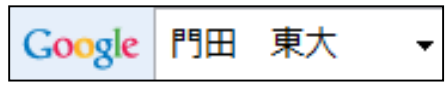
バイオインフォマティクス人材育成カリキュラム（次世代シークエンサ）速習コース
3. データ解析基礎 | 3-4. R Bioconductor I

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

- 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)

- イントロ | 一般 | 翻訳配列(translate)を取得

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ペ。

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_

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

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

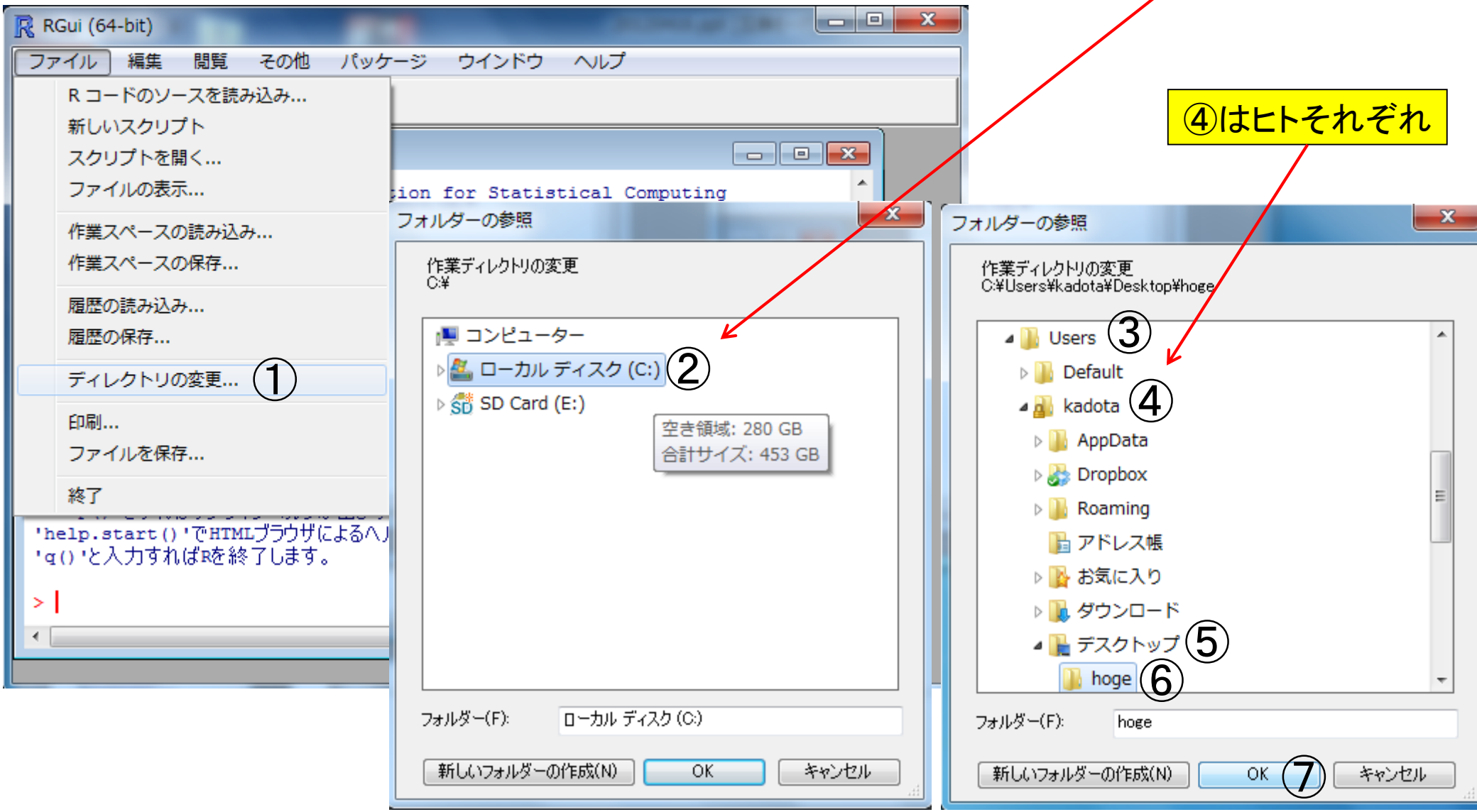
rcode_translate.txt

```
############################
###  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

#本番
hoge <- translate(fasta)               #fastaをアミノ酸配列に翻訳した結果をhog
names(hoge) <- names(fasta)            #現状では翻訳した結果のオブジェクトhoge
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
```

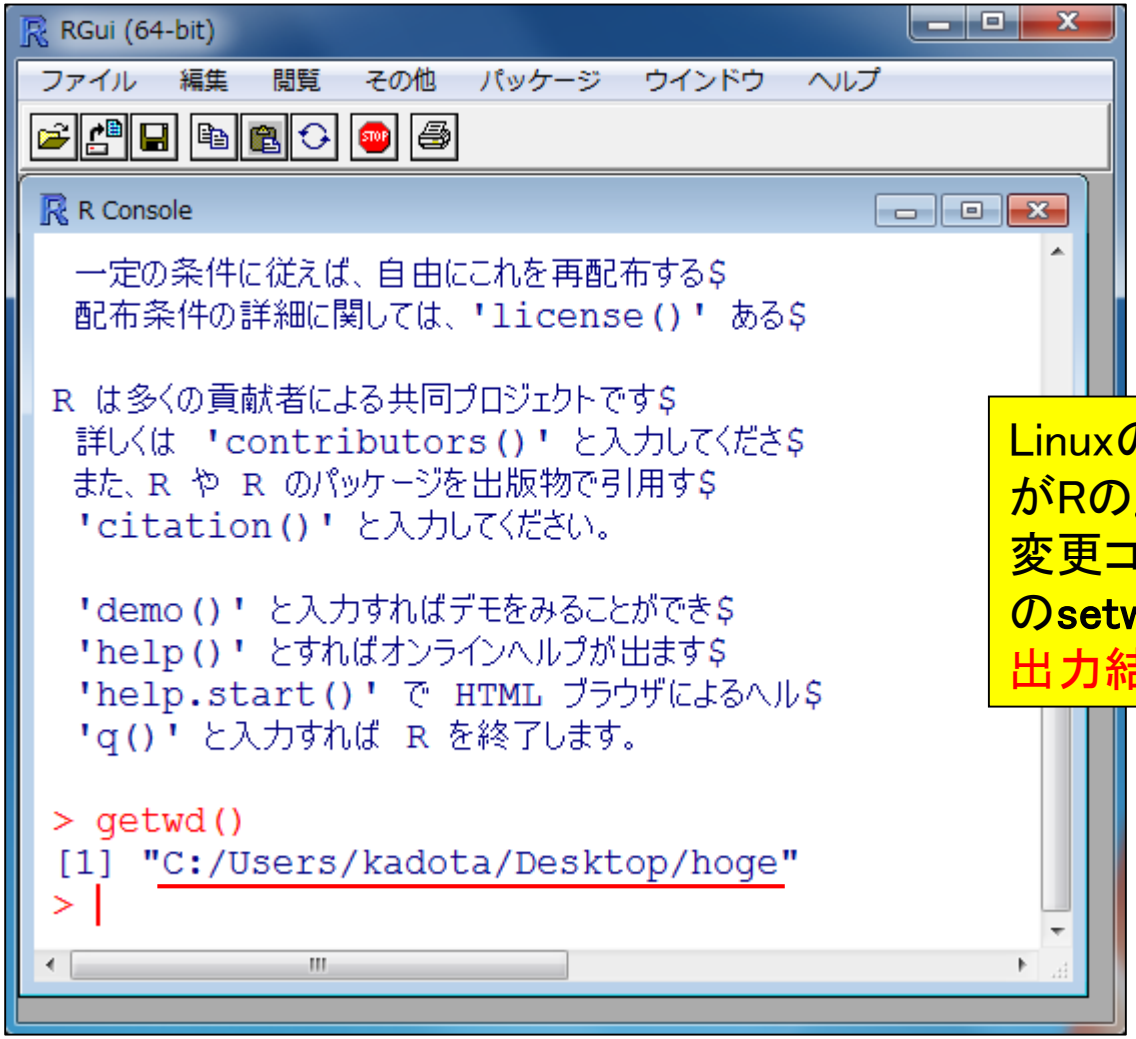
作業ディレクトリの変更

「Windows(C:)」となっている場合もあるが、気にしない

④はヒトそれぞれ



作業ディレクトリの変更



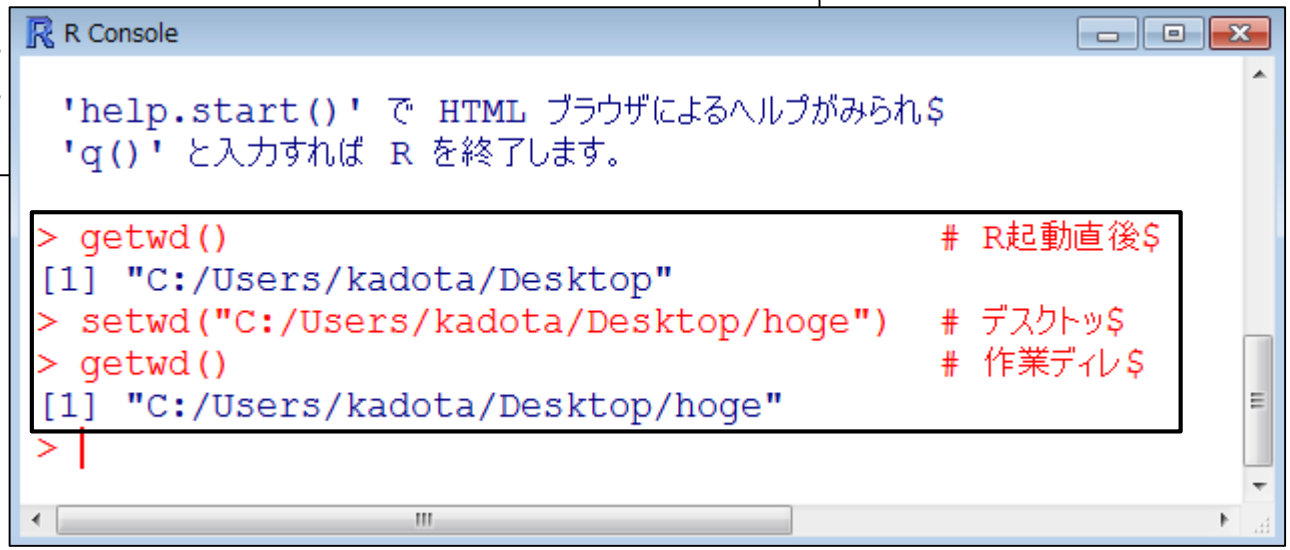
Linuxの**pwd**コマンドに相当するのがRの**getwd()**。Linuxのディレクトリ変更コマンド**cd**に相当するのがRの**setwd()**。重要なのは、getwd()の出力結果を把握しておくことです。

デスクトップのhogeと決めるなら…

ファイル名：rcode_20140909.txt

Rを再起動してコピペ。setwd関数実行によって、作業ディレクトリの変更がコピペ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")       #
```

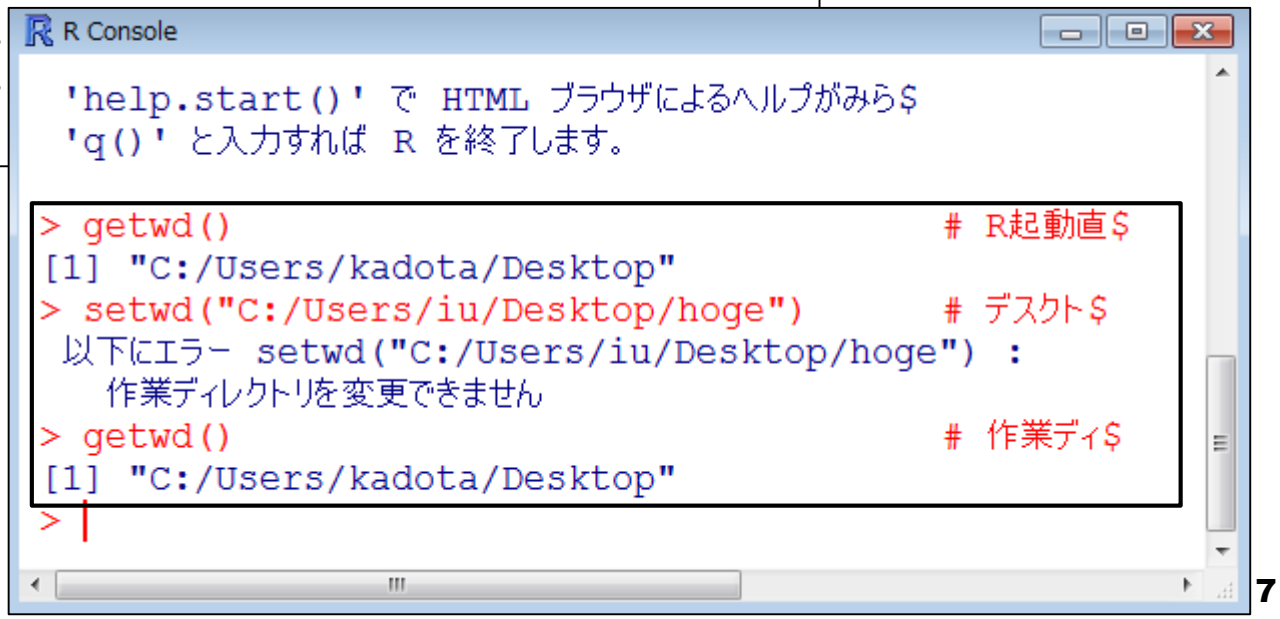


デスクトップのhogeと決めるなら…

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")       #
```



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を起動直後にコピペ**

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$
> 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"$
>
```

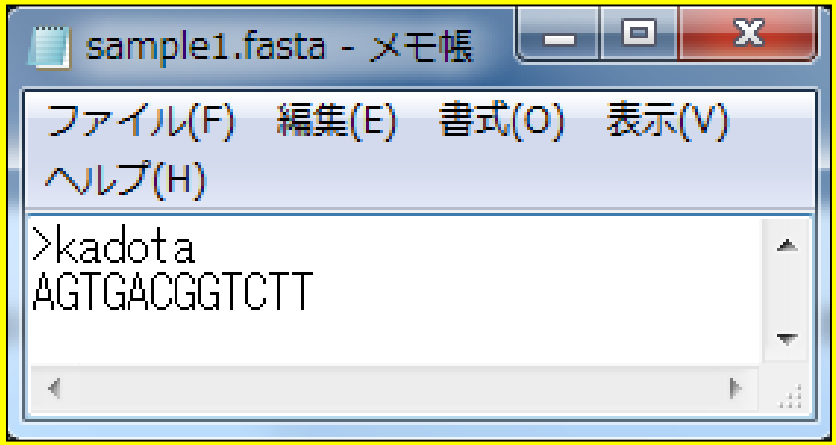
イントロ | 一般 | 翻訳配列(translate)を取得 NEW

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。

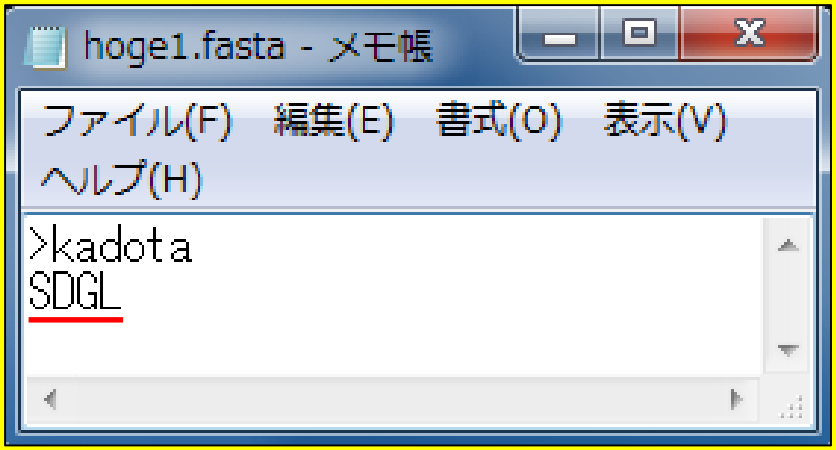
1. FASTA形式ファイル(sample1.fasta)の場合:

```
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
writeXStringSe
```

入力:塩基配列ファイル(sample1.fasta)



出力:アミノ酸配列ファイル(hoge1.fasta)



Contents

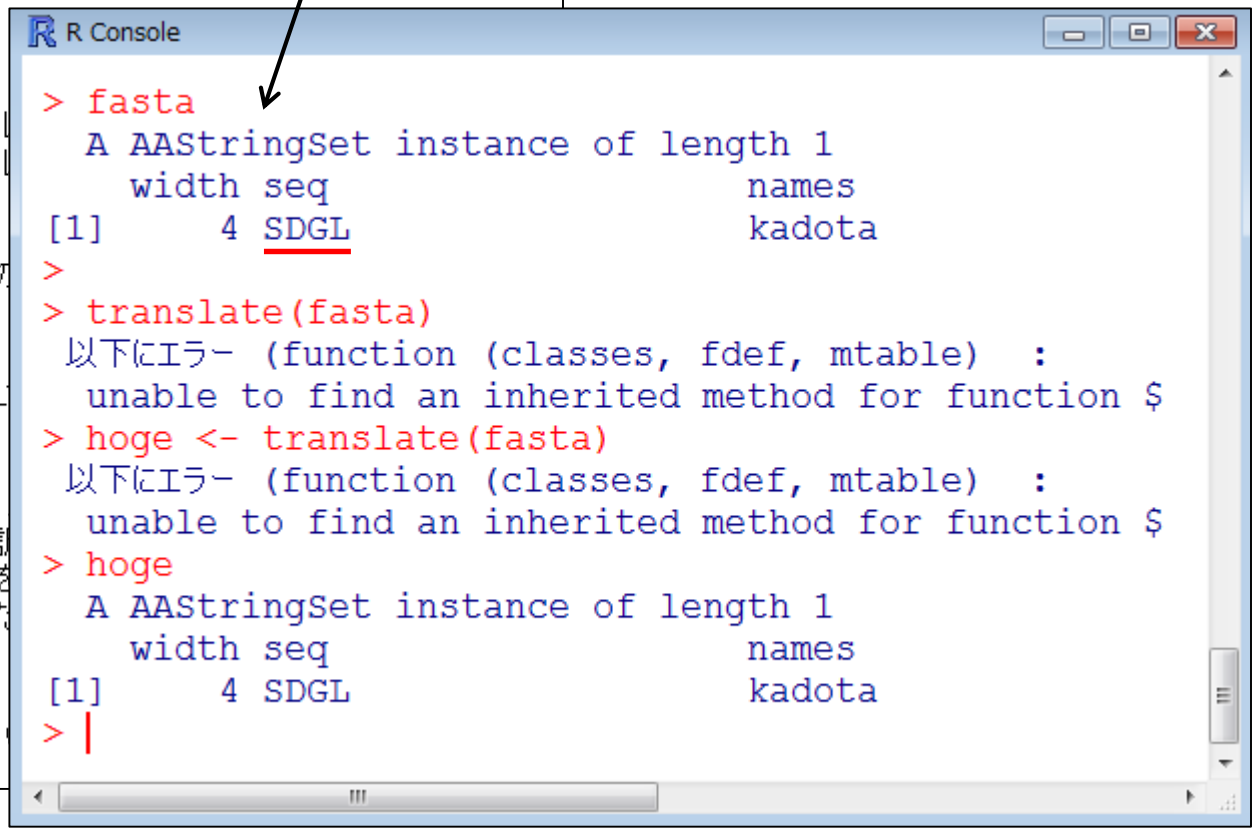
- 3-4. R Bioconductor I、2014/09/09 10:30-14:45、中級、実習
  - Tips：setwd関数を利用した効率的な作業ディレクトリの変更
  - **データの型1：translate関数の入力情報(AAStringSet)**
  - Tips：オブジェクトの消去
  - データの型2：翻訳配列取得のコードの中身を解説
  - バージョン情報把握とバージョンアップ
  - バージョン違いの影響
    - 過去から現在：BiostringsパッケージのreadDNAStringSet関数
    - 現在から未来：BSgenome.Hsapiens.UCSC.hg19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

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                                  #確認してるだ

#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というものまたは型という程度の理解でよい。

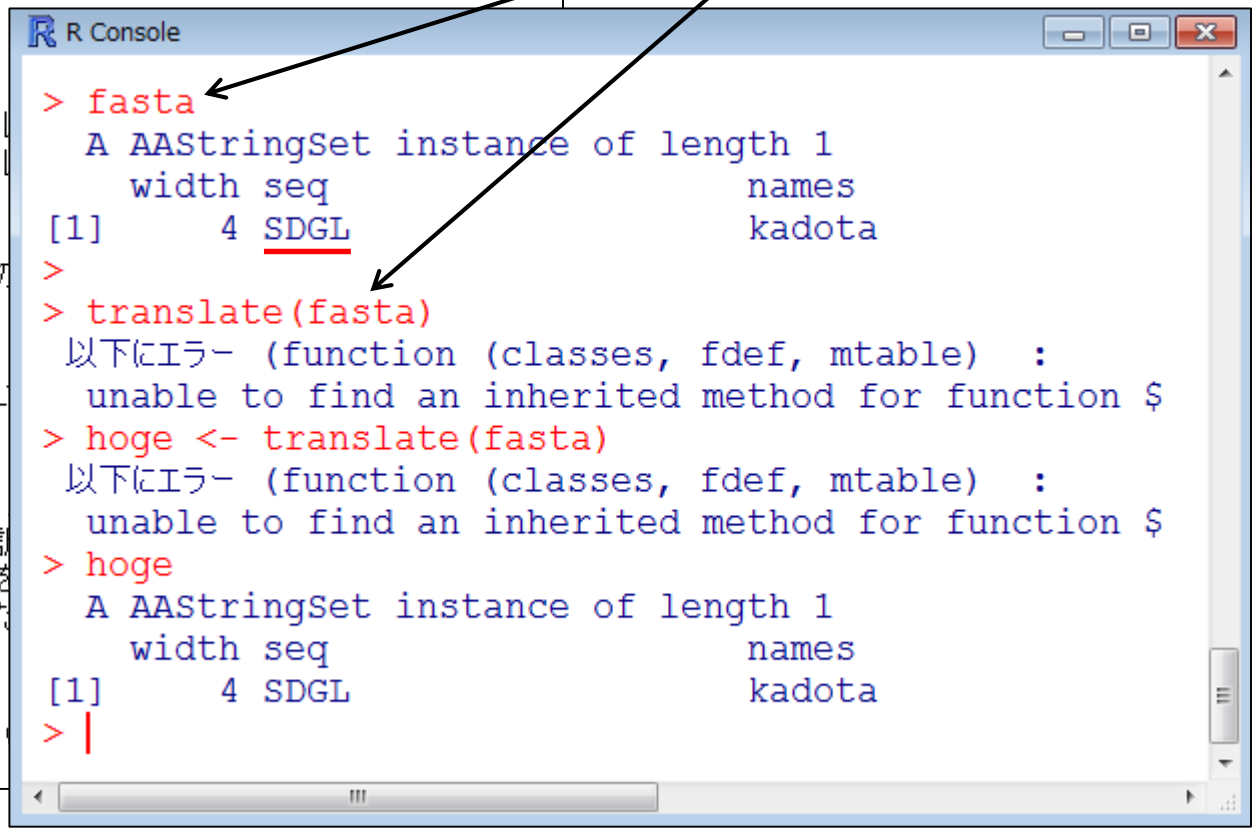


データの型1

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                                  #確認してるだ

#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。



データの型1

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。

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                                  #確認してるだ

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

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

############################
```

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」だと言われているのだろう、という対策方針を立てるべし

R Console

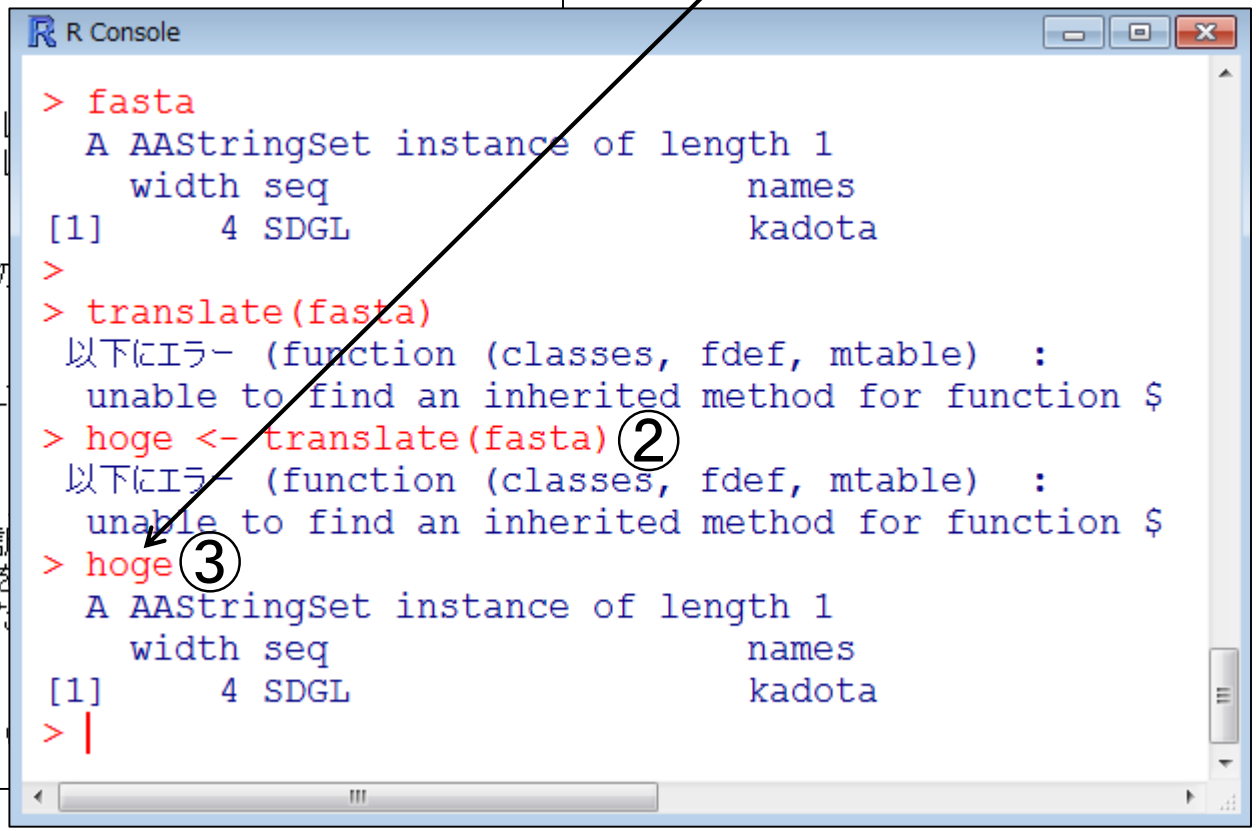
```
> fasta
  A AAStringSet instance of length 1
    width seq                  names
[1]     4 SDGL                 kadota
>
> translate(fasta)
 以下にエラー (function (classes, fdef, mtable)  :
  unable to find an inherited method for function 'translate' for signature '"AAStringSet"'
> hoge <- translate(fasta)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

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

#本番
hoge <- translate(fasta)  ①            #fastaをアミ
names(hoge) <- names(fasta)            #現状では翻訳
fasta <- hoge                          #hogeの中身を
fasta                                  #確認してるだ

#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。



Contents

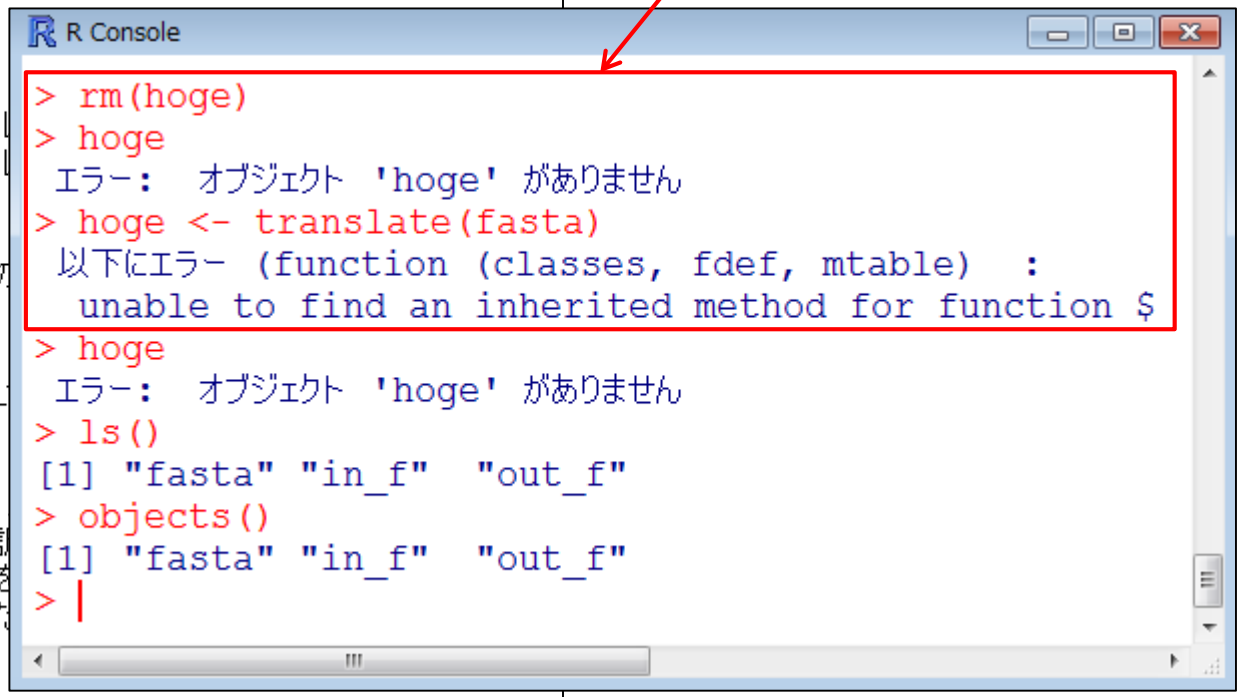
- 3-4. R Bioconductor I、2014/09/09 10:30-14:45、中級、実習
  - Tips：setwd関数を利用した効率的な作業ディレクトリの変更
  - データの型1：translate関数の入力情報(AAStringSet)
  - **Tips：オブジェクトの消去**
  - データの型2：翻訳配列取得のコードの中身を解説
  - バージョン情報把握とバージョンアップ
  - バージョン違いの影響
    - 過去から現在：BiostringsパッケージのreadDNAStringSet関数
    - 現在から未来：BSgenome.Hsapiens.UCSC.hg19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

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
#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                                  #確認してるだ
↓
#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
↓
```

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。

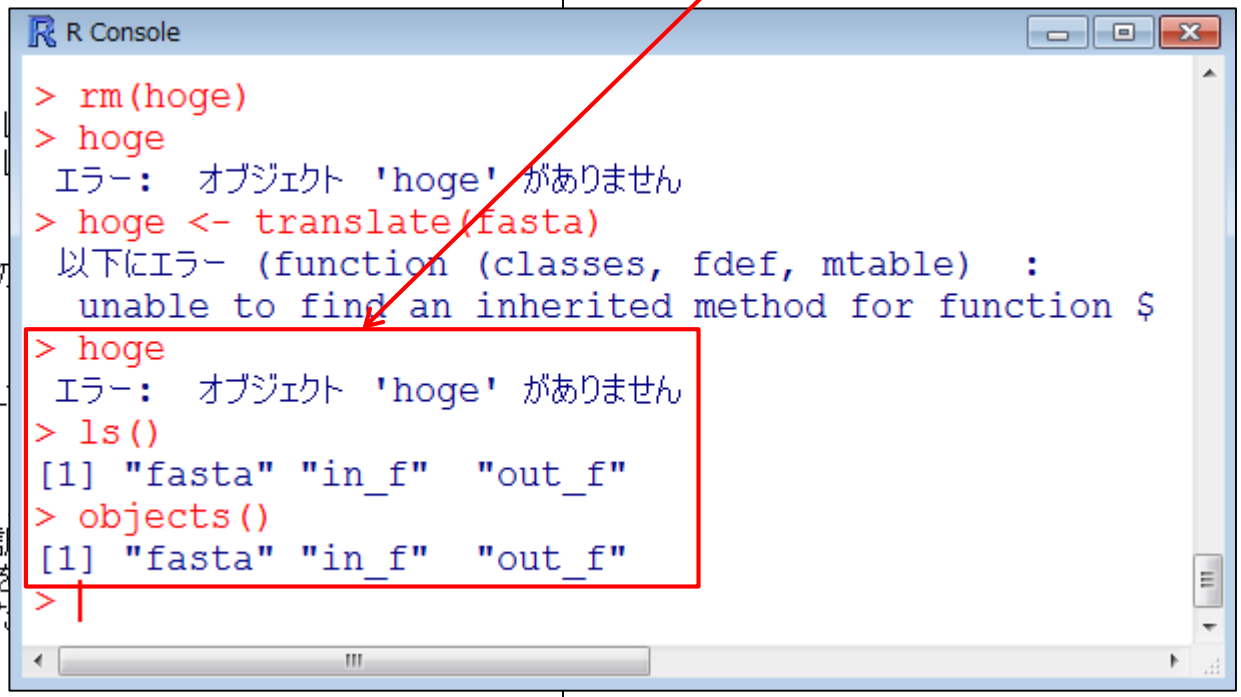


オブジェクトの消去はrm関数を利用

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
#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
↓
#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                                  #確認してるだ
↓
#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
↓
```

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。

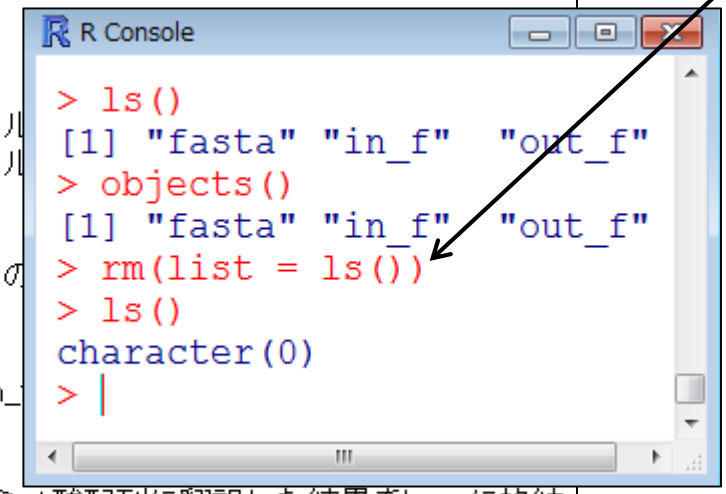


オブジェクトの消去はrm関数を利用

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
#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

#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
```

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。



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
names(hoge) <- names(fasta)            #現状では翻訳した結果のオブジェクトhogeのd
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
↓
```

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

「?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。

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

############################
```

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
>
```

R Console

```
> ?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

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
```

in_fで指定した入力ファイルを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
> |
```

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
```

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
>
```

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
```

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
>
```

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
>
```

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
```

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
>
```

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
```



```
>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.と基本的に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
```

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
```

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!**

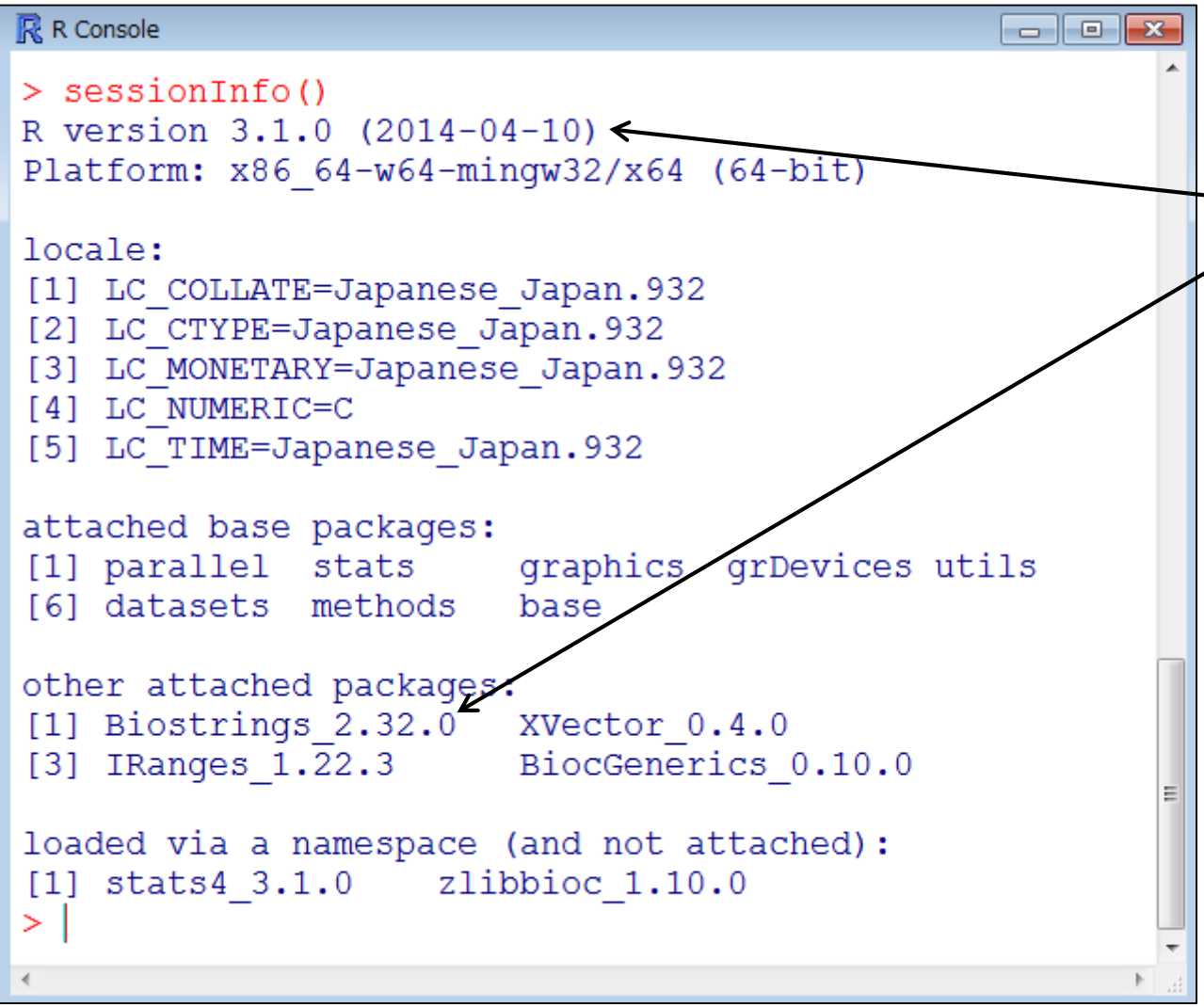
R Console

```
> #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 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

sessionInfo()で自分のR環境を把握



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[1], 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 t
R packages did not have normalization methods based on such a multi-step stra

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

CONCLUSION: DEGES in TCC is essential for accurate normalization of tag co
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 unb
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.

PMID: 23837715 [PubMed - indexed for MEDLINE] PMCID: PMC3716788 Free PMC Article

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[1], 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 t
R packages did not have normalization methods based on such a multi-step stra

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

CONCLUSION: DEGES in TCC is essential for accurate normalization of tag co
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 unb
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.

PMID: 23837715 [PubMed - indexed for MEDLINE] PMCID: PMC3716788 Free PMC Article

「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
    - …
  - 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

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。

(Rで)マイクロアレイデータ解析

(last modified 2014/07/24, since 2005)

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
  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 modifie 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

DFW (Chen et al., Bioinformatics, 2007)というAff
りの"階層的クラスタリング結果を導くような遺伝
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

- Rの昔のバージョンのインストール

Previous Releases of R for Windows

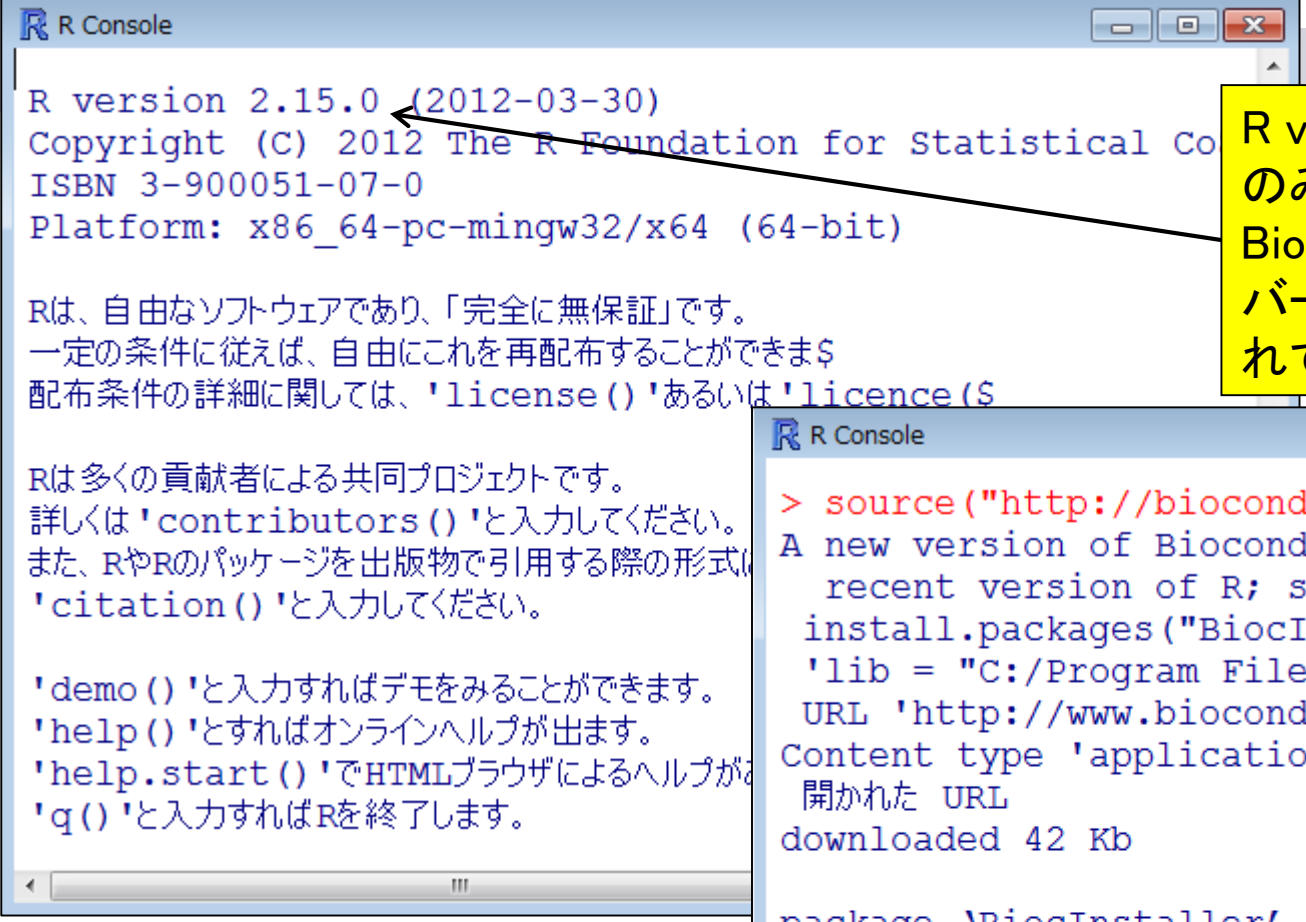
This directory contains previous binary releases of R to run on Windows 95, 98, ME, NT4.0, 2000 and XP or later on Intel/clone chips.

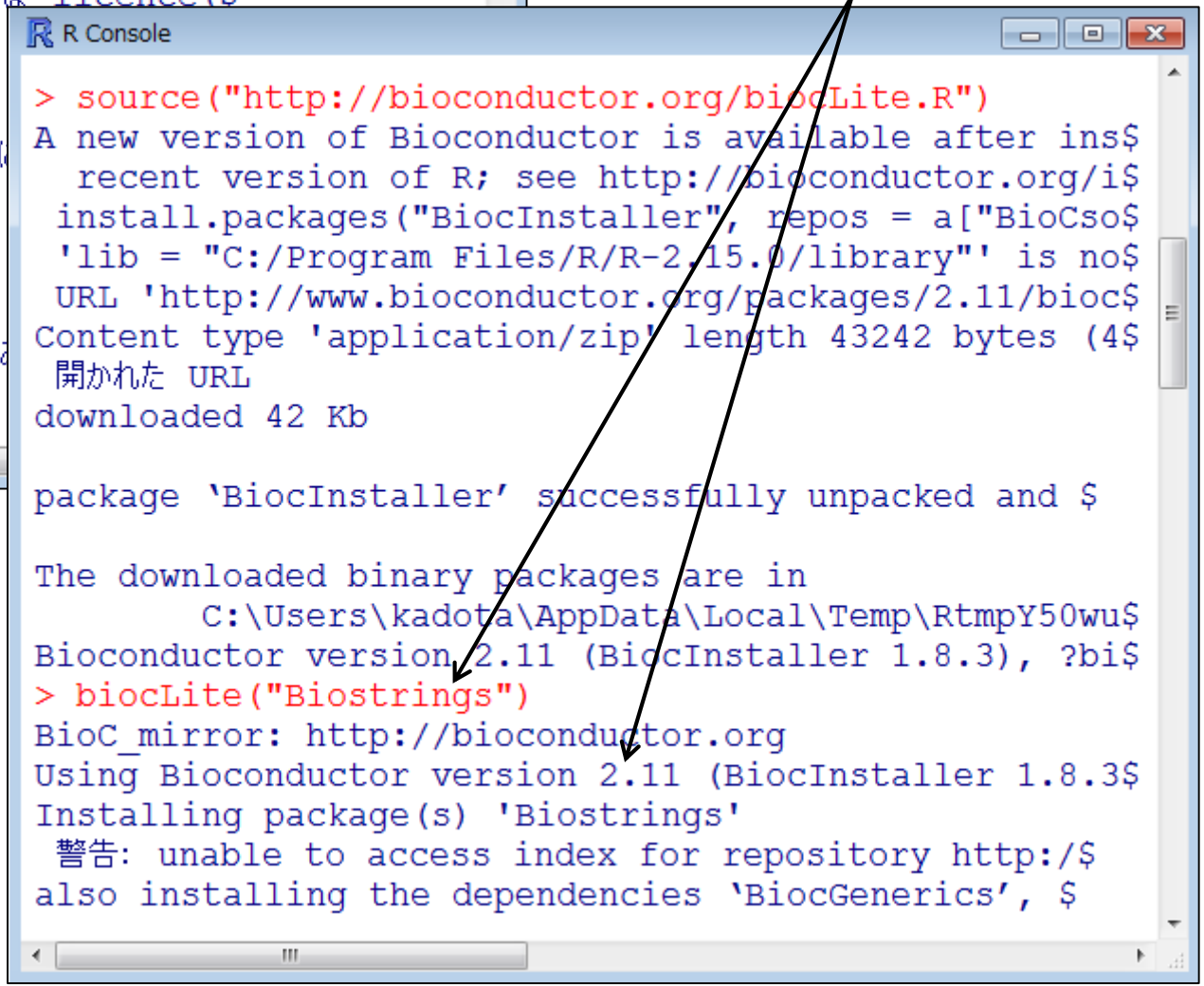
The current release, and links to development snapshots, are available here. Source code for these releases and others is available through the main CRAN page.

In this directory:

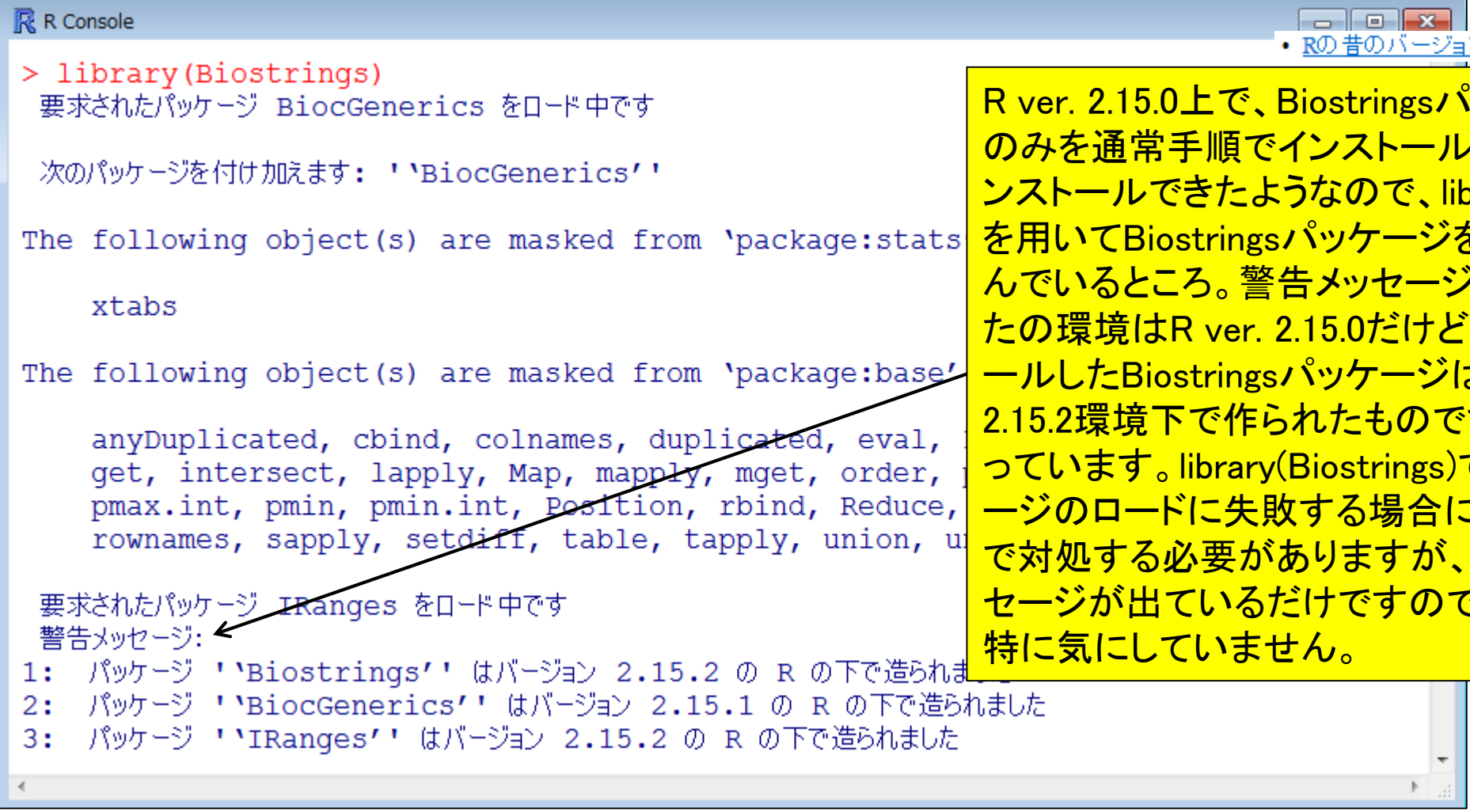
R 3.1.0 (April, 2014)
R 3.0.3 (March, 2014)
R 3.0.2 (September, 2013)
R 3.0.1 (May, 2013)
R 3.0.0 (April, 2013)
R 2.15.3 (March, 2013)
R 2.15.2 (October, 2012)
R 2.15.1 (June, 2012)
R 2.15.0 (March, 2012)
R 2.14.2 (February, 2012)
R 2.14.1 (December, 2011)
R 2.14.0 (November, 2011)
R 2.13.2 (September, 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。



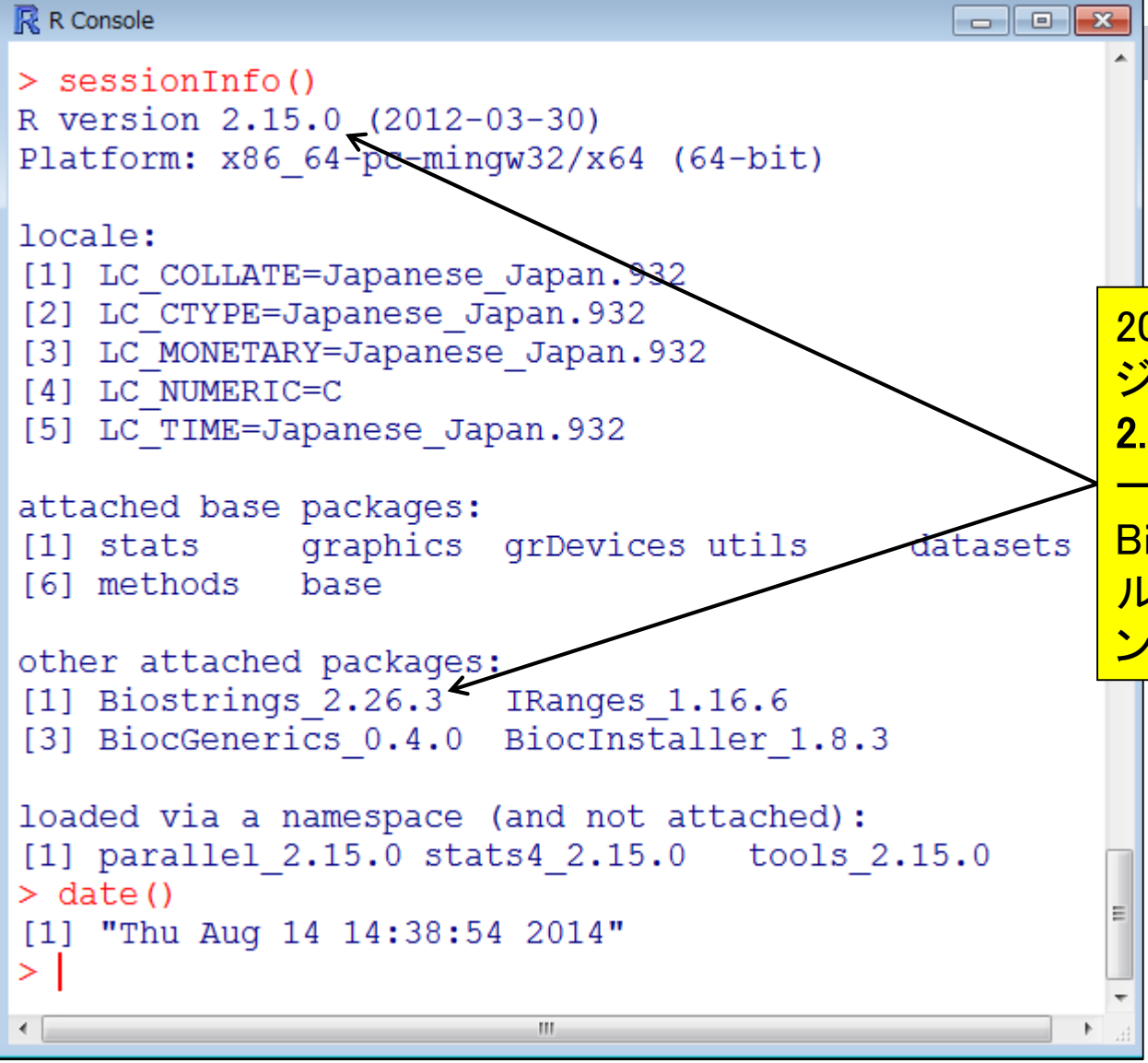


R ver. 2.15.0上で、Biostringsパッケージのみを通常手順でインストール。Bioconductor ver. 2.11でリリースされたバージョンのBiostringsがインストールされていることが分かる。



- Rの昔のバージョンのインストール

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ので、私は特に気にしていません。



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
    - …
  - 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

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。

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[1], 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. 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.

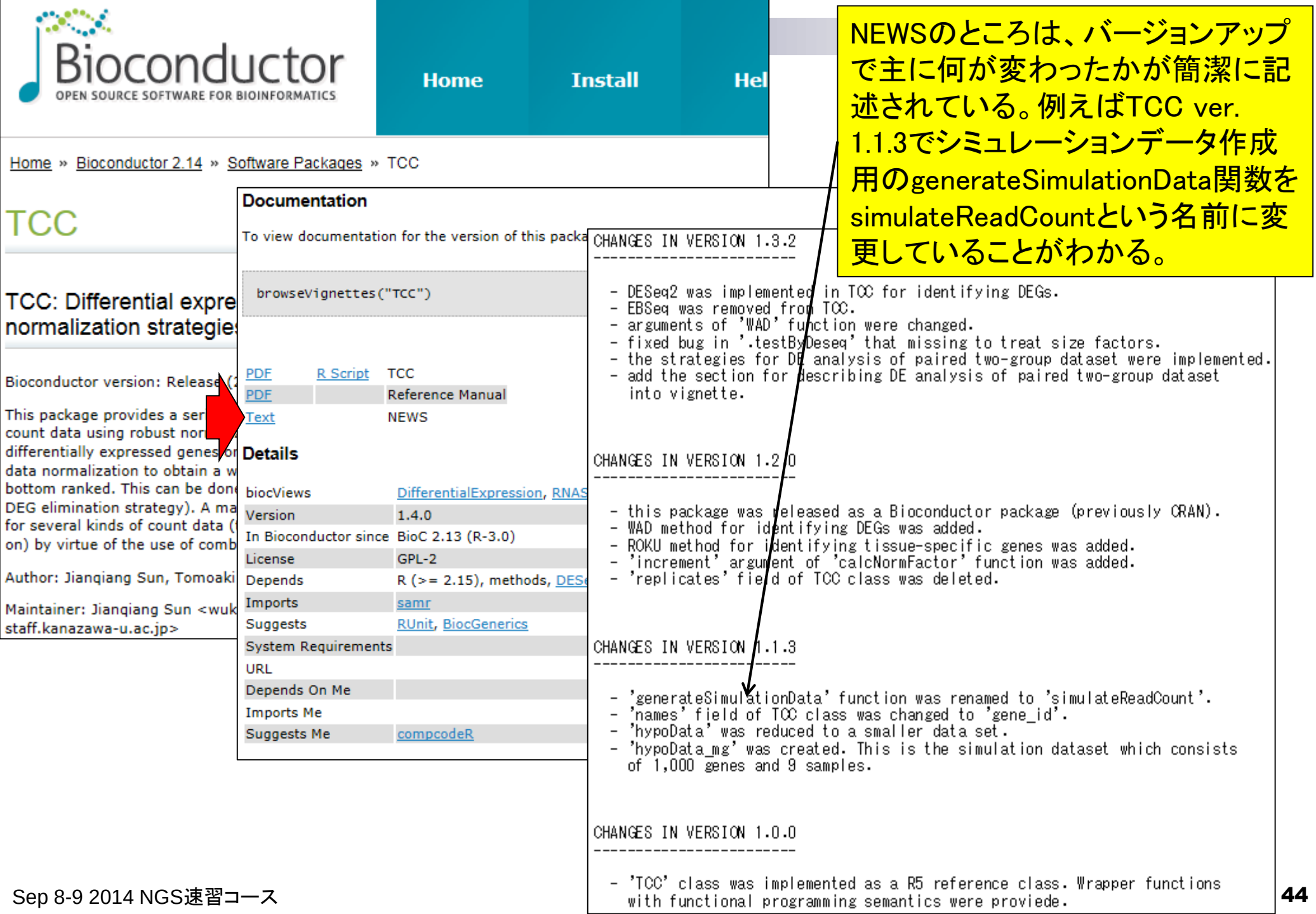
CONCLUSION: DEGES in TCC is essential for accurate normalization of tag count data, especially when up- and down-regulated DEGs in one of the samples are extremely biased in their number. TCC is useful for analyzing tag count data in various scenarios ranging from unbiased to extremely biased differential 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.

PMID: 23837715 [PubMed - indexed for MEDLINE] PMCID: PMC3716788 Free PMC Article

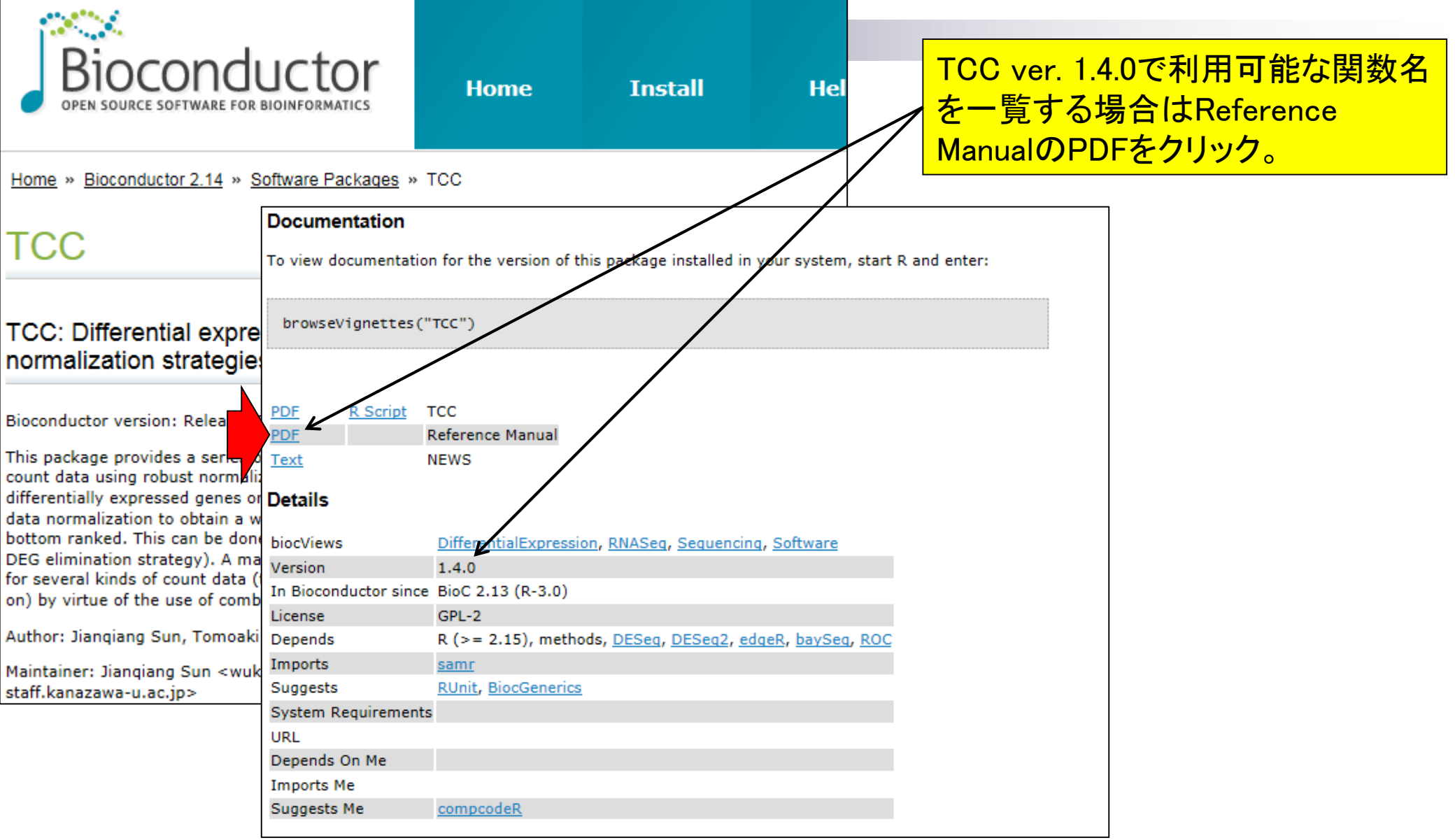
2013年7月の論文publish以降も継続的にアップデートしています：

- 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
解析 | 発現変動 | 2群間 | 対応なし | について NEW
実験デ
Aさん
Bさん
Cさん
Dさん
Eさん
Fさん
Gさん
2014年
R用:
R用:
DEGSeq: Wang et al., Bioinformatics, 2010
edgeR: Robinson et al., Bioinformatic
GPseq: Srivastava et al., Nucleic Acid
baySeq: Hardcastle and Kelly, BMC B
DESeq: Anders and Huber, Genome B
DESeq2: Anders and Huber, Genome
NBPSeq: Di et al., SAGMB, 2011
BBSeq: Zhou et al., Bioinformatics, 2
NOISeq: Tarazona et al., Genome Res
PoissonSeq: Li et al., Biostatistics, 20
SAMseq: Li and Tibshirani, Stat Meth
easyRNASeq: Delhomme et al., Bioin
DSGseq: Wang et al., Gene, 2013
sSeq: Yu et al., Bioinformatics, 2013
TCC: Sun et al., BMC Bioinformatics
tweeDEseq: Esnaola et al., BMC Bioi
NPEBseq: Bi et al., BMC Bioinforma
DER Finder: Frazee et al., Biostatistic
Characteristic Direction(CD): Clark et
edgeR-robust: Zhou et al., Nucleic Ac
ShrinkBayes: Van De Wiel et al., BM
Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS
Home
Install
Hel
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")):



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。


Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS
Home
Install
Hel
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 expre
normalization strategie
Bioconductor version: Relea
This package provides a ser
count data using robust norm
differentially expressed genes or
data normalization to obtain a w
bottom ranked. This can be don
DEG elimination strategy). A ma
for several kinds of count data (
on) by virtue of the use of comb
Author: Jianqiang Sun, Tomoaki
Maintainer: Jianqiang Sun <wuk
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

Documentation

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

```
browseVignettes("TC
```

PDF R Script TC
PDF Ref
Text NE

Details

biocViews	D
Version	1.
In Bioconductor since	Bi
License	G
Depends	R
Imports	sa
Suggests	R
System Requirements	
URL	
Depends On Me	
Imports Me	
Suggests Me	co

これが関数一覧

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.a.u-tokyo.ac.jp>, Tomoaki Nishiyama <tomoakin@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

Documentation

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

```
browseVignettes("TC
```

PDF	R Script	TC
PDF		Ref
Text		NE

Details

biocViews	D
Version	1.
In Bioconductor since	Bi
License	G
Depends	R
Imports	sa
Suggests	R
System Requirements	
URL	
Depends On Me	
Imports Me	
Suggests Me	co

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

20 / 33 100%

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

NBsample *Sampling from negative*

Description

This methods allow sampling from Negative Bino
ferentially 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)
```

215.9 x 279.4 mm

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。

R Console

```
> library(TCC)                    # TCCパッケージをロード
> ?NBsample                       # NBsample関数のマニュ
>
```

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 biomial 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)
```

[Package *TCC* version 1.4.0 Index]

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

(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。

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", 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

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。

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ペ。

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="f

#本番
hoge <- translate(fasta)                   #f
names(hoge) <- names(fasta)                #現
fasta <- hoge                              #h
fasta                                      #確

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

R Console

```
R version 2.12.0 (2010-10-15)
Copyright (C) 2010 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'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

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

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.

```
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 <- readDNAStringSe

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

#ファイルに保存
writeXStringSet(fasta,
```

R Console

```
> fasta <- readDNAStringSet(in_f, format="fasta")#in_fで指定したフ$
 エラー:  関数 "readDNAStringSet" 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 l$
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

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="fa

#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=
```

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readDNAStringSetでもread.DNAStringSet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$
>
> #入力ファイルの読み込み
> fasta <- readDNAStringSet(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。

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

```
> #入力ファイルの読み込み
> fasta <- read.DNAStringSet(in_f, format="fasta")#$
 警告メッセージ:
'read.DNAStringSet' is deprecated.
Use 'readDNAStringSet' 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_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

############################
###  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
```

ファイル名：rcode_20140909.txt

```
############################
###  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

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ペ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readDNAStringSet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 'li$

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$
>
> #入力ファイルの読み込み
> fasta <- readDNAStringSet(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readDNAStringSet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()' と入力してくださし
  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.DNAStringSet(in_f, format="fasta")#$
 エラー:  関数 "read.DNAStringSet" を見つけることが$
 wrapup 中にエラーが起こりました:  コネクションを開$
>
> #本番
> hoge <- translate(fasta)                    #fastaをア$
 wrapup 中にエラーが起こりました:  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 中にエラーが起こりました:  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 中にエラーが起こりました:  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 中にエラーが起こりました:  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     geneplotter_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

NGSデータ解析とR

ヒトやマウスなどの代表的な生物種については、ゲノム配列情報を含む生物種ごとのRパッケージが存在する。(ゲノム配列や)上流配列をmulti-FASTA形式ファイルとして保存可能。将来使えなくなるという警告がR ver. 3.1.0で出ている例を紹介します。

(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)

- イントロ | 一般 | 配列取得 | ゲノム配列 | 公共DBから (last modified 2014/05/28)
- イントロ | 一般 | 配列取得 | ゲノム配列 | BSgenome (last modified 2014/06/28) NEW
- イントロ | 一般 | 配列取得 | プロモーター配列 | 公共DBから (last modified 2014/04/02)
- イントロ | 一般 | 配列取得 | プロモーター配列 | BSgenome (last modified 2014/04/25)
- イントロ | 一般 | 配列取得 | プロモーター配列 | GenomicFeatures(Lawrence 2013) (last modi
- イントロ | 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 2014/04/0
- イントロ | 一般 | 配列取得 | トラン
- イントロ | NGS | 様々なプラットフォ
- イントロ | NGS | qPCRやmicroarra
- イントロ | NGS | 可視化(ゲノムブ
- イントロ | NGS | 配列取得 | FAST
- イントロ | NGS | 配列取得 | FAST
- イントロ | NGS | 配列取得 | シミュ
- イントロ | NGS | 配列取得 | シミュ
- イントロ | NGS | アノテーション情
- イントロ | NGS | アノテーション情
- イントロ | NGS | アノテーション情
- イントロ | NGS | アノテーション情
- イントロ | NGS | アノテーション情
- イントロ | NGS | アノテーション情
- イントロ | NGS | アノテーション情

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

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ペ。

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

#後処理（パッケージ名でだいたいわかるがproviderやversionを分割して表示したい場合）
installed.genomes(splitNameParts=TRUE) #インストール済みの生物種をリストアップ
```

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

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ペ。

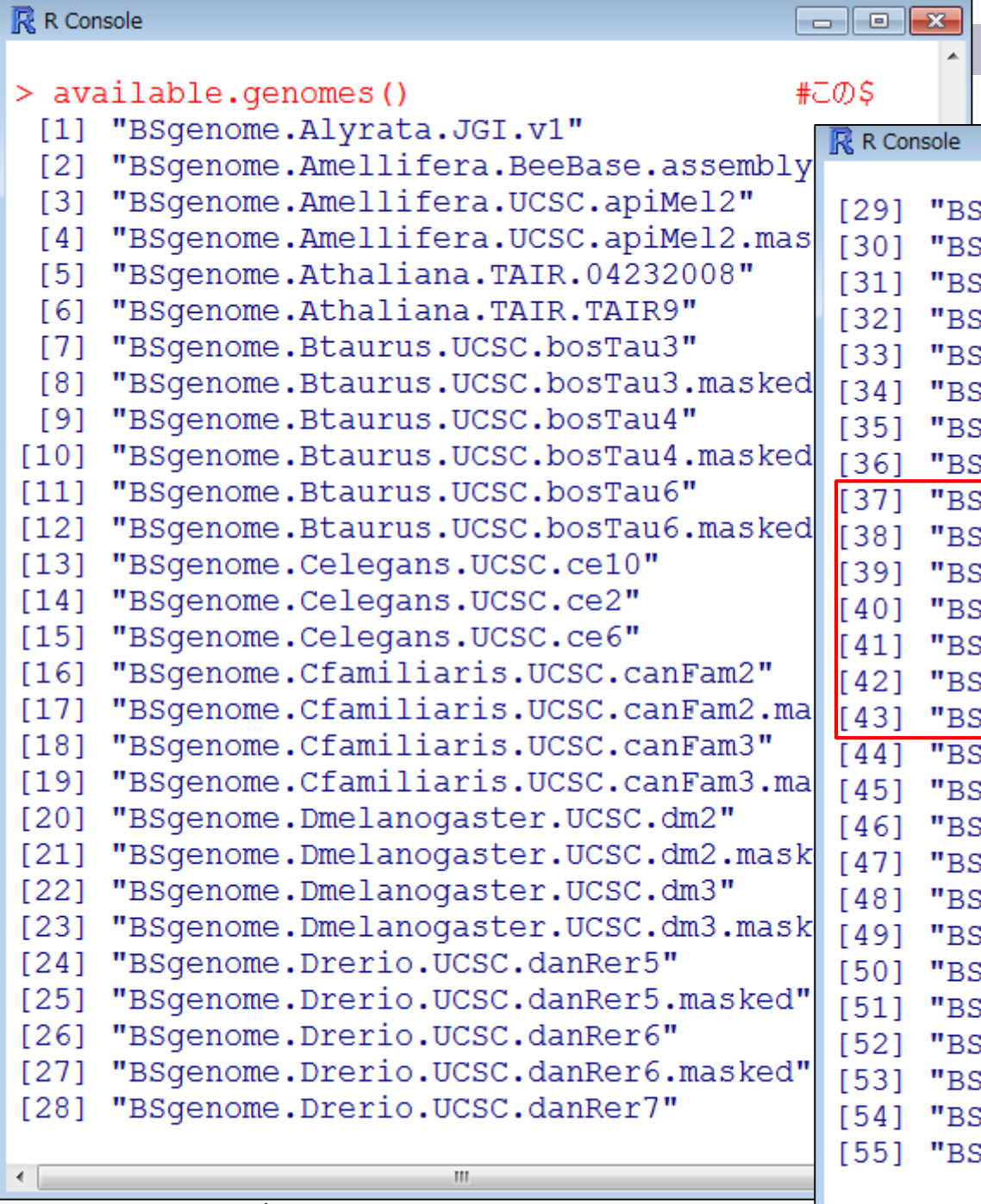
1. 利用可能な生物種とRにインストール済みの生物種をリストアップしたい場合:

```
#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
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)
```

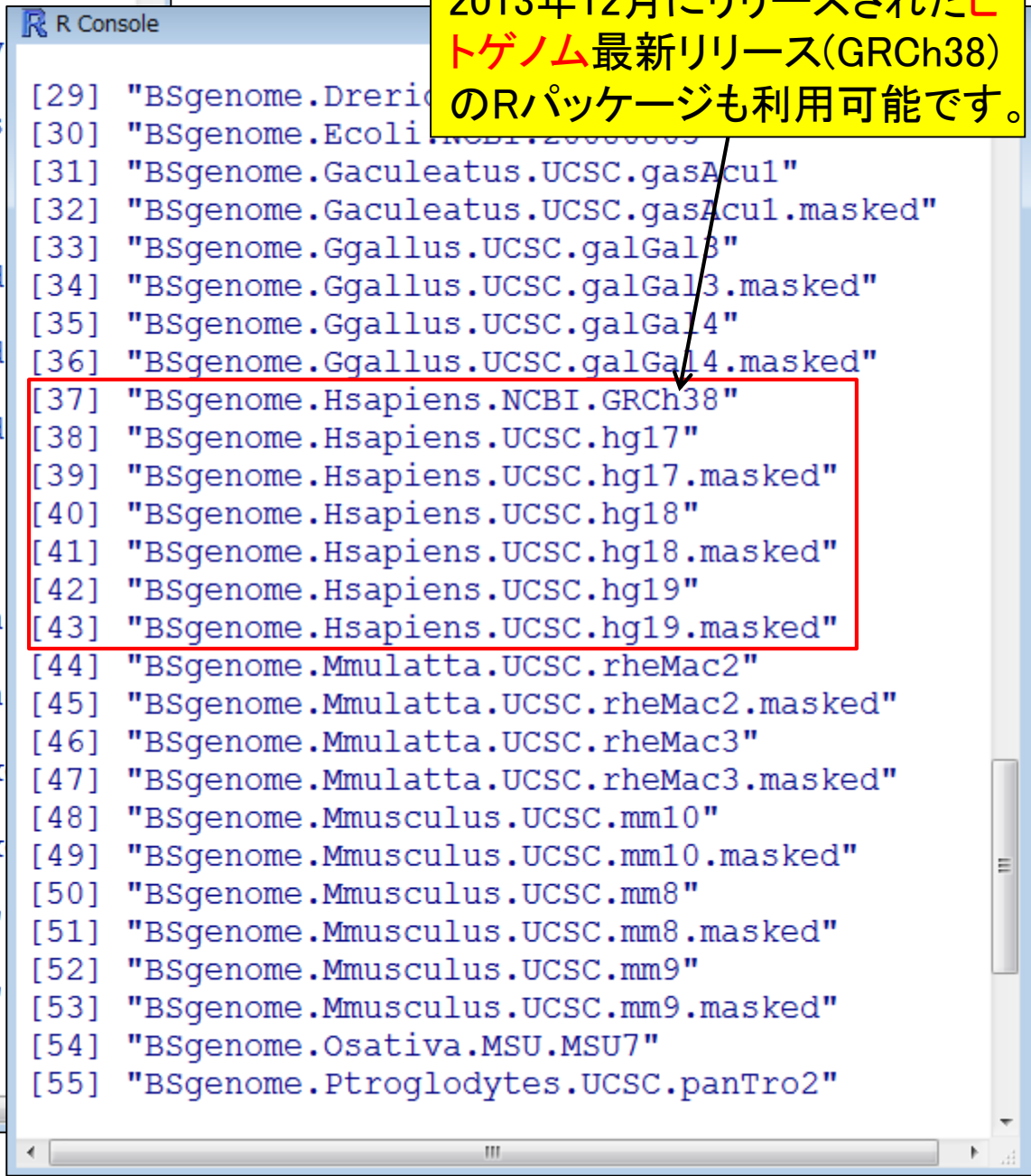
R ver. 3.1.0 (Bioconductor ver. 2.14)で利用可能な生物種のパッケージ名をリストアップ。70個あることが分かる。Rのバージョンが古いとおそらくパッケージ数は少なくなる。

R Console

```
 [55] "BSgenome.Ptroglodytes.UCSC.panTro2"
 [56] "BSgenome.Ptroglodytes.UCSC.panTro2.masked"
 [57] "BSgenome.Ptroglodytes.UCSC.panTro3"
 [58] "BSgenome.Ptroglodytes.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"
> |
```



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。



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

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ペ。

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

#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
```

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。

R Console

```
[65] "BSgenome.Scer
[66] "BSgenome.Sscr
[67] "BSgenome.Sscr
[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"
>
```

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
ます。注意点は、「Mac OS X のバージョンに関わらず R-3.1.0-snowleopar

1. Windows release版のインストールの場合:

1. Rのインストーラを「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
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
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てる
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「R3.X.Y(3
よって異なります)」または「R x64 3.X.Y (64 bitの場合)」アイコンを
5. 以下を、「Rコンソール画面上」でコピー&ペーストする。10GB程度
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

```
install.packages(available.packages()[,1], dependencies=TRUE)#CRAN中にある全てのパック
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
biocLite("BSgenome.Athaliana.TAIR.TAIR9", suppressUpdates=TRUE)#Bioconductor中にあるB
```

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。

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"
>
```

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

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。

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

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
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)
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
(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.tr
(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

1. 利用可能な生物種と

```
#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
installed.genome

#後処理（パッケー
installed.genome
```

2. ゼブラフィッシュ("BS

400MB程度あります..

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中のオブジェクト名をhogeに統一)
#tmp <- unlist(strsplit(param, ".", 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
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
```

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 C$
Platform: x86_64-w64-mingw32/x64 (64

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
'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
'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

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。

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(1000b
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
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
#tmp <- unlist(strsplit(pa
tmp <- ls(paste("package",
hoge <- eval(parse(text=tm
hoge

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

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

R Console

```
> fasta
  A DNAStringSet instance of length 28020
        width seq                                          names
    [1]  1000 CCACCTGGGGAAGCG...TGCCGTTCCTCTCCC 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...TCCCCCACCACCCCC NM_001127389_up_1...
[28017]  1000 AAAGACAGAGCGACG...CCACGCCCTCCCCCA NM_033178_up_1000...
[28018]  1000 AAAGACAGAGCGACG...CCACGCCCTCCCCCA NM_033178_up_1000...
[28019]  1000 TTGTATTTTTAGTAG...CCTCTAGCTGTGTGT NM_006625_up_1000...
[28020]  1000 TTGTATTTTTAGTAG...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(1000b
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
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

#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(param, ".", fixed=TRUE))[2]#param1で指定
tmp <- ls(paste("package", param1, sep=":"))#param1で指定したパ
hoge <- eval(parse(text=tmp))             #文字列tmpをRオブジェクト
hoge                                      #確認してるだけです(ここで

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

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

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
CCACCTGGGGAAGCGAGGCCGCTCCCGTGTGCGCTCGGCGGGGAGGACGC
GGTGGCGGCGGGACTCGAACTCGGTCGTCGAAGGGCGCCCCGCAGGGTCC
CGTCTGTAGCCCGGACGGCGAGCCCAGGTTGGAGTGGGAGGGAGCCGGCG
TGCACAGCTGTCTTTGGTCGGGCTCAGGCTTCCGCTCCCTGCCTGCTCCC
CTTTCCAGCCTCCCGCCCCAGAAACGATCTCGAGCGTTGCCAGTTTGATT
CCAGAGCCCCACTCGGGTGGGTTCTTTTGTTTCTTTGTTTTAATGACAGT
TCCCAGCCCTTCAGCATGTCATGCGCAATTAATTCCCTGGCTCTCACGAA
GGCAGCTGGGGTGAAATTTCTTCTGCATCATCCTTTTGGGGATGTTTATG
ATGTGACGTCAGTCGGATTGATTTTTCTCTCTTGAATGAAGGATGGGAGG
GGAGAAAGAGAGACGGAGAGAGAGAGAGACGCACAGATGTGCACGGAGGC
CACAGACACTGACATTTGGAATTCCTTCAGGGTAAAAGGACACCGGAATG
GGAGCTTAGAAGTGTGTTGCTAAGATTTCCGGCTGCACGGAATTATTAAG
TTTTTCTTAAAAAAACAAAAAAAGAAAGAAAGAAAGAAAAAGAAAAGAAC
CCCCTCCGCAGCGAGCCACTTAGGTGCTGCTTTCACGCCAGAGTCCCCTG
TTAAGGTGGCAGCCCTTGATAACTAATCTCGGGCACCCAGCCGCTTCTGT
AAGCTTAAGGAGACGACGAGGAGGGGCGGGGGAAGTGCGTCACCAGGTGG
GGAAGGGGCTGTGTATTTGGTGACAAGCGGGAGGCGATGGGGGTGGAGGG
GAATGGGGACGGGAAATAGGTTCTGTGTGCTCTCCGGGGGTATTGTGTCA
GGAGATGCAGGCTGGCTACCATGTGACGCGGTCCAAAGCTGAAGGGATTG
GCCGAGGCAGCGCAGGCGGTGCAGCTCGGCCAGCTTGCCGTTCCTCTCCC
>NM_013943_up_1000_chr1_25070760_f chr1:25070760-25071759
TGGACAACGACTTGGAAGTCCTTAGTGGCCTGCAGGCACTGAGATTGCAG
```

バージョン違いの影響

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。

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(1000b
したい場合:

上流1000bp以外に2000bp, 5000bpもあります...

```
out_f <- "hoge5.fasta"                          #出力ファイルを指定しておく
param1 <- "BSgenome.Hsapiens.UCSC.hg19"#パッ
param2 <- "upstream1000"                        #上流

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

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

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

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

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

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.Hsapie
param2 <- "upstream1000"

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

#前処理(param1で指定したパッ
#tmp <- unlist(strsplit(pa
tmp <- ls(paste("package",
hoge <- eval(parse(text=tm
hoge

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

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

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 (2
Copyright (C) 2014
Platform: x86_64-w

R は、自由なソフトウェアであ
 一定の条件に従えば、自由
 配布条件の詳細に関しては

R は多くの貢献者による共同
 詳しくは 'contributo
 また、R や R のパッケージ
 'citation()' と入力

 'demo()' と入力すれば
 'help()' とすればオン
 'help.start()' で
 'q()' と入力すれば R

> getwd()
[1] "C:/Users/kado
>
```

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(1000b
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                                               names
    [1]  1000 CCACCTGGGGAAGCGAG...CTTGCCGTTCCTCTCCC NM_032291_up_1000...
    [2]  1000 TGGACAACGACTTGGAA...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...CCTCCCCCACCACCCCC NM_001127389_up_1...
[28017]  1000 AAAGACAGAGCGACGCG...CACCACGCCCTCCCCCA NM_033178_up_1000...
[28018]  1000 AAAGACAGAGCGACGCG...CACCACGCCCTCCCCCA NM_033178_up_1000...
[28019]  1000 TTGTATTTTTAGTAGAG...GCCCTCTAGCTGTGTGT NM_006625_up_1000...
[28020]  1000 TTGTATTTTTAGTAGAG...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の情報は使うな
・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m
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
param1 <- "BSgenome.Hsapiens.UCSC.hg19"#パッケ
param2 <- "upstream1000"                        #上流10

#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(param, ".", fixed=TRUE
tmp <- ls(paste("package", param1, sep=":"))#p
hoge <- eval(parse(text=tmp))                   #文字列t
hoge                                            #確認し

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

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

R Console

```
> fasta <- eval(parse(text=tmp))                #文字$
警告メッセージ:
  Starting with BioC 2.14, upstream sequences a$
  However they can easily be extracted from the$
  sequences with something like (for example fo$

      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 ($
  that contains a gene model based on the exact$
  as the BSgenome object you pass to getSeq(). $
  your own custom TranscriptDb object from vari$
  See the makeTranscriptDbFromUCSC(), makeTrans$
  makeTranscriptDbFromGFF() functions in the Ge$
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fa$
>
```

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ペでうまくいきます。

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                                          names
    [1]  1000 ACACATGCTACCGCG...TTTTCTTTTCGTTAA 100287102
    [2]  1000 GCTATTATCACCTAT...GAAACGAATAACTCT 79501
    [3]  1000 GAATTAGGCTTCTGC...GAGAAAAAGGCGGGG 643837
    [4]  1000 CGGGGAGCCCCGAGG...CCCCAGCTTGGGCCA 148398
    [5]  1000 CGGCGGGGCTCCTAT...GGCGGGAGCGGCGGG 339451
    ...   ... ...
[23052]  1000 GGTGAGCCAATCCTG...GTGGAAATCTCAGCC 283788
[23053]  1000 AGCCCTCCACACAAG...TTCTTCCTCTCCAAC 100507412
[23054]  1000 CGGGGCCCAGGGAGT...AGGCCTCCTGGCTGC 728410
[23055]  1000 CGGGGCCCAGGGAGT...AGGCCTCCTGGCTGC 100653046
[23056]  1000 CAGGCTGAGCCCTGC...CCGGGGCTCACCGCG 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"#
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geno
library(param1, character.only=T)               #指定し
tmp <- ls(paste("package", param1, sep=":"))#
genome <- eval(parse(text=tmp))                 #文字列

library(param2, character.only=T)               #指定し
tmp <- ls(paste("package", param2, sep=":"))#
txdb <- eval(parse(text=tmp))                   #文字列

#本番
gn <- sort(genes(txdb))                         #遺伝子
hoge <- flank(gn, width=param3)                 #指定し
fasta <- getSeq(genome, hoge)                   #指定し

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

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                                names
    [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 AGCCCTCCACA...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でエラーがでます。

```
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
param2 <- "upstream1000"                  #上流1000bpを

#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h
#tmp <- unlist(strsplit(param, ".", fixed=TRUE))[2]
tmp <- ls(paste("package", param1, sep=":"))#param1
hoge <- eval(parse(text=tmp))             #文字列tmpを
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

#本番
tmp <- paste("hoge
fasta <- eval(pars

#ファイルに保存
writeXStringSet(fa
```

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

```
|   HSCHR19KIR_RP5_B_HAP_CTG3_1
|
| (use the '$' or '[[' operator to access a given
| sequence)
>
> #本番
> tmp <- paste("hoge$", 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**

```
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"#
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genc
library(param1, character.only=T)               #指定し
tmp <- ls(paste("package", param1, sep=":"))#
genome <- eval(parse(text=tmp))                 #文字列

library(param2, character.only=T)               #指定し
tmp <- ls(paste("package", param2, sep=":"))#
txdb <- eval(parse(text=tmp))                   #文字列

#本番
gn <- sort(genes(txdb))                         #遺伝子
hoge <- flank(gn, width=param3)                 #指定し
fasta <- getSeq(genome, hoge)                   #指定し

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

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                                names
    [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 AGCCCTCCACA...CCTCTCCAAC 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)

- イントロ | 一般 | Tips | 任意の拡張子でファイルを保存(last modified 2013/09/26)
- イントロ | 一般 | Tips | 拡張子は同じで任意の文字を追加して保存(last modified 20
- イントロ | 一般 | 配列取得 | ゲノム配列 | 公共DBから(last modified 2014/05/28)
- イントロ | 一般 | 配列取得 | ゲノム配列 | BSgenome(last modified 2014/06/28) NEW
- イントロ | 一般 | 配列取得 | プロモーター配列 | 公共DBから(last modified 2014/04/
- イントロ | 一般 | 配列取得 | プロモーター配列 | BSgenome(last modified 2014/04/25
- イントロ | 一般 | 配列取得 | プロモーター配列 | GenomicFeatures(Lawrence 2013)(
- イントロ | 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 2
- イントロ | 一般 | 配列取得 | トランスクリプトーム配列 | biomaRt(Durinck 2009)(last
- イントロ | NGS | 様々なプラットフォーム(last modified 2014/06/10)
- イントロ | NGS | qPCRやmicroarrayなどとの比較(last modified 2014/07/11) NEW
- イントロ | NGS | 可視化(ゲノムブラウザやViewer)(last modified 2014/06/25) NEW
- イントロ | NGS | 配列取得 | FASTQ or SRALite | 公共DBから(last modified 2014/06/28) NEW
- イントロ | NGS | 配列取得 | FASTQ or SRALite | SRAdb(Zhu 2013)(last modified 2014/06/26) NEW
- イントロ | 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
- イントロ | 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) N
- イントロ | NGS | アノテーション情報取得 | について(last modified 2014/03/26)
- イントロ | 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)
- イントロ | NGS | アノテーション情報取得 | refFlat形式ファイル(last modified 2013/09/25)
- イントロ | NGS | アノテーション情報取得 | biomaRt(Durinck 2009)(last modified 2013/09/26)
- イントロ | NGS | アノテーション情報取得 | TranscriptDb | について(last modified 2014/03/28)
- イントロ | NGS | アノテーション情報取得 | TranscriptDb | TxDb.*から(last modified 2013/10/08)
- イントロ | NGS | アノテーション情報取得 | TranscriptDb | GenomicFeatures(Lawrence 2013)(last modified 2013/10
- イントロ | 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)
- イントロ | NGS | 読み込み | FASTA形式 | 基本情報を取得(last modified 2014/05/29)
- イントロ | NGS | 読み込み | FASTA形式 | description行の記述を整形(last modified 2014/04/05)
- イントロ | NGS | 読み込み | FASTQ形式(last modified 2014/06/15)
- イントロ | NGS | 読み込み | FASTQ形式 | description行の記述を整形(last modified 2013/06/13)
- イントロ | NGS | 読み込み | Illuminaの* seq.txt(last modified 2013/06/13)
- イントロ | 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。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

(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
(last modified 2014/07/18, since 2010)

- イントロ | 一般 | 任意のキーワードを含む行を抽出(基礎)

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
- イントロ | 一般 | 任意のキーワードを含む行を抽出(基礎) (last modified 2014/04/11)
- イントロ | 一般 | ランダムな塩基配列を生成 (last modified 2014/06/16)
- 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にあり、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ペ。

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
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
out_f <- "hoge1.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
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
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
```

(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)

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. エラー一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。

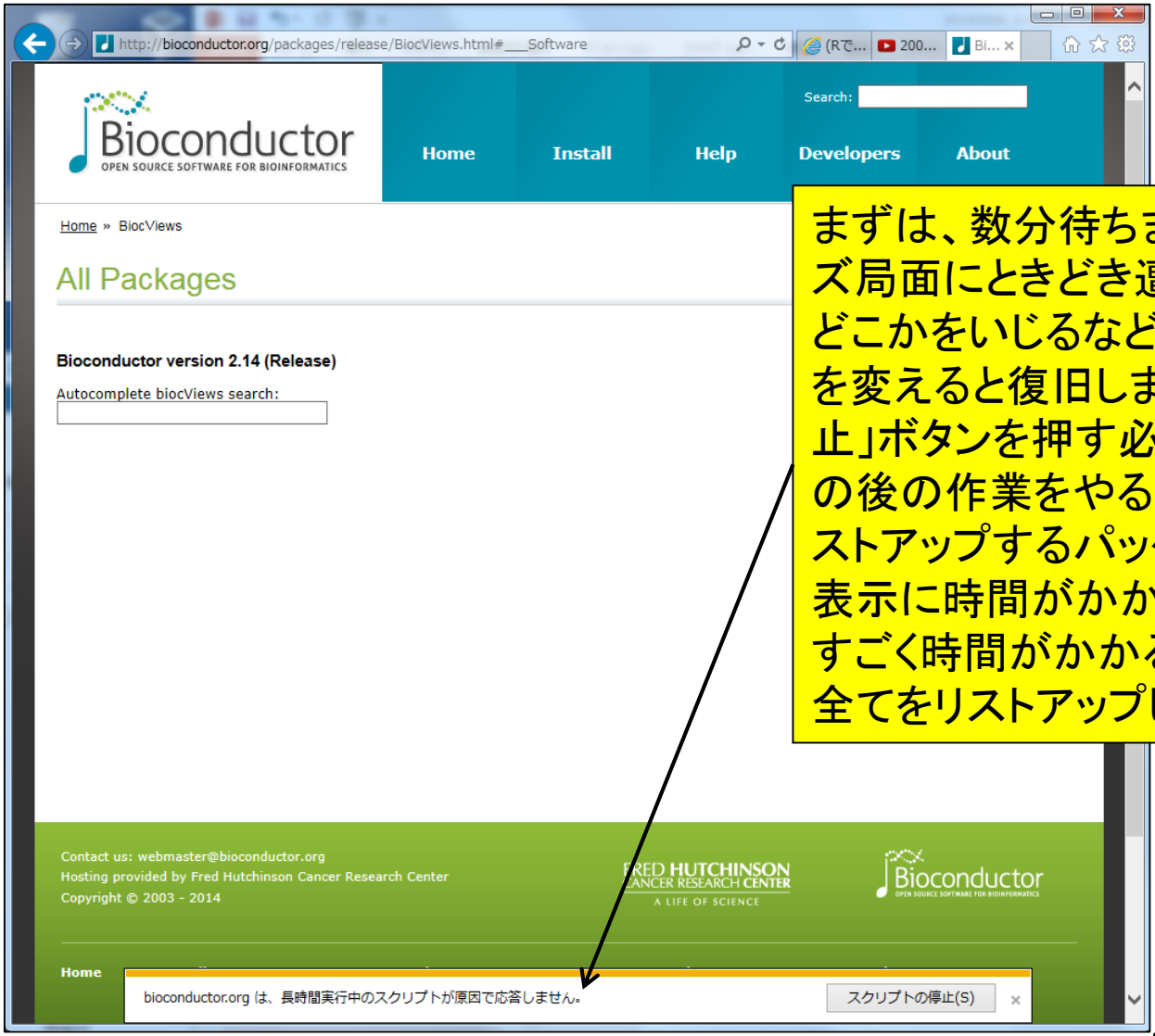
- イントロ | 一般 | Tips | 任意の拡張子でファイルを保存(last modified 2013/09/2
- イントロ | 一般 | Tips | 拡張子は同じで任意の文字を追加して保存(last modifi
- イントロ | 一般 | 配列取得 | ゲノム配列 | 公共DBから(last modified 2014/05/2
- イントロ | 一般 | 配列取得 | ゲノム配列 | BSgenome(last modified 2014/06/28)
- イントロ | 一般 | 配列取得 | プロモーター配列 | 公共DBから(last modified 201
- イントロ | 一般 | 配列取得 | プロモーター配列 | BSgenome(last modified 2014/
- イントロ | 一般 | 配列取得 | プロモーター配列 | GenomicFeatures(Lawrence 20
- イントロ | 一般 | 配列取得 | トランスクリプトーム配列 | 公共DBから(last modif
- イントロ | 一般 | 配列取得 | トランスクリプトーム配列 | biomaRt(Durinck 2009)
- イントロ | NGS | 様々なプラットフォーム(last modified 2014/06/10)
- イントロ | NGS | qPCRやmicroarrayなどとの比較(last modified 2014/07/11) N
- イントロ | NGS | 可視化(ゲノムブラウザやViewer)(last modified 2014/06/25)
- イントロ | NGS | 配列取得 | FASTQ or SRALite | 公共DBから(last modified 2014/06/28) NEW
- イントロ | NGS | 配列取得 | FASTQ or SRALite | SRAdb(Zhu 2013)(last modified 2014/06/26) NEW
- イントロ | 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
- イントロ | 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) N
- イントロ | NGS | アノテーション情報取得 | について(last modified 2014/03/26)
- イントロ | 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)
- イントロ | NGS | アノテーション情報取得 | refFlat形式ファイル(last modified 2013/09/25)
- イントロ | NGS | アノテーション情報取得 | biomaRt(Durinck 2009)(last modified 2013/09/26)
- イントロ | NGS | アノテーション情報取得 | TranscriptDb | について(last modified 2014/03/28)
- イントロ | NGS | アノテーション情報取得 | TranscriptDb | TxDb.*から(last modified 2013/10/08)
- イントロ | NGS | アノテーション情報取得 | TranscriptDb | GenomicFeatures(Lawrence 2013)(last modified 2013/10
- イントロ | 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)
- イントロ | NGS | 読み込み | FASTA形式 | 基本情報を取得(last modified 2014/05/29)
- イントロ | NGS | 読み込み | FASTA形式 | description行の記述を整形(last modified 2014/04/05)
- イントロ | NGS | 読み込み | FASTQ形式(last modified 2014/06/15)
- イントロ | NGS | 読み込み | FASTQ形式 | description行の記述を整形(last modified 2013/06/13)
- イントロ | NGS | 読み込み | Illuminaの* seq.txt(last modified 2013/06/13)
- イントロ | NGS | 読み込み | Illuminaの* qseq.txt(last modified 2013/06/17)

Bioconductor概観

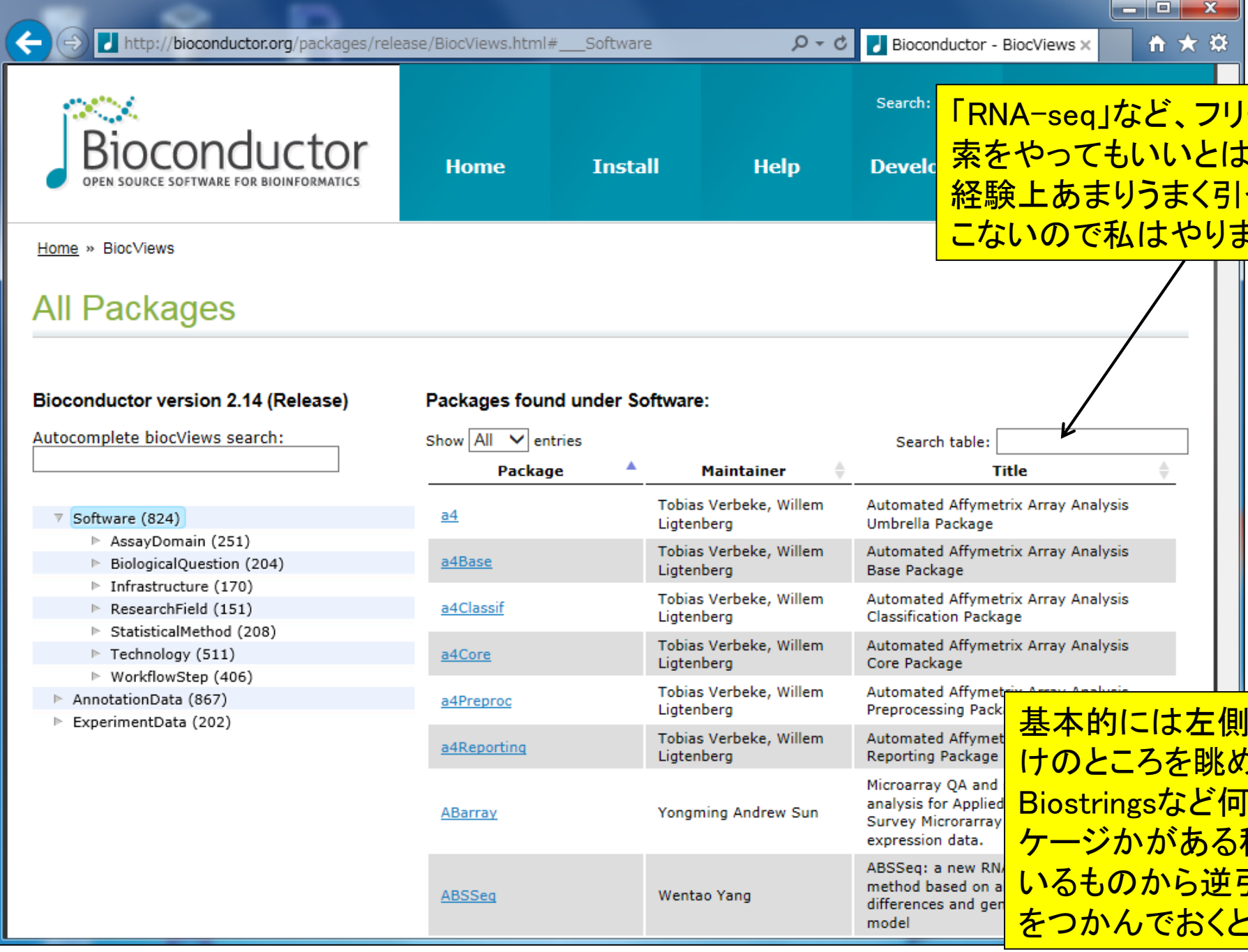
NGSに特化した内容ではないが、R Tipsは門田が独学でRを勉強し始めた2005年頃から今でも時々お世話になっているウェブサイト。

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概観

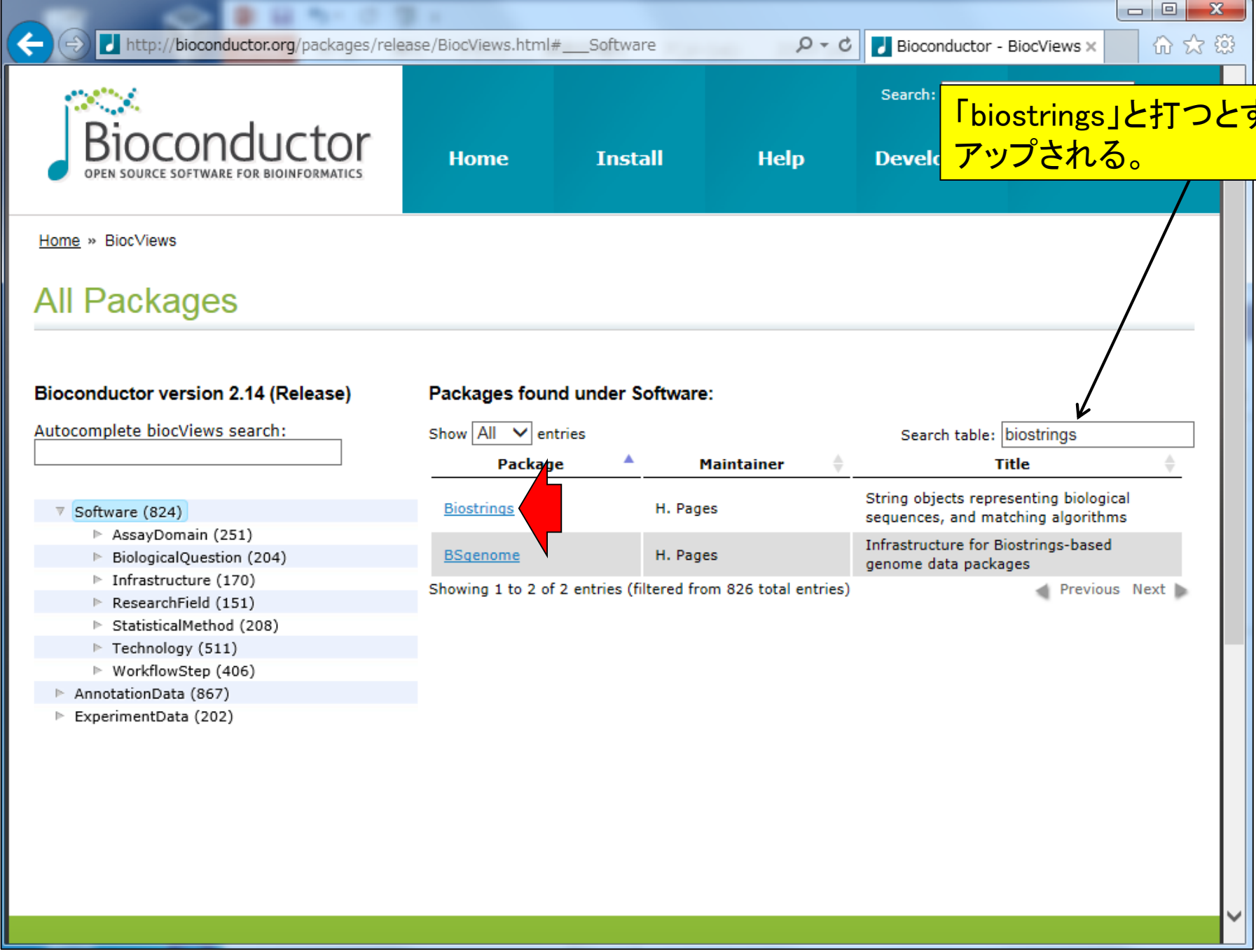


まずは、数分待ちます。このようなフリーズ局面にときどき遭遇しますが、四隅のどこかをいじるなどしてウィンドウサイズを変えると復旧します。「スクリプトの停止」ボタンを押す必要はありません。この後の作業をやると体感できますが、リストアップするパッケージ数に比例して表示に時間がかかるようです。最初にすごく時間がかかるのは824パッケージ全てをリストアップしているためです。

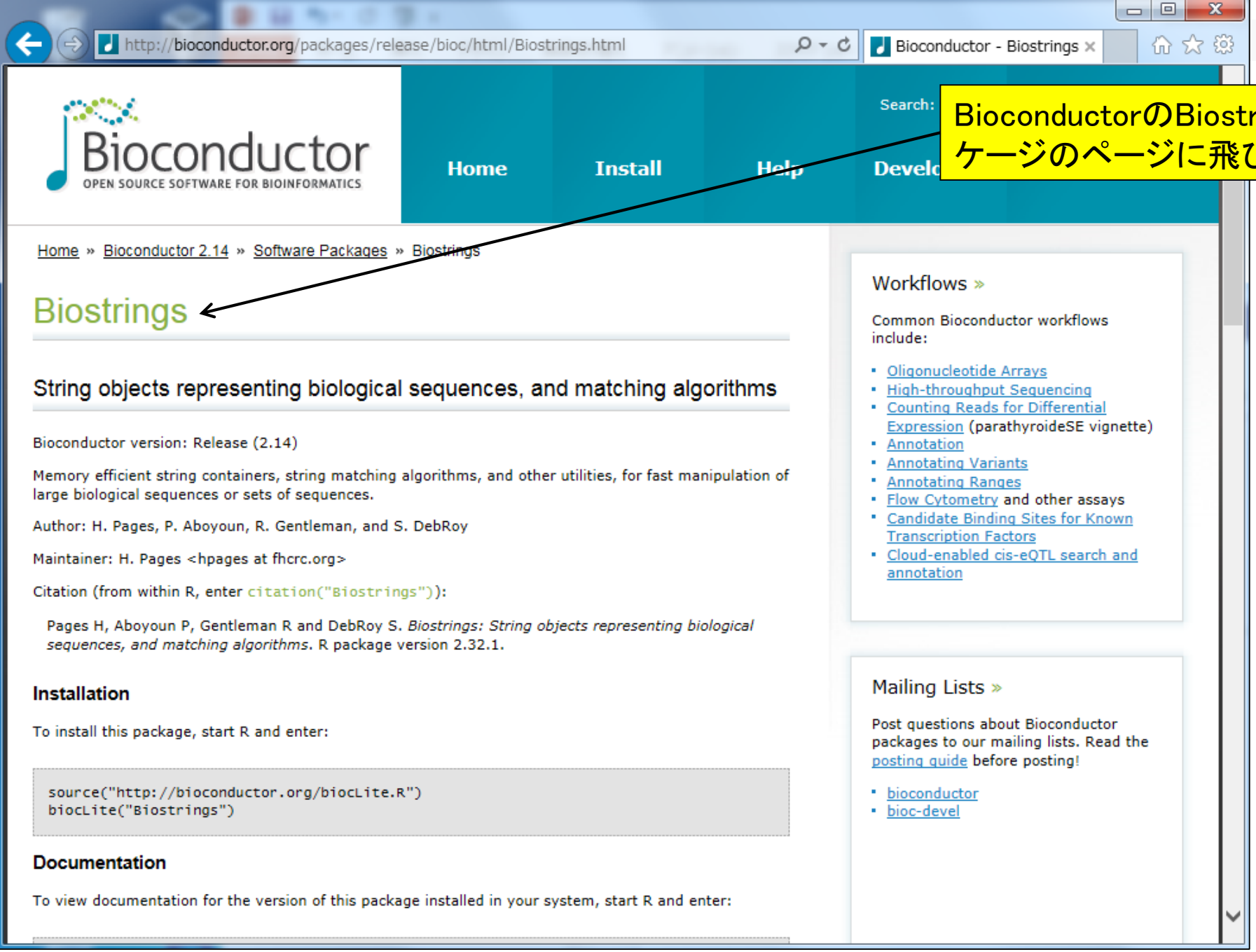


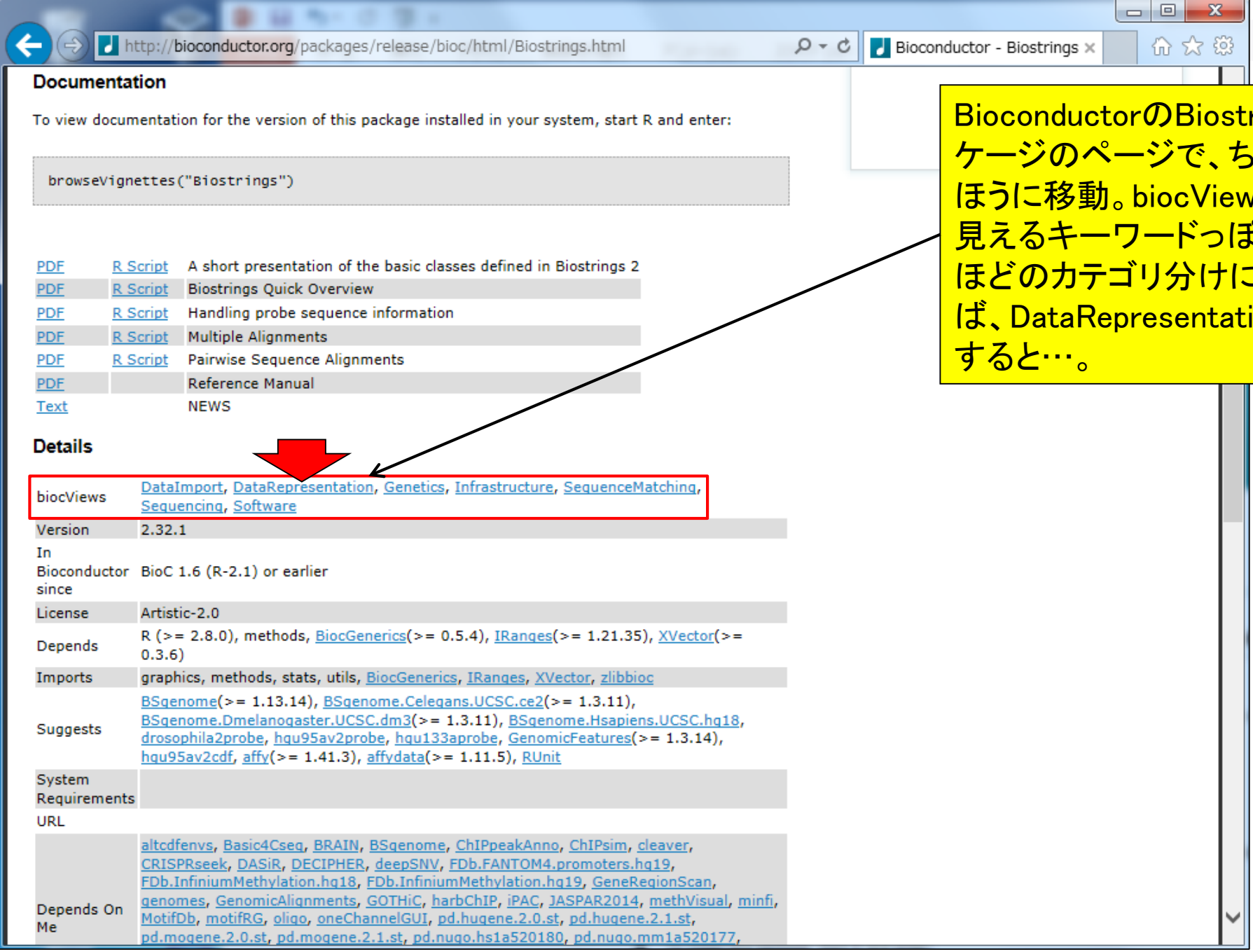
「RNA-seq」など、フリーワード検索をやってもいいとは思いますが、経験上あまりうまく引っかかってこないので私はやりません。

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。

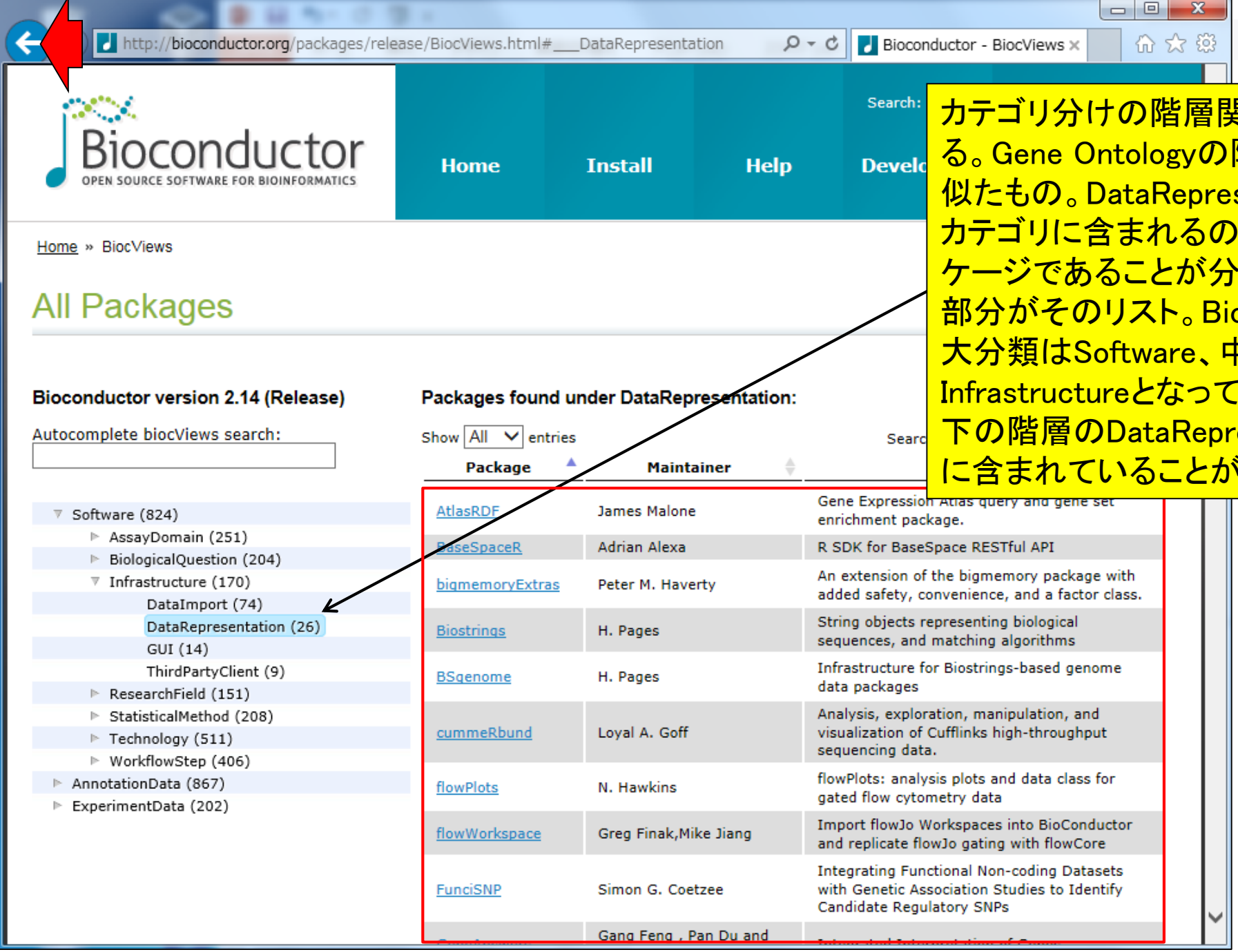


「biostrings」と打つとすぐにリストアップされる。


http://bioconductor.org/packages/release/bioc/html/Biostrings.html
Bioconductor - Biostrings
Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS
Search:
Home
Install
Help
Develc
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

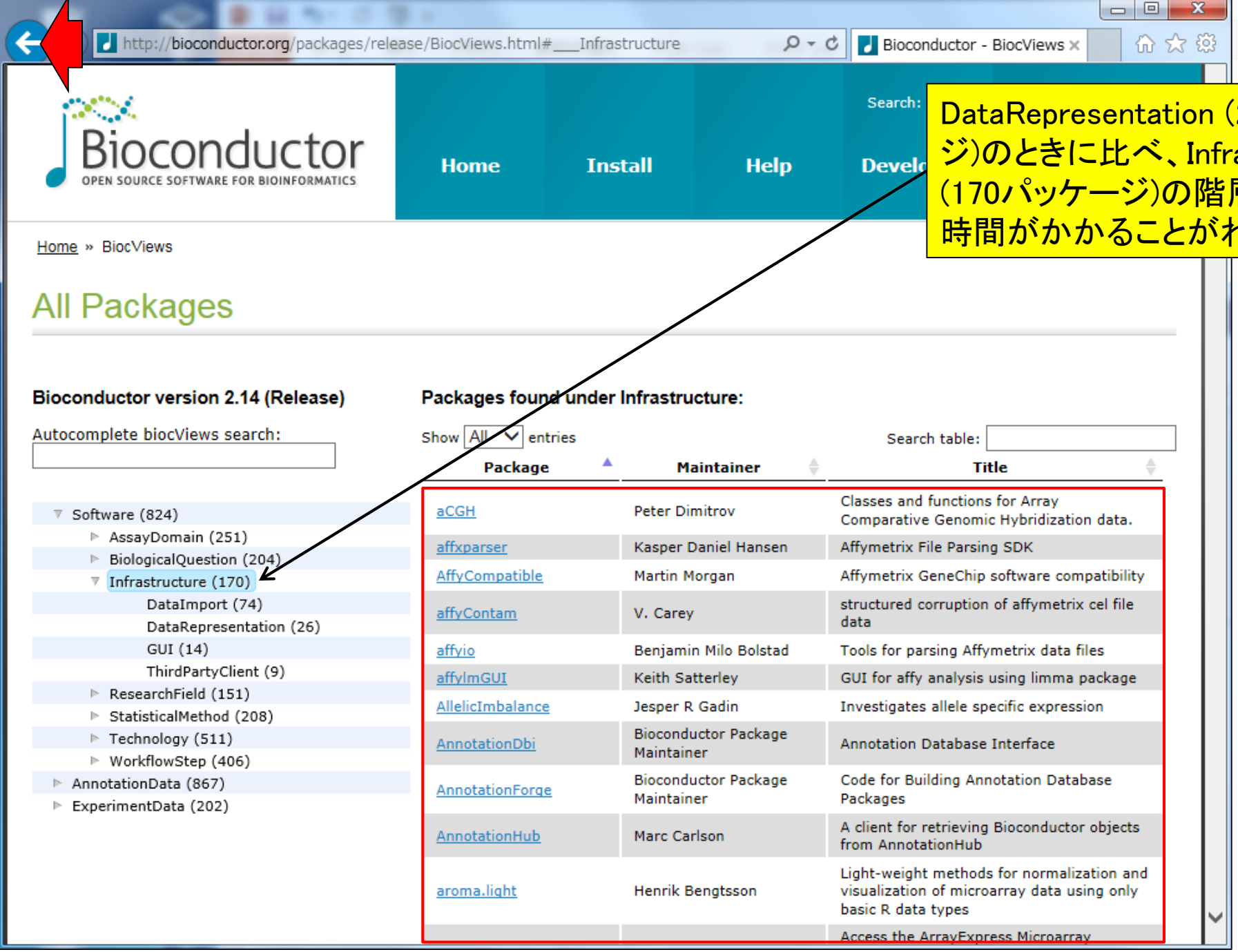


BioconductorのBiostringsパッケージのページで、ちょっと下のほうに移動。biocViewsのところで見えるキーワードっぽいのがさきほどのカテゴリ分けに相当。例えば、DataRepresentationをクリックすると…。



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。


http://bioconductor.org/packages/release/bioc/html/Biostrings.html
Bioconductor - Biostrings
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 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, BiocGenerics(>= 0.5.4), IRanges(>= 1.21.35), XVector(>= 0.3.6)
Imports graphics, methods, stats, utils, BiocGenerics, IRanges, XVector, zlibbioc
Suggests BSgenome(>= 1.13.14), BSgenome.Celegans.UCSC.ce2(>= 1.3.11), BSgenome.Dmelanogaster.UCSC.dm3(>= 1.3.11), BSgenome.Hsapiens.UCSC.hg18, drosophila2probe, hgu95av2probe, hgu133aprobe, GenomicFeatures(>= 1.3.14), hgu95av2cdf, affy(>= 1.41.3), affydata(>= 1.11.5), RUnit
System Requirements
URL
Depends On Me altcdfenvs, Basic4Cseq, BRAIN, BSgenome, ChIPpeakAnno, ChIPsim, cleaver, CRISPRseek, DASiR, DECIPHER, deepSNV, FDb.FANTOM4.promoters.hg19, FDb.InfiniumMethylation.hg18, FDb.InfiniumMethylation.hg19, GeneRegionScan, genomes, GenomicAlignments, GOTHiC, harbChIP, iPAC, JASPAR2014, methVisual, minfi, MotifDb, motifRG, oligo, oneChannelGUI, pd.hugene.2.0.st, pd.hugene.2.1.st, pd.mogene.2.0.st, pd.mogene.2.1.st, pd.nugo.hs1a520180, pd.nugo.mm1a520177,
大分類のSoftware、中分類のInfrastructureも存在する。



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。

http://bioconductor.org/packages/release/bioc/html/Biostrings.html

Bioconductor - Biostrings

Documentation

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

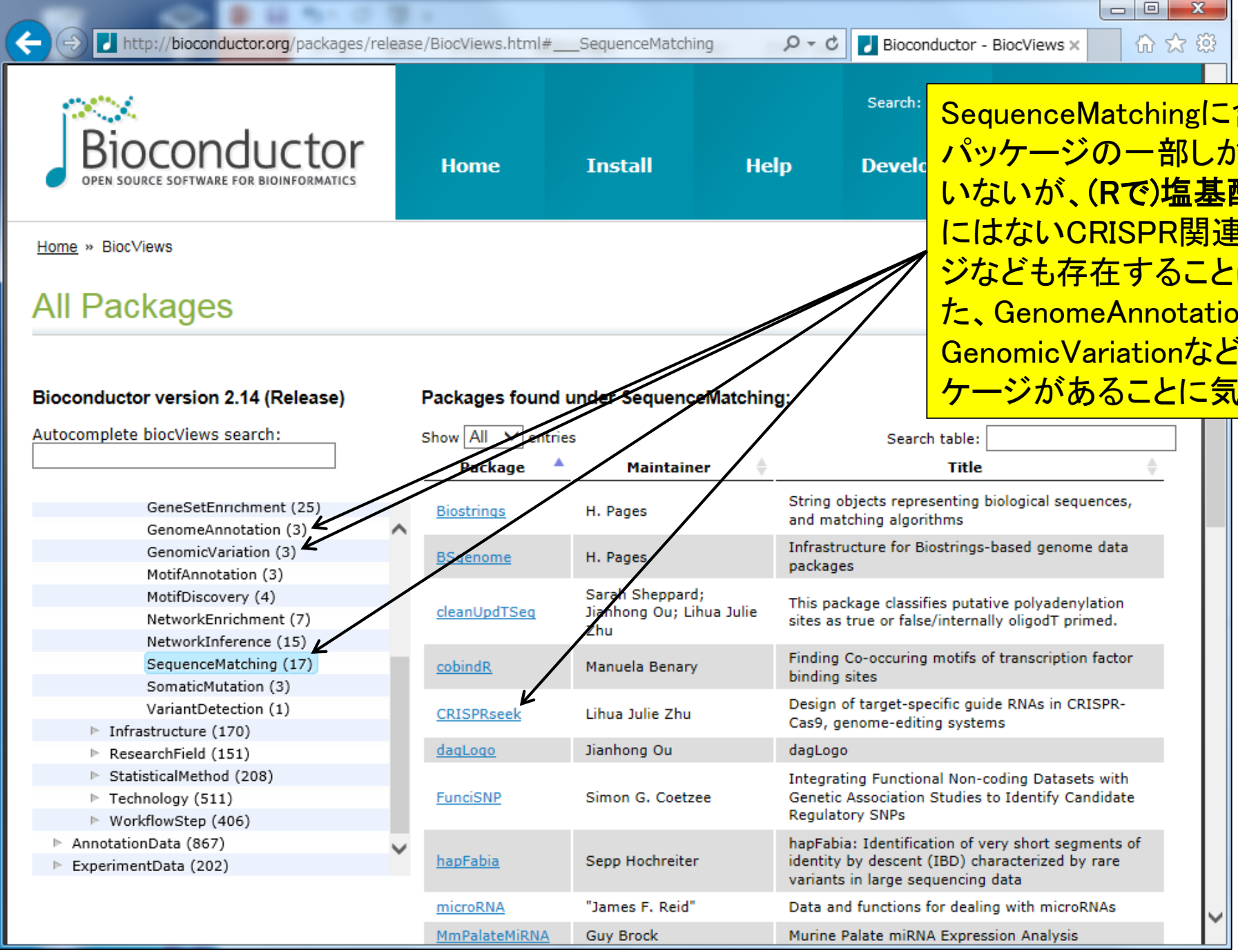
```
browseVignettes("Biostrings")
```

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

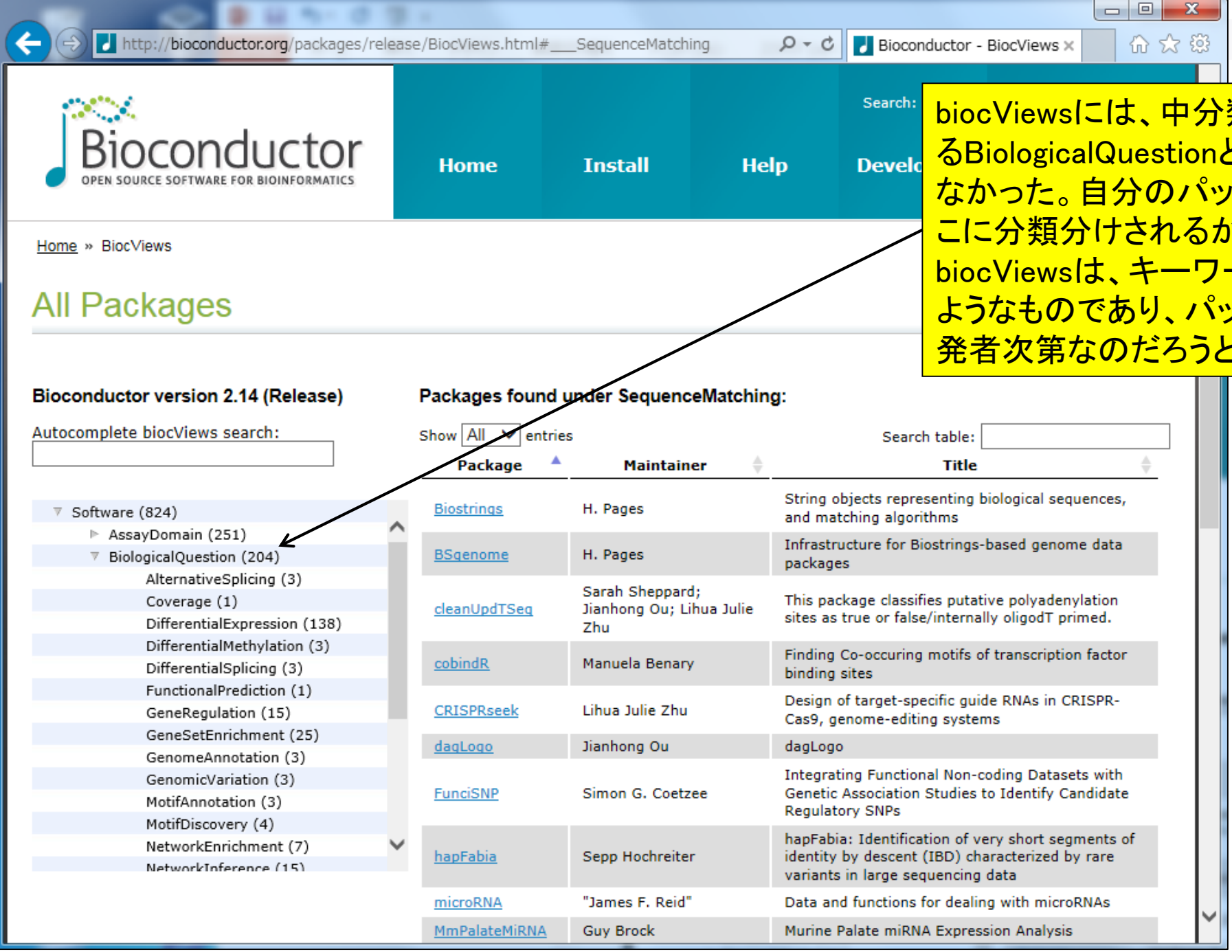
Details

biocViews	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, BiocGenerics(>= 0.5.4), IRanges(>= 1.21.35), XVector(>= 0.3.6)
Imports	graphics, methods, stats, utils, BiocGenerics, IRanges, XVector, zlibbioc
Suggests	BSgenome(>= 1.13.14), BSgenome.Celegans.UCSC.ce2(>= 1.3.11), BSgenome.Dmelanogaster.UCSC.dm3(>= 1.3.11), BSgenome.Hsapiens.UCSC.hg18, drosophila2probe, hgu95av2probe, hgu133aprobe, GenomicFeatures(>= 1.3.14), hgu95av2cdf, affy(>= 1.41.3), affydata(>= 1.11.5), RUnit
System Requirements	
URL	
Depends On Me	altcdfenvs, Basic4Cseq, BRAIN, BSgenome, ChIPpeakAnno, ChIPsim, cleaver, CRISPRseek, DASiR, DECIPHER, deepSNV, FDb.FANTOM4.promoters.hg19, FDb.InfiniumMethylation.hg18, FDb.InfiniumMethylation.hg19, GeneRegionScan, genomes, GenomicAlignments, GOTHiC, harbChIP, iPAC, JASPAR2014, methVisual, minfi, MotifDb, motifRG, oligo, oneChannelGUI, pd.hugene.2.0.st, pd.hugene.2.1.st, pd.mogene.2.0.st, pd.mogene.2.1.st, pd.nugo.hs1a520180, pd.nugo.mm1a520177,

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。



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。



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。