
setwd("C:/Users/kadota/Desktop/hoge") # デスクトップ上のhogeフォルダにしたい場合 ↓
#setwd("C:/Users/kadota/Desktop") # 「デスクトップ」にしたい場合 ↓
#setwd("C:/Users/kadota/Documents") # 「マイドキュメント」にしたい場合 ↓
↓
##### ↓
```

```
R Console
###
###
in
ou
↓
#
li
↓
#
fa
↓
#
ho
na
fa
fa
↓
#
wr
↓
> fasta
A AAStringSet instance of length 1
width seq          names
[1]      4 SDGL     kadota
>
> translate(fasta)
以下にエラー (function (classes, fdef, mtable) :
unable to find an inherited method for function 'translate' for signature "AAStringSet"
> hoge <- translate(fasta)
以下にエラー (function (classes, fdef, mtable) :
unable to find an inherited method for function 'translate' for signature "AAStringSet"
> hoge
A AAStringSet instance of length 1
width seq          names
[1]      4 SDGL     kadota
> |
```



```
> ?translate
starting httpd help server ... done
> |
```

```
translate {Biostrings}
```

## Translating DNA/RNA sequences

### Description

Functions for translating DNA or RNA sequences into amino acid sequences.

### Usage

```
## Translating DNA/RNA:
translate(x, genetic.code=GENETIC_CODE, if.fuzzy.codon="error")

## Extracting codons without translating them:
codons(x)
```

### Arguments

**x** A [DNAStringSet](#), [RNAStringSet](#), [DNAString](#), [RNAString](#), [MaskedDNAString](#) or [MaskedRNAString](#) object for translate.

A [DNAString](#), [RNAString](#), [MaskedDNAString](#) or [MaskedRNAString](#) object for codons.

**genetic.code** The genetic code to use for the translation of codons into Amino Acid letters. It must be represented as a named character vector of length 64 similar to predefined constant `GENETIC_CODE` i.e. it must contain 1-letter strings in the Amino Acid alphabet and its names must be identical to `names(GENETIC_CODE)`. The default value for `genetic.code` is `GENETIC_CODE` which represents The Standard Genetic Code. See `?AA_ALPHABET` for the Amino Acid alphabet and `?GENETIC_CODE` for The Standard Genetic Code and its known variants.

**if.fuzzy.codon** How fuzzy codons (i.e codon with IUPAC ambiguities) should be handled. Accepted values are:

translate関数が入力として受け付けるものはDNAStringSetやRNAStringSetなどであり、AAStringSetという記述がないことが分かります。UsageとArgumentsの記述でだいたい分かります。また、これらの記述を眺め、codons関数の存在に気づくことを通じて、幅を広げていきます。

# Contents

- 3-4. R Bioconductor I、2014/09/09 10:30-14:45、中級、実習
  - Tips: setwd関数を利用した効率的な作業ディレクトリの変更
  - データの型1: translate関数の入力情報(AAStringSet)
  - Tips: オブジェクトの消去
  - **データの型2: 翻訳配列取得のコードの中身を解説**
  - バージョン情報把握とバージョンアップ
  - バージョン違いの影響
    - 過去から現在: BiostringsパッケージのreadDNAStrngSet関数
    - 現在から未来: BSgenome.Hsapiens.UCSC.hg19パッケージでプロモーター配列取得
  - Bioconductor概観



塩基配列を読み込んでアミノ酸配列に翻訳するやり方を示します。

「ファイル」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

1. FASTA形式ファイル(sample1.fasta)の場合:

```

in_f <- "sample1.fasta" #入力ファイル名を指定してin_fに格納
out_f <- "hoge1.fasta" #出力ファイル名を指定してout_fに格納

#必要なパッケージをロード
library(Biostrings) #パッケージの読み込み

#入力ファイルの読み込み
fasta <- readDNASTringSet(in_f, format="fasta") #in_fで指定したファイルの読み込み

#本番
hoge <- translate(fasta) #fastaをアミノ酸配列に翻訳した結果をhogeに格納
names(hoge) <- names(fasta) #現状では翻訳した結果のオブジェクトhogeのdescript
fasta <- hoge #hogeの中身をfastaに格納
fasta #確認してるだけです

#ファイルに保存
writeXStringSet(fasta, file=out_f, format="fasta")
    
```

```

>kadota
AGTGACGGTCTT
    
```

in\_fで指定した入力ファイルを用いてreadDNASTringSet関数を用いて読み込んだ後のfastaオブジェクトは、translate関数が入力として受け付けるDNASTringSet形式であることが分かる

```

R Console
> in_f <- "sample1.fasta" #入力$
> out_f <- "hoge1.fasta" #出力$
>
> #必要なパッケージをロード
> library(Biostrings) #パッ$
>
> #入力ファイルの読み込み
> fasta <- readDNASTringSet(in_f, format="fast$
> fasta
A DNASTringSet instance of length 1
width seq names $
[1] 12 AGTGACGGTCTT kadota
> |
    
```

塩基配列を読み込んでアミノ酸配列に翻訳するやり方を示します。

「ファイル」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

## 1. FASTA形式ファイル(sample1.fasta)の場合:

```
in_f <- "sample1.fasta" #入力ファイル名を指定してin_fに格納
out_f <- "hoge1.fasta" #出力ファイル名を指定してout_fに格納

#必要なパッケージをロード
library(Biostrings) #パッケージの読み込み

#入力ファイルの読み込み
fasta <- readDNAStringSet(in_f, format="fasta") #in_fで指定したファイルの読み込み

#本番
hoge <- translate(fasta) #fastaをアミノ酸配列に翻訳した結果をhogeに格納
names(hoge) <- names(fasta) #現状では翻訳した結果のオブジェクトhogeのdescript
fasta <- hoge #hogeの中身をfastaに格納
fasta #確認してるだけです

#ファイルに保存
writeXStringSet(fasta, file=out_f, format="fasta")
```

```
>kadota
AGTGACGGTCTT
```

入力ファイルのdescription行の記述がnamesという場所、配列長がwidthという場所に格納されていることもわかる。

```
R Console
> in_f <- "sample1.fasta" #入力$
> out_f <- "hoge1.fasta" #出力$
>
> #必要なパッケージをロード
> library(Biostrings) #パッ$
>
> #入力ファイルの読み込み
> fasta <- readDNAStringSet(in_f, format="fast$
> fasta
A DNAStringSet instance of length 1
width seq names $
[1] 12 AGTGACGGTCTT kadota
> |
```

塩基配列を読み込んでアミノ酸配列に翻訳するやり方を示します。

「ファイル」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

## 1. FASTA形式ファイル(sample1.fasta)の場合:

```

in_f <- "sample1.fasta" #入力ファイル名を指定してin_fに格納
out_f <- "hoge1.fasta" #出力ファイル名を指定してout_fに格納

#必要なパッケージをロード
library(Biostrings) #パッケージの読み込み

#入力ファイルの読み込み
fasta <- readDNAStringSet(in_f, format="fasta") #in_fで指定したファイルの読み込み

#本番
hoge <- translate(fasta) #fastaをアミノ酸配列に翻訳した結果をhogeに格納
names(hoge) <- names(fasta) #現状では翻訳した結果のオブジェクトhogeのdescription
fasta <- hoge #hogeの中身をfastaに格納
fasta #確認してるだけです

#ファイルに保存
writeXStringSet(fasta, file=out_f, format="fasta")
    
```

```
>kadota
AGTGACGGTCTT
```

入力ファイルのdescription行の記述がnamesという関数、配列長がwidthという関数で取り出せることが分かる。ただしseq関数は、存在はするがイメージと異なり塩基配列は返さない…。

```

R Console
> fasta
  A DNAStringSet instance of length 1
    width seq          names
[1]    12 AGTGACGGTCTT   kadota
> names(fasta)
[1] "kadota"
> width(fasta)
[1] 12
> as.character(fasta)
      kadota
"AGTGACGGTCTT"
> seq(fasta)
[1] 1
> |
    
```

塩基配列を読み込んでアミノ酸配列に翻訳するやり方を示します。

「ファイル」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピペ。

## 1. FASTA形式ファイル(sample1.fasta)の場合:

```
in_f <- "sample1.fasta" #入力ファイル名を指定してin_fに格納
out_f <- "hoge1.fasta" #出力ファイル名を指定してout_fに格納

#必要なパッケージをロード
library(Biostrings) #パッケージの読み込み

#入力ファイルの読み込み
fasta <- readDNAStringSet(in_f, format="fasta") #in_fで指定したファイルの読み込み

#本番
hoge <- translate(fasta) #fastaをアミノ酸配列に翻訳した結果をhogeに格納
names(hoge) <- names(fasta) #現状では翻訳した結果のオブジェクトhogeのdescript
fasta <- hoge #hogeの中身をfastaに格納
fasta #確認してるだけです

#ファイルに保存
writeXStringSet(fasta, file=out_f, format="fasta")
```

```
>kadota
AGTGACGGTCTT
```

translate関数実行後のhogeオブジェクトは、AAStringSet形式に変わっていることがわかる

```
R Console
> #本番
> hoge <- translate(fasta)
> hoge
  A AAStringSet instance of length 1
    width seq          names
[1]     4 SDGL          kadota
>
```

塩基配列を読み込んでアミノ酸配列に翻訳するやり方を示します。

「ファイル」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

1. FASTA形式ファイル(sample1.fasta)の場合:

```
in_f <- "sample1.fasta" #入力ファイル名を指定してin_fに格納
out_f <- "hoge1.fasta" #出力ファイル名を指定してout_fに格納

#必要なパッケージをロード
library(Biostrings) #パッケージの読み込み

#入力ファイルの読み込み
fasta <- readDNAStringSet(in_f, format="fasta") #in_fで指定したファイルの読み込み

#本番
hoge <- translate(fasta) #fastaをアミノ酸配列に翻訳した結果をhogeに格納
names(hoge) <- names(fasta) #現状では翻訳した結果のオブジェクトhogeのdescript
fasta <- hoge #hogeの中身をfastaに格納
fasta #確認してるだけです

#ファイルに保存
writeXStringSet(fasta, file=out_f, format="fasta")
```

```
>kadota
AGTGACGGTCTT
```

少なくともR ver. 2.15.2までは必要だったが、講義資料作成時のR環境(R ver. 3.1.0とBioconductor ver. 2.14)では、この作業はもはや必要ない。

```
R Console
> #本番
> hoge <- translate(fasta)
> hoge
  A AAStringSet instance of length 1
    width seq          names
[1]     4 SDGL         kadota
>
> names(hoge) <- names(fasta) #現状$
> fasta <- hoge              #hoge$
> fasta                       #確認$
  A AAStringSet instance of length 1
    width seq          names
[1]     4 SDGL         kadota
> |
```

塩基配列を読み込んでアミノ酸配列に翻訳するやり方を示します。  
「ファイル」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

1. FASTA形式ファイル(sample1.fasta)の場合:

```
in_f <- "sample1.fasta" #入力ファイル名を指定してin_fに格納
out_f <- "hoge1.fasta" #出力ファイル名を指定してout_fに格納

#必要なパッケージをロード
library(Biostrings) #パッケージの読み込み

#入力ファイルの読み込み
fasta <- readDNASTringSet(in_f, format="fasta") #in_fで指定したファイルの読み込み

#本番
hoge <- translate(fasta) #fastaをアミノ酸配列に翻訳した結果をhogeに格納
names(hoge) <- names(fasta) #現状では翻訳した結果のオブジェクトhogeのdescription
fasta <- hoge #hogeの中身をfastaに格納
fasta #確認してるだけです

#ファイルに保存
writeXStringSet(fasta, file=out_f)
```

```
>kadota
AGTGACGGTCTT
```

NGS速習コース終了後に関連個所をこのように修正するか、バージョンの違いの影響を説明する資料として残すかは考え中

6. FASTA形式ファイル(sample1.fasta)の場合:

1と基本的と同じ結果が得られますが、AAStringSetオブジェクトでdescription行が消えてしまうバグが修正されたようなので、すっきりさせたものです。

```
in_f <- "sample1.fasta" #入力ファイル名を指定してin_fに格納
out_f <- "hoge6.fasta" #出力ファイル名を指定してout_fに格納

#必要なパッケージをロード
library(Biostrings) #パッケージの読み込み

#入力ファイルの読み込み
fasta <- readDNASTringSet(in_f, format="fasta") #in_fで指定したファイルの読み込み

#本番
fasta <- translate(fasta) #fastaをアミノ酸配列に翻訳した結果をfastaに格納
fasta #確認してるだけです

#ファイルに保存
writeXStringSet(fasta, file=out_f, format="fasta", width=50) #fastaの中身を指定したファ
```

塩基配列を読み込んでアミノ酸配列に翻訳するやり方を示します。

「ファイル」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

1. FASTA形式ファイル(sample1.fasta)の場合:

```
in_f <- "sample1.fasta" #入力ファイル名を指定してin_fに格納
out_f <- "hoge1.fasta" #出力ファイル名を指定してout_fに格納

#必要なパッケージをロード
library(Biostrings) #パッケージの読み込み

#入力ファイルの読み込み
fasta <- readDNAStringSet(in_f, format="fasta") #in_fで指定したファイルの読み込み

#本番
hoge <- translate(fasta) #fastaをアミノ酸配列に翻訳した結果をhogeに格納
names(hoge) <- names(fasta) #現状では翻訳した結果のオブジェクトhogeのdescription
fasta <- hoge #hogeの中身をfastaに格納
fasta #確認してるだけです

#ファイルに保存
writeXStringSet(fasta, file=out_f, format="fasta")
```

```
>kadota
AGTGACGGTCTT
```

講義資料作成時のR環境(R ver. 3.1.0とBioconductor ver. 2.14)では、この作業はもはや必要ないと2014/08/13に気づいたが、一連の手順作成当時はこうする必要があったのです。自分のR環境を把握しバージョンの違いに気をつけるべし!

```
R Console
> #本番
> hoge <- translate(fasta)
> hoge
  A AAStringSet instance of length 1
    width seq          names
[1]     4 SDGL         kadota
>
> names(hoge) <- names(fasta) #現状$
> fasta <- hoge              #hoge$
> fasta                       #確認$
  A AAStringSet instance of length 1
    width seq          names
[1]     4 SDGL         kadota
> |
```

# Contents

- 3-4. R Bioconductor I、2014/09/09 10:30-14:45、中級、実習
  - Tips: setwd関数を利用した効率的な作業ディレクトリの変更
  - データの型1: translate関数の入力情報(AAStringSet)
  - Tips: オブジェクトの消去
  - データの型2: 翻訳配列取得のコードの中身を解説
  - バージョン情報把握とバージョンアップ
  - バージョン違いの影響
    - 過去から現在: BiostringsパッケージのreadDNAStrngSet関数
    - 現在から未来: BSgenome.Hsapiens.UCSC.hg19パッケージでプロモーター配列取得
  - Bioconductor概観



# sessionInfo()で自分のR環境を把握

```
R Console
> sessionInfo()
R version 3.1.0 (2014-04-10)
Platform: x86_64-w64-mingw32/x64 (64-bit)

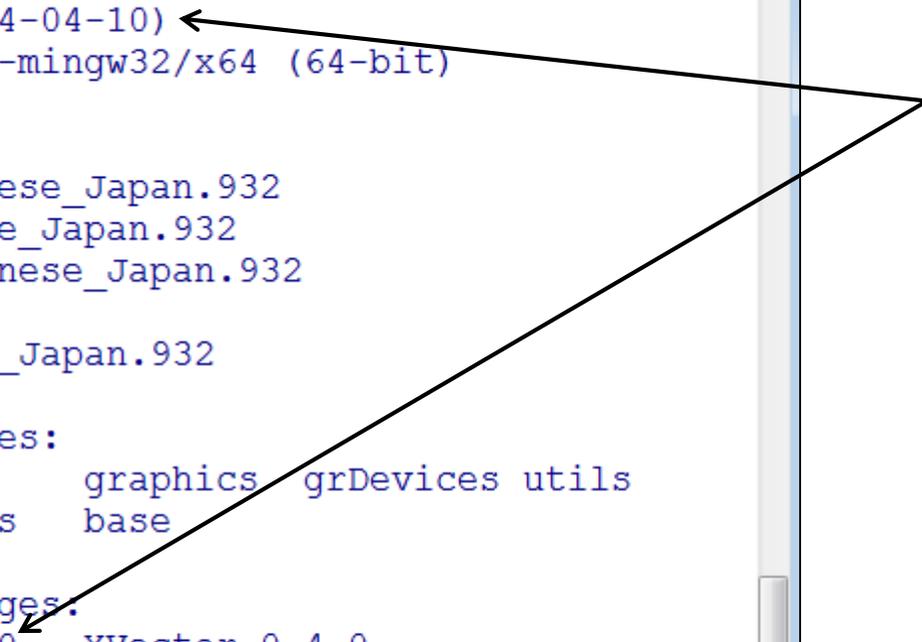
locale:
[1] LC_COLLATE=Japanese_Japan.932
[2] LC_CTYPE=Japanese_Japan.932
[3] LC_MONETARY=Japanese_Japan.932
[4] LC_NUMERIC=C
[5] LC_TIME=Japanese_Japan.932

attached base packages:
[1] parallel stats graphics grDevices utils
[6] datasets methods base

other attached packages:
[1] Biostrings_2.32.0 XVector_0.4.0
[3] IRanges_1.22.3 BiocGenerics_0.10.0

loaded via a namespace (and not attached):
[1] stats4_3.1.0 zlibbioc_1.10.0
> |
```

門田の解析環境は、R ver. 3.1.0であることがわかる。また、Biostringsパッケージのバージョンは2.32.0と読み取る。



# 投稿論文へのバージョン情報記載例

BMC Bioinformatics. 2013 Jul 9;14:219. doi: 10.1186/1471-2105-14-219.

## TCC: an R package for comparing tag count data with robust normalization strategies.

Sun J<sup>1</sup>, Nishiyama T, Shimizu K, Kadota K.

### Author information

#### Abstract

**BACKGROUND:** Differential expression analysis based on "next-generation" sequencing technologies is a fundamental means of studying RNA expression. We recently developed a multi-step normalization method (called TbT) for two-group RNA-seq data with replicates and demonstrated that the statistical methods available in four R packages (*edgeR*, *DESeq*, *baySeq*, and *NBPSeq*) together with TbT can produce a well-ranked gene list in which true differentially expressed genes (DEGs) are top-ranked and non-DEGs are bottom ranked. However, the advantages of the current TbT method come at the cost of a huge computation time. The current R packages did not have normalization methods based on such a multi-step strategy.

**RESULTS:** TCC (an acronym for Tag Count Comparison) is an R package that provides functions for differential expression analysis of tag count data. The package includes normalization methods, whose strategy is to remove potential DEGs before performing normalization. The normalization function based on this DEG elimination strategy consists of (i) the original TbT method based on DEGES for two-group data with or without replicates, (ii) faster methods for two-group data with or without replicates, and (iii) methods for comparison. TCC provides a simple unified interface to perform such analyses with the functions provided by *edgeR*, *DESeq*, and *baySeq*. Additionally, a function for gene selection under various conditions and alternative DEGES procedures consisting of functions from existing packages are provided. Bioinformatics scientists can use TCC to evaluate their data and biologists familiar with other R packages can easily learn what is done in TCC.

**CONCLUSION:** DEGES in TCC is essential for accurate normalization of tag count data. When up- and down-regulated DEGs in one of the samples are extremely biased, TCC is useful for analyzing tag count data in various scenarios ranging from unbalanced to biased differential expression. TCC is available at <http://www.iu.a.u-tokyo.ac.jp/~kadota/TCC/> and will appear in Bioconductor (<http://bioconductor.org/>) from ver. 2.13.

R ver. 3.1.0の場合は、ここを ver. 3.1.0に変更すればよい

the functions implemented in *edgeR*, *DESeq*, and *baySeq*. Here, we demonstrate that the DEGES-based normalization methods are more effective than the methods implemented in the other packages. All analyses were performed using R (ver. 2.15.2) and Bioconductor [20]. Execution times were measured on a Linux system (CentOS release 6.2 (Final), Intel® Xeon® E5-4617 (2.9 GHz) 24 CPU, and 512 GB memory). The versions of major R libraries were TCC ver. 1.1.99, *edgeR* ver. 3.0.4, *DESeq* ver. 1.10.1, and *baySeq* ver. 1.12.0.

# 投稿論文へのバージョン情報記載例

BMC Bioinformatics, 2013 Jul 9;14:219. doi: 10.1186/1471-2105-14-219.

## TCC: an R package for comparing tag count data with robust normalization strategies.

Sun J<sup>1</sup>, Nishiyama T, Shimizu K, Kadota K.

### Author information

#### Abstract

**BACKGROUND:** Differential expression analysis based on "next-generation" sequencing technologies is a fundamental means of studying RNA expression. We recently developed a multi-step normalization method (called TbT) for two-group RNA-seq data with replicates and demonstrated that the statistical methods available in four R packages (*edgeR*, *DESeq*, *baySeq*, and *NBPSeg*) together with TbT can produce a well-ranked gene list in which true differentially expressed genes (DEGs) are top-ranked and non-DEGs are bottom ranked. However, the advantages of the current TbT method come at the cost of a huge computation time. The other R packages did not have normalization methods based on such a multi-step strategy.

**RESULTS:** TCC (an acronym for Tag Count Comparison) is an R package that provides functions for differential expression analysis of tag count data. The package includes normalization methods, whose strategy is to remove potential DEGs before performing normalization. The normalization function based on this DEG elimination strategy consists of (i) the original TbT method based on DEGES for two-group data with or without replicates, (ii) faster methods for two-group data with or without replicates, and (iii) methods for comparison.

TCC provides a simple unified interface to perform such analyses via functions provided by *edgeR*, *DESeq*, and *baySeq*. Additionally, a function for gene selection under various conditions and alternative DEGES procedures consisting of functions from existing packages are provided. Bioinformatics scientists can use TCC to evaluate their methods and biologists familiar with other R packages can easily learn what is done in TCC.

**CONCLUSION:** DEGES in TCC is essential for accurate normalization of tag count data when up- and down-regulated DEGs in one of the samples are extremely biased. TCC is useful for analyzing tag count data in various scenarios ranging from unbalanced to unbiased differential expression. TCC is available at <http://www.iu.a.u-tokyo.ac.jp/~kadota/TCC/> and will appear in Bioconductor (<http://bioconductor.org/>) from ver. 2.13.

「R and Bioconductor」や「R/Bioconductor」など、2大リポジトリの一つであるBioconductorがセットで記述される場合が多いのは、TCCやBiostringsパッケージを含むバイオインフォ系パッケージの多くがBioconductorから配布されているから。

the functions implemented in *edgeR*, *DESeq*, and *baySeq*. Here, we demonstrate that the DEGES-based normalization methods are more effective than the methods implemented in the other packages. All analyses were performed using R (ver. 2.15.2) and Bioconductor [20]. Execution times were measured on a Linux system (CentOS release 6.2 (Final), Intel® Xeon® E5-4617 (2.9 GHz) 24 CPU, and 512 GB memory). The versions of major R libraries were TCC ver. 1.1.99, edgeR ver. 3.0.4, DESeq ver. 1.10.1, and baySeq ver. 1.12.0.

# 定期的なバージョンアップをお勧め

## ■ R本体もBioconductorも定期的にバージョンアップがなされている

### □ R (<http://www.r-project.org/>)

- 2014-07-10にver. 3.1.1をリリース
- 2014-04-10にver. 3.1.0をリリース
- 2014-03-06にver. 3.0.3をリリース
- ...
- 2012-03-30にver. 2.15.0をリリース
- ...

2014年8月14日現在のR本体の最新バージョンは3.1.1、Biostrings最新バージョンは2.32.1。しかし、R ver. 2.15.0など本体のバージョンが古いと、2014年8月14日にBiostringsパッケージを個別にインストールしてもver. 2.26.3と昔のバージョンがインストールされる点に注意が必要です。

### □ Bioconductor (<http://bioconductor.org/>)は半年ごとにリリース

- 2014-04にver. 2.14をリリース (R ver. 3.1.0で動作確認)、提供パッケージ数: 824
- 2013-10にver. 2.13をリリース (R ver. 3.0で動作確認)、提供パッケージ数: 750
- 2013-04にver. 2.12をリリース (R ver. 3.0で動作確認)、提供パッケージ数: 672
- 2012-10にver. 2.11をリリース (R ver. 2.15.1で動作確認)、提供パッケージ数: 608
- 2012-04にver. 2.10をリリース (R ver. 2.15.0で動作確認)、提供パッケージ数: 553
- 2011-11にver. 2.9をリリース (R ver. 2.14.0で動作確認)、提供パッケージ数: 517
- 2011-04にver. 2.8をリリース (R ver. 2.13.0で動作確認)、提供パッケージ数: 466



## What's new?

- 門田幸二 著 [シリーズ Useful R 第7巻トランスクリプトーム解析](#)刊行(共立出版)や、ROKU法 (Kadota et al., 2006)、WAD法 (Kadota et al., 2008)などについてレイ解析部分のRコードについては、このページの「書籍|トランスクリプトーム
- お知らせは主に([Rで](#))塩基配列解析で行っておりますのでそちらをご覧ください。ども([Rで](#))塩基配列解析中の[参考資料\(講義、講習会、本など\)](#)から辿れます。

- [はじめに](#) (last modified 2014/05/14)
- [過去のお知らせ](#) (last modified 2014/03/03)
- [Rのインストールと起動](#) (last modified 2014/05/14)
- [Rの昔のバージョンのインストール](#) (last modified 2012/04/07)
- [使用例\(初心者向け\)](#) (last modified 2011/09/15)
- [サンプルデータ](#) (last modified 2014/05/02)

## Rの昔のバージョンのインストール

[DFW \(Chen et al., Bioinformatics, 2007\)](#) というAffyの"階層的クラスタリング結果を導くような遺伝子データ用の正規化法(嫌味ではなくいい方法)なんは、R2.7.2あたりだと正常に動作していましたが、く動いてくれません。そのような場合でも、Rの昔ることによって、DFWを利用することができます。の昔のバージョンをインストールするやり方をR2.

1. [ここ](#)をクリックして、任意のRのバージョンの
2. R-X.Y.Z-win32.exe(例えば [R-2.7.2-win32](#)、がままに押す

## Previous Releases of R for Windows

This directory contains previous binary releases of R to run on Windows 95, 98, ME, NT4.0, 2000 and XP or later on Intel/clone chips.

The current release, and links to development snapshots, are available [here](#). Source code for these releases and others is available through [the main CRAN page](#).

In this directory:

- [R 3.1.0](#) (April, 2014)
- [R 3.0.3](#) (March, 2014)
- [R 3.0.2](#) (September, 2013)
- [R 3.0.1](#) (May, 2013)
- [R 3.0.0](#) (April, 2013)
- [R 2.15.3](#) (March, 2013)
- [R 2.15.2](#) (October, 2012)
- [R 2.15.1](#) (June, 2012)
- [R 2.15.0](#) (March, 2012)
- [R 2.14.2](#) (February, 2012)
- [R 2.14.1](#) (December, 2011)
- [R 2.14.0](#) (November, 2011)
- [R 2.13.2](#) (September, 2011)
- [R 2.13.1](#) (July, 2011)
- [R 2.13.0](#) (April, 2011)
- [R 2.12.2](#) (February, 2011)
- [R 2.12.1](#) (December, 2010)
- [R 2.12.0](#) (October, 2010)
- [R 2.11.1](#) (May, 2010)
- [R 2.11.0](#) (April, 2010)
- [R 2.10.1](#) (December, 2009)
- [R 2.10.0](#) (October, 2009)

Rの昔のバージョンのインストール手順。Windows版のver. 2.15.0の場合。



```
> library(Biostrings)
```

```
要求されたパッケージ BiocGenerics をロード中です
```

```
次のパッケージを付け加えます: '`BiocGenerics`'
```

```
The following object(s) are masked from `package:stats`:  
xtabs
```

```
The following object(s) are masked from `package:base`:  
anyDuplicated, cbind, colnames, duplicated, eval,  
get, intersect, lapply, Map, mapply, mget, order,  
pmax.int, pmin, pmin.int, Position, rbind, Reduce,  
rownames, sapply, setdiff, table, tapply, union, u
```

```
要求されたパッケージ IRanges をロード中です
```

```
警告メッセージ:
```

- 1: パッケージ '`Biostrings`' はバージョン 2.15.2 の R の下で造られました
- 2: パッケージ '`BiocGenerics`' はバージョン 2.15.1 の R の下で造られました
- 3: パッケージ '`IRanges`' はバージョン 2.15.2 の R の下で造られました

R ver. 2.15.0上で、Biostringsパッケージのみを通常手順でインストール。無事インストールできたようなので、library関数を用いてBiostringsパッケージを読み込んでいるところ。警告メッセージは「あなたの環境はR ver. 2.15.0だけど、インストールしたBiostringsパッケージはR ver. 2.15.2環境下で作られたものです」と言っています。library(Biostrings)でパッケージのロードに失敗する場合には本気で対処する必要がありますが、警告メッセージが出ているだけです。私は特に気にしていません。

```
> sessionInfo()
R version 2.15.0 (2012-03-30)
Platform: x86_64-pc-mingw32/x64 (64-bit)

locale:
[1] LC_COLLATE=Japanese_Japan.932
[2] LC_CTYPE=Japanese_Japan.932
[3] LC_MONETARY=Japanese_Japan.932
[4] LC_NUMERIC=C
[5] LC_TIME=Japanese_Japan.932

attached base packages:
[1] stats      graphics  grDevices  utils      datasets
[6] methods    base

other attached packages:
[1] Biostrings_2.26.3  IRanges_1.16.6
[3] BiocGenerics_0.4.0 BiocInstaller_1.8.3

loaded via a namespace (and not attached):
[1] parallel_2.15.0 stats4_2.15.0  tools_2.15.0
> date()
[1] "Thu Aug 14 14:38:54 2014"
> |
```

2014年8月14日現在のR本体の最新バージョンは3.1.1、Biostrings最新バージョンは2.32.1。しかし、R ver. 2.15.0など本体のバージョンが古いと、2014年8月14日にBiostringsパッケージを個別にインストールしてもver. 2.26.3と昔のバージョンがインストールされる点に注意が必要です。

# 定期的なバージョンアップをお勧め

## ■ R本体もBioconductorも定期的にバージョンアップがなされている

### □ R (<http://www.r-project.org/>)

- 2014-07-10にver. 3.1.1をリリース
- 2014-04-10にver. 3.1.0をリリース
- 2014-03-06にver. 3.0.3をリリース
- ...
- 2012-03-30にver. 2.15.0をリリース
- ...

基本的にはバグの修正や新たな機能がどんどん追加されているので最新版の利用をお勧め。毎年5月初めと11月初めごろにBioconductorを覗きにいくとよいだろう。

### □ Bioconductor (<http://bioconductor.org/>)は半年ごとにリリース

- 2014-04にver. 2.14をリリース (R ver. 3.1.0で動作確認)、提供パッケージ数: 824
- 2013-10にver. 2.13をリリース (R ver. 3.0で動作確認)、提供パッケージ数: 750
- 2013-04にver. 2.12をリリース (R ver. 3.0で動作確認)、提供パッケージ数: 672
- 2012-10にver. 2.11をリリース (R ver. 2.15.1で動作確認)、提供パッケージ数: 608
- 2012-04にver. 2.10をリリース (R ver. 2.15.0で動作確認)、提供パッケージ数: 553
- 2011-11にver. 2.9をリリース (R ver. 2.14.0で動作確認)、提供パッケージ数: 517
- 2011-04にver. 2.8をリリース (R ver. 2.13.0で動作確認)、提供パッケージ数: 466



# バージョンアップの意義：具体例

BMC Bioinformatics. 2013 Jul 9;14:219. doi: 10.1186/1471-2105-14-219.

## TCC: an R package for comparing tag count data with robust normalization strategies.

Sun J<sup>1</sup>, Nishiyama T, Shimizu K, Kadota K.

### Author information

#### Abstract

**BACKGROUND:** Differential expression analysis based on "next-generation" sequencing technologies is a fundamental means of studying RNA expression. We recently developed a multi-step normalization method (called TbT) for two-group RNA-seq data with replicates and demonstrated that the statistical methods available in four R packages (edgeR, DESeq, baySeq, and NBPSeg) together with TbT can produce a well-ranked gene list in which true differentially expressed genes (DEGs) are top-ranked and non-DEGs are bottom ranked. However, the advantages of the current TbT method come at the cost of a huge computation time. Moreover, the R packages did not have normalization methods based on such a multi-step strategy.

**RESULTS:** TCC (an acronym for Tag Count Comparison) is an R package that provides a series of functions for differential expression analysis of tag count data. The package incorporates multi-step normalization methods, whose strategy is to remove potential DEGs before performing the data normalization. The normalization function based on this DEG elimination strategy (DEGES) includes (i) the original TbT method based on DEGES for two-group data with or without replicates, (ii) much faster methods for two-group data with or without replicates, and (iii) methods for multi-group comparison. TCC provides a simple unified interface to perform such analyses with combinations of functions provided by edgeR, DESeq, and baySeq. Additionally, a function for generating simulation data under various conditions and alternative DEGES procedures consisting of functions in the existing packages are provided. Bioinformatics scientists can use TCC to evaluate their methods, and biologists familiar with other R packages can easily learn what is done in TCC.

**CONCLUSION:** DEGES in TCC is essential for accurate normalization of tag count data, especially when up- and down-regulated DEGs in one of the samples are extremely biased in their number. TCC is useful for analyzing tag count data in various scenarios ranging from unbiased to extremely biased differential expression. TCC is available at <http://www.iu.a.u-tokyo.ac.jp/~kadota/TCC/> and will appear in Bioconductor (<http://bioconductor.org/>) from ver. 2.13.

2013年7月の論文publish以降も継続的にアップデートしています:

- 多群間比較やpaired dataへの対応など、解析可能な実験デザインを拡張
- DESeq2対応もほぼ完了
- サンプル間クラスタリング用関数やマイクロアレイデータ用組織特異的発現パターン検出法ROKUの実装
- ドキュメントが充実(TCC ver. 1.4.0で74ページに!)

- 解析 | 発現変動 | について (last modified 2014/07/10) NEW
- 解析 | 発現変動 | 2群間 | 対応なし | について (last modified 2014/07/10) NEW
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | TCC (Sun 2013) (last modified 2014/07/10) NEW

解析 | 発現変動 | 2群間 | 対応なし | について NEW

- 実験データ
- 解析 Aさん
- 解析 Bさん
- 解析 Cさん
- 解析 Dさん
- 解析 Eさん
- 解析 Fさん
- 解析 Gさん

R用:

- [DEGSeq: Wang et al., Bioinformatics, 2010](#)
- [edgeR: Robinson et al., Bioinformatics, 2010](#)
- [GPseq: Srivastava et al., Nucleic Acid Res, 2012](#)
- [baySeq: Hardcastle and Kelly, BMC Bioinformatics, 2010](#)
- [DESeq: Anders and Huber, Genome Biology, 2010](#)
- [DESeq2: Anders and Huber, Genome Biology, 2014](#)
- [NBPSeq: Di et al., SAGMB, 2011](#)
- [BBSeq: Zhou et al., Bioinformatics, 2012](#)
- [NOISeq: Tarazona et al., Genome Res, 2012](#)
- [PoissonSeq: Li et al., Biostatistics, 2010](#)
- [SAMseq: Li and Tibshirani, Stat Meth, 2012](#)
- [easyRNASeq: Delhomme et al., Bioinformatics, 2012](#)
- [DSGseq: Wang et al., Gene, 2013](#)
- [sSeq: Yu et al., Bioinformatics, 2013](#)
- [TCC: Sun et al., BMC Bioinformatics, 2013](#)
- [tweeDEseq: Esnaola et al., BMC Bioinformatics, 2013](#)
- [NPEBseq: Bi et al., BMC Bioinformatics, 2013](#)
- [DER Finder: Frazee et al., Biostatistics, 2013](#)
- [Characteristic Direction\(CD\): Clark et al., BMC Bioinformatics, 2013](#)
- [edgeR-robust: Zhou et al., Nucleic Acid Res, 2013](#)
- [ShrinkBayes: Van De Wiel et al., BMC Bioinformatics, 2013](#)

2014年

R用:



Home Install Help

Home » [Bioconductor 2.14](#) » [Software Packages](#) » [TCC](#)

# TCC

**TCC: Differential expression analysis for tag count data with robust normalization strategies**

Bioconductor version: Release (2.14)

This package provides a series of functions for performing differential expression analysis from RNA-seq count data using robust normalization strategy (called DEGES). The basic idea of DEGES is that potential differentially expressed genes or transcripts (DEGs) among compared samples should be removed before data normalization to obtain a well-ranked gene list where true DEGs are top-ranked and non-DEGs are bottom ranked. This can be done by performing a multi-step normalization strategy (called DEGES for DEG elimination strategy). A major characteristic of TCC is to provide the robust normalization methods for several kinds of count data (two-group with or without replicates, multi-group/multi-factor, and so on) by virtue of the use of combinations of functions in depended packages.

Author: Jianqiang Sun, Tomoaki Nishiyama, Kentaro Shimizu, and Koji Kadota

Maintainer: Jianqiang Sun <wukong at bi.a.u-tokyo.ac.jp>, Tomoaki Nishiyama <tomoakin at staff.kanazawa-u.ac.jp>

Citation (from within R, enter `citation("TCC")`):



# TCC

## TCC: Differential expression normalization strategies

Bioconductor version: Release (2.14.0)

This package provides a series of functions for normalizing count data using robust normalization methods for differentially expressed genes or data normalization to obtain a weighted bottom ranked. This can be done for several kinds of count data (on) by virtue of the use of comb

Author: Jianqiang Sun, Tomoaki

Maintainer: Jianqiang Sun <wukstaff.kanazawa-u.ac.jp>

### Documentation

To view documentation for the version of this package

browseVignettes("TCC")		
<a href="#">PDF</a>	<a href="#">R Script</a>	TCC
<a href="#">PDF</a>		Reference Manual
<a href="#">Text</a>		NEWS

### Details

biocViews	<a href="#">DifferentialExpression</a> , <a href="#">RNAS</a>
Version	1.4.0
In Bioconductor since	BioC 2.13 (R-3.0)
License	GPL-2
Depends	R (>= 2.15), methods, <a href="#">DESeq2</a>
Imports	<a href="#">samr</a>
Suggests	<a href="#">RUnit</a> , <a href="#">BiocGenerics</a>
System Requirements	
URL	
Depends On Me	
Imports Me	
Suggests Me	<a href="#">compcodeR</a>

### CHANGES IN VERSION 1.3.2

- DESeq2 was implemented in TCC for identifying DEGs.
- EBSeq was removed from TCC.
- arguments of 'WAD' function were changed.
- fixed bug in '.testByDeseq' that missing to treat size factors.
- the strategies for DE analysis of paired two-group dataset were implemented.
- add the section for describing DE analysis of paired two-group dataset into vignette.

### CHANGES IN VERSION 1.2.0

- this package was released as a Bioconductor package (previously CRAN).
- WAD method for identifying DEGs was added.
- ROKU method for identifying tissue-specific genes was added.
- 'increment' argument of 'calcNormFactor' function was added.
- 'replicates' field of TCC class was deleted.

### CHANGES IN VERSION 1.1.3

- 'generateSimulationData' function was renamed to 'simulateReadCount'.
- 'names' field of TCC class was changed to 'gene\_id'.
- 'hypoData' was reduced to a smaller data set.
- 'hypoData\_mg' was created. This is the simulation dataset which consists of 1,000 genes and 9 samples.

### CHANGES IN VERSION 1.0.0

- 'TCC' class was implemented as a R5 reference class. Wrapper functions with functional programming semantics were provided.

NEWSのところは、バージョンアップで主に何が変わったかが簡潔に記述されている。例えばTCC ver. 1.1.3でシミュレーションデータ作成用のgenerateSimulationData関数をsimulateReadCountという名前に変更していることがわかる。



TCC ver. 1.4.0で利用可能な関数名を一覧する場合はReference ManualのPDFをクリック。

Home » [Bioconductor 2.14](#) » [Software Packages](#) » TCC

# TCC

## TCC: Differential expression normalization strategies

Bioconductor version: Released  
 This package provides a series of methods for normalizing count data using robust normalization strategies to identify differentially expressed genes or data normalization to obtain a weighted bottom ranked. This can be done using a method (DEG elimination strategy). A method is available for several kinds of count data (e.g., RNA-seq) by virtue of the use of combination of methods.  
 Author: Jianqiang Sun, Tomoaki Kanazawa  
 Maintainer: Jianqiang Sun <wukun@staff.kanazawa-u.ac.jp>

**Documentation**

To view documentation for the version of this package installed in your system, start R and enter:

```
browseVignettes("TCC")
```

[PDF](#) [R Script](#) TCC  
[PDF](#) Reference Manual  
[Text](#) NEWS

**Details**

biocViews [DifferentialExpression](#), [RNASeq](#), [Sequencing](#), [Software](#)  
 Version 1.4.0  
 In Bioconductor since BioC 2.13 (R-3.0)  
 License GPL-2  
 Depends R (>= 2.15), methods, [DESeq](#), [DESeq2](#), [edgeR](#), [baySeq](#), [ROC](#)  
 Imports [samr](#)  
 Suggests [RUnit](#), [BiocGenerics](#)  
 System Requirements  
 URL  
 Depends On Me  
 Imports Me  
 Suggests Me [compcoder](#)



To view documentation for the version of this package installed in your system, start R and enter:

これが関数一覧

browseVignettes ("TC

PDF R Script TCC  
PDF Ref  
Text NE

**Details**

biocViews D  
Version 1.  
In Bioconductor since B  
License G  
Depends R  
Imports se  
Suggests R  
System Requirements  
URL  
Depends On Me  
Imports Me  
Suggests Me cc

http://bioconductor.org/packages/release/bioc/manuals/TCC/man/TCC.pdf

bioconductor.org

1 / 33 61.7%

ツール 署名 注釈

しおり

- arab
- calcAUCValue
- calcNormFactors
- clusterSample
- do\_TbT
- estimateDE
- exactTestafterTbT
- filterLowCountGenes
- getNormalizedData
- getResult
- hypoData
- hypoData\_mg
- hypoData\_ts
- MAplot
- nakai
- NBsample
- plot
- plotFCPseudocolor
- ROKU
- simulateReadCounts
- TCC
- TCC-class
- WAD
- Index

### Package 'TCC'

August 13, 2014

**Type** Package

**Title** TCC: Differential expression analysis for tag count data with robust normalization strategies

**Version** 1.4.0

**Author** Jianqiang Sun, Tomoaki Nishiyama, Kentaro Shimizu, and Koji Kadota

**Maintainer** Jianqiang Sun <wukong@bi.i.u.-tokyo.ac.jp>, Tomoaki Nishiyama <tomoaki.in@staff.kanazawa-u.ac.jp>

**Description** This package provides a series of functions for performing differential expression analysis from RNA-seq count data using robust normalization strategy (called DEGES). The basic idea of DEGES is that potential differentially expressed genes or transcripts (DEGs) among compared samples should be removed before data normalization to obtain a well-ranked gene list where true DEGs are top-ranked and non-DEGs are bottom ranked. This can be done by performing a multi-step normalization strategy (called DEGES for DEG elimination strategy). A major characteristic of TCC is to provide the robust normalization methods for several kinds of count data (two-group with or without replicates, multi-group/multi-factor, and so on) by virtue of the use of combinations of functions in depended packages.

**Depends** R (>= 2.15), methods, DESeq, DESeq2, edgeR, baySeq, ROC

**Imports** samr

**Suggests** RUnit, BiocGenerics

**Enhances** snow

**License** GPL-2

**Copyright** Authors listed above

**biocViews** Sequencing, DifferentialExpression, RNASeq

1

2 arab

To view documentation for the version of this package installed in your system, start R and enter:

The screenshot shows a web browser displaying the Bioconductor manual for the `NBSample` function. The browser address bar shows the URL: `http://bioconductor.org/packages/release/bioc/manuals/TCC/man/TCC.pdf`. The left sidebar contains a list of functions, with `NBSample` highlighted and a red arrow pointing to it. The main content area displays the function's description, usage, arguments, and examples.

**Yellow Callout Box:** TCCに限らず、(開発者側の論理で)現在は非推奨の関数が残っている場合がある。大抵、赤の下線部分のように消去予定だという通告か、こっちを使え的なアドバイスが書かれています。いつのリリースで消すかは開発者次第。

**Function Details:**

- Description:** This methods allow sampling from Negative Binomially differentially expressed genes having specified level of differential expression in terms of fold change. The proportion of upregulated are also specified. The distribution of original expression levels are generated by resampling real data of *Arabidopsis* RNA-seq data from `arab`. This function will be obsolete. Use `simulateReadCounts` instead.
- Usage:**

```
NBSample(DEG_foldchange = 4, repA = 3, repB = 3,
         Ngene = 3000, PDEG = 0.15, PA = 0.2)
```
- Arguments:**
  - `DEG_foldchange`: Fold change value of differentially expressed genes
  - `repA`: Replicate number for sample A
  - `repB`: Replicate number for sample B
  - `Ngene`: Number of genes to produce
  - `PDEG`: Proportion of differentially expressed genes
  - `PA`: Proportion of upregulated genes in sample A among differentially expressed genes (DEGs)
- Examples:**

```
## Not run:
sample <- NBSample()

## End(Not run)
```

```
> library(TCC)
> ?NBsample
> |
```

```
# TCCパッケージをロード
# NBsample関数のマニユ
```

Reference Manual中の記述と「?関数名」で得られる記述は同じです。

NBsample {TCC}

R Documentation

## Sampling from negative binomial distribution

### Description

This methods allow sampling from Negative Binomial distribution with specified proportion of differentially expressed genes having specified level of differential expression in terms of fold change. The proportion of upregulated are also specified. The distribution of original expression levels are generated by resampling real data of *Arabidopsis* RNA-seq data from [arab](#). This function will be obsoleted. Use [simulateReadCounts](#) instead.

### Usage

```
NBsample(DEG_foldchange = 4, repA = 3, repB = 3,
         Ngene = 3000, PDEG = 0.15, PA = 0.2)
```

### Arguments

DEG_foldchange	Fold change value of differentially expressed genes
repA	Replicate number for sample A
repB	Replicate number for sample B
Ngene	Number of genes to produce
PDEG	Proportion of differentially expressed genes
PA	Proportion of upregulated genes in sample A among differentially expressed genes (DEGs)

### Examples

```
## Not run:
sample <- NBsample()

## End(Not run)
```

# Contents

- 3-4. R Bioconductor I、2014/09/09 10:30-14:45、中級、実習
  - Tips: setwd関数を利用した効率的な作業ディレクトリの変更
  - データの型1: translate関数の入力情報(AAStringSet)
  - Tips: オブジェクトの消去
  - データの型2: 翻訳配列取得のコードの中身を解説
  - バージョン情報把握とバージョンアップ
  - **バージョン違いの影響**
    - 過去から現在: BiostringsパッケージのreadDNAStringSet関数
    - 現在から未来: BSgenome.Hsapiens.UCSC.hg19パッケージでプロモーター配列取得
  - Bioconductor概観



# バージョン違いの影響

## ■ 過去から現在: Biostringsパッケージ

### イントロ | 一般 | [翻訳配列\(translate\)を取得](#) **NEW**

(Rで)塩基配列解析中のコードは比較的新しいR環境を想定しています。つまり定期的なアップデートを行っているという前提。

塩基配列を読み込んでアミノ酸配列に翻訳するやり方を示します。  
「ファイル」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

#### 1. FASTA形式ファイル([sample1.fasta](#))の場合:

```

in_f <- "sample1.fasta"      #入力ファイル名を指定してin_fに格納
out_f <- "hoge1.fasta"      #出力ファイル名を指定してout_fに格納

#必要なパッケージをロード
library(Biostrings)        #パッケージの読み込み

#入力ファイルの読み込み
fasta <- readDNASTringSet(in_f, format="fasta")#in_fで指定したファイルの読み込み

#本番
hoge <- translate(fasta)    #fastaをアミノ酸配列に翻訳した結果をhogeに格納
names(hoge) <- names(fasta) #現状では翻訳した結果のオブジェクトhogeのdescript
fasta <- hoge               #hogeの中身をfastaに格納
fasta                      #確認してるだけです

#ファイルに保存
writeXStringSet(fasta, file=out_f, format="fasta", width=50)#fastaの中身を指定したファ
    
```



# バージョン違いの影響

## ■ 過去から現在 : Biostringsパッケージ

### イントロ | 一般 | 翻訳配列(translate)を取得 **NEW**

2010年10月リリースのR ver. 2.12.0で実行するとreadDNAStringSet関数のところでエラーが出ます。

塩基配列を読み込んでアミノ酸配列に翻訳するやり方を示します。

「ファイル」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し、以下をコピー。

#### 1. FASTA形式ファイル(sample1.fasta)の場合:

```

in_f <- "sample1.fasta"      #入力ファイル名
out_f <- "hoge1.fasta"      #出力ファイル名

#必要なパッケージをロード
library(Biostrings)        #Biostringsパッケージをロード

#入力ファイルの読み込み
fasta <- readDNAStringSet(in_f, format="fasta")

#本番
hoge <- translate(fasta)    #アミノ酸配列に翻訳
names(hoge) <- names(fasta) #アミノ酸配列の名前をFASTAファイルの名前に変更
fasta <- hoge               #アミノ酸配列をfastaオブジェクトに代入

#ファイルに保存
writeXStringSet(fasta, file=out_f, format="fasta")
    
```

```

R Console
R version 2.12.0 (2010-10-15)
Copyright (C) 2010 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-pc-mingw32/x64 (64-bit)

Rは、自由なソフトウェアであり、「完全に無保証」です。
一定の条件に従えば、自由にこれを再配布することができます。
配布条件の詳細に関しては、'license()'あるいは'licen$

Rは多くの貢献者による共同プロジェクトです。
詳しくは'contributors()'と入力してください。
また、RやRのパッケージを出版物で引用する際の形式に$
'citation()'と入力してください。

'demo()'と入力すればデモをみることができます。
'help()'とすればオンラインヘルプが出ます。
'help.start()'でHTMLブラウザによるヘルプがみられま$
'q()'と入力すればRを終了します。
    
```



# バージョン違いの影響

## ■ 過去から現在: Biostringsパッケージ

### イントロ | 一般 | 翻訳配列(translate)を取得 **NEW**

理由は、R ver. 2.12.0の頃はreadDNAStrngSetという関数名ではなくread.DNAStrngSetだったからです。

塩基配列を読み込んでアミノ酸配列に翻訳するやり方を示します。

「ファイル」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

#### 1. FASTA形式ファイル(sample1.fasta)

```
in_f <- "sample1.fasta"
out_f <- "hoge1.fasta"
```

```
#必要なパッケージをロード
library(Biostrings)
```

```
#入力ファイルの読み込み
fasta <- readDNAStrngSet(in_f)
```

```
#本番
hoge <- translate(fasta)
names(hoge) <- names(fasta)
fasta <- hoge
fasta
```

```
#ファイルに保存
writeXStringSet(fasta, out_f)
```

```
R Console
> fasta <- readDNAStrngSet(in_f, format="fasta") #in_fで指定したフ$
エラー: 関数 "readDNAStrngSet" を見つけることができませんでした
>
> #本番
> hoge <- translate(fasta) #fastaをアミノ酸配列に翻訳$
以下にエラー translate(fasta) :
  引数 'x' をみつけられませんでした (関数 'translate' に対するメソ$)
> names(hoge) <- names(fasta) #現状では翻訳した結果のオ$
エラー: オブジェクト 'fasta' がありません
> fasta <- hoge #hogeの中身をfastaに格納
エラー: オブジェクト 'hoge' がありません
> fasta #確認してるだけです
エラー: オブジェクト 'fasta' がありません
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", width=50) #fast$
エラー: 関数 "writeXStringSet" を見つけることができませんでした
>
```

# バージョン違いの影響

## ■ 過去から現在 : Biostringsパッケージ

search()で表示されるパッケージリストには、確かにBiostringsが存在する。?readDNAStringSetでNo documentationとなる一方で、?read.DNAStringSetではマニュアルがちゃんと開く。

```
R Console
> search()
[1] ".GlobalEnv"          "package:Biostrings" "package:IRange"
[4] "package:stats"       "package:graphics"  "package:grDevices"
[7] "package:utils"       "package:datasets"  "package:methods"
[10] "Autoloads"           "package:base"

> ?readDNAStringSet
No documentation for 'readDNAStringSet' in specified packages and 1$
you could try '??readDNAStringSet'

> ?read.DNAStringSet
> |
```

XStringSet-io {Biostrings} R Documentation

**Read/write an XStringSet object from/to a file**

**Description**

Functions to read/write an [XStringSet](#) object from/to a file.

**Usage**

```
## Read FASTA (or FASTQ) files in an XStringSet object:
read.BStringSet(filepath, format="fasta",
                 nrec=-1L, skip=0L, use.names=TRUE)
read.DNAStringSet(filepath, format="fasta",
                  nrec=-1L, skip=0L, use.names=TRUE)
read.RNAStringSet(filepath, format="fasta",
                  nrec=-1L, skip=0L, use.names=TRUE)
read.AAStringSet(filepath, format="fasta",
                  nrec=-1L, skip=0L, use.names=TRUE)
```

# バージョン違いの影響

## ■ 過去から現在 : Biostringsパッケージ

2010年10月リリースのR ver. 2.12.0の頃は、Biostrings ver. 2.18.4です。

### イントロ | 一般 | 翻訳配列(translate)を取得 **NEW**

塩基配列を読み込んでアミノ酸配列に翻訳するやり方を示す  
「ファイル」-「ディレクトリの変更」で解析したいファイルを置

#### 1. FASTA形式ファイル(sample1.fasta)の場合:

```

in_f <- "sample1.fasta"      #入
out_f <- "hoge1.fasta"      #出

#必要なパッケージをロード
library(Biostrings)         #パ

#入力ファイルの読み込み
fasta <- readDNASTringSet(in_f, format="fasta")

#本番
hoge <- translate(fasta)    #fa
names(hoge) <- names(fasta) #現
fasta <- hoge              #ho
fasta                      #確

#ファイルに保存
writeXStringSet(fasta, file=out_f, format="fasta")
    
```

```

R Console
> sessionInfo()
R version 2.12.0 (2010-10-15)
Platform: x86_64-pc-mingw32/x64 (64-bit)

locale:
[1] LC_COLLATE=Japanese_Japan.932
[2] LC_CTYPE=Japanese_Japan.932
[3] LC_MONETARY=Japanese_Japan.932
[4] LC_NUMERIC=C
[5] LC_TIME=Japanese_Japan.932

attached base packages:
[1] stats      graphics  grDevices  utils
[5] datasets  methods   base

other attached packages:
[1] Biostrings_2.18.4 IRanges_1.8.9

loaded via a namespace (and not attached):
[1] Biobase_2.10.0 tools_2.12.0
> |
    
```

# バージョン違いの影響

2012年3月リリースのR ver. 2.15.0  
で実行するとエラーは出ません。  
この当時はreadDNAStrngSetで  
もread.DNAStrngSetでもどちらで  
も受け入れてくれていた。

```
R Console
R version 2.15.0 (2012-03-30)
Copyright (C) 2012 The R Foundation for Statistical
ISBN 3-900051-07-0
Platform: x86_64-pc-mingw32/x64 (64-bit)

Rは、自由なソフトウェアであり、「完全に無保証」です$
一定の条件に従えば、自由にこれを再配布することがで$
配布条件の詳細に関しては、'license()'あるいは$

Rは多くの貢献者による共同プロジェクトです。
詳しくは'contributors()'と入力してください。
また、RやRのパッケージを出版物で引用する際の形式に$
'citation()'と入力してください。

'demo()'と入力すればデモをみることができます。
'help()'とすればオンラインヘルプが出ます。
'help.start()'でHTMLブラウザによるヘルプがみ$
'q()'と入力すればRを終了します。
```

```
R Console
> library(Biostrings) #パッケージ$
>
> #入力ファイルの読み込み
> fasta <- readDNAStrngSet(in_f, format="fasta")#i$
>
> #本番
> hoge <- translate(fasta) #fastaをア$
> names(hoge) <- names(fasta) #現状では$
> fasta <- hoge #hogeの中$
> fasta #確認して$
A AAStringSet instance of length 1
width seq names $
[1] 4 SDGL kadota
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta"$
>
> |
```

# バージョン違いの影響

2012年3月リリースのR ver. 2.15.0  
 で実行するとエラーは出ません。  
 この当時はreadDNAStrngSetでも  
 read.DNAStrngSetでもどちらでも  
 受け入れてくれていた。そのため、  
 下記の使えなくなるぞという  
 警告を気にもしていなかった。

```
R Console
R version 2.15.0 (2012-03-30)
Copyright (C) 2012 The R Foundation for Statistical
ISBN 3-900051-07-0
Platform: x86_64-pc-mingw32/x64 (64-bit)
```

Rは、自由なソフトウェアであり、「完全に無保証」です。\$  
 一定の条件に従えば、自由にこれを再配布することができ\$  
 配布条件の詳細に関しては、'license()'あるいは

Rは多くの貢献者による共同プロジェクトです。  
 詳しくは'contributors()'と入力してください。  
 また、RやRのパッケージを出版物で引用する際の形式に\$  
 'citation()'と入力してください。

'demo()'と入力すればデモをみることができます。  
 'help()'とすればオンラインヘルプが出ます。  
 'help.start()'でHTMLブラウザによるヘルプがみ\$  
 'q()'と入力すればRを終了します。

```
R Console
> #入力ファイルの読み込み
> fasta <- read.DNAStrngSet(in_f, format="fasta")#$
警告メッセージ:
'read.DNAStrngSet' is deprecated.
Use 'readDNAStrngSet' instead.
See help("Deprecated")
>
> #本番
> hoge <- translate(fasta) #fastaをア$
> names(hoge) <- names(fasta) #現状では$
> fasta <- hoge #hogeの中$
> fasta #確認して$
A AAStringSet instance of length 1
width seq names
[1] 4 SDGL kadota
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta"$
> |
```

# バージョン違いの影響

```
R Console
> ?readDNAStringSet
> ?read.DNAStringSet
> sessionInfo()
R version 2.15.0 (2012-03-30)
Platform: x86_64-pc-mingw32/x64 (64-bit)

locale:
[1] LC_COLLATE=Japanese_Japan.932
[2] LC_CTYPE=Japanese_Japan.932
[3] LC_MONETARY=Japanese_Japan.932
[4] LC_NUMERIC=C
[5] LC_TIME=Japanese_Japan.932

attached base packages:
[1] stats      graphics  grDevices  utils
[5] datasets  methods   base

other attached packages:
[1] Biostrings_2.26.3  IRanges_1.16.6
[3] BiocGenerics_0.4.0 BiocInstaller_1.8.3

loaded via a namespace (and not attached):
[1] parallel_2.15.0 stats4_2.15.0  tools_2.15.0
> |
```

2012年3月リリースのR ver. 2.15.0で実行するとエラーは出ません。この当時はreadDNAStringSetでもread.DNAStringSetでもどちらでも受け入れてくれていた。そのため、下記の使えなくなるぞという警告を気にもしていなかった。この頃は、?readDNAStringSetでも?read.DNAStringSetでもマニュアルがちゃんと開いていた。

## ファイル名 : rcode\_20140909.txt

## ファイル名 : rcode\_translate3.txt

```
#####↓
### 作業ディレクトリの変更↓
#####↓
setwd("C:/Users/kadota/Desktop/hoge") # デスクトップ上の
#setwd("C:/Users/kadota/Desktop") # 「デスクトップ」
#setwd("C:/Users/kadota/Documents") # 「マイドキュメント」
↓
#####↓
### オブジェクトの消去↓
#####↓
rm(list = ls())↓
↓
#####↓
### 翻訳配列取得↓
#####↓
in_f <- "sample1.fasta" #入力ファイル名を指定してin_fに格納↓
out_f <- "hoge1.fasta" #出力ファイル名を指定してout_fに格納↓
↓
#必要なパッケージをロード↓
library(Biostrings) #パッケージの読み込み↓
↓
#入力ファイルの読み込み↓
fasta <- readDNASTringSet(in_f, format="fasta")#in_fで指定したファイルの読み込み↓
↓
#本番↓
hoge <- translate(fasta) #fastaをアミノ酸配列に翻訳した結果をhogeに格納↓
names(hoge) <- names(fasta) #現状では翻訳した結果のオブジェクトhogeのdescri
fasta <- hoge #hogeの中身をfastaに格納↓
fasta #確認してるだけです↓
↓
#ファイルに保存↓
writeXStringSet(fasta, file=out_f, format="fasta", width=50)#fastaの中身を指定したファ
```

```
#####↓
### Tips (Biostrings/パッケージ)↓
#####↓
source("http://bioconductor.org/biocLite.R")# Biostrings/パッケージのイ
biocLite("Biostrings") # Biostrings/パッケージのインストール
↓
library(Biostrings) # Biostrings/パッケージをロード↓
↓
?readDNASTringSet # マニュアルを表示↓
?read.DNASTringSet # マニュアルを表示↓
sessionInfo() # 自分のR環境を確認↓
```

これらのテンプレートコードのコピーでスライド作成を行っています。

# バージョン違いの影響

2014年4月リリースのR ver. 3.1.0  
では、完全にreadDNAStrngSetし  
か受け入れなくなっている。

```
R Console
R version 3.1.0 (2014-04-10) ← "Spring Dance"
Copyright (C) 2014 The R Foundation for Statistical$
Platform: x86_64-w64-mingw32/x64 (64-bit)

R は、自由なソフトウェアであり、「完全に無保証」で$
一定の条件に従えば、自由にこれを再配布することがで$
配布条件の詳細に関しては、'license()' あるいは 'lic$

R は多くの貢献者による共同プロジェクトです。
詳しくは 'contributors()' と入力してください。
また、R や R のパッケージを出版物で引用する際の形
'citation()' と入力してください。

'demo()' と入力すればデモをみることができます。
'help()' とすればオンラインヘルプが出ます。
'help.start()' で HTML ブラウザによるヘル
'q()' と入力すれば R を終了します。
```

```
R Console
> library(Biostrings) #パッケージ$
>
> #入力ファイルの読み込み
> fasta <- readDNAStrngSet(in_f, format="fasta")#i$
>
> #本番
> hoge <- translate(fasta) #fastaをア$
> names(hoge) <- names(fasta) #現状では$
> fasta <- hoge #hogeの中$
> fasta #確認して$
A AAStringSet instance of length 1
  width seq          names
[1]    4 SDGL          kadota
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta"$
>
> |
```

# バージョン違いの影響

2014年4月リリースのR ver. 3.1.0  
では、完全にreadDNAStrngSetし  
か受け入れなくなっている。

```
R Console
R version 3.1.0 (2014-04-10) ← "Spring Dance"
Copyright (C) 2014 The R Foundation for Statistical$
Platform: x86_64-w64-mingw32/x64 (64-bit)
```

R は、自由なソフトウェアであり、「完全に無保証」で\$  
一定の条件に従えば、自由にこれを再配布することができ\$  
配布条件の詳細に関しては、'license()' があるし

R は多くの貢献者による共同プロジェクトです。  
詳しくは 'contributors()' と入力してください\$  
また、R や R のパッケージを出版物で引用する際の形\$  
'citation()' と入力してください。

'demo()' と入力すればデモをみることができます。  
'help()' とすればオンラインヘルプが出ます。  
'help.start()' で HTML ブラウザによるヘル\$  
'q()' と入力すれば R を終了します。

```
R Console
> #入力ファイルの読み込み
> fasta <- read.DNAStrngSet(in_f, format="fasta")#$
エラー: 関数 "read.DNAStrngSet" を見つけることが$
wrapup 中にエラーが起きました: コネクションを開$
>
> #本番
> hoge <- translate(fasta) #fastaをア$
wrapup 中にエラーが起きました: コネクションを開$
以下にエラー translate(fasta) :
  引数 'x' の評価中にエラーが起きました (関数 'tra$
wrapup 中にエラーが起きました: コネクションを開$
> names(hoge) <- names(fasta) #現状では$
エラー: オブジェクト 'fasta' がありません
wrapup 中にエラーが起きました: コネクションを開$
> fasta <- hoge #hogeの中$
エラー: オブジェクト 'hoge' がありません
wrapup 中にエラーが起きました: コネクションを開$
> fasta #確認して$
エラー: オブジェクト 'fasta' がありません
```

# バージョン違いの影響

2014年4月リリースのR ver. 3.1.0  
では、完全にreadDNAStringSetし  
か受け入れなくなっている。

R Console

```
> ?readDNAStringSet # マニュアルを表示
> ?read.DNAStringSet # マニュアルを表示
No documentation for 'read.DNAStringSet' in specified packages and librari$
you could try '??read.DNAStringSet'
> sessionInfo() # 自分のR環境を確認
R version 3.1.0 (2014-04-10)
Platform: x86_64-w64-mingw32/x64 (64-bit)

locale:
[1] LC_COLLATE=Japanese_Japan.932 LC_CTYPE=Japanese_Japan.932
[3] LC_MONETARY=Japanese_Japan.932 LC_NUMERIC=C
[5] LC_TIME=Japanese_Japan.932

attached base packages:
[1] parallel stats graphics grDevices utils datasets methods
[8] base

other attached packages:
 [1] TCC_1.4.0 ROC_1.40.0 baySeq_1.18.0
 [4] edgeR_3.6.0 limma_3.20.1 DESeq2_1.4.0
 [7] RcppArmadillo_0.4.200.0 Rcpp_0.11.1 GenomicRanges_1.16.1
[10] GenomeInfoDb_1.0.2 DESeq_1.16.0 lattice_0.20-29
[13] locfit_1.5-9.1 Biobase_2.24.0 Biostrings_2.32.0
[16] XVector_0.4.0 IRanges_1.22.3 BiocGenerics_0.10.0

loaded via a namespace (and not attached):
 [1] annotate_1.42.0 AnnotationDbi_1.26.0 DBI_0.2-7
 [4] genefilter_1.46.0 genplotter_1.42.0 grid_3.1.0
```

# バージョン違いの影響

- 過去から現在 : Biostringsパッケージ
  - R ver. 2.12.0 (2010年10月リリース)
    - Bioconductor ver. 2.7; Biostrings ver. 2.18.4): read.DNAStringSet関数のみ
  - R ver. 2.15.0 (2012年3月リリース)
    - Bioconductor ver. 2.11; Biostrings ver. 2.26.3): 移行期
  - R ver. 3.1.0 (2014年4月リリース)
    - Bioconductor ver. 2.14; Biostrings ver. 2.32.0): readDNAStringSet関数のみ

これは過去の出来事ですが、  
現在進行形の事例もあります。

# バージョン違いの影響

- 現在から未来: BSgenome.Hsapiens.UCSC.hg19パッケージ
  - R ver. 3.0.3 (2014年3月リリース)
    - Bioconductor ver. 2.13: パッケージ内に上流配列情報が格納?!されている
  - R ver. 3.1.0 (2014年4月リリース)
    - Bioconductor ver. 2.14: 移行期(Transcript DB形式のオブジェクト利用を推奨)
  - R ver. 3.X.Y (2014年10月リリース?!)
    - Bioconductor ver. 2.15: ...

R ver. 3.1.0で移行期に入っている事例を紹介

# Contents

- 3-4. R Bioconductor I、2014/09/09 10:30-14:45、中級、実習
  - Tips: setwd関数を利用した効率的な作業ディレクトリの変更
  - データの型1: translate関数の入力情報(AAStringSet)
  - Tips: オブジェクトの消去
  - データの型2: 翻訳配列取得のコードの中身を解説
  - バージョン情報把握とバージョンアップ
  - バージョン違いの影響
    - 過去から現在: BiostringsパッケージのreadDNAStrngSet関数
    - 現在から未来: BSgenome.Hsapiens.UCSC.hg19パッケージでプロモーター配列取得
  - Bioconductor概観





**BSgenome**パッケージを用いて様々な生物種のプロモーター配列(転写開始点近傍配列; 上流配列)を取得するやり方を示します。シロイヌナズナ(*A.thaliana*)、ウシ(*B.taurus*)、線虫(*C.elegans*)、犬(*C.familiaris*)、キイロショウジョウバエ(*D.melanogaster*)、ゼブラフィッシュ(*D.rerio*)、大腸菌(*E.coli*)、イトヨ(*G.aculeatus*)、セキショクヤケイ(*G.gallus*)、ヒト(*H.sapiens*)、アカゲザル(*M.mulatta*)、マウス(*M.musculus*)、チンパンジー(*P.troglodytes*)、ラット(*R.norvegicus*)、出芽酵母(*S.cerevisiae*)、トキソプラズマ(*T.gondii*)と実に様々な生物種が利用可能であることがわかりますが、生物種によって、上流配列情報がないものもあります(例:シロイヌナズナ)。「ファイル」-「ディレクトリの変更」でファイルを保存したいディレクトリに移動し以下をコピー。

1. 利用可能な生物種とRにインストール済みの生物種をリストアップしたい場合:

```
#必要なパッケージをロード
library(BSgenome)
```

#パッケージの読み込み

```
#本番 (利用可能なリストアップ; インストール済みとは限らない)
available.genomes()
```

#このパッケージ中で利用可能なゲノムをリストアップ

```
#本番 (インストール済みの生物種をリストアップ)
installed.genomes()
```

```
#後処理 (パッケージ名でだいたいわかるがpr
installed.genomes(splitNameParts=TRUE)
```

R ver. 3.1.0 (Bioconductor ver. 2.14)で利用可能な生物種のパッケージ名をリストアップ。70個あることが分かる。Rのバージョンが古いとおそらくパッケージ数は少なくなる。

```
R Console

[55] "BSgenome.Ptroglydytes.UCSC.panTro2"
[56] "BSgenome.Ptroglydytes.UCSC.panTro2.masked"
[57] "BSgenome.Ptroglydytes.UCSC.panTro3"
[58] "BSgenome.Ptroglydytes.UCSC.panTro3.masked"
[59] "BSgenome.Rnorvegicus.UCSC.rn4"
[60] "BSgenome.Rnorvegicus.UCSC.rn4.masked"
[61] "BSgenome.Rnorvegicus.UCSC.rn5"
[62] "BSgenome.Rnorvegicus.UCSC.rn5.masked"
[63] "BSgenome.Scerevisiae.UCSC.sacCer1"
[64] "BSgenome.Scerevisiae.UCSC.sacCer2"
[65] "BSgenome.Scerevisiae.UCSC.sacCer3"
[66] "BSgenome.Sscrofa.UCSC.susScr3"
[67] "BSgenome.Sscrofa.UCSC.susScr3.masked"
[68] "BSgenome.Tgondii.ToxoDB.7.0"
[69] "BSgenome.Tguttata.UCSC.taeGut1"
[70] "BSgenome.Tguttata.UCSC.taeGut1.masked"
> |
```

```
> available.genomes()
[1] "BSgenome.Alyrata.JGI.v1"
[2] "BSgenome.Amellifera.BeeBase.assembly"
[3] "BSgenome.Amellifera.UCSC.apiMel2"
[4] "BSgenome.Amellifera.UCSC.apiMel2.mas"
[5] "BSgenome.Athaliana.TAIR.04232008"
[6] "BSgenome.Athaliana.TAIR.TAIR9"
[7] "BSgenome.Btaurus.UCSC.bosTau3"
[8] "BSgenome.Btaurus.UCSC.bosTau3.masked"
[9] "BSgenome.Btaurus.UCSC.bosTau4"
[10] "BSgenome.Btaurus.UCSC.bosTau4.masked"
[11] "BSgenome.Btaurus.UCSC.bosTau6"
[12] "BSgenome.Btaurus.UCSC.bosTau6.masked"
[13] "BSgenome.Celegans.UCSC.ce10"
[14] "BSgenome.Celegans.UCSC.ce2"
[15] "BSgenome.Celegans.UCSC.ce6"
[16] "BSgenome.Cfamiliaris.UCSC.canFam2"
[17] "BSgenome.Cfamiliaris.UCSC.canFam2.ma"
[18] "BSgenome.Cfamiliaris.UCSC.canFam3"
[19] "BSgenome.Cfamiliaris.UCSC.canFam3.ma"
[20] "BSgenome.Dmelanogaster.UCSC.dm2"
[21] "BSgenome.Dmelanogaster.UCSC.dm2.mask"
[22] "BSgenome.Dmelanogaster.UCSC.dm3"
[23] "BSgenome.Dmelanogaster.UCSC.dm3.mask"
[24] "BSgenome.Drerio.UCSC.danRer5"
[25] "BSgenome.Drerio.UCSC.danRer5.masked"
[26] "BSgenome.Drerio.UCSC.danRer6"
[27] "BSgenome.Drerio.UCSC.danRer6.masked"
[28] "BSgenome.Drerio.UCSC.danRer7"
```

#この\$

R Console

```
[29] "BSgenome.Drerio.UCSC.danRer5"
[30] "BSgenome.Ecoli.NCBI.286898"
[31] "BSgenome.Gaculeatus.UCSC.gasAcu1"
[32] "BSgenome.Gaculeatus.UCSC.gasAcu1.masked"
[33] "BSgenome.Ggallus.UCSC.galGal3"
[34] "BSgenome.Ggallus.UCSC.galGal3.masked"
[35] "BSgenome.Ggallus.UCSC.galGal4"
[36] "BSgenome.Ggallus.UCSC.galGal4.masked"
[37] "BSgenome.Hsapiens.NCBI.GRCh38"
[38] "BSgenome.Hsapiens.UCSC.hg17"
[39] "BSgenome.Hsapiens.UCSC.hg17.masked"
[40] "BSgenome.Hsapiens.UCSC.hg18"
[41] "BSgenome.Hsapiens.UCSC.hg18.masked"
[42] "BSgenome.Hsapiens.UCSC.hg19"
[43] "BSgenome.Hsapiens.UCSC.hg19.masked"
[44] "BSgenome.Mmulatta.UCSC.rheMac2"
[45] "BSgenome.Mmulatta.UCSC.rheMac2.masked"
[46] "BSgenome.Mmulatta.UCSC.rheMac3"
[47] "BSgenome.Mmulatta.UCSC.rheMac3.masked"
[48] "BSgenome.Mmusculus.UCSC.mm10"
[49] "BSgenome.Mmusculus.UCSC.mm10.masked"
[50] "BSgenome.Mmusculus.UCSC.mm8"
[51] "BSgenome.Mmusculus.UCSC.mm8.masked"
[52] "BSgenome.Mmusculus.UCSC.mm9"
[53] "BSgenome.Mmusculus.UCSC.mm9.masked"
[54] "BSgenome.Osativa.MSU.MSU7"
[55] "BSgenome.Ptroglydytes.UCSC.panTro2"
```

2013年12月にリリースされたヒトゲノム最新リリース(GRCh38)のRパッケージも利用可能です。

BSgenomeパッケージを用いて様々な生物種のプロモーター配列(転写開始点近傍配列; 上流配列)を取得するやり方を示します。シロイヌナズナ(*A.thaliana*)、ウシ(*B.taurus*)、線虫(*C.elegans*)、犬(*C.familiaris*)、キイロショウジョウバエ(*D.melanogaster*)、ゼブラフィッシュ(*D.rerio*)、大腸菌(*E.coli*)、イトヨ(*G.acuteatus*)、セキショクヤケイ(*G.gallus*)、ヒト(*H.sapiens*)、アカゲザル(*M.mulatta*)、マウス(*M.musculus*)、チンパンジー(*P.troglodytes*)、ラット(*R.norvegicus*)、出芽酵母(*S.cerevisiae*)、トキソプラズマ(*T.gondii*)と実に様々な生物種が利用可能であることがわかりますが、生物種によって、上流配列情報がないものもあります(例:シロイヌナズナ)。

「ファイル」→「ディレクトリの変更」でファイルを保存したいディレクトリに移動し以下をコピー。

1. 利用可能な生物種とRにインストール済みの生物種をリストアップしたい

```
#必要なパッケージをロード
library(BSgenome) #パッケージの読み込み

#本番 (利用可能なリストアップ; インストール済みとは限らない)
available.genomes() #このパッケージ中で

#本番 (インストール済みの生物種をリストアップ)
installed.genomes() #インストール済みの

#後処理 (パッケージ名でだいたいわかるがproviderやversionを分
installed.genomes(splitNameParts=TRUE) #インストール済みの
```

```
R Console

[65] "BSgenome.Scerevisiae.UCSC.taeGut1"
[66] "BSgenome.Sscerevisiae.UCSC.taeGut1"
[67] "BSgenome.Sscerevisiae.UCSC.taeGut1.masked"
[68] "BSgenome.Tgondii.ToxoDB.7.0"
[69] "BSgenome.Tguttata.UCSC.taeGut1"
[70] "BSgenome.Tguttata.UCSC.taeGut1.masked"
> installed.genomes() #イン$

[1] "BSgenome.Athaliana.TAIR.TAIR9"
[2] "BSgenome.Celegans.UCSC.ce2"
[3] "BSgenome.Drerio.UCSC.danRer7"
[4] "BSgenome.Ecoli.NCBI.20080805"
[5] "BSgenome.Hsapiens.NCBI.GRCh38"
[6] "BSgenome.Hsapiens.UCSC.hg19"
[7] "BSgenome.Mmusculus.UCSC.mm9"
> |
```

実際にインストール済みのものは(このPC環境では)7パッケージであることがわかる。植物のシロイヌナズナ(*Arabidopsis thaliana*)のパッケージは推奨手順通りにインストール作業をしたヒトは存在するはず。

# NGSデータ解析とR

理由:このパッケージはデフォルトではインストールされないが、「Rのインストールと起動」のところで明示的に個別にインストールをしていたからです。

## Rのインストールと起動 **NEW**

基本的には[こちら](#)または[こちら](#)をご覧ください。

よく分からない人でWindowsユーザーの方は以下を参考にしてください。2のインストール手順は[こちら](#)。2014年5月14日にアップデートしたMac版のインストール手順は[こちら](#)。注意点は、「Mac OS Xのバージョンに関わらず R-3.1.0-snowleopard」

### 1. Windows release版のインストールの場合:

1. [Rのインストール](#)を「実行」
2. 聞かれるがままに「次へ」などを押してとにかくインストールを完了
3. **Windows Vista**の人は(パッケージのインストール中に書き込み権限に「コントロールパネル」-「ユーザーアカウント」-「ユーザーアカウント制御(UAC)を使ってコンピュータの保護に役立ちます」のメッセージが強くお勧めします。
4. インストールが無事完了したら、デスクトップに出現する「R 3.X.Y(32 bit)」または「R x64 3.X.Y(64 bitの場合)」アイコンをダブルクリック
5. 以下を、「R コンソール画面」でコピー&ペーストする。10GB程度必要です。(どこからダウンロードするか?と聞かれるので、その場合は自

```
R Console
[65] "BSgenome.Scerevisiae.UCSC.sacCer3"
[66] "BSgenome.SsCrofa.UCSC.susScr3"
[67] "BSgenome.SsCrofa.UCSC.susScr3.masked"
[68] "BSgenome.Tgondii.ToxoDB.7.0"
[69] "BSgenome.Tguttata.UCSC.taeGut1"
[70] "BSgenome.Tguttata.UCSC.taeGut1.masked"
> installed.genomes() #イン$
[1] "BSgenome.Athaliana.TAIR.TAIR9"
[2] "BSgenome.Celegans.UCSC.ce2"
[3] "BSgenome.Drerio.UCSC.danRer7"
[4] "BSgenome.Ecoli.NCBI.20080805"
[5] "BSgenome.Hsapiens.NCBI.GRCh38"
[6] "BSgenome.Hsapiens.UCSC.hg19"
[7] "BSgenome.Mmusculus.UCSC.mm9"
> |
```

```
install.packages(available.packages()[,1], dependencies=TRUE)#CRAN中にある全てのパッケージをインストール
source("http://www.bioconductor.org/biocLite.R")#おまじない
biocLite(all_group()) #Bioconductor中にある全てのパッケージをインストール
biocLite("BSgenome.Athaliana.TAIR.TAIR9", suppressUpdates=TRUE)#Bioconductor中にあるパッケージをインストール
```

6. 「コントロールパネル」-「デスクトップのカスタマイズ」-「フォルダオプション」-「表示(タブ)」-「詳細設定」のところで、「登録されている拡張子は表示しない」のチェックを外してください。

# Contents

- 3-4. R Bioconductor I、2014/09/09 10:30-14:45、中級、実習
  - Tips: setwd関数を利用した効率的な作業ディレクトリの変更
  - データの型1: translate関数の入力情報(AAStringSet)
  - Tips: オブジェクトの消去
  - データの型2: 翻訳配列取得のコードの中身を解説
  - バージョン情報把握とバージョンアップ
  - バージョン違いの影響
    - 過去から現在: BiostringsパッケージのreadDNAStrngSet関数
    - 現在から未来: BSgenome.Hsapiens.UCSC.hg19パッケージでプロモーター配列取得
  - Bioconductor概観



# バージョン違いの影響

イントロ | 一般 | 配列取得 | プロモーター配列 | BSgenome NEW

ヒトゲノム(BSgenome.Hsapiens.UCSC.hg19)の上流1000bp (upstream1000)の配列を取得し、hoge5.fastaというファイル名で保存するコードをR ver. 3.0.3とR ver. 3.1.0で実行し、挙動の違いを眺めます。

BSgenomeパッケージを用いて様々な生物種のプロモーター配列(転写開始点近傍配列; 上流配列)を示します。シロイヌナズナ(A.thaliana)、ウシ(B.taurus)、線虫(C.elegans)、犬(C.familiaris)、ウバエ(D.melanogaster)、ゼブラフィッシュ(D.rerio)、大腸菌(E.coli)、イトヨ(G.acuteatus)、セキショウ(G.gallus)、ヒト(H.sapiens)、アカゲザル(M.mulatta)、マウス(M.musculus)、チンパンジー(P.troglodytes)、ノルウェーシロネ(R.norvegicus)、出芽酵母(S.cerevisiae)、トキンプラズマ(T.gondii)と実に様々な生物種が利用可能ですが、生物種によって、上流配列情報がないものもあります(例:シロイヌナズナ)。

5. インストール済みのヒト("BSgenome.Hsapiens.UCSC.hg19")の転写開始点上流配列(1000bp)をmulti-FASTAファイルで保存したい場合:

## 1. 利用可能な生物種と

上流1000bp以外に2000bp, 5000bpもあります ...

#必要なパッケージ  
library(BSgenome)

#本番 (利用可能な)  
available.genome

#本番 (インストール済)  
installed.genome

#後処理 (パッケージインストール済)  
installed.genome

```

out_f <- "hoge5.fasta" #出力ファイル名を指定してout_fに格納
param1 <- "BSgenome.Hsapiens.UCSC.hg19" #パッケージ名を指定
param2 <- "upstream1000" #上流1000bpを指定

#必要なパッケージをロード
library(param1, character.only=T) #param1で指定したパッケージの読み込み

#前処理(param1で指定したパッケージ中のオブジェクト名をhogeに統一)
#tmp <- unlist(strsplit(param1, ".", fixed=TRUE))[2] #param1で指定した文字列からオブジェクト名を取得した
tmp <- ls(paste("package", param1, sep=":")) #param1で指定したパッケージで利用可能なオブジェクト名を取得
hoge <- eval(parse(text=tmp)) #文字列tmpをRオブジェクトとしてhogeに格納(パッケージ中にはオブジェクト名)
hoge #確認してるだけです(ここで、multiple sequencesのところにparam2

#本番
tmp <- paste("hoge$", param2, sep="") #param2で指定した文字列を含むオブジェクト名を作成した結果をtmpに
fasta <- eval(parse(text=tmp)) #文字列tmpをRオブジェクトとしてfastaに格納

#ファイルに保存
writeXStringSet(fasta, file=out_f, format="fasta", width=50) #fastaの中身を指定したファイル名で保存
    
```

## 2. ゼブラフィッシュ("BSgenome.Drerio.UCSC.hg19")

400MB程度あります...

# バージョン違いの影響

R ver. 3.0.3での実行例

```
R Console
R version 3.0.3 (2014-03-06) -- "Warm Puppy"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
```

R は、自由なソフトウェアであり、「完全に無保証」です。\$  
 一定の条件に従えば、自由にこれを再配布することができます。\$  
 配布条件の詳細に関しては、'license()' あるいは  
 'demo()' と入力すればデモをみることができます。  
 'help()' とすればオンラインヘルプが出ます。  
 'help.start()' で HTML ブラウザによるヘルプが  
 'q()' と入力すれば R を終了します。  
 > getwd()  
 [1] "C:/Users/kadota/Desktop"  
 > |

```
R Console
| chrUn_gl000237          chrUn_gl000238
| chrUn_gl000239          chrUn_gl000240
| chrUn_gl000241          chrUn_gl000242
| chrUn_gl000243          chrUn_gl000244
| chrUn_gl000245          chrUn_gl000246
| chrUn_gl000247          chrUn_gl000248
| chrUn_gl000249
|
| multiple sequences (see '?mseqnames'):  

|   upstream1000 upstream2000 upstream5000
|
| (use the '$' or '[[' operator to access a given  

| sequence)
>
> #本番
> tmp <- paste("hoge$", param2, sep="") #param2で指$
> fasta <- eval(parse(text=tmp))      #文字列tmpを$
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", $
> |
```

# バージョン違いの影響

5. インストール済みのヒト("BSgenome.Hsapiens.UCSC.hg19")の転写開始点上流配列(1000bp)を取得したい場合:

上流1000bp以外に2000bp, 5000bpもあります...

```
out_f <- "hoge5.fasta" #出力ファイル名を指定してout_fに格納
param1 <- "BSgenome.Hsapiens.UCSC.hg19" #パッケージ名を指定
param2 <- "upstream1000" #上流1000bpを指定
```

```
#必要なパッケージをロード
library(param1, character.only=T) #param1で指定したパッケージの読み込み
```

```
#前処理(param1で指定したパッケージ)
#tmp <- unlist(strsplit(paste("package", param1), "."))
tmp <- ls(paste("package", param1))
hoge <- eval(parse(text=tmp))
hoge
```

```
#本番
tmp <- paste("hoge$", param2)
fasta <- eval(parse(text=tmp))
```

```
#ファイルに保存
writeXStringSet(fasta, file=out_f)
```

getwd()でみられる作業ディレクトリ上に、out\_fで指定した出力ファイル(hoge5.fasta)が作成されているはずです。width列の数値が全て1000になっていることから、上流1,000bpを切り出せていることが推察できます。

```
R Console
> fasta
A DNAStringSet instance of length 28020
      width seq
[1] 1000 CCACCTGGGGAAGCG...TGCCGTTCTCTCCC NM_032291_up_1000...
[2] 1000 TGGACAACGACTTGG...CGTGCTCCTGCCGCC NM_013943_up_1000...
[3] 1000 GAGGCAGAGGTTGCA...ACTATGGGCGGGGCC NM_052998_up_1000...
[4] 1000 ATGTGAGAGAGTTCA...GTCCCTCCCGCCCTC NM_032785_up_1000...
[5] 1000 ATCAGAAGTTTGGGA...GCTGCCGCTCTAGCC NM_001145277_up_1...
...
[28016] 1000 AGCGACGCGGGGACT...TCCCCACCACCCCC NM_001127389_up_1...
[28017] 1000 AAAGACAGAGCGACG...CCACGCCCTCCCCCA NM_033178_up_1000...
[28018] 1000 AAAGACAGAGCGACG...CCACGCCCTCCCCCA NM_033178_up_1000...
[28019] 1000 TTGTATTTTGTAGTAG...CCTCTAGCTGTGTGT NM_006625_up_1000...
[28020] 1000 TTGTATTTTGTAGTAG...CCTCTAGCTGTGTGT NM_054016_up_1000...

> getwd()
[1] "C:/Users/kadota/Desktop"
> |
```

# バージョン違いの影響

5. インストール済みのヒト("BSgenome.Hsapiens.UCSC.hg19")の転写開始点上流配列(1000bp)を取得したい場合:

上流1000bp以外に2000bp, 5000bpもあります...

```

out_f <- "hoge5.fasta" #出力ファイル名を指定してout_fに格納
param1 <- "BSgenome.Hsapiens.UCSC.hg19" #パッケージ名を指定
param2 <- "upstream1000" #上流1000bpを指定

#必要なパッケージをロード
library(param1, character.only=T) #param1で指定したパッケージをロード

#前処理(param1で指定したパッケージ中のオブジェクト名をhoge1に統一)
#tmp <- unlist(strsplit(param1, ".", fixed=TRUE))[2] #param1で指定したパッケージ名を抽出
tmp <- ls(paste("package", param1, sep=":")) #param1で指定したパッケージ中のオブジェクト名を抽出
hoge <- eval(parse(text=tmp)) #文字列tmpをRオブジェクトに変換
hoge #確認してるだけです(ここでは省略)

#本番
tmp <- paste("hoge$", param2, sep="") #param2で指定した文字列を文字列に結合
fasta <- eval(parse(text=tmp)) #文字列tmpをRオブジェクトに変換

#ファイルに保存
writeXStringSet(fasta, file=out_f, format="fasta", width=50) #fastaオブジェクトをファイルに保存
    
```

慣れてくると、実物 (hoge5.fasta)を眺めて安心するよりも、fastaオブジェクトを眺めるほうが全体像をつかみやすいと思います。出力ファイルの行あたりの塩基数が50になっているのは、widthオプションを50にしているから。

```

>NM_032291 up 1000 chr1 66998825 f chr1:66998825-66999824
CCACCTGGGGAAGCGAGGCCGCTCCCGTGTGCGCTCGGCCGGGAGGACGC↓
GCTGGCGGCGGGACTCGAACTCGGTCGTCGAAGGGCGCCCCGAGGGTCC↓
CGTCTGTAGCCCGGACGGCGAGCCCAGGTTGGAGTGGGAGGGAGCCGGCG↓
TGACAGCTGTCTTTGGTGGGCTCAGGCTTCCGCTCCCTGCCTGCTCCC↓
CTTCCAGCCTCCCGCCCCAGAAACGATCTCGAGCGTTGCCAGTTTGATT↓
CCAGAGCCCCACTCGGGTGGGTTCTTTTGTCTTTTGTTTAATGACAGT↓
TCCCAGCCCTTCAGCATGTCATGCGCAATTAATCCCTGGCTCTCACGAA↓
GGCAGCTGGGGTGAATTTCTTCTGCATCATCTTTTGGGGATGTTTATG↓
ATGTGACGTCAGTCGATTGATTTTTCTCTTGAATGAAGGATGGGAGG↓
GGAGAAAGAGAGACGGAGAGAGAGAGACGCACAGATGTGCACGGAGGC↓
CACAGACTGACATTTGGAATTCCTTCAGGGTAAAAGGACACCGGAATG↓
GGAGCTTAGAAGTGTGTTGCTAAGATTTCCGGCTGCACGGAATTATTAAG↓
TTTTTCTTAAAAAACAAAAAAGAAAGAAAGAAAGAAAAAGAAAAGAAC↓
CCCCTCCGCAGCGAGCCACTTAGGTGCTGCTTTCACGCCAGAGTCCCCTG↓
TTAAGGTGGCAGCCCTTGATAACTAATCTCGGGCACCCAGCCGCTTCTGT↓
AAGCTTAAGGAGACGACGAGGAGGGGGGGGGAAGTGCCTACCAGGTGG↓
GGAAGGGGCTGTGATTTGGTGACAAGCGGGAGGCGATGGGGGTGGAGGG↓
GAATGGGGACGGGAAATAGGTTCTGTGTGCTCTCCGGGGGATTGTGTCA↓
GGAGATGCAGGCTGGCTACCATGTGACGGCTCCAAAGCTGAAGGGATTG↓
GCCGAGGCAGCGCAGGCGGTGCAGCTCGGCCAGCTTGCCGTTCCCTCTCCC↓
>NM_013943 up 1000 chr1 25070760 f chr1:25070760-25071759
TGGACAACGACTTGGAAAGTCTTAGTGGCCTGCAGGCACTGAGATTGCAG↓
TTCAACATCTTCAGCTTTCAGCTTCTTACTCTCCAAACCCCTACCTCACA↓
    
```

# バージョン違いの影響

5. インストール済みのヒト("BSgenome.Hsapiens.UCSC hg19")の転写開始点上流配列(1000bp)を取得したい場合:

上流1000bp以外に2000bp, 5000bpもあります...

```

out_f <- "hoge5.fasta"
param1 <- "BSgenome.Hsapiens.UCSC.hg19"
param2 <- "upstream1000"

#必要なパッケージをロード
library(param1, character.only=T)

#前処理(param1で指定したパッケージ中のオブジェクト)
#tmp <- unlist(strsplit(param1, ".", fixed=T))
tmp <- ls(paste("package", param1, sep=":"))
hoge <- eval(parse(text=tmp))

#本番
tmp <- paste("hoge$", param2, sep="")
fasta <- eval(parse(text=tmp))

#ファイルに保存
writeXStringSet(fasta, file=out_f, format="fasta")
    
```

R ver. 3.0.3では、特に何の問題もなく上流配列を取得できます。理由は、param2で指定したupstream1000情報がこのパッケージのバージョン(ver. 1.3.19)中に存在するからです。

```

R Console
| chrUn_gl1000237      chrUn_gl1000238
| chrUn_gl1000239      chrUn_gl1000240
| chrUn_gl1000241      chrUn_gl1000242
| chrUn_gl1000243      chrUn_gl1000244
| chrUn_gl1000245      chrUn_gl1000246
| chrUn_gl1000247      chrUn_gl1000248
| chrUn_gl1000249
|
| multiple sequences (see '?mseqnames'):
|   upstream1000 upstream2000 upstream5000
|
| (use the '$' or '[[ ' operator to access a given
| sequence)
>
> #本番
> tmp <- paste("hoge$", param2, sep="") #param2で指$
> fasta <- eval(parse(text=tmp))       #文字列tmpを$
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", $
> |
    
```

# バージョン違いの影響

sessionInfo()でR環境情報を取得

5. インストール済みのヒト("BSgenome.Hsapiens.UCSC.hg19")の転写開始点上流配列(1000bp)をmulti-FASTAファイルで保存したい場合:

上流1000bp以外に2000bp, 5000bp

```
out_f <- "hoge5.fasta"
param1 <- "BSgenome.Hsapiens.UCSC.hg19"
param2 <- "upstream1000"
```

```
#必要なパッケージをロード
library(param1, character.only=TRUE)
```

```
#前処理(param1で指定したパッケージをリストアップ)
#tmp <- unlist(strsplit(param1, "\\."))
tmp <- ls(paste("package:", tmp))
hoge <- eval(parse(text=tmp))
hoge
```

```
#本番
tmp <- paste("hoge$", tmp)
fasta <- eval(parse(text=tmp))
```

```
#ファイルに保存
writeXStringSet(fasta, file=out_f)
```

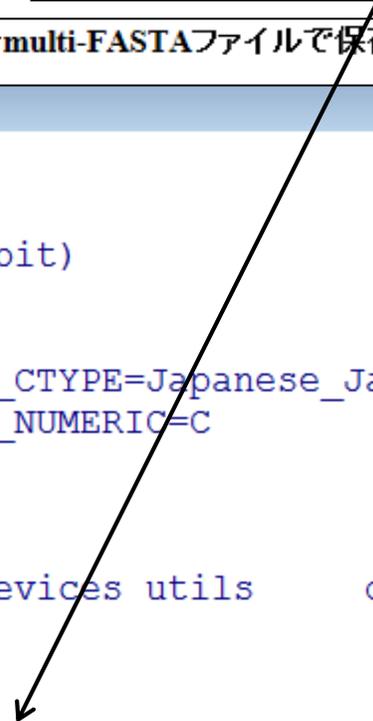
```
R Console
> sessionInfo()
R version 3.0.3 (2014-03-06)
Platform: x86_64-w64-mingw32/x64 (64-bit)

locale:
[1] LC_COLLATE=Japanese_Japan.932  LC_CTYPE=Japanese_Japan.932
[3] LC_MONETARY=Japanese_Japan.932 LC_NUMERIC=C
[5] LC_TIME=Japanese_Japan.932

attached base packages:
[1] parallel stats graphics grDevices utils datasets
[7] methods base

other attached packages:
[1] BSgenome.Hsapiens.UCSC.hg19_1.3.19
[2] BSgenome_1.30.0
[3] Biostrings_2.30.1
[4] GenomicRanges_1.14.4
[5] XVector_0.2.0
[6] IRanges_1.20.7
[7] BiocGenerics_0.8.0

loaded via a namespace (and not attached):
[1] stats4_3.0.3
> |
```



# バージョン違いの影響

R ver. 3.1.0での実行例。R ver. 3.0.3のときと違って、警告メッセージが出ていることが分かります。BioC 2.14は、2014年4月リリースのBioconductor ver. 2.14のこと。

```
R Console
R version 3.1.0 <2014.05.16>
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64

R は、自由なソフトウェアであり、ほとんどの場合、特定の条件に  
従えば、自由に再配布することができます。配布条件の詳細に  
関しては、http://www.R-project.org/contributor.html を見てください。  
R は多くの貢献者による共同プロジェクトです。詳しくは 'contributor  
'citation()' と入力すればご覧いただけます。'demo()' と入力すれば  
'help()' とすればオンラインヘルプを見られます。'help.start()' で  
'q()' と入力すれば R を終了します。'?' と入力すればヘルプを  
見ます。'demo()' と入力すればデモを実行します。

> getwd()
[1] "C:/Users/kadokawa/Desktop"
> |
```

```
R Console
> #本番
> tmp <- paste("hoge$", param2, sep="") #
> fasta <- eval(parse(text=tmp)) #
警告メッセージ:
Starting with BioC 2.14, upstream sequences are deprecated.
However they can easily be extracted from the full genome
sequences with something like (for example for hg19):

library(TxDb.Hsapiens.UCSC.hg19.knownGene)
txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
gn <- sort(genes(txdb))
up1000 <- flank(gn, width=1000)
library(BSgenome.Hsapiens.UCSC.hg19)
genome <- BSgenome.Hsapiens.UCSC.hg19
up1000seqs <- getSeq(genome, up1000)

IMPORTANT: Make sure you use a TxDb package (or TranscriptDb object)
that contains a gene model based on the exact same reference genome
as the BSgenome object you pass to getSeq(). Note that you can make
your own custom TranscriptDb object from various annotation resources.
See the makeTranscriptDbFromUCSC(), makeTranscriptDbFromBiomart(), and
makeTranscriptDbFromGFF() functions in the GenomicFeatures package.

>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", width=50)#fastaの$
> |
```

# バージョン違いの影響

警告メッセージは出るものの、上流配列自体は取得できるようです(R ver. 3.1.0)。1年後には使えなくなるかも…。

5. インストール済みのヒト("BSgenome.Hsapiens.UCSC.hg19")の転写開始点上流配列(1000bp)を取得したい場合:

上流1000bp以外に2000bp, 5000bpもあります...

```
out_f <- "hoge5.fasta" #出力ファイル名を指定してout_fに格納
param1 <- "BSgenome.Hsapiens.UCSC.hg19" #パッケージ名を指定
param2 <- "upstream1000" #上流1000bpを指定
```

```
#必要なパッケージを
library(param1, c

#前処理(param1で指
>
#tmp <- unlist(st
tmp <- ls(paste("
hoge <- eval(pars
hoge

#本番
tmp <- paste("hog
fasta <- eval(par

#ファイルに保存
writeXStringSet(f
```

```
R Console
See the makeTranscriptDbFromUCSC(), makeTranscriptDbFromBiomart(), and
makeTranscriptDbFromGFF() functions in the GenomicFeatures package.
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", width=50)#fastaの$
> fasta
A DNASTringSet instance of length 28020
      width seq
[1] 1000 CCACCTGGGGAAGCGAG...CTTGCCGTTCTCTCCC NM_032291_up_1000...
[2] 1000 TGGACAACGACTTGAA...CCCGTGCTCCTGCCGCC NM_013943_up_1000...
[3] 1000 GAGGCAGAGGTTGCAGT...GCACTATGGGCGGGGCC NM_052998_up_1000...
[4] 1000 ATGTGAGAGAGTTCAAG...GCGTCCCTCCCGCCCTC NM_032785_up_1000...
[5] 1000 ATCAGAAGTTTGGGATC...GAGCTGCCGCTCTAGCC NM_001145277_up_1...
... ..
[28016] 1000 AGCGACGCGGGGACTGG...CCTCCCCCACCACCCC NM_001127389_up_1...
[28017] 1000 AAAGACAGAGCGACGCG...CACCACGCCCTCCCCCA NM_033178_up_1000...
[28018] 1000 AAAGACAGAGCGACGCG...CACCACGCCCTCCCCCA NM_033178_up_1000...
[28019] 1000 TTGTATTTTGTAGTAGAG...GCCCTCTAGCTGTGTGT NM_006625_up_1000...
[28020] 1000 TTGTATTTTGTAGTAGAG...GCCCTCTAGCTGTGTGT NM_054016_up_1000...
> |
```

# バージョン違いの影響

警告メッセージを要約すると...

- upstreamの情報は使わない
- TranscriptDbオブジェクトを使え

5. インストール済みのヒト("BSgenome.Hsapiens.UCSC.hg19")の転写開始点上流配列(1000bp)をみたい場合:

上流1000bp以外に2000bp, 5000bpもあります...

```

out_f <- "hoge5.fasta" #出力ファイル名
param1 <- "BSgenome.Hsapiens.UCSC.hg19" #パッケージ名
param2 <- "upstream1000" #上流1000bp

#必要なパッケージをロード
library(param1, character.only=T) #param1

#前処理(param1で指定したパッケージ中のオブジェクト)
#tmp <- unlist(strsplit(param1, ".", fixed=TRUE))
tmp <- ls(paste("package:", param1, sep=":")) #パッケージ中のオブジェクト
hoge <- eval(parse(text=tmp)) #文字列をRコードとして実行
hoge #確認

#本番
tmp <- paste("hoge$", param2, sep="") #param2
fasta <- eval(parse(text=tmp)) #文字列をRコードとして実行

#ファイルに保存
writeXStringSet(fasta, file=out_f, format="fasta")
    
```

```

R Console
> fasta <- eval(parse(text=tmp)) #文字列
警告メッセージ:
Starting with BioC 2.14, upstream sequences are deprecated.
However they can easily be extracted from the BSgenome object using
sequences with something like (for example for hg19)
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
gn <- sort(genes(txdb))
up1000 <- flank(gn, width=1000)
library(BSgenome.Hsapiens.UCSC.hg19)
genome <- BSgenome.Hsapiens.UCSC.hg19
up1000seqs <- getSeq(genome, up1000)

IMPORTANT: Make sure you use a TxDb package (BSgenome.Hsapiens.UCSC.hg19.knownGene)
that contains a gene model based on the exact same genome assembly
as the BSgenome object you pass to getSeq(). If you use a custom
your own custom TranscriptDb object from variouseq, you can use
See the makeTranscriptDbFromUCSC(), makeTranscriptDbFromEnsembl(),
makeTranscriptDbFromGFF() functions in the GenomicFeatures package.

>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta")
>
    
```

# バージョン違いの影響

こんな感じで上流配列を取得できますという**赤枠**のテンプレートコードは基本的にコピーでうまくいきます。

ファイル名: rcode\_20140909.txt

```
#####↓
### R ver. 3.1.0の推奨手順で上流配列取得
#####↓
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
gn <- sort(genes(txdb))↓
up1000 <- flank(gn, width=1000)↓
library(BSgenome.Hsapiens.UCSC.hg19)↓
genome <- BSgenome.Hsapiens.UCSC.hg19↓
up1000seqs <- getSeq(genome, up1000)↓
↓
```

```
R Console
> library(TxDb.Hsapiens.UCSC.hg19.knownGene)
> txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
> gn <- sort(genes(txdb))
> up1000 <- flank(gn, width=1000)
> library(BSgenome.Hsapiens.UCSC.hg19)
> genome <- BSgenome.Hsapiens.UCSC.hg19
> up1000seqs <- getSeq(genome, up1000)
> up1000seqs
A DNAStringSet instance of length 23056
      width seq
[1] 1000 ACACATGCTACCGCG...TTTCTTTTCGTTAA 100287102
[2] 1000 GCTATTATCACCTAT...GAAACGAATAACTCT 79501
[3] 1000 GAATTAGGCTTCTGC...GAGAAAAAGGCGGGG 643837
[4] 1000 CGGGGAGCCCCGAGG...CCCAGCTTGGGCCA 148398
[5] 1000 CGGCGGGGCTCCTAT...GGCGGGAGCGGCGGG 339451
...
[23052] 1000 GGTGAGCCAATCCTG...GTGGAAATCTCAGCC 283788
[23053] 1000 AGCCCTCCACACAAG...TTCTTCTCTCCAAC 100507412
[23054] 1000 CGGGGCCAGGGAGT...AGGCCTCCTGGCTGC 728410
[23055] 1000 CGGGGCCAGGGAGT...AGGCCTCCTGGCTGC 100653046
[23056] 1000 CAGGCTGAGCCCTGC...CCGGGGCTCACC GCG 100288687
> |
```

```
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
gn <- sort(genes(txdb))
up1000 <- flank(gn, width=1000)
library(BSgenome.Hsapiens.UCSC.hg19)
genome <- BSgenome.Hsapiens.UCSC.hg19
up1000seqs <- getSeq(genome, up1000)
```

(Rで)塩基配列解析の記述法。  
上流配列の塩基数を任意に  
設定できて便利。

8. インストール済みのヒト("BSgenome.Hsapiens.UCSC.hg19")の転写開始点上流配列(1000bp)をmulti-FASTAファイルで保存したい場合:

2014年4月リリースのBioconductor 2.14での推奨手順です。ゲノムのパッケージ(例:BSgenome.Hsapiens.UCSC.hg19)と対応するアノテーションパッケージ(例:TxDb.Hsapiens.UCSC.hg19.knownGene)を読み込んで実行しています。

```
out_f <- "hoge8.fasta" #出力ファイル名を指定してout_fに格納
param1 <- "BSgenome.Hsapiens.UCSC.hg19" #パッケージ名を指定(BSgenome系のゲノムパッケージ)
param2 <- "TxDb.Hsapiens.UCSC.hg19.knownGene" #アノテーションパッケージ
param3 <- 1000 #上流 x

#前処理(指定したパッケージ中のオブジェクト名をgenomeに格納)
library(param1, character.only=T) #指定したパッケージをロード
tmp <- ls(paste("package", param1, sep=":")) #パッケージ中のオブジェクト名をリスト化
genome <- eval(parse(text=tmp)) #文字列をオブジェクトに変換

library(param2, character.only=T) #指定したパッケージをロード
tmp <- ls(paste("package", param2, sep=":")) #パッケージ中のオブジェクト名をリスト化
txdb <- eval(parse(text=tmp)) #文字列をオブジェクトに変換

#本番
gn <- sort(genes(txdb)) #遺伝子名をソート
hoge <- flank(gn, width=param3) #指定した幅で上流配列を抽出
fasta <- getSeq(genome, hoge) #指定したゲノムから配列を取得

#ファイルに保存
writeXStringSet(fasta, file=out_f, format="fasta")
```

```
R Console
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta")
> fasta
A DNASTringSet instance of length 23056
      width seq
[1] 1000 ACACATGCTAC...TTTTCGTTAA 100287102
[2] 1000 GCTATTATCAC...GAATAACTCT 79501
[3] 1000 GAATTAGGCTT...AAAGGCGGGG 643837
[4] 1000 CGGGGAGCCCC...GCTTGGGCCA 148398
[5] 1000 CGGCGGGGCTC...GAGCGGCGGG 339451
... ..
[23052] 1000 GGTGAGCCAAT...AATCTCAGCC 283788
[23053] 1000 AGCCCTCACA...CCTCTCCAAC 100507412
[23054] 1000 CGGGGCCCAGG...TCCTGGCTGC 728410
[23055] 1000 CGGGGCCCAGG...TCCTGGCTGC 100653046
[23056] 1000 CAGGCTGAGCC...GCTCACCGCG 100288687
> |
```

# バージョン違いの影響

sessionInfo()でR環境情報を取得

```
> sessionInfo()
R version 3.1.0 (2014-04-10)
Platform: x86_64-w64-mingw32/x64 (64-bit)

locale:
[1] LC_COLLATE=Japanese_Japan.932  LC_CTYPE=Japanese_Japan.932
[3] LC_MONETARY=Japanese_Japan.932 LC_NUMERIC=C
[5] LC_TIME=Japanese_Japan.932

attached base packages:
[1] parallel  stats          graphics  grDevices  utils      datasets  methods   base

other attached packages:
 [1] TxDb.Hsapiens.UCSC.hg19.knownGene_2.14.0  GenomicFeatures_1.16.0
 [3] AnnotationDbi_1.26.0                      Biobase_2.24.0
 [5] BSgenome.Hsapiens.UCSC.hg19_1.3.99        BSgenome_1.32.0
 [7] Biostrings_2.32.0                          XVector_0.4.0
 [9] GenomicRanges_1.16.1                      GenomeInfoDb_1.0.2
[11] IRanges_1.22.3                             BiocGenerics_0.10.0

loaded via a namespace (and not attached):
 [1] BatchJobs_1.2                BBmisc_1.5                    BiocParallel_0.6.0
 [4] biomaRt_2.20.0              bitops_1.0-6                  brew_1.0-6
 [7] codetools_0.2-8            DBI_0.2-7                     digest_0.6.4
[10] fail_1.2                    foreach_1.4.2                 GenomicAlignments_1.0.0
[13] iterators_1.0.7            plyr_1.8.1                    Rcpp_0.11.1
[16] RCurl_1.95-4.1             Rsamtools_1.16.0             RSQLite_0.11.4
[19] rtracklayer_1.24.0        sendmailR_1.1-2              stats4_3.1.0
[22] stringr_0.6.2              tools_3.1.0                   XML_3.98-1.1
[25] zlibbioc_1.10.0
```

# バージョン違いの影響

- 現在から未来: Bsgenome.Hsapiens.UCSC.hg19パッケージ
  - R ver. 3.0.3 (2014年3月リリース)
    - Bioconductor ver. 2.13: パッケージ内に上流配列情報が格納?!されている
  - R ver. 3.1.0 (2014年4月リリース)
    - Bioconductor ver. 2.14: 移行期(Transcript DB形式のオブジェクト利用を推奨)
  - R ver. 3.X.Y (2014年10月リリース?!)
    - Bioconductor ver. 2.15:

R ver. 3.1.0では、2009年2月リリースのヒトゲノム(hg19)パッケージでは警告は出るもののupstream1000で取得できる。しかし、2013年12月リリースのヒトゲノム(GRCh38)パッケージではすでに取得できなくなっているなど混沌としています。

6. インストール済みのヒト("BSgenome.Hsapiens.NCBI.GRCh38")の転写開始点上流配列(1000bp)をmulti-FASTAファイルで保存したい場合:

2013年12月にリリースされたGenome Reference Consortium GRCh38のRパッケージです。上流配列情報を含まないのがエラーがでます。

```

out_f <- "hoge6.fasta" #出力ファイル名を指定してout_fに格納
param1 <- "BSgenome.Hsapiens.NCBI.GRCh38" #パッケージ名
param2 <- "upstream1000" #上流1000bpを指定

#必要なパッケージをロード
library(param1, character.only=T) #param1で指定

#前処理(param1で指定したパッケージ中のオブジェクト名をhogeとする)
#tmp <- unlist(strsplit(param1, ".", fixed=TRUE))[2]
tmp <- ls(paste("package", param1, sep=":")) #param1で指定したパッケージ中のオブジェクト名
hoge <- eval(parse(text=tmp)) #文字列tmpをRオブジェクトとして評価
hoge #確認してるだけです(ここで、multiple sequencesのところにparam2で指定した文字列)

#本番
tmp <- paste("hoge$", hoge, param2, sep="")
fasta <- eval(parse(text=tmp))

#ファイルに保存
writeXStringSet(fasta, out_f)

```

R ver. 3.1.0では、2009年2月リリースのヒトゲノム(hg19)パッケージでは警告は出るもののupstream1000で取得できる。しかし、2013年12月リリースのヒトゲノム(GRCh38)パッケージではすでに取得できなくなっているなど混沌としています、**の実例。**

```

R Console
|   HSCR19KIR_RP5_B_HAP_CTG3_1
|
| (use the '$' or '[' operator to access a given
| sequence)
>
> #本番
> tmp <- paste("hoge$", hoge, param2, sep="") #param2で指定した文字列を含む$
> fasta <- eval(parse(text=tmp)) #文字列tmpをRオブジェクトとし$
以下にエラー x[[name]] : no such sequence
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", width=50) #fastaの$
以下にエラー is(x, "XStringSet") : オブジェクト 'fasta' がありません
> |

```

# Contents

- 3-4. R Bioconductor I、2014/09/09 10:30-14:45、中級、実習
  - Tips: setwd関数を利用した効率的な作業ディレクトリの変更
  - データの型1: translate関数の入力情報(AAStringSet)
  - Tips: オブジェクトの消去
  - データの型2: 翻訳配列取得のコードの中身を解説
  - バージョン情報把握とバージョンアップ
  - バージョン違いの影響
    - 過去から現在: BiostringsパッケージのreadDNAStrngSet関数
    - 現在から未来: BSgenome.Hsapiens.UCSC.hg19パッケージでプロモーター配列取得
  - Bioconductor概観



```
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
gn <- sort(genes(txdb))
up1000 <- flank(gn, width=1000)
library(BSgenome.Hsapiens.UCSC.hg19)
genome <- BSgenome.Hsapiens.UCSC.hg19
up1000seqs <- getSeq(genome, up1000)
```

TxDbというパッケージは、R内での統一規格でアノテーション情報を格納したもの。一般にはGTF/GFF形式のアノテーションファイルが用いられる。

### 8. インストール済みのヒト("BSgenome.Hsapiens.UCSC.hg19")の転写開始点上流配列

2014年4月リリースのBioconductor 2.14での推奨手順です。ゲノムのパッケージ(例:BSgenome.Hsapiens.UCSC.hg19)と対応するアノテーションパッケージ(例:TxDb.Hsapiens.UCSC.hg19.knownGene)を読み込んで実行しています。

```
out_f <- "hoge8.fasta" #出力ファイル名を指定してout_fに格納
param1 <- "BSgenome.Hsapiens.UCSC.hg19" #パッケージ名を指定(BSgenome系のゲノムパッケージ)
param2 <- "TxDb.Hsapiens.UCSC.hg19.knownGene" #アノテーションパッケージ名を指定
param3 <- 1000 #上流x下流の長さ

#前処理(指定したパッケージ中のオブジェクト名をgenomeに格納)
library(param1, character.only=T) #指定したパッケージを読み込む
tmp <- ls(paste("package", param1, sep=":")) #パッケージ中のオブジェクト名をリスト化
genome <- eval(parse(text=tmp)) #文字列をRオブジェクトに変換

library(param2, character.only=T) #指定したパッケージを読み込む
tmp <- ls(paste("package", param2, sep=":")) #パッケージ中のオブジェクト名をリスト化
txdb <- eval(parse(text=tmp)) #文字列をRオブジェクトに変換

#本番
gn <- sort(genes(txdb)) #遺伝子
hoge <- flank(gn, width=param3) #指定した長さで flank
fasta <- getSeq(genome, hoge) #指定したパッケージで配列取得

#ファイルに保存
writeXStringSet(fasta, file=out_f, format="fasta")
```

```
R Console
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta")
> fasta
A DNAStringSet instance of length 23056
      width seq
[1] 1000 ACACATGCTAC...TTTTCGTTAA 100287102
[2] 1000 GCTATTATCAC...GAATAACTCT 79501
[3] 1000 GAATTAGGCTT...AAAGGCGGGG 643837
[4] 1000 CGGGGAGCCCC...GCTTGGGCCA 148398
[5] 1000 CGGCGGGGCTC...GAGCGGCGGG 339451
... ..
[23052] 1000 GGTGAGCCAAT...AATCTCAGCC 283788
[23053] 1000 AGCCCTCACA...CCTCTCAAC 100507412
[23054] 1000 CGGGGCCCAGG...TCCTGGCTGC 728410
[23055] 1000 CGGGGCCCAGG...TCCTGGCTGC 100653046
[23056] 1000 CAGGCTGAGCC...GCTCACCGCG 100288687
> |
```

# (Rで)塩基配列解析

～NGS、RNA-seq、ゲノム、トランスクリプトーム、正規化、発現変動、統計、モデル、バイオインフォマティクス～

(last modified 2014/07/14, since 2010)

What's  
この  
すの  
2014  
ンフ  
2014  
門田  
マツ  
ニック  
した  
2014  
東大  
申込  
)m  
参考

- インタロ | 一般 | Tips | [任意の拡張子でファイルを保存](#) (last modified 2013/09/26)
- インタロ | 一般 | Tips | [拡張子は同じで任意の文字を追加して保存](#) (last modified 2013/09/26)
- インタロ | 一般 | 配列取得 | ゲノム配列 | [公共DBから](#) (last modified 2014/05/28)
- インタロ | 一般 | 配列取得 | ゲノム配列 | [BSgenome](#) (last modified 2014/06/28) **NEW**
- インタロ | 一般 | 配列取得 | プロモーター配列 | [公共DBから](#) (last modified 2014/04/01)
- インタロ | 一般 | 配列取得 | プロモーター配列 | [BSgenome](#) (last modified 2014/04/25)
- インタロ | 一般 | 配列取得 | プロモーター配列 | [GenomicFeatures\(Lawrence 2013\)](#) (last modified 2013/10/08)
- インタロ | 一般 | 配列取得 | トランスクリプトーム配列 | [公共DBから](#) (last modified 2014/04/01)
- インタロ | 一般 | 配列取得 | トランスクリプトーム配列 | [biomaRt\(Durinck 2009\)](#) (last modified 2013/09/26)
- インタロ | NGS | [様々なプラットフォーム](#) (last modified 2014/06/10)
- インタロ | NGS | [qPCRやmicroarrayなどとの比較](#) (last modified 2014/07/11) **NEW**
- インタロ | NGS | [可視化\(ゲノムブラウザやViewer\)](#) (last modified 2014/06/25) **NEW**
- インタロ | NGS | 配列取得 | FASTQ or SRALite | [公共DBから](#) (last modified 2014/06/28) **NEW**
- インタロ | NGS | 配列取得 | FASTQ or SRALite | [SRADB\(Zhu 2013\)](#) (last modified 2014/06/26) **NEW**
- インタロ | NGS | 配列取得 | シミュレーションデータ | [シミュレーションデータについて](#) (last modified 2014/06/25) **NEW**
- インタロ | NGS | 配列取得 | シミュレーションデータ | [ランダムな塩基配列の生成から](#) (last modified 2014/06/23) **NEW**
- **インタロ | NGS | アノテーション情報取得 | [アノテーション情報取得について](#) (last modified 2014/03/26)**
- インタロ | NGS | アノテーション情報取得 | [GFF/GTF形式ファイル](#) (last modified 2014/04/11)
- インタロ | NGS | アノテーション情報取得 | [refFlat形式ファイル](#) (last modified 2013/09/25)
- インタロ | NGS | アノテーション情報取得 | [biomaRt\(Durinck 2009\)](#) (last modified 2013/09/26)
- **インタロ | NGS | アノテーション情報取得 | [TranscriptDb](#) | [TranscriptDbについて](#) (last modified 2014/03/28)**
- インタロ | NGS | アノテーション情報取得 | TranscriptDb | [TxDb.\\*から](#) (last modified 2013/10/08)
- インタロ | NGS | アノテーション情報取得 | TranscriptDb | [GenomicFeatures\(Lawrence 2013\)](#) (last modified 2013/10/08)
- インタロ | NGS | アノテーション情報取得 | TranscriptDb | [GFF/GTF形式ファイルから](#) (last modified 2014/04/01)
- インタロ | NGS | 読み込み | FASTA形式 | [基本情報を取得](#) (last modified 2014/05/29)
- インタロ | NGS | 読み込み | FASTA形式 | [description行の記述を整形](#) (last modified 2014/04/05)
- インタロ | NGS | 読み込み | FASTQ形式 | [FASTQ形式](#) (last modified 2014/06/15)
- インタロ | NGS | 読み込み | FASTQ形式 | [description行の記述を整形](#) (last modified 2013/06/13)
- インタロ | NGS | 読み込み | [Illuminaの \\* seq.txt](#) (last modified 2013/06/13)
- インタロ | NGS | 読み込み | [Illuminaの \\* qseq.txt](#) (last modified 2013/06/17)

TxDbというパッケージは、R内での統一規格でアノテーション情報を格納したものである。一般にはGTF/GFF形式のアノテーションファイルが提供されているので、それらを読み込んでTxDbオブジェクトに変換するための関数なども用意されている。

# GFF/GTF形式ファイルの例

遺伝子ごとに、どの染色体のどの座標上に存在するのかなどの情報を含むタブ区切りテキストファイル

GFF3形式ファイルの例 (シロイヌナズナ; TAIR10\_GFF3\_genes.gff)

▲	A	B	C	D	E	F	G	H	I
1	Chr1	TAIR10	chromosome	1	30427671	.	.	.	ID=Chr1;Name=Chr1
2	Chr1	TAIR10	gene	3631	5899	.	+	.	ID=AT1G01010;Note=protein_coding_gene;Name=AT1G01010
3	Chr1	TAIR10	mRNA	3631	5899	.	+	.	ID=AT1G01010.1;Parent=AT1G01010;Name=AT1G01010.1;Index=1
4	Chr1	TAIR10	protein	3760	5630	.	+	.	ID=AT1G01010.1-Protein;Name=AT1G01010.1;Derives_from=AT1G01010.1
5	Chr1	TAIR10	exon	3631	3913	.	+	.	Parent=AT1G01010.1
6	Chr1	TAIR10	five_prime_UTR	3631	3759	.	+	.	Parent=AT1G01010.1
7	Chr1	TAIR10	CDS	3760	3913	.	+	0	Parent=AT1G01010.1,AT1G01010.1-Protein;
8	Chr1	TAIR10	exon	3996	4276	.	+	.	Parent=AT1G01010.1
9	Chr1	TAIR10	CDS	3996	4276	.	+	2	Parent=AT1G01010.1,AT1G01010.1-Protein;
10	Chr1	TAIR10	exon	4486	4605	.	+	.	Parent=AT1G01010.1
11	Chr1	TAIR10	CDS	4486	4605	.	+	0	Parent=AT1G01010.1,AT1G01010.1-Protein;
12	Chr1	TAIR10	exon	4706	5095	.	+	.	Parent=AT1G01010.1

GTF形式ファイルの例 (ゼブラフィッシュ; Danio\_rerio.Zv9.75.gtf)

▲	A	B	C	D	E	F	G	H	I
1									#!genome-build Zv9
2									#!genome-version Zv9
3									#!genome-date 2010-04
4									#!genome-build-accession NCBI:GCA_000002035.2
5									#!genebuild-last-updated 2014-02
6	7	protein_coding	gene	100958	101715	.	+	.	gene_id "ENSDARG00000076051"; gene_name "CABZ01062994.1"; gene
7	7	protein_coding	transcript	100958	101715	.	+	.	gene_id "ENSDARG00000076051"; transcript_id "ENSDART00000113409
8	7	protein_coding	exon	100958	100975	.	+	.	gene_id "ENSDARG00000076051"; transcript_id "ENSDART00000113409
9	7	protein_coding	CDS	100958	100975	.	+	0	gene_id "ENSDARG00000076051"; transcript_id "ENSDART00000113409
10	7	protein_coding	exon	101077	101715	.	+	.	gene_id "ENSDARG00000076051"; transcript_id "ENSDART00000113409
11	7	protein_coding	CDS	101077	101715	.	+	0	gene_id "ENSDARG00000076051"; transcript_id "ENSDART00000113409
12	7	protein_coding	gene	116160	117573	.	+	.	gene_id "ENSDARG00000088691"; gene_name "BX511027.1"; gene_sour
13	7	protein_coding	transcript	116160	117573	.	+	.	gene_id "ENSDARG00000088691"; transcript_id "ENSDART00000129330

任意の染色体のみ、遺伝子名をもつもののみ抽出したい場合は、この項目をテンプレートとして利用可能。

- ・ 書籍 | 日本乳酸菌学会誌 | [第1回イントロダクション](#) (last modified 2014/07/07) NEW
- ・ イントロ | 一般 | [ランダムに行を抽出](#) (last modified 2014/07/17) NEW
- ・ イントロ | 一般 | [任意の文字列を行の最初に挿入](#) (last modified 2014/07/17) NEW
- ・ このウェブサイトで、[任意のキーワードを含む行を抽出\(基礎\)](#) (last modified 2014/04/11)
- ・ [ランダムな塩基配列を生成](#) (last modified 2014/06/16)
- ・ [任意の長さの可能な全ての塩基配列を作成](#) (last modified 2013/06/14)

## イントロ | 一般 | 任意のキーワードを含む行を抽出(基礎)

例えばタブ区切りテキストファイルが手元があり、この中からリストファイル中の文字列を含む行を抽出するやり方を示します。Linux (UNIX)のgrepコマンドのようなものであり、perlのハッシュのようなものです。

「ファイル」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

1. 目的のタブ区切りテキストファイル([annotation.txt](#))中の第1列目をキーとして、リストファイル([genelist1.txt](#))中のものが含まれる行全体を出力したい場合:

```

in_f1 <- "annotation.txt" #入力ファイル名を指定してin_f1に格納(アノテーション)
in_f2 <- "genelist1.txt" #入力ファイル名を指定してin_f2に格納(リストファイル)
out_f <- "hogel.txt" #出力ファイル名を指定してout_fに格納
param <- 1 #アノテーションファイル中の検索したい列番号を指定

#入力ファイルの読み込み
data <- read.table(in_f1, header=TRUE, sep="\t", quote="") #in_f1で指定したファイルの読み込み
keywords <- readLines(in_f2) #in_f2で指定したファイルの読み込み
dim(data) #オブジェクトdataの行数と列数を表示

#本番
obj <- is.element(as.character(data[,param]), keywords) #条件を満たすかどうかを判定した結果
out <- data[obj,] #objがTRUEとなる行のみ抽出した結果をoutに格納
dim(out) #オブジェクトoutの行数と列数を表示

#ファイルに保存
write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F) #outの中身を指定したファイルに保存

```

# (Rで)塩基配列解析

～NGS、RNA-seq、ゲノム、トランスクリプトーム、正規化、発現変動、統計、モデル、バイオインフォマティクス～

(last modified 2014/07/14, since 2010)

What's  
この  
すの  
2014  
ンフ  
2014  
門田  
マツ  
ニック  
した  
2014  
東大  
申込  
)m  
参考

- インタロ | 一般 | Tips | [任意の拡張子でファイルを保存](#) (last modified 2013/09/2)
- インタロ | 一般 | Tips | [拡張子は同じで任意の文字を追加して保存](#) (last modified 2013/09/2)
- インタロ | 一般 | 配列取得 | [ゲノム配列](#) | [公共DBから](#) (last modified 2014/05/2)
- インタロ | 一般 | 配列取得 | [ゲノム配列](#) | [BSgenome](#) (last modified 2014/06/28)
- インタロ | 一般 | 配列取得 | [プロモーター配列](#) | [公共DBから](#) (last modified 2014/06/28)
- インタロ | 一般 | 配列取得 | [プロモーター配列](#) | [BSgenome](#) (last modified 2014/06/28)
- インタロ | 一般 | 配列取得 | [プロモーター配列](#) | [GenomicFeatures\(Lawrence 2013\)](#) (last modified 2014/06/28)
- インタロ | 一般 | 配列取得 | [トランスクリプトーム配列](#) | [公共DBから](#) (last modified 2014/06/28)
- インタロ | 一般 | 配列取得 | [トランスクリプトーム配列](#) | [biomaRt\(Durinck 2009\)](#) (last modified 2014/06/28)
- インタロ | NGS | [様々なプラットフォーム](#) (last modified 2014/06/10)
- インタロ | NGS | [qPCRやmicroarrayなどとの比較](#) (last modified 2014/07/11) NEW
- インタロ | NGS | [可視化\(ゲノムブラウザやViewer\)](#) (last modified 2014/06/25) NEW
- インタロ | NGS | 配列取得 | [FASTQ or SRALite](#) | [公共DBから](#) (last modified 2014/06/28) NEW
- インタロ | NGS | 配列取得 | [FASTQ or SRALite](#) | [SRADB\(Zhu 2013\)](#) (last modified 2014/06/26) NEW
- インタロ | NGS | 配列取得 | [シミュレーションデータ](#) | [について](#) (last modified 2014/06/25) NEW
- インタロ | NGS | 配列取得 | [シミュレーションデータ](#) | [ランダムな塩基配列の生成から](#) (last modified 2014/06/23) NEW
- [インタロ](#) | NGS | [アノテーション情報取得](#) | [について](#) (last modified 2014/03/26)
- インタロ | NGS | [アノテーション情報取得](#) | [GFF/GTF形式ファイル](#) (last modified 2014/04/11)
- インタロ | NGS | [アノテーション情報取得](#) | [refFlat形式ファイル](#) (last modified 2013/09/25)
- インタロ | NGS | [アノテーション情報取得](#) | [biomaRt\(Durinck 2009\)](#) (last modified 2013/09/26)
- [インタロ](#) | NGS | [アノテーション情報取得](#) | [TranscriptDb](#) | [について](#) (last modified 2014/03/28)
- インタロ | NGS | [アノテーション情報取得](#) | [TranscriptDb](#) | [TxDb.\\*から](#) (last modified 2013/10/08)
- インタロ | NGS | [アノテーション情報取得](#) | [TranscriptDb](#) | [GenomicFeatures\(Lawrence 2013\)](#) (last modified 2013/10/08)
- インタロ | NGS | [アノテーション情報取得](#) | [TranscriptDb](#) | [GFF/GTF形式ファイルから](#) (last modified 2014/04/01)
- インタロ | NGS | 読み込み | [FASTA形式](#) | [基本情報を取得](#) (last modified 2014/05/29)
- インタロ | NGS | 読み込み | [FASTA形式](#) | [description行の記述を整形](#) (last modified 2014/04/05)
- インタロ | NGS | 読み込み | [FASTQ形式](#) (last modified 2014/06/15)
- インタロ | NGS | 読み込み | [FASTQ形式](#) | [description行の記述を整形](#) (last modified 2013/06/13)
- インタロ | NGS | 読み込み | [Illuminaの \\* seq.txt](#) (last modified 2013/06/13)
- インタロ | NGS | 読み込み | [Illuminaの \\* qseq.txt](#) (last modified 2013/06/17)

1. (Rで)塩基配列解析で提供されている様々な項目を眺め、多くの例題をこなし、できることの全体像を把握。
2. エラー例とその対処法を身につける。基本戦略は「?関数名」や検索エンジンなどを駆使。やりたいことの多くは基本テクニックの組合せで可能。
3. Bioconductorのパッケージリストを眺め、知識やテクニックの幅を広げる。

# Bioconductor概観

## (Rで)塩基配列解析

～NGS、RNA-seq、ゲノム、トランスクリプトーム、正規化、発現変動、

(last modified 2012/09/10)

作図 | [M-A plot\(基本編\)](#) (last modified 2012/10/01)

作図 | [M-A plot](#)

作図 | [ROC曲線](#)

作図 | [SplicingG](#)

パイプライン | [パイプライン](#)

### リンク集

- [R](#)
- [Bioconductor: Gentleman et al., Genome Biol., 2004](#)
- [CRAN](#)
- [RipWiki](#)
- [R Tips\(竹澤様\)](#)
- [BioEdit\(フリーの配列編集ソフト\)](#)
- [BioMart: Smedley et al., BMC](#)
- [DDBJ Read Annotation Pipeline](#)
- [EMBOSS explorer \(EMBOSS\)](#)
- [Biostar: Parnell et al., PLoS Co](#)
- [SEQanswers: Li et al., Bioinfor](#)
- [NGS WikiBook: Li et al., Brief](#)
- [HT Sequence Analysis with R](#)

NGSに特化した内容ではないが、R Tipsは門田が独学でRを勉強し始めた2005年頃から今でも時々お世話になっているウェブサイト。

PubMed上で「R Bioconductor」でキーワード検索し原著論文があるパッケージのみ探すのも一つの戦略ですが、原著論文公開前のパッケージも見つかります。

Bioconductor  
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Home Install Help Developers About

Search:

### About Bioconductor

Bioconductor provides tools for the analysis and comprehension of high-throughput genomic data. Bioconductor uses the R statistical programming language, and is open source and open development. It has two releases each year, [824 software packages](#), and an active user community. Bioconductor is also available as an [Amazon Machine Image \(AMI\)](#).

Install »  
Get started with *Bioconductor*

- [Install Bioconductor](#)
- [Explore packages](#)
- [Mailing list](#)
- [Latest newsletter](#)
- [Follow us on Twitter](#)
- [Using R](#)

Learn »  
Master *Bioconductor* tools

- [Recent courses](#)
- [Package vignettes](#)
- [Common work flows](#)
- [FAQ](#)
- [Community resources](#)
- [Videos](#)

Use »  
Create bioinformatic solutions with *Bioconductor*

- [Software, Annotation, and Experiment packages](#)
- [Amazon Machine Image](#)
- [Latest release announcement](#)
- [Subscribe to the mailing list](#)

Develop »  
Contribute to *Bioconductor*

- [Use Bioc `devel`](#)
- [`Devel` Software, Annotation and Experiment packages](#)
- [Package guidelines](#)
- [New package submission](#)
- [Developer resources](#)
- [Build reports](#)

# Bioconductor概観

The screenshot shows the Bioconductor website interface. The browser address bar displays the URL: `http://bioconductor.org/packages/release/BiocViews.html#___Software`. The page header includes the Bioconductor logo and navigation links: Home, Install, Help, Developers, and About. A search bar is visible in the top right. The main content area shows the breadcrumb `Home » BiocViews` and the heading `All Packages`. Below this, it indicates `Bioconductor version 2.14 (Release)` and an `Autocomplete biocViews search:` input field. At the bottom of the page, a green footer contains contact information and logos for Fred Hutchinson Cancer Research Center and Bioconductor. A red arrow points to a JavaScript error message in the browser's console at the bottom of the page: `bioconductor.org は、長時間実行中のスクリプトが原因で応答しません。 スクリプトの停止(S) x`.

まずは、数分待ちます。このようなフリーズ局面にときどき遭遇しますが、四隅のどこかをいじるなどしてウィンドウサイズを変えると復旧します。「スクリプトの停止」ボタンを押す必要はありません。この後の作業をやると体感できますが、リストアップするパッケージ数に比例して表示に時間がかかるようです。最初にすごく時間がかかるのは824パッケージ全てをリストアップしているためです。



Home

Install

Help

Develop

「RNA-seq」など、フリーワード検索をやってもいいとは思いますが、経験上あまりうまく引っかけっこないので私はやりません。

Home » BiocViews

## All Packages

Bioconductor version 2.14 (Release)

Autocomplete biocViews search:

Packages found under Software:

Show  entries

Search table:

- Software (824)
  - AssayDomain (251)
  - BiologicalQuestion (204)
  - Infrastructure (170)
  - ResearchField (151)
  - StatisticalMethod (208)
  - Technology (511)
  - WorkflowStep (406)
- AnnotationData (867)
- ExperimentData (202)

Package	Maintainer	Title
<a href="#">a4</a>	Tobias Verbeke, Willem Ligtenberg	Automated Affymetrix Array Analysis Umbrella Package
<a href="#">a4Base</a>	Tobias Verbeke, Willem Ligtenberg	Automated Affymetrix Array Analysis Base Package
<a href="#">a4Classif</a>	Tobias Verbeke, Willem Ligtenberg	Automated Affymetrix Array Analysis Classification Package
<a href="#">a4Core</a>	Tobias Verbeke, Willem Ligtenberg	Automated Affymetrix Array Analysis Core Package
<a href="#">a4Preproc</a>	Tobias Verbeke, Willem Ligtenberg	Automated Affymetrix Array Analysis Preprocessing Package
<a href="#">a4Reporting</a>	Tobias Verbeke, Willem Ligtenberg	Automated Affymetrix Array Analysis Reporting Package
<a href="#">ABarray</a>	Yongming Andrew Sun	Microarray QA and analysis for Applied Survey Microarray expression data.
<a href="#">ABSSeq</a>	Wentao Yang	ABSSeq: a new RNA-seq method based on a differences and gene model

基本的には左側のカテゴリ分けのところを眺めますが、Biostringsなど何を行うパッケージかがある程度分かっているものから逆引きして感覚をつかんでおくとよいでしょう。

http://bioconductor.org/packages/release/BiocViews.html#\_\_\_Software

Bioconductor - BiocViews x

Bioconductor  
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Home Install Help

Search: **biostrings**

Home » BiocViews

## All Packages

**Bioconductor version 2.14 (Release)**

Autocomplete biocViews search:

▼ Software (824)

- ▶ AssayDomain (251)
- ▶ BiologicalQuestion (204)
- ▶ Infrastructure (170)
- ▶ ResearchField (151)
- ▶ StatisticalMethod (208)
- ▶ Technology (511)
- ▶ WorkflowStep (406)
- ▶ AnnotationData (867)
- ▶ ExperimentData (202)

**Packages found under Software:**

Show  entries

Search table:

Package	Maintainer	Title
<a href="#">Biostrings</a>	H. Pages	String objects representing biological sequences, and matching algorithms
<a href="#">BSgenome</a>	H. Pages	Infrastructure for Biostrings-based genome data packages

Showing 1 to 2 of 2 entries (filtered from 826 total entries)

◀ Previous Next ▶

「biostrings」と打つとすぐにリストアップされる。



BioconductorのBiostringsパッケージのページに飛びます。

Home » Bioconductor 2.14 » Software Packages » Biostrings

# Biostrings

## String objects representing biological sequences, and matching algorithms

Bioconductor version: Release (2.14)

Memory efficient string containers, string matching algorithms, and other utilities, for fast manipulation of large biological sequences or sets of sequences.

Author: H. Pages, P. Aboyoun, R. Gentleman, and S. DebRoy

Maintainer: H. Pages <hpages at fhcrc.org>

Citation (from within R, enter `citation("Biostrings")`):

Pages H, Aboyoun P, Gentleman R and DebRoy S. *Biostrings: String objects representing biological sequences, and matching algorithms*. R package version 2.32.1.

### Installation

To install this package, start R and enter:

```
source("http://bioconductor.org/biocLite.R")
biocLite("Biostrings")
```

### Documentation

To view documentation for the version of this package installed in your system, start R and enter:

**Workflows >>**

Common Bioconductor workflows include:

- [Oligonucleotide Arrays](#)
- [High-throughput Sequencing](#)
- [Counting Reads for Differential Expression](#) (parathyroideSE vignette)
- [Annotation](#)
- [Annotating Variants](#)
- [Annotating Ranges](#)
- [Flow Cytometry](#) and other assays
- [Candidate Binding Sites for Known Transcription Factors](#)
- [Cloud-enabled cis-eQTL search and annotation](#)

**Mailing Lists >>**

Post questions about Bioconductor packages to our mailing lists. Read the [posting guide](#) before posting!

- [bioconductor](#)
- [bioc-devel](#)

### Documentation

To view documentation for the version of this package installed in your system, start R and enter:

```
browseVignettes("Biostrings")
```

- [PDF](#) [R Script](#) A short presentation of the basic classes defined in Biostrings 2
- [PDF](#) [R Script](#) Biostrings Quick Overview
- [PDF](#) [R Script](#) Handling probe sequence information
- [PDF](#) [R Script](#) Multiple Alignments
- [PDF](#) [R Script](#) Pairwise Sequence Alignments
- [PDF](#) Reference Manual
- [Text](#) NEWS

### Details

biocViews	<a href="#">DataImport</a> , <a href="#">DataRepresentation</a> , <a href="#">Genetics</a> , <a href="#">Infrastructure</a> , <a href="#">SequenceMatching</a> , <a href="#">Sequencing</a> , <a href="#">Software</a>
Version	2.32.1
In Bioconductor since	BioC 1.6 (R-2.1) or earlier
License	Artistic-2.0
Depends	R (>= 2.8.0), methods, <a href="#">BiocGenerics</a> (>= 0.5.4), <a href="#">IRanges</a> (>= 1.21.35), <a href="#">XVector</a> (>= 0.3.6)
Imports	graphics, methods, stats, utils, <a href="#">BiocGenerics</a> , <a href="#">IRanges</a> , <a href="#">XVector</a> , <a href="#">zlibbioc</a>
Suggests	<a href="#">BSgenome</a> (>= 1.13.14), <a href="#">BSgenome.Celegans.UCSC.ce2</a> (>= 1.3.11), <a href="#">BSgenome.Dmelanogaster.UCSC.dm3</a> (>= 1.3.11), <a href="#">BSgenome.Hsapiens.UCSC.hq18</a> , <a href="#">drosophila2probe</a> , <a href="#">hqu95av2probe</a> , <a href="#">hqu133aprobe</a> , <a href="#">GenomicFeatures</a> (>= 1.3.14), <a href="#">hqu95av2cdf</a> , <a href="#">affy</a> (>= 1.41.3), <a href="#">affydata</a> (>= 1.11.5), <a href="#">RUnit</a>
System Requirements	
URL	
Depends On Me	<a href="#">altcdfenvs</a> , <a href="#">Basic4Cseq</a> , <a href="#">BRAIN</a> , <a href="#">BSgenome</a> , <a href="#">ChIPpeakAnno</a> , <a href="#">ChIPsim</a> , <a href="#">cleaver</a> , <a href="#">CRISPRseek</a> , <a href="#">DASiR</a> , <a href="#">DECIPHER</a> , <a href="#">deepSNV</a> , <a href="#">FDb.FANTOM4.promoters.hq19</a> , <a href="#">FDb.InfiniumMethylation.hq18</a> , <a href="#">FDb.InfiniumMethylation.hq19</a> , <a href="#">GeneRegionScan</a> , <a href="#">genomes</a> , <a href="#">GenomicAlignments</a> , <a href="#">GOTHic</a> , <a href="#">harbChIP</a> , <a href="#">iPAC</a> , <a href="#">JASPAR2014</a> , <a href="#">methVisual</a> , <a href="#">minfi</a> , <a href="#">MotifDb</a> , <a href="#">motifRG</a> , <a href="#">oligo</a> , <a href="#">oneChannelGUI</a> , <a href="#">pd.hugene.2.0.st</a> , <a href="#">pd.hugene.2.1.st</a> , <a href="#">pd.moqene.2.0.st</a> , <a href="#">pd.moqene.2.1.st</a> , <a href="#">pd.nuqo.hs1a520180</a> , <a href="#">pd.nuqo.mm1a520177</a> ,



BioconductorのBiostringsパッケージのページで、ちょっと下のほうに移動。biocViewsのところで見えるキーワードっぽいのがさきほどのカテゴリ分けに相当。例えば、DataRepresentationをクリックすると…。

Home » BiocViews

# All Packages

Bioconductor version 2.14 (Release)

Autocomplete biocViews search:

Packages found under DataRepresentation:

Show  entries

- ▼ Software (824)
  - ▶ AssayDomain (251)
  - ▶ BiologicalQuestion (204)
  - ▼ Infrastructure (170)
    - DataImport (74)
    - DataRepresentation (26)**
    - GUI (14)
    - ThirdPartyClient (9)
    - ▶ ResearchField (151)
    - ▶ StatisticalMethod (208)
    - ▶ Technology (511)
    - ▶ WorkflowStep (406)
  - ▶ AnnotationData (867)
  - ▶ ExperimentData (202)

Package	Maintainer	Description
<a href="#">AtlasRDF</a>	James Malone	Gene Expression Atlas query and gene set enrichment package.
<a href="#">BaseSpaceR</a>	Adrian Alexa	R SDK for BaseSpace RESTful API
<a href="#">bigmemoryExtras</a>	Peter M. Haverty	An extension of the bigmemory package with added safety, convenience, and a factor class.
<a href="#">Biostrings</a>	H. Pages	String objects representing biological sequences, and matching algorithms
<a href="#">BSgenome</a>	H. Pages	Infrastructure for Biostrings-based genome data packages
<a href="#">cummeRbund</a>	Loyal A. Goff	Analysis, exploration, manipulation, and visualization of Cufflinks high-throughput sequencing data.
<a href="#">flowPlots</a>	N. Hawkins	flowPlots: analysis plots and data class for gated flow cytometry data
<a href="#">flowWorkspace</a>	Greg Finak, Mike Jiang	Import flowJo Workspaces into BioConductor and replicate flowJo gating with flowCore
<a href="#">FunciSNP</a>	Simon G. Coetzee	Integrating Functional Non-coding Datasets with Genetic Association Studies to Identify Candidate Regulatory SNPs
	Gang Feng, Pan Du and	Extended Substitution of Gene

カテゴリ分けの階層関係がわかる。Gene Ontologyの階層分類と似たもの。DataRepresentationのカテゴリに含まれるのは26パッケージであることが分かる。赤枠部分がそのリスト。Biostringsは大分類はSoftware、中分類はInfrastructureとなっており、その下の階層のDataRepresentationに含まれていることがわかる。

### Documentation

To view documentation for the version of this package installed in your system, start R and enter:

```
browseVignettes("Biostrings")
```

- [PDF](#) [R Script](#) A short presentation of the basic classes defined in Biostrings 2
- [PDF](#) [R Script](#) Biostrings Quick Overview
- [PDF](#) [R Script](#) Handling probe sequence information
- [PDF](#) [R Script](#) Multiple Alignments
- [PDF](#) [R Script](#) Pairwise Sequence Alignments
- [PDF](#) Reference Manual
- [Text](#) NEWS

### Details

biocViews	<a href="#">DataImport</a> , <a href="#">DataRepresentation</a> , <a href="#">Genetics</a> , <a href="#">Infrastructure</a> , <a href="#">SequenceMatching</a> , <a href="#">Sequencing</a> , <a href="#">Software</a>
Version	2.32.1
In Bioconductor since	BioC 1.6 (R-2.1) or earlier
License	Artistic-2.0
Depends	R (>= 2.8.0), methods, <a href="#">BiocGenerics</a> (>= 0.5.4), <a href="#">IRanges</a> (>= 1.21.35), <a href="#">XVector</a> (>= 0.3.6)
Imports	graphics, methods, stats, utils, <a href="#">BiocGenerics</a> , <a href="#">IRanges</a> , <a href="#">XVector</a> , <a href="#">zlibbioc</a>
Suggests	<a href="#">BSgenome</a> (>= 1.13.14), <a href="#">BSgenome.Celegans.UCSC.ce2</a> (>= 1.3.11), <a href="#">BSgenome.Dmelanogaster.UCSC.dm3</a> (>= 1.3.11), <a href="#">BSgenome.Hsapiens.UCSC.hq18</a> , <a href="#">drosophila2probe</a> , <a href="#">hqu95av2probe</a> , <a href="#">hqu133aprobe</a> , <a href="#">GenomicFeatures</a> (>= 1.3.14), <a href="#">hqu95av2cdf</a> , <a href="#">affy</a> (>= 1.41.3), <a href="#">affydata</a> (>= 1.11.5), <a href="#">RUnit</a>
System Requirements	
URL	
Depends On Me	<a href="#">altcdfenvs</a> , <a href="#">Basic4Cseq</a> , <a href="#">BRAIN</a> , <a href="#">BSgenome</a> , <a href="#">ChIPpeakAnno</a> , <a href="#">ChIPsim</a> , <a href="#">cleaver</a> , <a href="#">CRISPRseek</a> , <a href="#">DASiR</a> , <a href="#">DECIPHER</a> , <a href="#">deepSNV</a> , <a href="#">FDb.FANTOM4.promoters.hq19</a> , <a href="#">FDb.InfiniumMethylation.hq18</a> , <a href="#">FDb.InfiniumMethylation.hq19</a> , <a href="#">GeneRegionScan</a> , <a href="#">genomes</a> , <a href="#">GenomicAlignments</a> , <a href="#">GOTHic</a> , <a href="#">harbChIP</a> , <a href="#">iPAC</a> , <a href="#">JASPAR2014</a> , <a href="#">methVisual</a> , <a href="#">minfi</a> , <a href="#">MotifDb</a> , <a href="#">motifRG</a> , <a href="#">oligo</a> , <a href="#">oneChannelGUI</a> , <a href="#">pd.hugene.2.0.st</a> , <a href="#">pd.hugene.2.1.st</a> , <a href="#">pd.moqene.2.0.st</a> , <a href="#">pd.moqene.2.1.st</a> , <a href="#">pd.nuqo.hs1a520180</a> , <a href="#">pd.nuqo.mm1a520177</a> ,

大分類のSoftware、中分類のInfrastructureも存在する。



[DataImport](#), [DataRepresentation](#), [Genetics](#), [Infrastructure](#), [SequenceMatching](#), [Sequencing](#), [Software](#)

DataRepresentation (26パッケージ)のときに比べ、Infrastructure (170パッケージ)の階層表示には時間がかかることがわかる。

Home » BiocViews

# All Packages

Bioconductor version 2.14 (Release)

Packages found under Infrastructure:

Autocomplete biocViews search:

Show  entries

Search table:

	Package	Maintainer	Title
▼ Software (824)			
▶ AssayDomain (251)			
▶ BiologicalQuestion (204)			
▼ Infrastructure (170)	<a href="#">aCGH</a>	Peter Dimitrov	Classes and functions for Array Comparative Genomic Hybridization data.
	<a href="#">affxparser</a>	Kasper Daniel Hansen	Affymetrix File Parsing SDK
	<a href="#">AffyCompatible</a>	Martin Morgan	Affymetrix GeneChip software compatibility
	<a href="#">affyContam</a>	V. Carey	structured corruption of affymetrix cel file data
	<a href="#">affyio</a>	Benjamin Milo Bolstad	Tools for parsing Affymetrix data files
	<a href="#">affyImGUI</a>	Keith Satterley	GUI for affy analysis using limma package
	<a href="#">AllelicImbalance</a>	Jesper R Gadin	Investigates allele specific expression
	<a href="#">AnnotationDbi</a>	Bioconductor Package Maintainer	Annotation Database Interface
	<a href="#">AnnotationForge</a>	Bioconductor Package Maintainer	Code for Building Annotation Database Packages
	<a href="#">AnnotationHub</a>	Marc Carlson	A client for retrieving Bioconductor objects from AnnotationHub
	<a href="#">aroma.light</a>	Henrik Bengtsson	Light-weight methods for normalization and visualization of microarray data using only basic R data types
			Access the ArrayExpress Microarray

### Documentation

To view documentation for the version of this package installed in your system, start R and enter:

```
browseVignettes("Biostrings")
```

- [PDF](#) [R Script](#) A short presentation of the basic classes defined in Biostrings 2
- [PDF](#) [R Script](#) Biostrings Quick Overview
- [PDF](#) [R Script](#) Handling probe sequence information
- [PDF](#) [R Script](#) Multiple Alignments
- [PDF](#) [R Script](#) Pairwise Sequence Alignments
- [PDF](#) Reference Manual
- [Text](#) NEWS

### Details



bioViews [DataImport](#), [DataRepresentation](#), [Genetics](#), [Infrastructure](#), [SequenceMatching](#), [Sequencing](#), [Software](#)

Version	2.32.1
In Bioconductor since	BioC 1.6 (R-2.1) or earlier
License	Artistic-2.0
Depends	R (>= 2.8.0), methods, <a href="#">BiocGenerics</a> (>= 0.5.4), <a href="#">IRanges</a> (>= 1.21.35), <a href="#">XVector</a> (>= 0.3.6)
Imports	graphics, methods, stats, utils, <a href="#">BiocGenerics</a> , <a href="#">IRanges</a> , <a href="#">XVector</a> , <a href="#">zlibbioc</a>
Suggests	<a href="#">BSgenome</a> (>= 1.13.14), <a href="#">BSgenome.Celegans.UCSC.ce2</a> (>= 1.3.11), <a href="#">BSgenome.Dmelanogaster.UCSC.dm3</a> (>= 1.3.11), <a href="#">BSgenome.Hsapiens.UCSC.hq18</a> , <a href="#">drosophila2probe</a> , <a href="#">hqu95av2probe</a> , <a href="#">hqu133aprobe</a> , <a href="#">GenomicFeatures</a> (>= 1.3.14), <a href="#">hqu95av2cdf</a> , <a href="#">affy</a> (>= 1.41.3), <a href="#">affydata</a> (>= 1.11.5), <a href="#">RUnit</a>
System Requirements	
URL	
Depends On Me	<a href="#">altcdfenvs</a> , <a href="#">Basic4Cseq</a> , <a href="#">BRAIN</a> , <a href="#">BSgenome</a> , <a href="#">ChIPpeakAnno</a> , <a href="#">ChIPsim</a> , <a href="#">cleaver</a> , <a href="#">CRISPRseek</a> , <a href="#">DASiR</a> , <a href="#">DECIPHER</a> , <a href="#">deepSNV</a> , <a href="#">FDb.FANTOM4.promoters.hq19</a> , <a href="#">FDb.InfiniumMethylation.hq18</a> , <a href="#">FDb.InfiniumMethylation.hq19</a> , <a href="#">GeneRegionScan</a> , <a href="#">genomes</a> , <a href="#">GenomicAlignments</a> , <a href="#">GOTHIC</a> , <a href="#">harbChIP</a> , <a href="#">iPAC</a> , <a href="#">JASPAR2014</a> , <a href="#">methVisual</a> , <a href="#">minfi</a> , <a href="#">MotifDb</a> , <a href="#">motifRG</a> , <a href="#">oligo</a> , <a href="#">oneChannelGUI</a> , <a href="#">pd.hugene.2.0.st</a> , <a href="#">pd.hugene.2.1.st</a> , <a href="#">pd.moqene.2.0.st</a> , <a href="#">pd.moqene.2.1.st</a> , <a href="#">pd.nuqo.hs1a520180</a> , <a href="#">pd.nuqo.mm1a520177</a> ,

Biostringsは塩基配列の切り出しや文字列検索系関数も提供しているので、SequenceMatchingがあるのも妥当。

Home » BiocViews

## All Packages

Bioconductor version 2.14 (Release)

Autocomplete biocViews search:

Packages found under SequenceMatching:

Show  entries

Search table:

Package	Maintainer	Title
<a href="#">Biostrings</a>	H. Pages	String objects representing biological sequences, and matching algorithms
<a href="#">BSgenome</a>	H. Pages	Infrastructure for Biostrings-based genome data packages
<a href="#">cleanUpdTSeq</a>	Sarah Sheppard; Jianhong Ou; Lihua Julie Zhu	This package classifies putative polyadenylation sites as true or false/internally oligodT primed.
<a href="#">cobindR</a>	Manuela Benary	Finding Co-occurring motifs of transcription factor binding sites
<a href="#">CRISPRseek</a>	Lihua Julie Zhu	Design of target-specific guide RNAs in CRISPR-Cas9, genome-editing systems
<a href="#">dagLogo</a>	Jianhong Ou	dagLogo
<a href="#">FunciSNP</a>	Simon G. Coetzee	Integrating Functional Non-coding Datasets with Genetic Association Studies to Identify Candidate Regulatory SNPs
<a href="#">hapFabia</a>	Sepp Hochreiter	hapFabia: Identification of very short segments of identity by descent (IBD) characterized by rare variants in large sequencing data
<a href="#">microRNA</a>	"James F. Reid"	Data and functions for dealing with microRNAs
<a href="#">MmPalateMiRNA</a>	Guy Brock	Murine Palate miRNA Expression Analysis

SequenceMatchingに含まれる17パッケージの一部しか表示されていないが、(Rで)塩基配列解析中にはないCRISPR関連のパッケージなども存在することに気づく。また、GenomeAnnotationやGenomicVariationなど様々なパッケージがあることに気づく。



Home » BiocViews

# All Packages

Bioconductor version 2.14 (Release)

Packages found under SequenceMatching:

Autocomplete biocViews search:

Show **All** entries Search table:

- ▼ Software (824)
  - ▶ AssayDomain (251)
  - ▼ BiologicalQuestion (204)
    - AlternativeSplicing (3)
    - Coverage (1)
    - DifferentialExpression (138)
    - DifferentialMethylation (3)
    - DifferentialSplicing (3)
    - FunctionalPrediction (1)
    - GeneRegulation (15)
    - GeneSetEnrichment (25)
    - GenomeAnnotation (3)
    - GenomicVariation (3)
    - MotifAnnotation (3)
    - MotifDiscovery (4)
    - NetworkEnrichment (7)
    - NetworkInference (15)

Package	Maintainer	Title
<a href="#">Biostrings</a>	H. Pages	String objects representing biological sequences, and matching algorithms
<a href="#">BSgenome</a>	H. Pages	Infrastructure for Biostrings-based genome data packages
<a href="#">cleanUpdTSeq</a>	Sarah Sheppard; Jianhong Ou; Lihua Julie Zhu	This package classifies putative polyadenylation sites as true or false/internally oligodT primed.
<a href="#">cobindR</a>	Manuela Benary	Finding Co-occurring motifs of transcription factor binding sites
<a href="#">CRISPRseek</a>	Lihua Julie Zhu	Design of target-specific guide RNAs in CRISPR-Cas9, genome-editing systems
<a href="#">dagLogo</a>	Jianhong Ou	dagLogo
<a href="#">FunciSNP</a>	Simon G. Coetzee	Integrating Functional Non-coding Datasets with Genetic Association Studies to Identify Candidate Regulatory SNPs
<a href="#">hapFabia</a>	Sepp Hochreiter	hapFabia: Identification of very short segments of identity by descent (IBD) characterized by rare variants in large sequencing data
<a href="#">microRNA</a>	"James F. Reid"	Data and functions for dealing with microRNAs
<a href="#">MmPalateMiRNA</a>	Guy Brock	Murine Palate miRNA Expression Analysis

biocViewsには、中分類に相当するBiologicalQuestionという記述はなかった。自分のパッケージがどこに分類分けされるかを指定するbiocViewsは、キーワード指定のようなものであり、パッケージ開発者次第なのだろうと妄想する。

