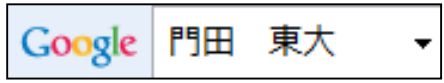
バイオインフォマティクス人材育成カリキュラム（次世代シークエンサ）速習コース
3. データ解析基礎 | 3-3. R各種パッケージ

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

- 3-3. R 各種パッケージ、2014/09/08 15:00-18:15、中級、実習
  - パッケージ
    - 「(Rで)塩基配列解析」のインストール手順おさらい
    - CRANとBioconductor
    - 代表的なパッケージBiostringsの利用法：library, search, objects関数
    - 作業スペース(workspace)の概念
    - Biostringsパッケージで利用可能な関数を概観
  - source関数の利用

パッケージ

(Rで)塩基配列解析の推奨インストール手順をおさらい。大まかな手順は、①R本体のインストール、および②CRANおよびBioconductorから提供されている各種パッケージのインストール

(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

(last modified 2014/08/08, since 2010)

What's new?

- このウェブページはフリーソフトRと必要な
  1. Rのインストールと起動および2. 基本的
- 2014年10月 HPCIワークショップ「医
  フォマティクス入門講習会@仙台国際セン
- 門田幸二 著シリーズ Useful R 第7巻トラ

- 日本乳酸菌学会誌のNGS関連連載の第1
- 参考資料(講義、講習会、本など)の項目を

- はじめに (last modified 2014/01/30)
- 参考資料(講義、講習会、本など) (last mo
- 過去のお知らせ (last modified 2014/08/03)
- Rのインストールと起動 (last modified 201
- 基本的な利用法 (last modified 2014/07/20
- サンプルデータ (last modified 2014/07/17)
- バイオインフォマティクス人材育成カリキュ
- 書籍|トランスクリプトームについて (last m

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。2014年7月31日にアップデートしたWindows用のインストール手順はこちら。2014年5月14日にアップデートしたMac版のインストール手順こちら(by 孫建強氏)もあります。注意点は、「Mac OS X のバージョンに関わらず R-3.1.0-snowleopard.pkg をインストールしたほうがよい」です。

1. Windows release版のインストールの場合:

①
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させる
3. **Windows Vistaの人**は(パッケージのインストール中に書き込み権限に関するエラーが出るのを避けるために)「コントロールパネル」-「ユーザーアカウント」-「ユーザーアカウント制御の有効化または無効化」で、「ユーザーアカウント制御(UAC)を使ってコンピュータの保護に役立たせる」の**チェックをあらかじめ外しておくことを強くお勧め**します。

②
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* (32 bitの場合; *X*や*Y*中の数値はバージョンによって異なります)」または「R x64 3.*X*.*Y* (64 bitの場合)」アイコンをダブルクリックして起動
5. 以下を、「Rコンソール画面上」でコピー&ペーストする。10GB程度のディスク容量を要しますが一番お手軽です。(どこからダウンロードするか？と聞かれるので、その場合は自分のいる場所から近いサイトを指定)

```
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本体とパッケージの関係

- パソコンを購入しただけの状態では、できることが限られています。
  - 通常は、Officeやウイルス撃退ソフトなどをインストールして利用します。
- Linuxをインストールしただけの状態では、できることが限られています。
  - 通常は、マッピングなど各種プログラムをインストールして利用します。
- R本体をインストールしただけの状態では、できることが限られています。
  - NGS解析を行う各種パッケージ（またはライブラリ）をインストールして利用します。

「R本体」と「パッケージ」の関係は、「パソコン」と「ソフト」、「Microsoft EXCEL」と「アドイン」、「Cytoscape」と「プラグイン」のようなものという理解でよい。

CRANとBioconductor

- R上で利用可能なパッケージの2大リポジトリ(貯蔵庫)
  - CRAN (The Comprehensive R Archive Network):5,802パッケージ
  - Bioconductor:824パッケージ

CRANは生命科学分野に限らず様々な分野で利用されるパッケージを含む。NGS解析は、主にBioconductorから提供されているパッケージを利用します。

パッケージ

- 「(Rで)塩基配列解析」のインストール手順おさらい

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
14日にアップデートしたMac版のインストール手順こちら(by
ンストールしたほうがよい」です。

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させる
3. **Windows Vistaの人**は(パッケージのインストール中に書き込み権限に関するエラーが出るのを避けるために)「コントロールパネル」-「ユーザーアカウント」-「ユーザーアカウント制御の有効化または無効化」で、「ユーザーアカウント制御(UAC)を使ってコンピュータの保護に役立たせる」の**チェックをあらかじめ外しておくことを強くお勧め**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* (32 bitの場合; *X*や*Y*中の数値はバージョンによって異なります)」または「R x64 3.*X*.*Y* (64 bitの場合)」アイコンをダブルクリックして起動
5. 以下を、「Rコンソール画面上」でコピー&ペーストする。10GB程度のディスク容量を要しますが一番お手軽です。(どこからダウンロードするか？と聞かれるので、その場合は自分のいる場所から近いサイトを指定)

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

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で)塩基配列解析の推奨インストール手順は、CRANおよびBioconductorから提供されている各種パッケージを予めインストールしておくこと。たった3行のコードで2つのリポジトリから提供されているパッケージ群を一度にインストール。有線LAN接続環境でも数時間程度かかるのは数千ものパッケージをダウンロードしてインストールする作業に相当する部分だから。

- 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 mod
- 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
- イントロ | 一般 | 逆鎖(reverse)を取得(last modified 2013/06/14)
- イントロ | 一般 | 2連続塩基の出現頻度情報を取得(last modified 2014/07/18) NEW
- イントロ | 一般 | 3連続塩基の出現頻度情報を取得(last modified 2013/06/14)
- イントロ | 一般 |
- イントロ | 一般 |
- イントロ | 一般 |
- イントロ | 一般 |
- イントロ | 一般 |
- イントロ | 一般 |
- イントロ | 一般 |
- イントロ | 一般 |
- イントロ | 一般 |
- イントロ | 一般 |
- イントロ | NGS |
- イントロ | NGS |
- イントロ | NGS |
- イントロ | NGS |
- イントロ | NGS |

- イントロ | 一般 | 翻訳配列(translate)を取得

塩基配列を入力としてコピペで翻訳配列を得ることができたのは、Bioconductorから提供されているBiostringsというパッケージを予めインストールしておいたからです。

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.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
```

イントロ | 一般 | 翻訳配列(translate)を取得 NEW

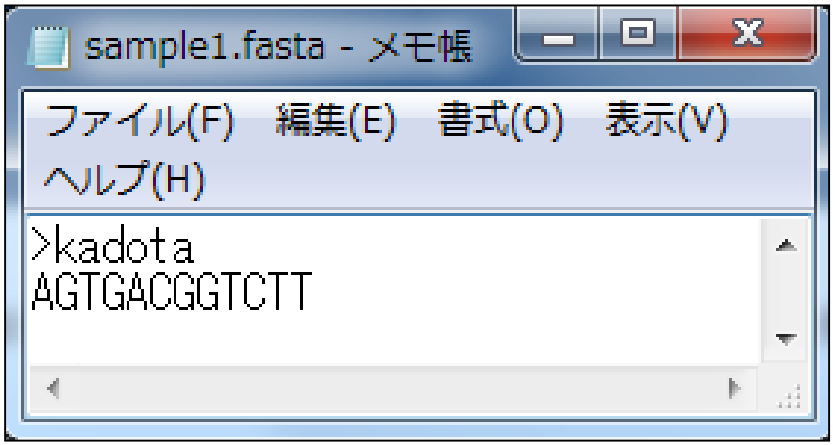
塩基配列を読み込んでアミノ酸配列に翻訳するやり方を示します。
「ファイル」-「ディレクトリの変更」で解析したいファイルを置いて

塩基配列を入力としてコピペで翻訳配列を得ることができたのは、Bioconductorから提供されているBiostringsというパッケージを予めインストールしておいたからです。

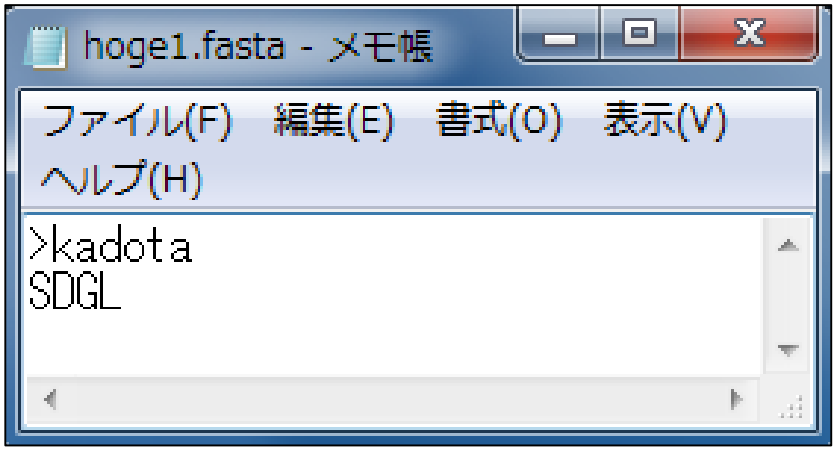
1. FASTA形式ファイル(sample1.fasta)の場合:

```
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
writeXStrin
```

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)



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」ー「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
out_f <- "hoge1.fasta"                     #出力ファイル名を指

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

#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
```

Biostringsパッケージが提供している様々な関数を利用するという宣言。Biostringsパッケージのインストールができていない状態でlibrary関数を実行しても、「Biostringsという名前のパッケージはありません」という類のエラーが出て、目的の結果は得られません。

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
「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

1. FASTA形式ファイル(sample1.fasta)の場合:

```
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)

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

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

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

Biostringsパッケージがインストールされていても、library関数でBiostringsパッケージを読み込まないと、Biostringsパッケージで提供されている関数やサンプルデータを利用することができない。以下のコードは、library(Biostrings)の左側に#を入れて、実行されないように変更したもの。

```
############################↓
###  翻訳配列取得(Biostringsパッケージの読み込みなし)↓
############################↓
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
↓
#本番↓
hoge <- translate(fasta)                   #fastaをアミノ酸配列に翻訳した結果をh
names(hoge) <- names(fasta)                #現状では翻訳した結果のオブジェクトho
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
↓
↓
```

原因既知状態でエラーを出す

（すでに開いているRがあれば一旦終了し）
R起動直後の状態で下記コードをコピペ

R Console

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

R は、自由なソフトウェアであり、「完全
 一定の条件に従えば、自由にこれを再
 配布条件の詳細に関しては、'lic

R は多くの貢献者による共同プロジェク
 詳しくは 'contributors()'
 また、R や R のパッケージを出版物
 'citation()' と入力してくださ

 'demo()' と入力すればデモをみ
 'help()' とすればオンラインヘル
 'help.start()' で HTML
 'q()' と入力すれば R を終了し

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

```
############################
###  翻訳配列取得(Biostringsパッケージの読み込みなし)
############################
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

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

原因既知状態でエラーを出す

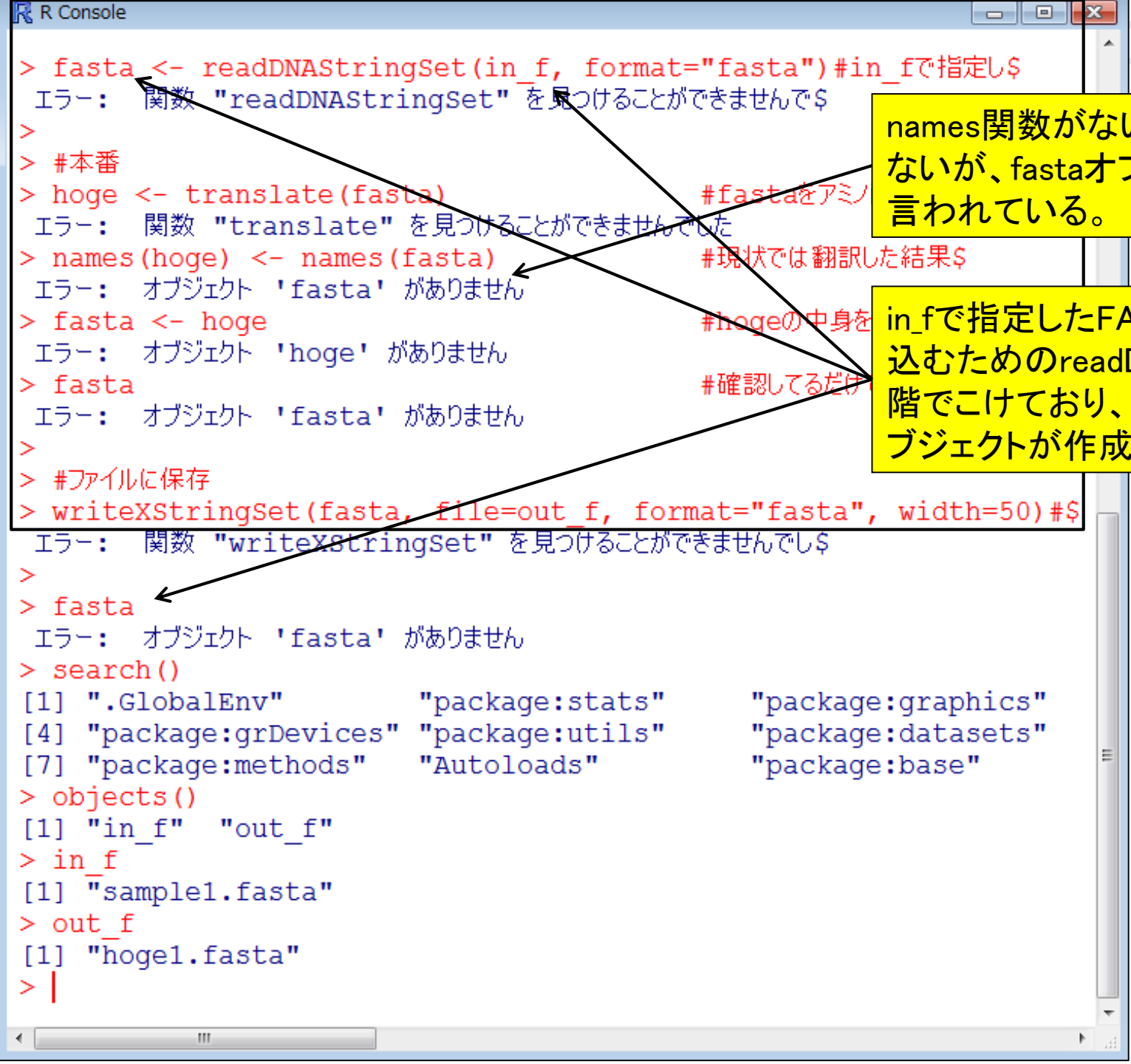
ファイル名：rcode_20140908.txt

```
############################↓
###  翻訳配列取得(Biostrings/
############################↓
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
↓
↓
```

R Console

```
> in_f <- "sample1.fasta"
> out_f <- "hoge1.fasta"
>
> #必要なパッケージをロード
> #library(Biostrings)                    #パッケージの読み$
>
> #入力ファイルの読み込み
> fasta <- readDNAStringSet(in_f, format="fasta")#in_fで指定$
 エラー:  関数 "readDNAStringSet" を見つけることができません$
>
> #本番
> hoge <- translate(fasta)                #fastaをアミノ酸配$
 エラー:  関数 "translate" を見つけることができませんでした
> names(hoge) <- names(fasta)             #現状では翻訳した結$
 エラー:  オブジェクト 'fasta' がありません
> fasta <- hoge                           #hogeの中身をfasta$
 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=5$
 エラー:  関数 "writeXStringSet" を見つけることができません$
>
```

3つの関数(readDNAStringSet, translate, and writeXStringSet)がBiostringsパッケージから提供されているものであることが分かる。また、names関数はBiostringsパッケージとは無関係であることもわかる。



• イントロ | 一般 | 翻訳配列(translate)を取得

names関数がないと文句を言われてはいないが、fastaオブジェクトがないと文句を言われている。

in_fで指定したFASTA形式ファイルを読み込むためのreadDNAStringSet関数実行段階でこけており、読み込み結果のfastaオブジェクトが作成されていないため。

R Console

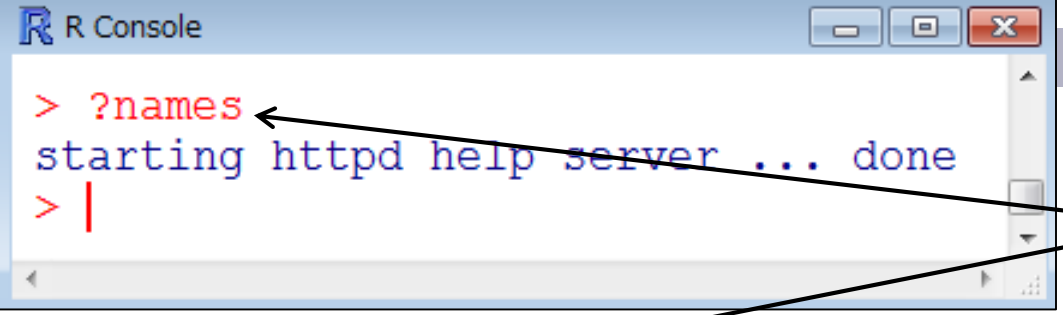
```
> fasta <- readDNAStringSet(in_f, format="fasta")#in_fで指定し$
 エラー:  関数 "readDNAStringSet" を見つけることができませんで$
>
> #本番
> hoge <- translate(fasta)                    #fastaをアミノ
 エラー:  関数 "translate" を見つけることができませんでした
> names(hoge) <- names(fasta)                 #現状では翻訳し
 エラー:  オブジェクト 'fasta' がありません
> fasta <- hoge                               #hogeの中身を
 エラー:  オブジェクト 'hoge' がありません
> fasta                                       #確認してるだけ
 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)#$
 エラー:  関数 "writeXStringSet" を見つけることができませんでし$
>
> fasta
 エラー:  オブジェクト 'fasta' がありません
> search()
[1] ".GlobalEnv"        "package:stats"     "package:graphics"
[4] "package:grDevices" "package:utils"     "package:datasets"
[7] "package:methods"   "Autoloads"         "package:base"
> objects()
[1] "in_f"  "out_f"
> in_f
[1] "sample1.fasta"
> out_f
[1] "hoge1.fasta"
>
```

search関数は現在読み込んでいるパッケージをリストアップ。このR環境は、明示的に何も読み込んでいないのにいくつか表示されている。これは、R起動時に基本的なパッケージ(stats, graphics, utils, baseなど)を自動的に読み込んでいるから。ちなみにnames関数はbaseパッケージで提供されているのでいつでも利用可能。

R Console

```
> fasta <- readDNAStringSet(in_f, format="fasta")#in_fで指定し$
 エラー:  関数 "readDNAStringSet" を見つけることができませんで$
>
> #本番
> hoge <- translate(fasta)                    #fastaをアミノ
 エラー:  関数 "translate" を見つけることができませんでした
> names(hoge) <- names(fasta)                 #現状では翻訳し
 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$
 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)#$
 エラー:  関数 "writeXStringSet" を見つけることができませんでし$
>
> fasta
 エラー:  オブジェクト 'fasta' がありません
> search()
[1] ".GlobalEnv"        "package:stats"     "package:graphics"
[4] "package:grDevices" "package:utils"     "package:datasets"
[7] "package:methods"   "Autoloads"         "package:base"
> objects()
[1] "in_f"  "out_f"
> in_f
[1] "sample1.fasta"
> out_f
[1] "hoge1.fasta"
>
```

translate関数もないしfastaオブジェクトもないので、translate(fasta)実行結果のhogeオブジェクトもない。ではどんなオブジェクトが利用可能かを調べるのがobjects関数。ls関数でもよい。



?namesを実行することで、names関数がbaseパッケージから提供されていることがわかる。

names {base} R Documentation

The Names of an Object

Description

Functions to get or set the names of an object.

Usage

```
names(x)
names(x) <- value
```

Arguments

x	an R object.
value	a character vector of up to the same length as x, or NULL.

Details

names is a generic accessor function, and names<- is a generic replacement function. The default methods get and set the "names" attribute of a vector (including a list) or pairlist.

If value is shorter than x, it is extended by character NAs to the length of x.

R Console

```
> search()
[1] ".GlobalEnv"        "package:stats"
[3] "package:graphics"  "package:grDevices"
[5] "package:utils"     "package:datasets"
[7] "package:methods"   "Autoloads"
[9] "package:base"
> ?readDNAStringSet
No documentation for 'readDNAStringSet' in specified packages and libraries:
you could try '??readDNAStringSet'
> library(Biostrings)
 要求されたパッケージ BiocGenerics をロード中です
 要求されたパッケージ parallel をロード中です

 次のパッケージを付け加えます: 'BiocGenerics'

 以下のオブジェクトはマスクされています (from 'package:parallel') :

     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
    clusterExport, clusterMap, parApply, parCapply, parLapply,
    parLapplyLB, parRapply, parSapply, parSapplyLB

 以下のオブジェクトはマスクされています (from 'package:stats') :

     xtabs
```

?関数名でhtmlマニュアルを見ることができるは、**search()**で表示されているパッケージで提供されている関数のみ。Biostringsパッケージを読み込んでない段階では、Biostringsが提供している関数のマニュアルを見ることはできない。

R Console

```
     anyDuplicated, append, as.data.frame, as.
    cbind, colnames, do.call, duplicated, eval
    Filter, Find, get, intersect, is.unsorted,
    Map, mapply, match, mget, order, paste, pm
    pmax.int, pmin, pmin.int, Position, rank,
    Reduce, rep.int, rownames, sapply, setdiff
    table, tapply, union, unique, unlist

 要求されたパッケージ IRanges をロード中です
 要求されたパッケージ XVector をロード中です
> search()
 [1] ".GlobalEnv"            "package:Biostrings"
 [3] "package:XVector"       "package:IRanges"
 [5] "package:BiocGenerics" "package:parallel"
 [7] "package:stats"         "package:graphics"
 [9] "package:grDevices"     "package:utils"
[11] "package:datasets"      "package:methods"
[13] "Autoloads"             "package:base"
>
```

library(Biostrings)実行によって、search()で表示されているパッケージリストにBiostringsを含むいくつかのパッケージが追加されていることが分かる。この段階で?readDNAStringSetが有効となり、htmlマニュアルが開くようになる。

R Console

```
> search()
 [1] ".GlobalEnv"            "package:Biostrings"
 [3] "package:XVector"       "package:IRanges"
 [5] "package:BiocGenerics"  "package:parallel"
 [7] "package:stats"         "package:graphics"
 [9] "package:grDevices"     "package:utils"
[11] "package:datasets"      "package:methods"
[13] "Autoloads"             "package:base"
> ?readDNAStringSet
starting httpd help server ... done
>
```

library(Biostrings)実行によって、search()で表示されているパッケージリストにBiostringsを含むいくつかのパッケージが追加されていることが分かる。この段階で?readDNAStringSetが有効となり、htmlマニュアルが開くようになる。

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:
readBStringSet(filepath, format="fasta",
               nrec=-1L, skip=0L, seek.first.rec=FALSE, use.names=TRUE)
readDNAStringSet(filepath, format="fasta",
               nrec=-1L, skip=0L, seek.first.rec=FALSE, use.names=TRUE)
readRNAStringSet(filepath, format="fasta",
               nrec=-1L, skip=0L, seek.first.rec=FALSE, use.names=TRUE)
readAAStringSet(filepath, format="fasta",
               nrec=-1L, skip=0L, seek.first.rec=FALSE, use.names=TRUE)

## Extract basic information about FASTA (or FASTQ) files
## without actually loading the sequence data:
```

もう一度コピペ

ファイル名：rcode_20140908.txt

```
############################↓
###  翻訳配列取得(Biostringsパッケージの読み込みなし)↓
############################
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(
↓
#本番↓
hoge <- translate(fasta)
names(hoge) <- names(fasta
fasta <- hoge
fasta
↓
#ファイルに保存↓
writeXStringSet(fasta, fil
↓
↓
```

#でコメントアウトしているにも関わらずエラーが出ないのは、以前にBiostringsパッケージを読み込んでいるから。

R Console

```
> in_f <- "sample1.fasta"                    #入力ファイル名を指定して$
> out_f <- "hoge1.fasta"                     #出力ファイル名を指定して$
>
> #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$
>
> #本番
> hoge <- translate(fasta)                   #fastaをアミノ酸配列に翻$
> names(hoge) <- names(fasta)                #現状では翻訳した結果のオ$
> fasta <- hoge                              #hogeの中身をfastaに格納
> fasta                                      #確認してるだけです
  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", width=50)#fas$
>
```

利用可能なオブジェクトを表示

R Console

```
> hoge <- translate(fasta)                    #fastaをアミノ酸配列に翻$
> names(hoge) <- names(fasta)                 #現状では翻訳した結果のオ$
> fasta <- hoge                               #hogeの中身をfastaに格納
> fasta                                       #確認してるだけです
  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", width=50)#fas$
> objects()
[1] "fasta" "hoge"  "in_f"  "out_f"
> fasta
  A AAStringSet instance of length 1
    width seq                                names
[1]     4 SDGL                               kadota
> hoge
  A AAStringSet instance of length 1
    width seq                                names
[1]     4 SDGL                               kadota
>
```

翻訳配列取得がエラーなく終了すると4つのオブジェクトが利用可能なようです。

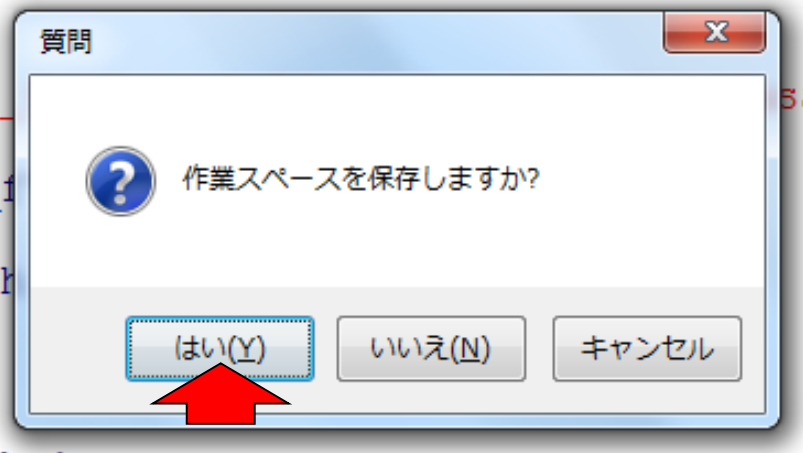
Contents

- 3-3. R 各種パッケージ、2014/09/08 15:00-18:15、中級、実習
  - パッケージ
    - 「(Rで)塩基配列解析」のインストール手順おさらい
    - CRANとBioconductor
    - 代表的なパッケージBiostringsの利用法：library, search, objects関数
    - **作業スペース(workspace)の概念**
    - Biostringsパッケージで利用可能な関数を概観
  - source関数の利用

Rの終了と作業スペース

R Console

```
> names(hoge) <- names(fasta)                    #現状では翻訳した結果のオ$
> fasta <- hoge                                  #hogeの中身をfastaに格納
> fasta                                          #確認してるだけです
  A AAStringSet instance of length 1
    width seq                      names
[1]     4 SDGL                     kadota
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_                                    s$
> objects()
[1] "fasta" "hoge"  "in_f"  "out_f
> fasta
  A AAStringSet instance of length
    width seq
[1]     4 SDGL
> hoge
  A AAStringSet instance of length 1
    width seq                                  names
[1]     4 SDGL                                 kadota
> q()
```

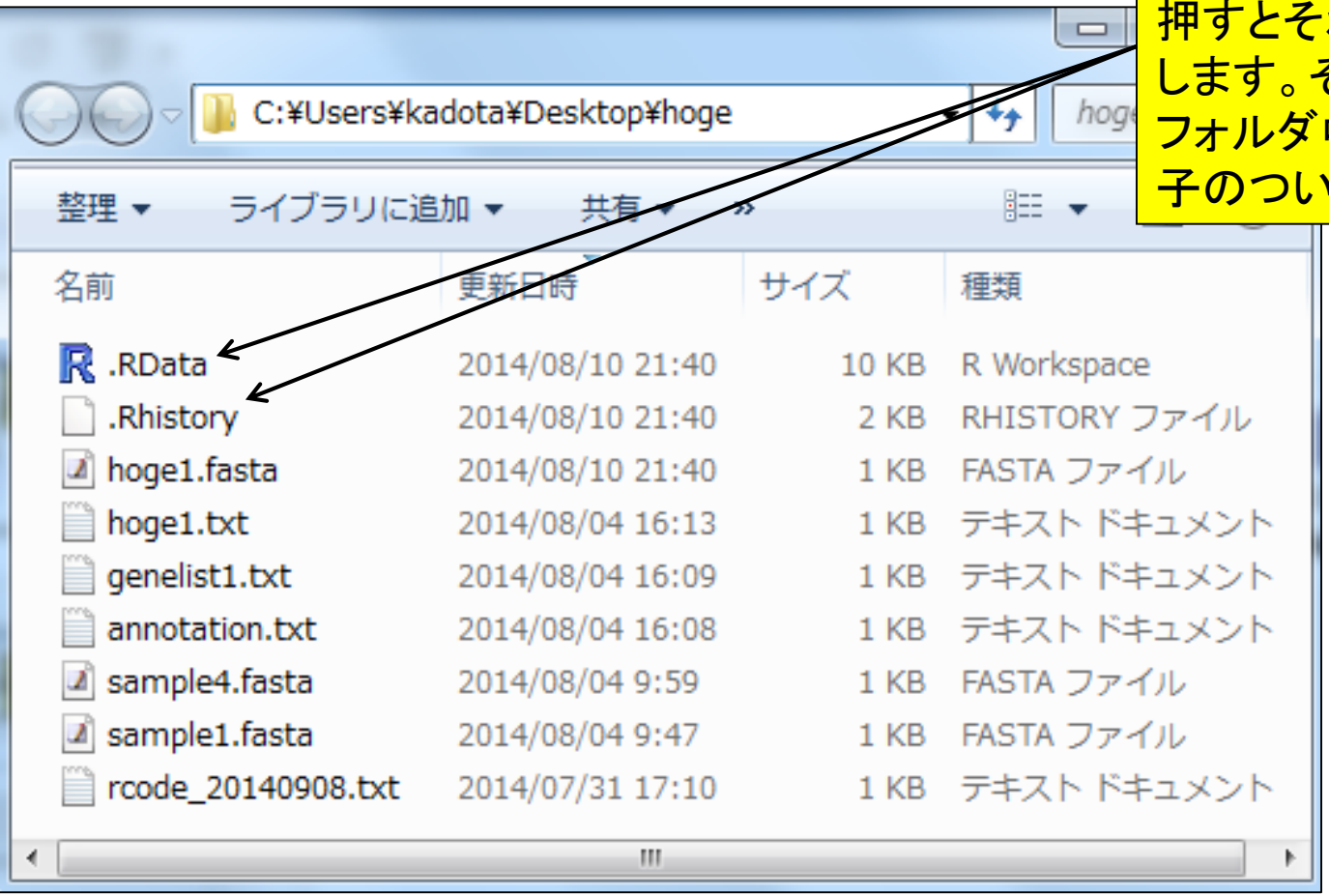


R Gui画面右上の×ボタン以外にq()
でもRを終了させることができます。

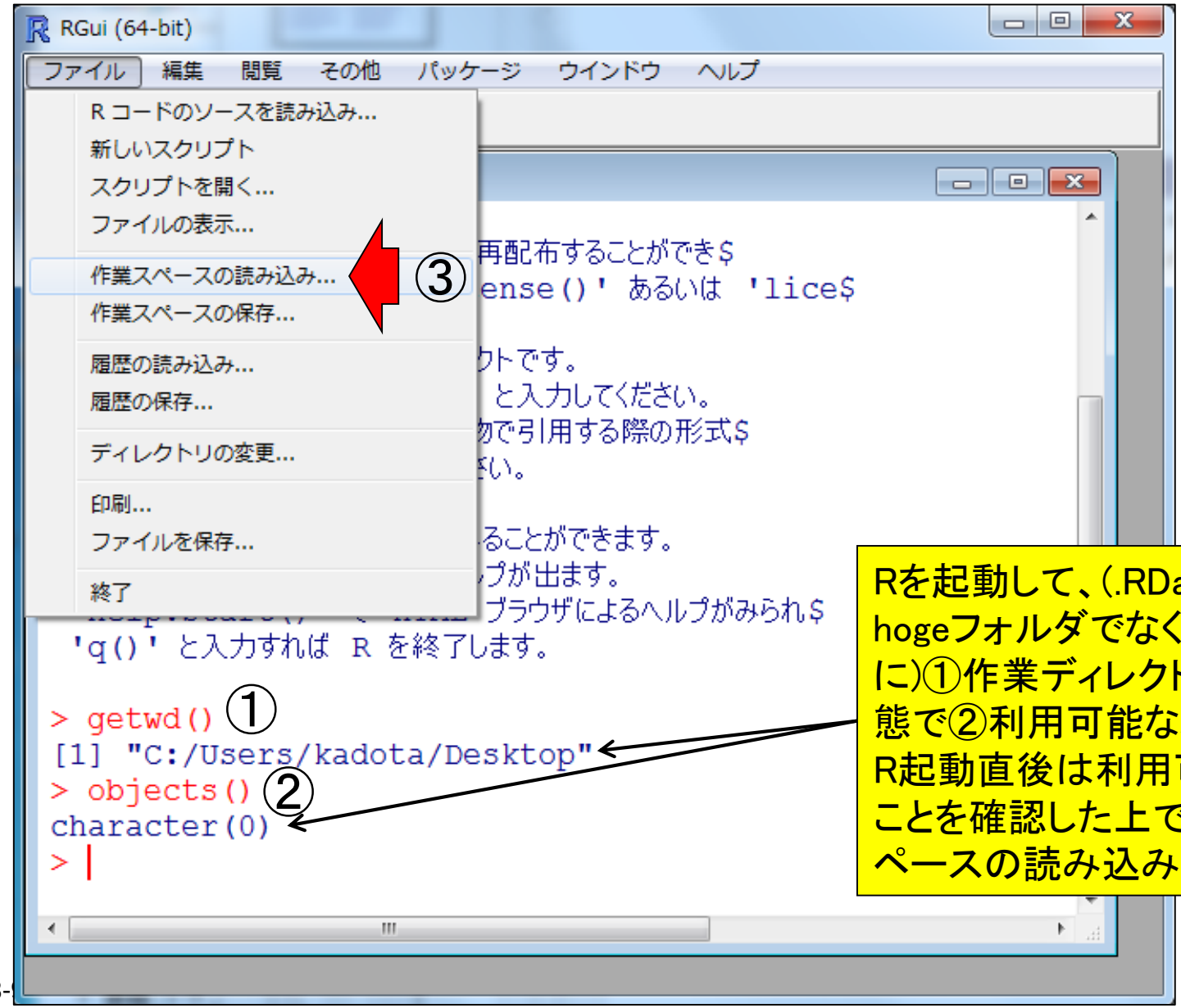
作業スペース(workspace)
保存の意味を説明します

Rの終了と作業スペース

R環境(バージョンやOS)によって若干違うかもしれませんが、私の環境では「はい」を押すとそれ以上何も聞かれることなく終了します。そして、作業ディレクトリだったhogeフォルダ中に.RDataと.Rhistoryという拡張子のついたファイルが自動生成されます。

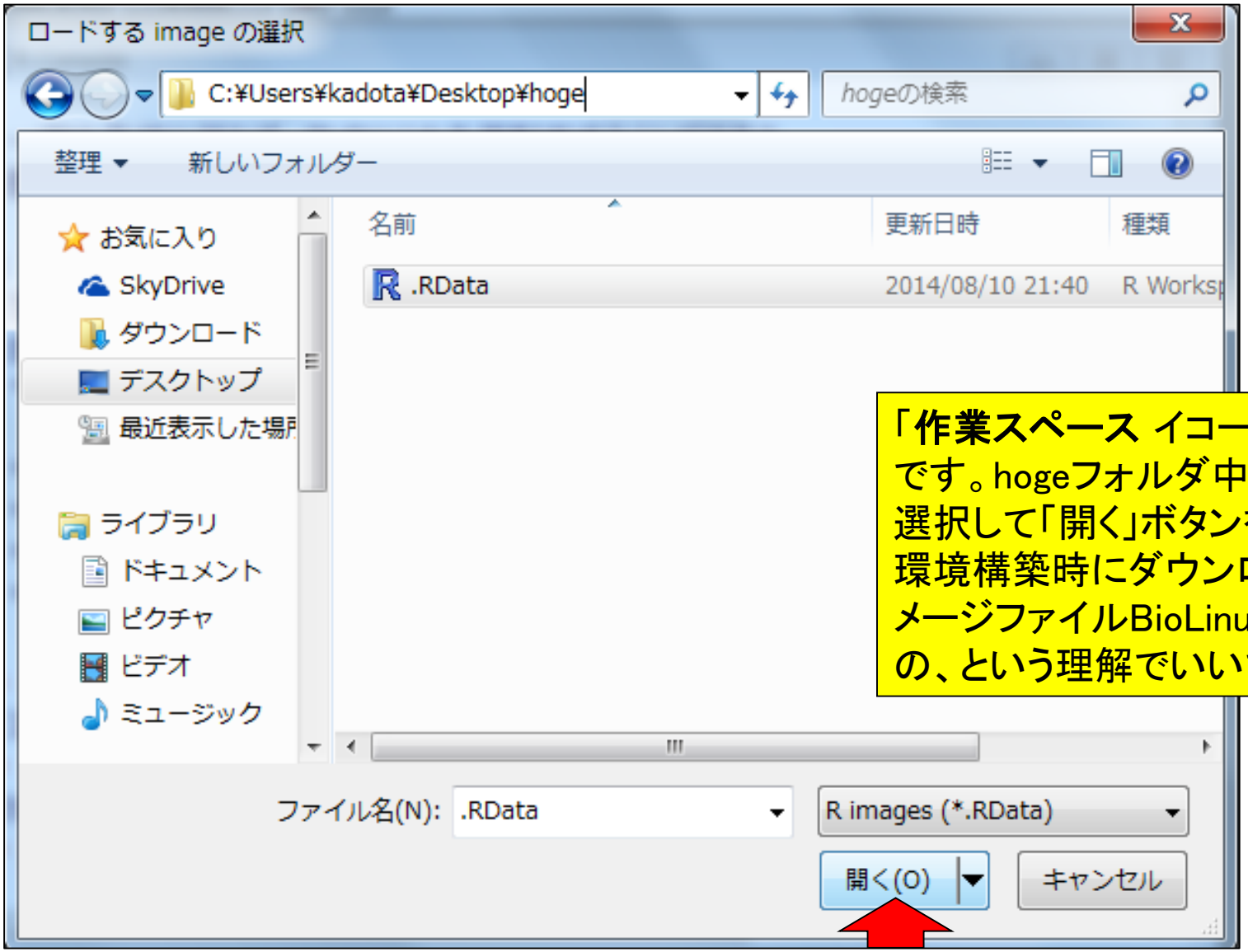


Rの起動と作業スペースの読み込み



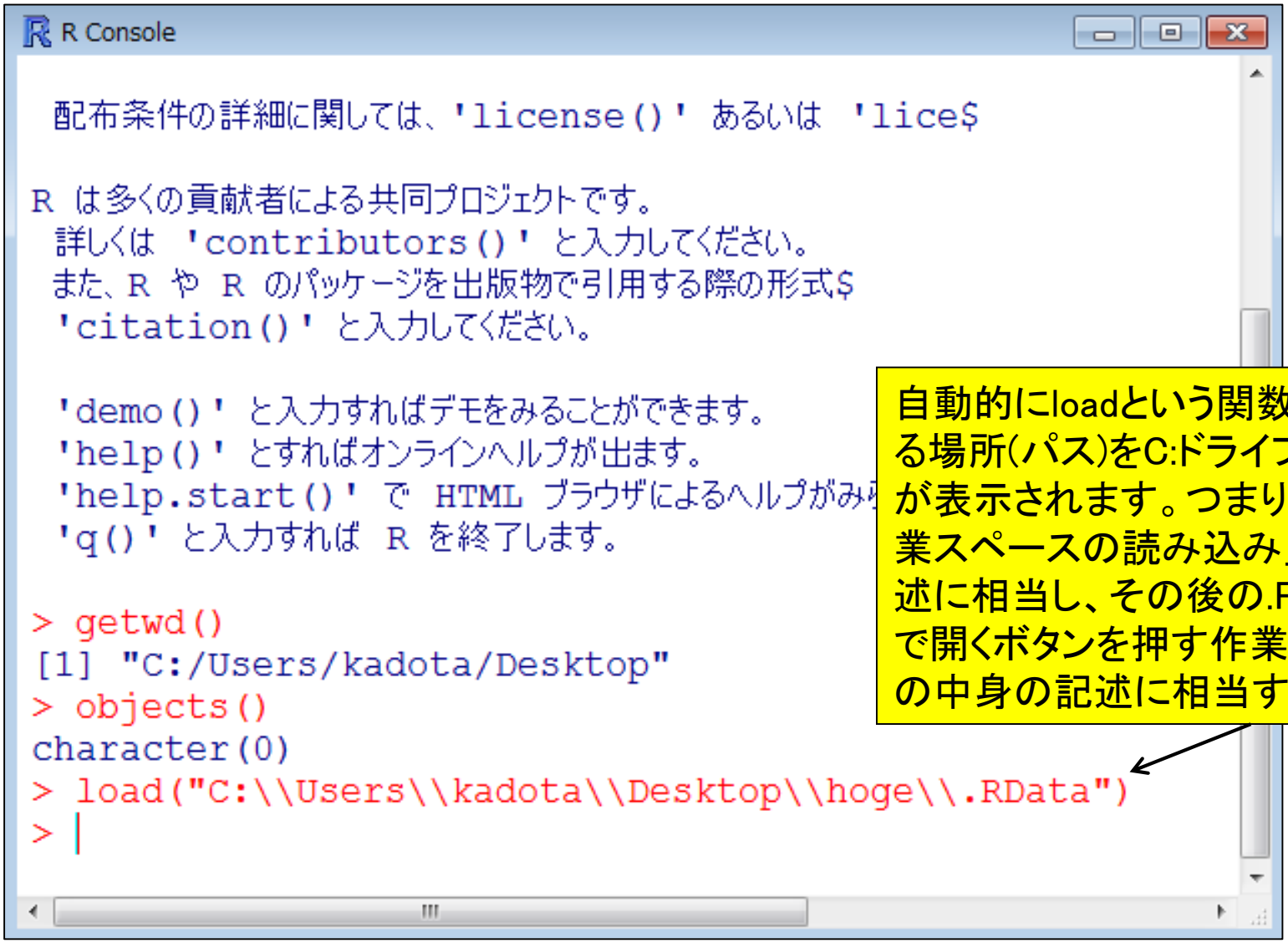
Rを起動して、(.RDataファイルが存在するhogeフォルダでなくてもいいことを示すために)①作業ディレクトリを**デスクトップ**にした状態で②利用可能なオブジェクトを調べます。R起動直後は利用可能なオブジェクトがないことを確認した上で、③「ファイル – 作業スペースの読み込み」を選択します。

Rの起動と作業スペースの読み込み



「作業スペース イコール .RDataファイル」です。hogeフォルダ中の.RDataファイルを選択して「開く」ボタンを押す。これはLinux環境構築時にダウンロードしてもらったイメージファイルBioLinux.ovaと同じようなもの、という理解でいいです。

作業スペースをロードした直後の状態



自動的にloadという関数と.RDataが存在する場所(パス)をC:ドライブから指示したものが表示されます。つまり、「ファイル – 作業スペースの読み込み」がload関数の記述に相当し、その後の.Rdataの場所を選んで開くボタンを押す作業がload関数の括弧の中身の記述に相当する、ということです。

作業スペースロード後の状態を確認

R Console

```
> getwd()
[1] "C:/Users/kadota/Desktop"
> objects()
character(0)
> load("C:\\Users\\kadota\\Desktop\\hoge\\.RData")
> getwd()
[1] "C:/Users/kadota/Desktop"
> objects()
[1] "fasta" "hoge"  "in_f"  "out_f"
> in_f
[1] "sample1.fasta"
> out_f
[1] "hoge1.fasta"
> search()
[1] ".GlobalEnv"        "package:stats"
[3] "package:graphics"  "package:grDevices"
[5] "package:utils"     "package:datasets"
[7] "package:methods"   "Autoloads"
[9] "package:base"
>
```

作業スペースロード後でも、作業ディレクトリ自体は以前行っていたhogeフォルダに変更されるわけではない。

作業スペースをロードする主なご利益は、**以前の作業スペースで利用可能だったオブジェクトが今の作業スペース中でも利用可能になる**点です。得るのに非常に計算時間がかかるオブジェクトを再利用したい場合などに便利。

search()実行結果からわかるように、以前行っていた**library(Biostrings)**は反映されないようです。

作業スペースロード後の状態を確認

```
R Console
> pre <- search()
> pre
[1] ".GlobalEnv"        "package:stats"
[3] "package:graphics"  "package:grDevices"
[5] "package:utils"     "package:datasets"
[7] "package:methods"   "Autoloads"
[9] "package:base"
> fasta
Loading required package: Biostrings
Loading required package: BiocGenerics
Loading required package: parallel

Attaching package: 'BiocGenerics'

  以下のオブジェクトはマスクされています (from 'pac$

      clusterApply, clusterApplyLB,
    clusterCall, clusterEvalQ, clusterExport,
    clusterMap, parApply, parCapply,
    parLapply, parLapplyLB, parRapply,
    parSapply, parSapplyLB

  以下のオブジェクトはマスクされています (from 'pac$
```

```
R Console
      xta

  以下のオブ

      any
     as.v
     dupl
     get,
     mapp
     pmax
     rbind, Reduce, rep.int, rownames, sapply,
     setdiff, sort, table, tapply, union,
     unique, unlist

Loading required package: IRanges
Loading required package: XVector
  A AAStringSet instance of length 1
    width seq                    names
[1]     4 SDGL                   kadota
> fasta
  A AAStringSet instance of length 1
    width seq                    names
[1]     4 SDGL                   kadota
>
```

fastaオブジェクトも以前と同じものが利用可能であることが分かる。fastaオブジェクトはBiostringsパッケージ中の関数を用いて得られたものであり、Biostringsパッケージ中で定義された専用の形式(AAStringSet)で記述されている。それゆえ、fastaの中身を表示する際に、自動的に必要なパッケージをロードしてくれるようだ。

作業スペースロード後の状態を確認

R Console

```
> post <- search()
> post
 [1] ".GlobalEnv"          "package:Biostrings"   "package:XVector"
 [4] "package:IRanges"     "package:BiocGenerics" "package:parallel"
 [7] "package:stats"       "package:graphics"     "package:grDevices"
[10] "package:utils"       "package:datasets"     "package:methods"
[13] "Autoloads"           "package:base"
> pre
[1] ".GlobalEnv"        "package:stats"     "package:graphics"
[4] "package:grDevices" "package:utils"     "package:datasets"
[7] "package:methods"   "Autoloads"         "package:base"
> setdiff(post, pre)
[1] "package:Biostrings"   "package:XVector"      "package:IRanges"
[4] "package:BiocGenerics" "package:parallel"
> setdiff(pre, post)
character(0)
> intersect(pre, post)
[1] ".GlobalEnv"        "package:stats"     "package:graphics"
[4] "package:grDevices" "package:utils"     "package:datasets"
[7] "package:methods"   "Autoloads"         "package:base"
>
```

fastaオブジェクト表示の際に必要なパッケージをロードする前後のパッケージリストをpreおよびpostオブジェクトで保持している。集合演算は、ロード後にどんなパッケージが利用可能になっているかを調べる場合にも便利。setdiff関数は、左側のベクトル中で右側のベクトル中の要素として含まれないものを表示。intersect関数は共通要素を表示。

Contents

- 3-3. R 各種パッケージ、2014/09/08 15:00-18:15、中級、実習
  - パッケージ
    - 「(Rで)塩基配列解析」のインストール手順おさらい
    - CRANとBioconductor
    - 代表的なパッケージBiostringsの利用法：library, search, objects関数
    - 作業スペース(workspace)の概念
    - **Biostringsパッケージで利用可能な関数を概観**
  - source関数の利用

Biostringsで利用可能な関数を概観

- 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descri…n)の…
- イントロ | 一般 | 翻訳配列(translate)を取得 (…mod…
- イントロ | 一般 | 相補鎖(complement)を取得 (…st mo…
- イントロ | 一般 | 逆相補鎖(reverse complement)を取…
- イントロ | 一般 | 逆鎖(reverse)を取得 (last modified 2…
- イントロ | 一般 | 2連続塩基の 出現頻度 情報を取得 (…
- イントロ | 一般 | 3連続塩基の 出現頻度 情報を取得 (…
- イントロ | 一般 | 任意の長さの 連続塩基の 出現頻度…
- イントロ | 一般 | Tips | 任意の 拡張子でファイルを保…
- イントロ | 一般 | Tips | 拡張子は同じで任意の文字を…
- イントロ | 一般 | 配列取得 | ゲノム配列 | 公共DBか…
- イントロ | 一般 | 配列取得 | ゲノム配列 | BSgenome(…
- イントロ | 一般 | 配列取得 | プロモーター配列 | 公共…
- イントロ | 一般 | 配列取得 | プロモーター配列 | BSge…

Biostringsパッケージは、翻訳配列取得以外にも様々な塩基配列解析用の関数を提供しています。

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
```

Biostringsで利用可能な関数を概観

- 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
- イントロ | 一般 | 翻訳配列(translate)を取得 (la od
- イントロ | 一般 | 相補鎖(complement)を取得 (mo
- イントロ | 一般 | 逆相補鎖(reverse complement)を取
- イントロ | 一般 | 逆鎖(reverse)を取得 (last modified 2
- イントロ | 一般 | 2連続塩基の 出現頻度情報を取得
- イントロ | 一般 | 3連続塩基の 出現頻度情報を取得
- イントロ | 一般 | 任意の長さの 連続塩基の出現頻度
- イントロ | 一般 | Tips | 任意の拡張子でファイルを保
- イントロ | 一般 | Tips | 拡張子は同じで任意の文字を
- イントロ | 一般 | 配列取得 | ゲノム配列 | 公共DBか
- イントロ | 一般 | 配列取得 | ゲノム配列 | BSgenome
- イントロ | 一般 | 配列取得 | プロモーター配列 | 公共
- イントロ | 一般 | 配列取得 | プロモーター配列 | BSge

相補鎖を取得する場合も、基本的な枠組みは翻訳配列取得と同じ。主要な関数がtranslateからcomplementに変わっただけです。

イントロ | 一般 | 相補鎖(complement)を取得

FASTA形式ファイルを読み込んで相補鎖を得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
fasta                                      #確認してるだけです

#本番
fasta <- complement(fasta)                 #fastaオブジェクトの相補鎖を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
```

Biostringsで利用可能な関数を概観

- 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 | 逆鎖(reverse)を取得 (last modified 2013/06/14)
- イントロ | 一般 | 2連続塩基の 出現頻度情報を取得 (last modified 2014/07/18) NEW
- イントロ | 一般 | 3連続塩基の 出現頻度情報を取得 (last modified 2013/06/14)
- イントロ | 一般 | 任意の長さの 連続塩基の出現頻度情報を取得 (last modified 2013/06/14)
- 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 2013/09/26)
- 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 modified 2014/04/23)
- 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 2014/04/02)
- 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 modified 2013/09/25)
- 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 2014/06/2

(Rで)塩基配列解析上で示している項目はごく一部です。
Biostringsパッケージが提供している全ての関数を網羅しているわけでもありません。

Biostringsで利用可能な関数を概観

イントロ | 一般 | 相補鎖(complement)を取得

FASTA形式ファイルを読み込んで相補鎖を得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
fasta                                   #確認してるだけです

#本番
fasta <- complement(fasta)              #fastaオブジェクトの相補鎖を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
```

2. multi-FASTA形式ファイル(h_rna.fasta)の場合:

```
in_f <- "h_rna.fasta"                   #入力ファイル名を指定してin_fに格納
out_f <- "hoge2.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
fasta                                   #確認してるだけです

#本番
fasta <- complement(fasta)              #fastaオブジェクトの相補鎖を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
```

◦ Biostrings

「必要なパッケージをロード」のところで、Biostringsを明示的に読み込んでいる場合は、項目の最後のところにBioconductorのBiostringsパッケージのウェブサイトにリンクを張っている。そこを眺めるとBiostringsが提供している関数のリストを概観できます。

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Search:

Home | Install | Help | Developers | About

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:

```
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

Workflows »

Common Bioconductor workflo
include:

- Oligonucleotide Arrays
- High-throughput Sequencin
- 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

(実習用PC環境はインストール済みだが)Biostringsパッケージを個別にインストールしたい場合

下から2番目のReference Manualは、おそらくどのパッケージにも存在する。一番下のNEWSがないパッケージもある。

Biostringsはここが5つ存在するが、パッケージによっては1つだけの場合もある。ここは開発者次第。

Biostringsで利用可能な関数を概観

Biostrings Quick Overview

Hervé Pagès
Fred Hutchinson Cancer Research Center
Seattle, WA

July 4, 2014

Please note that *most* but *not all* the functionalities provided by the Biostrings package are listed in this document.

Function	Description
length	Return the number of sequences in an object.
names	Return the names of the sequences in an object.
[	Extract sequences from an object.
head, tail	Extract the first or last sequences from an object.
rev	Re
c	Co
width, nchar	Re ob
==, !=	El
match, %in%	Ar
duplicated, unique	Ar
sort, order	Ar or lo
relist, split, extractList	Ar th tr ar

Table 1: **Low-level manipulatio**

DNA配列を入力としてアミノ酸配列を出力することが分かっている**translate**関数を眺めるとFunction列とDescription列の意味がよくわかる。

Function	Description
reverse complement reverseComplement	Compute the reverse, complement, or reverse-complement, of a set of DNA sequences.
translate	Translate a set of DNA sequences into a set of Amino Acid sequences.
chartr	Translate the letters in a set of sequences.
subseq, subseq<- extractAt, replaceAt	Extract/replace arbitrary substrings from/in a string or set of strings.
replaceLetterAt	Replace the letters specified by a set of positions by new letters.
padAndClip, stackStrings	Pad and clip strings.
unstrsplit	A fast implementation of `sapply(x, paste0, collapse=sep)` for collapsing the list elements of a *DNAStringSetList* or *AAStringSetList* object.

Table 3: **Sequence transformation and editing.**

Contents

- 3-3. R 各種パッケージ、2014/09/08 15:00-18:15、中級、実習
  - パッケージ
    - 「(Rで)塩基配列解析」のインストール手順おさらい
    - CRANとBioconductor
    - 代表的なパッケージBiostringsの利用法：library, search, objects関数
    - 作業スペース(workspace)の概念
    - Biostringsパッケージで利用可能な関数を概観
  - **source関数の利用**

source関数を利用

RでもLinuxのシェルスクリプトっぽいことができます。基本手順は①コピペする一連のコードからなるファイル(rcode_translate.txt)を入力ファイル(sample1.fasta)と同じフォルダ上に保存し、②作業ディレクトリの変更を行って、③source("rcode_translate.txt")を実行。

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

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
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

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

#本番
hoge <- translate(fasta)                 #fastaをア
names(hoge) <- names(fasta)              #現状では翻
fasta <- hoge                            #hogeの中
fasta                                    #確認してる

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

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
```

source関数を利用

①rcode_translate.txtと入力ファイルsample1.fastaをデスクトップ上のhoge2フォルダ中に保存し、②Rを起動して作業ディレクトリの変更を行って、③source("rcode_translate.txt")を実行した結果。Source関数実行後に出力ファイルhoge1.fastaが生成されていることがわかる

R Console

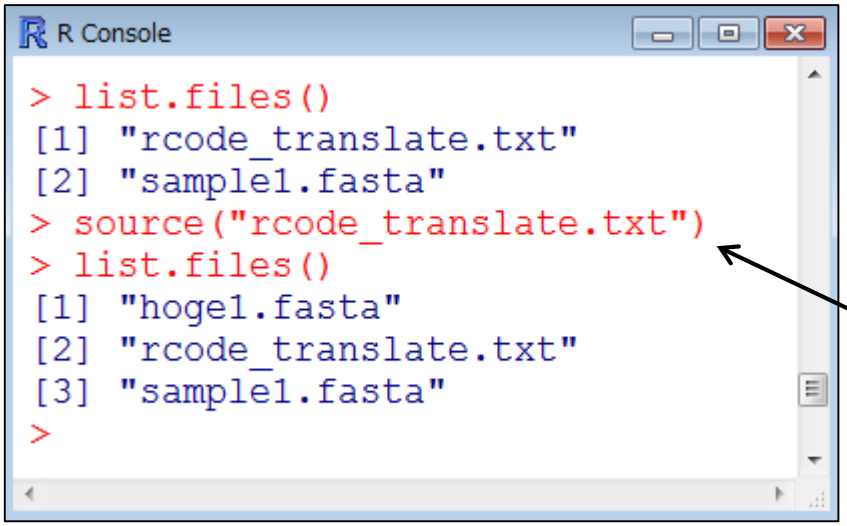
```
> getwd()
[1] "C:/Users/kadota/Desktop/hoge2"
> list.files()
[1] "rcode_translate.txt" "sample1.fasta"
> source("rcode_translate.txt")
 要求されたパッケージ BiocGenerics をロード中です
 要求されたパッケージ parallel をロード中です

 次のパッケージを付け加えます: 'BiocGenerics'

 以下のオブジェクトはマスクされています (from $

     clusterApply, clusterApplyLB,
    clusterCall, clusterEvalQ,
    clusterExport, clusterMap, parApply,
    parCapply, parLapply, parLapplyLB,
    parRapply, parSapply, parSapplyLB

 以下のオブジェクトはマスクされています (from $

     xtabs

 以下のオブジェクトはマスクされています (from $
```

R Cons

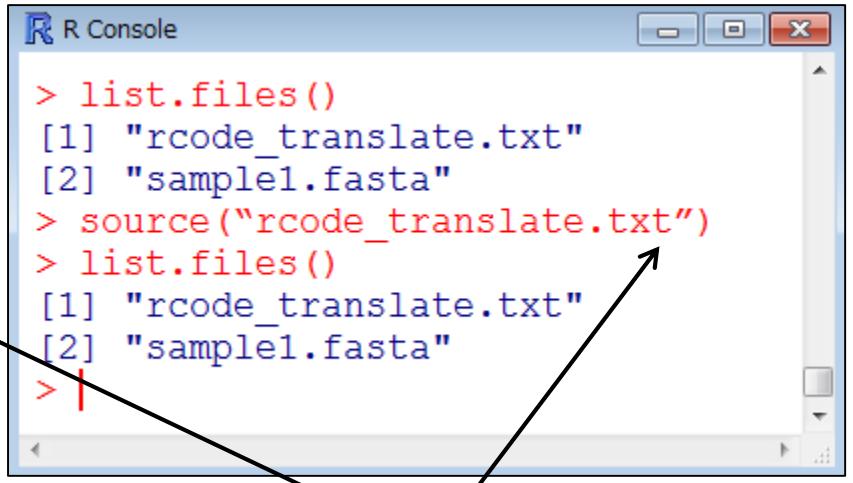
```
 以下の

 以下のオブジェクトはマスクされています (from $

     anyDuplicated, append, as.data.frame,
    as.vector, cbind, colnames, do.call,
    duplicated, eval, evalq, Filter, Find,
    get, intersect, is.unsorted, lapply,
    Map, mapply, match, mget, order,
    paste, pmax, pmax.int, pmin, pmin.int,
    Position, rank, rbind, Reduce,
    rep.int, rownames, sapply, setdiff,
    sort, table, tapply, union, unique,
    unlist

 要求されたパッケージ IRanges をロード中です
 要求されたパッケージ XVector をロード中です
> list.files()
[1] "hoge1.fasta"          "rcode_translate.txt"
[3] "sample1.fasta"
>
```

source関数を利用

うまくいく例



うまくいかない例



二重クォーテーションに注意!!
「全角」ではなく「半角」です。丸括弧もそうです。エラーメッセージが出ないので、こういうこともあるという経験をしておかないと対処法が分からず苦悩します。

source関数を利用

- Biostringsのようにパッケージを作成して提供する以外に、自作のR関数をウェブ上で公開しているヒトもいる。
  - http://www.iu.a.u-tokyo.ac.jp/~kadota/R/R_functions.R
    - 組織特異的発現パターン検出法ROKU (Kadota et al., *BMC Bioinformatics*, 2006)を実行するためのkadota_2006_bmc_bioinformatics関数
    - RNA-seq正規化法やシミュレーションデータ作成関数(Kadota et al., *AMB*, 2012)
    - etc…

R Console

```
> source("http://www.iu.a.u-tokyo.ac.jp/~kadota/R/R_functions.R")
> kadota_2006_bmc_bioinformatics
function(x){
    x <- x[(!is.na(x))]                          #NA以外の要素をベクトル$
    x_length <- length(x)                        #ベクトルxの要素数をx_l$
    x <- x[(x != 0)]                             #要素が0でないものをxに$
    if(sum(x) <= 0) {                            #xの和が0以下の場合は
        return(log(x_length, base=2))            #最大値を与える
    } else if(sd(x) == 0){                       #xの要素が全て0の場合も$
        return(log(x_length, base=2))            #最大値を与える
    } else {                                     #それ以外の場合
        x_prime <- abs(x-tukey.biweight(x))#Tukey's biweightで引い$
        p <- x_prime/sum(x_prime)                #ベクトルpの和が1になる$
        e <- sum(p * log(p, base=2))
        return(-e)
    }
}
> |
```

自分で関数を作成することもできる。TCCパッケージ ver. 1.4 (Sun et al., 2013)で別の関数を実装済みで、現在は不使用。

(Rで)マイクロアレイデータ解析

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

- 解析 | 発現変動 | 多群間 | について(last modified 2013/6/2)
- 解析 | 発現変動 | 多群間 | SpeCond(Cavalli_2011) (last mod
- 解析 | 発現変動 | 多群間 | ROKU(Kadota_2006) (modifi
- 解析 | 発現変動 | 多群間 | Sprent's non-parametric method(G
- 解析 | 発現変動 | 多群間 | Schug's H(x) statistic(Schug_2005
- 解析 | 発現変動 | 多群間 | Schug's Q statistic(Schug_2005) (l
- 解析 | 発現変動 | 多群間 | Ueda's AIC-based method(Kadota
- 解析 | 発現変動 | 多群間 | パターンマッチング法(テンプレー
- 解析 | 発現変動 | 時系列 | について(last modified 2013/6/10)
- 解析 | 発現変動 | 時系列 | Periodic genes | M-estimator (Ahd
- 解析 | 発現変動 | 時系列 | Periodic genes | Lomb-Scargle per
- 解析 | 発現変動 | 時系列 | non-periodic genes | randomized B
- 解析 | 発現変動 | 時系列 | non-periodic genes | IPCA (Yao_2
- 解析 | 発現変動 | 時系列 | non-periodic genes | gptk (Kalaitzi
- 解析 | 発現変動 | 時系列 | non-periodic genes | betr (Aryee_2
- 解析 | 発現変動 | 時系列 | non-periodic genes | maSigPro (Co
- 解析 | 発現変動 | 時系列 | non-periodic genes | SAM (Tusher
- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | について(la
- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | GAGE (Luc
- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | GSA (Efror

1. TCCパッケージ中のROKU関数を用いるやり方
2. source関数で読み込んでkadota_2006_bmc_bioinformatics関数を用いるやり方

解析 | 発現変動 | 多群間 | ROKU (Kadota_2006)

TCCパッケージで提供しているROKU法(Kadota et al., 2006)を用いて、遺伝子発現行列中の遺伝子を全体的な組織特異性の度合いでランキングします。出力ファイル中の"modH"列の数値は、「ROKU論文中のAdditional file 1(Suppl.xls)の"H(x')"列の数値」と対応しています。つまり、データ変換後のエントロピー値です。"ranking"列は、modHの値でランキングした結果です。"ranking"列で昇順にソートすることで、全体的な組織特異性の度合いでランキングしていることになります。つまり、上位が「(どの組織で特異的かはこのスコアだけでは分からないが)組織特異性が高い遺伝子」ということになります。残りの結果は「1:特異的高発現、-1:特異的低発現、0:その他」からなる「外れ値行列」です。例えば、組織AとBで1, それ以外の組織で0を示す遺伝子(群)は「AとB特異的高発現遺伝子」と判断します。
「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. サンプルデータ21のsample21.txtの場合:

log2変換後のデータであるという前提です。

```
in_f <- "sample21.txt"                  #入力ファイル名を指定してin_fに格納
out_f <- "hoge1.txt"                    #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(TCC)                            #パッケージの読み込み

#入力ファイルの読み込み
data <- read.table(in_f, header=TRUE, row.names=1, sep="\t", quote="")#in_fで指定したファイルの読み込み

#本番
hoge <- ROKU(data)                      #ROKUを実行した結果をhogeに格納
outlier <- hoge$outlier                 #外れ値行列をoutlierに格納
modH <- hoge$modH                       #データ変換後のエントロピー値をmodHに格納(原著論文のH(x')の値に相当)
ranking <- hoge$rank                    #modHでランキングした結果をrankingに格納

#ファイルに保存
tmp <- cbind(rownames(data), outlier, modH, ranking)#左端の列が遺伝子ID, 次にサンプル数分だけの列からなる「
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定したファイル名で保存
```

2. サンプルデータ5のsample5.txtの場合:

おまけのところで、「心臓特異的発現パターン」を示す遺伝子群を抽出するための"理想的なパターン(テンプレート)"を含むファイル(GDS1096_cl_heart.txt)を読み込んでいます。

```
in_f <- "sample5.txt"                   #入力ファイル名を指定してin_fに格納
out_f <- "hoge2.txt"                    #出力ファイル名を指定してout_fに格納

#必要な関数などをロード
source("http://www.iu.a.u-tokyo.ac.jp/~kadota/R/R_functions.R")#ROKUを実行するkadota_2006_bmc_bioinfor
library(affy)                           #Tukey's Biweightを計算するためのtukey.biweight関数が含まれている
library(som)                            #各遺伝子発現ベクトルを正規化（平均=0, 標準偏差=1）するためのnormal

#入力ファイルの読み込み
data <- read.table(in_f, header=TRUE, row.names=1, sep="\t", quote="")#in_fで指定したファイルの読み込み

#本番
data.z <- normalize(data, byrow=TRUE)  #正規化を実行し、結果をdata.zに格納
out <- t(apply(data.z, 1, kadota_2003_physiol_genomics_0.25))#各遺伝子発現ベクトルについて、Ueda's AIC-b
colnames(out) <- colnames(data)         #列名を付与
entropy_score <- apply(data, 1, kadota_2006_bmc_bioinformatics)#一行（一遺伝子）づつ、遺伝子発現ベクトル

#ファイルに保存
tmp <- cbind(rownames(data), out, entropy_score)#左端の列が遺伝子ID, 次にサンプル数分だけの列からなる「外
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定したファイル名で保存

#以下は（こんなこともできますという）おまけ
```

source関数を利用

http://www.bioconductor.org/biocLite.R をsource関数で読み込むことで、ネット経由で個別のパッケージのインストールを行うためのbiocLite関数を利用できる。

(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 2014/08/08, since 2010)

What's new?

- このウェブページはフリーソフトRと必要な
  1. Rのインストールと起動および2. 基本的
- 2014年10
- 門田幸二 著シリーズ Useful R 第7巻トラ

- 日本乳酸菌学会誌のNGS関連連載の第1
- 参考資料(講義、講習会、本など)の項目を

- はじめに (last modified 2014/01/30)
- 参考資料(講義、講習会、本など) (last mo
- 過去のお知らせ (last modified 2014/08/03)
- Rのインストールと起動 (last modified 201
- 基本的な利用法 (last modified 2014/07/20
- サンプルデータ (last modified 2014/07/17)
- バイオインフォマティクス人材育成カリキュ
- 書籍|トランスクリプトームについて (last m

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。2014年7月31日にアップデートしたWindows用のインストール手順はこちら。2014年5月14日にアップデートしたMac版のインストール手順こちら(by 孫建強氏)もあります。注意点は、「Mac OS X のバージョンに関わらず R-3.1.0-snowleopard.pkg をインストールしたほうがよい」です。

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させる
3. **Windows Vistaの人**は(パッケージのインストール中に書き込み権限に関するエラーが出るのを避けるために)「コントロールパネル」-「ユーザーアカウント」-「ユーザーアカウント制御の有効化または無効化」で、「ユーザーアカウント制御(UAC)を使ってコンピュータの保護に役立たせる」の**チェックをあらかじめ外しておくことを強くお勧め**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* (32 bitの場合; *X*や*Y*中の数値はバージョンによって異なります)」または「R x64 3.*X*.*Y* (64 bitの場合)」アイコンをダブルクリックして起動
5. 以下を、「Rコンソール画面上」でコピー&ペーストする。10GB程度のディスク容量を要しますが一番お手軽です。(どこからダウンロードするか？と聞かれるので、その場合は自分のいる場所から近いサイトを指定)

```
install.packages(available.packages()[,1], dependencies=TRUE)#CRAN中にある全てのパッケ
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
```

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。