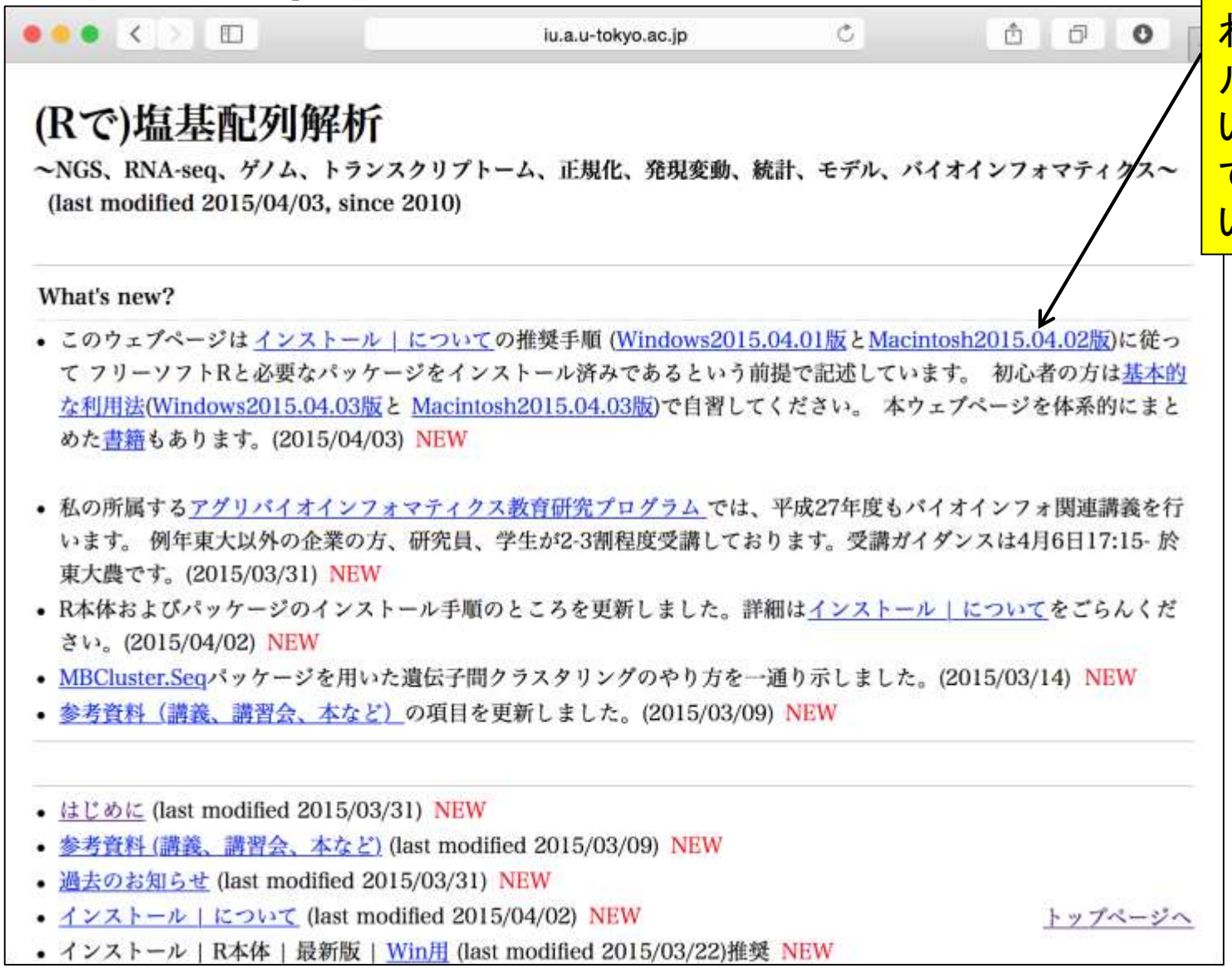
到達目標:このスライドに書かれている程度のことは自在にできるようにしてエラーへの対処法を身につける。
1. 必要なパッケージのインストールが正しくできているかどうかの自力での判定、および個別のパッケージのインストール
2. 作業ディレクトリの変更
3. テキストエディタで自在に入出力ファイル名の変更(どんなファイル名のものがどこに生成されるかという全体像の把握)
4.「ありがちなミス」のところで示しているエラーメッセージとその原因をきっちり理解

(Rで)塩基配列解析 基本的な利用法 Macintosh版

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プログラム
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/

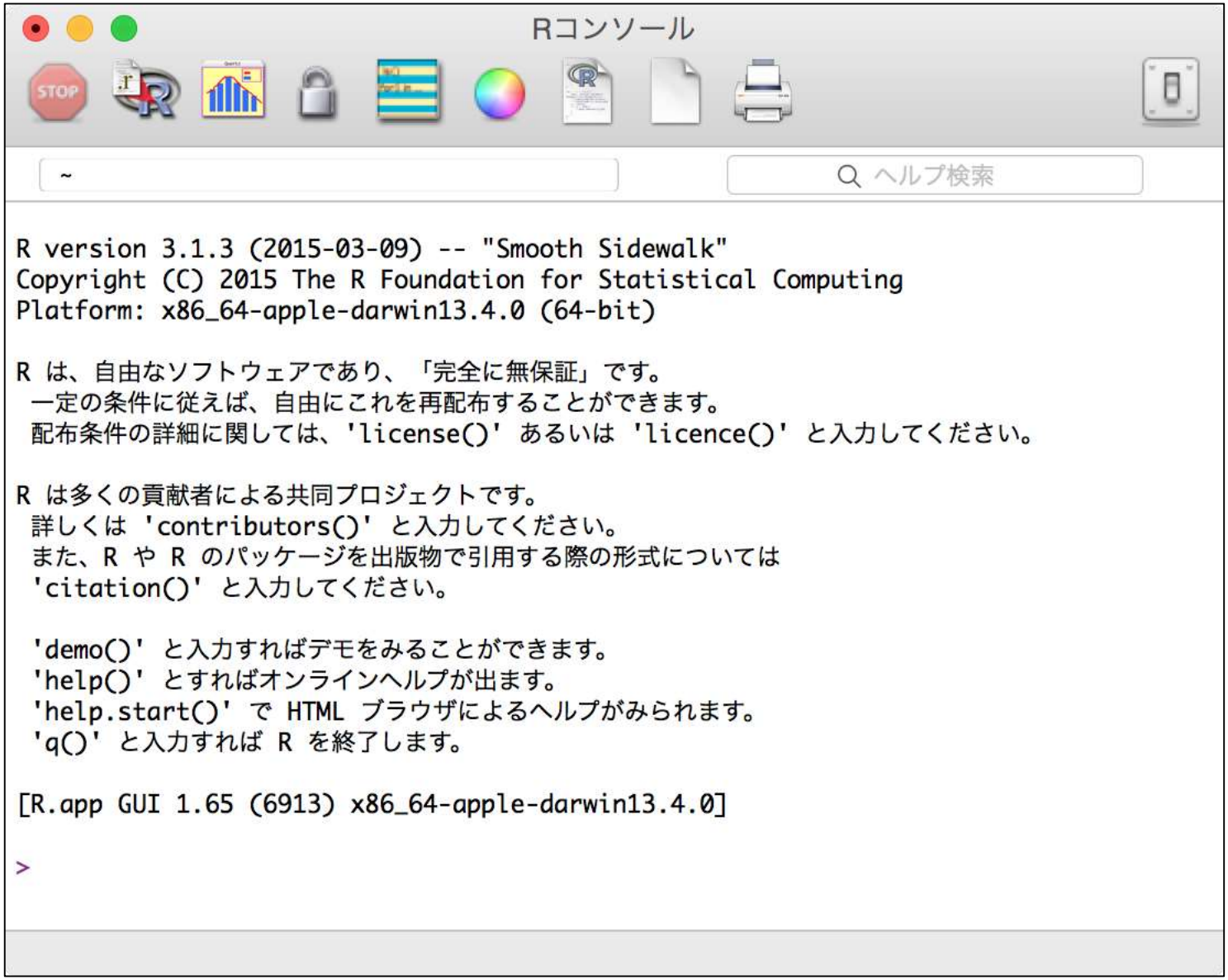
前提条件



インストールができているつもりでも、実際にはできていなかったという事例が散見されます。パッケージ名のスペルミスもよく見受けられます。いくつかのパッケージについて適切にインストールされているか確認しておきましょう。

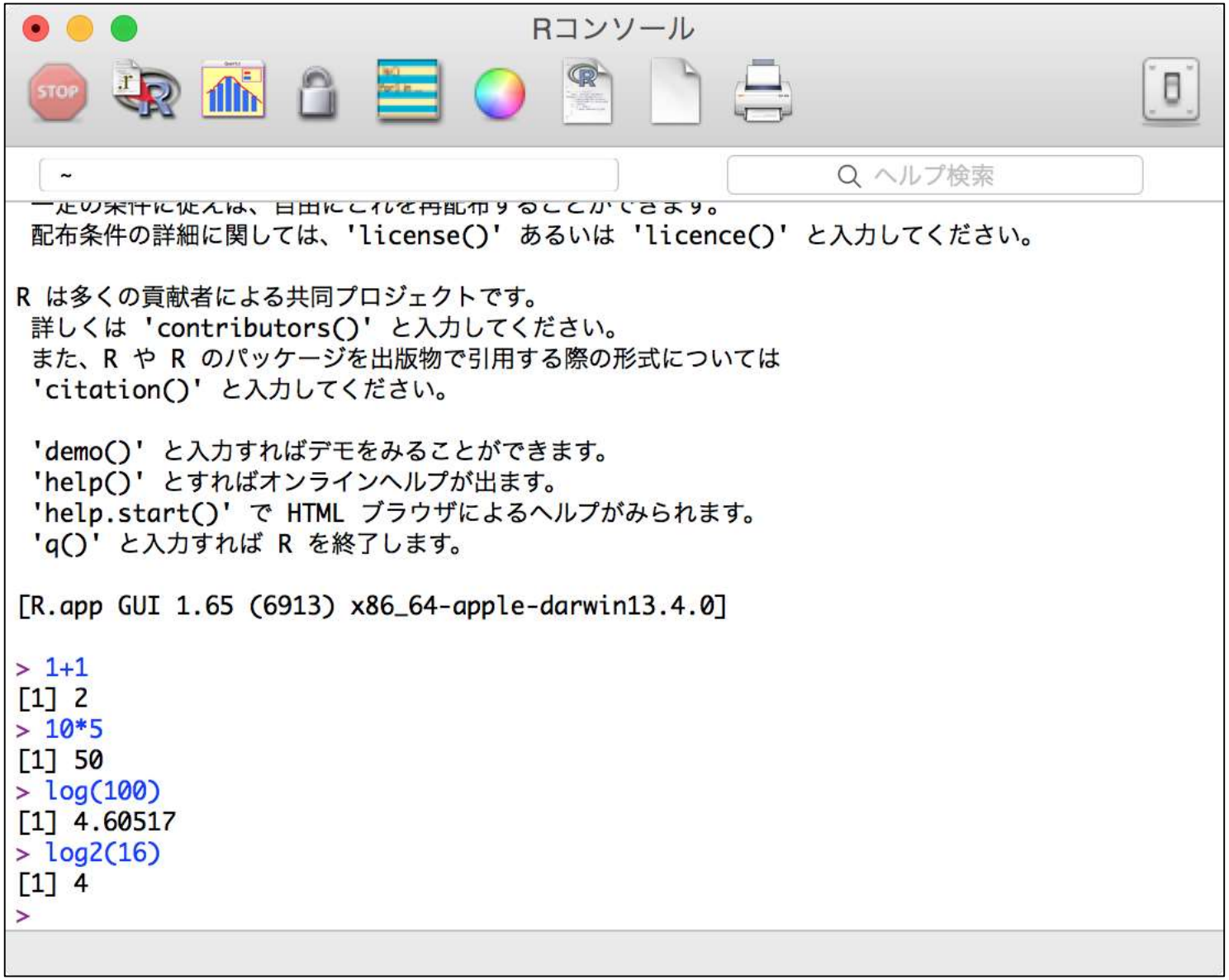
起動直後の状態です

Rの起動



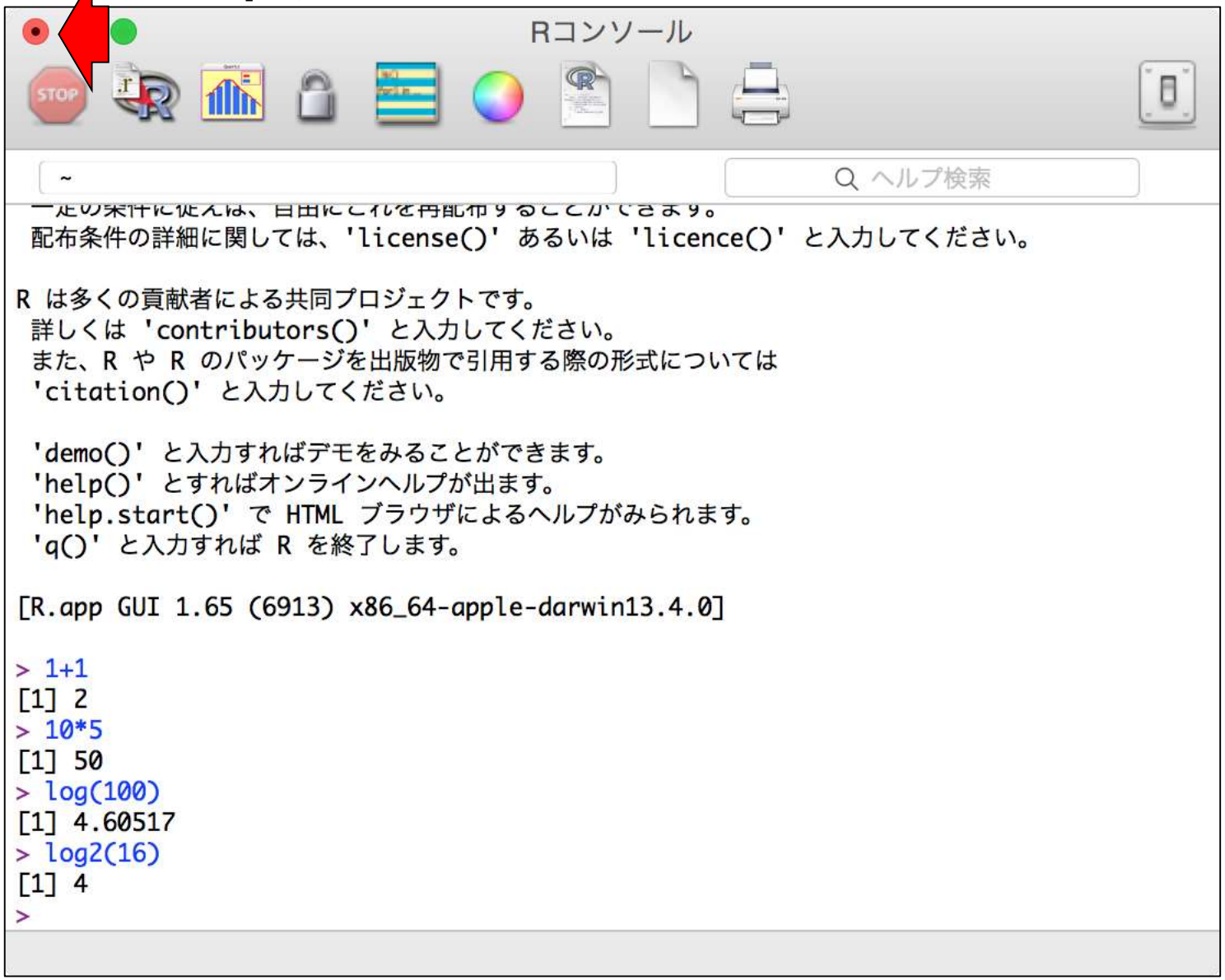
数値計算ができます

基本的な利用法

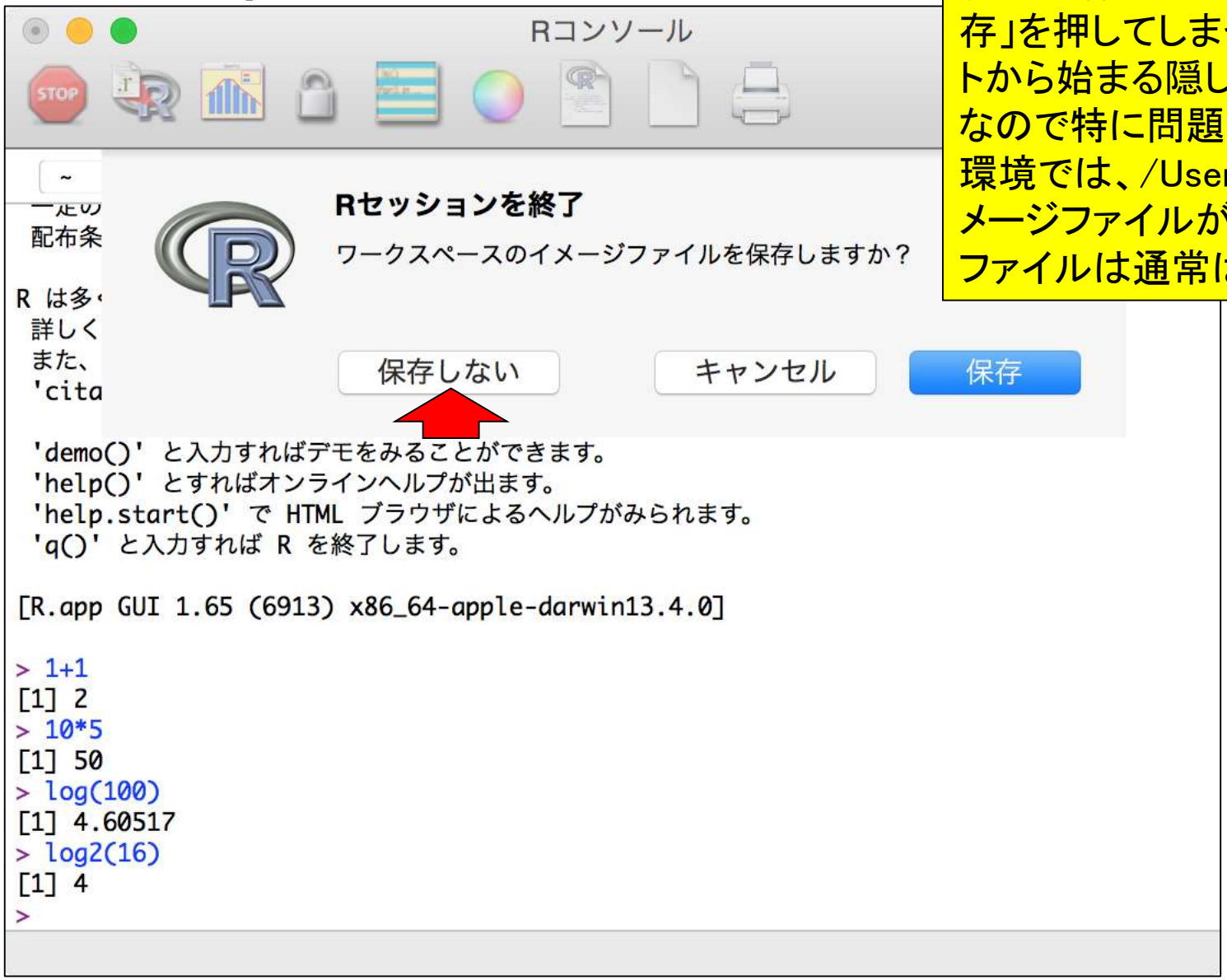


Rの終了

通常のソフトウェアと同様、左上の赤丸ボタンを押せばよい。



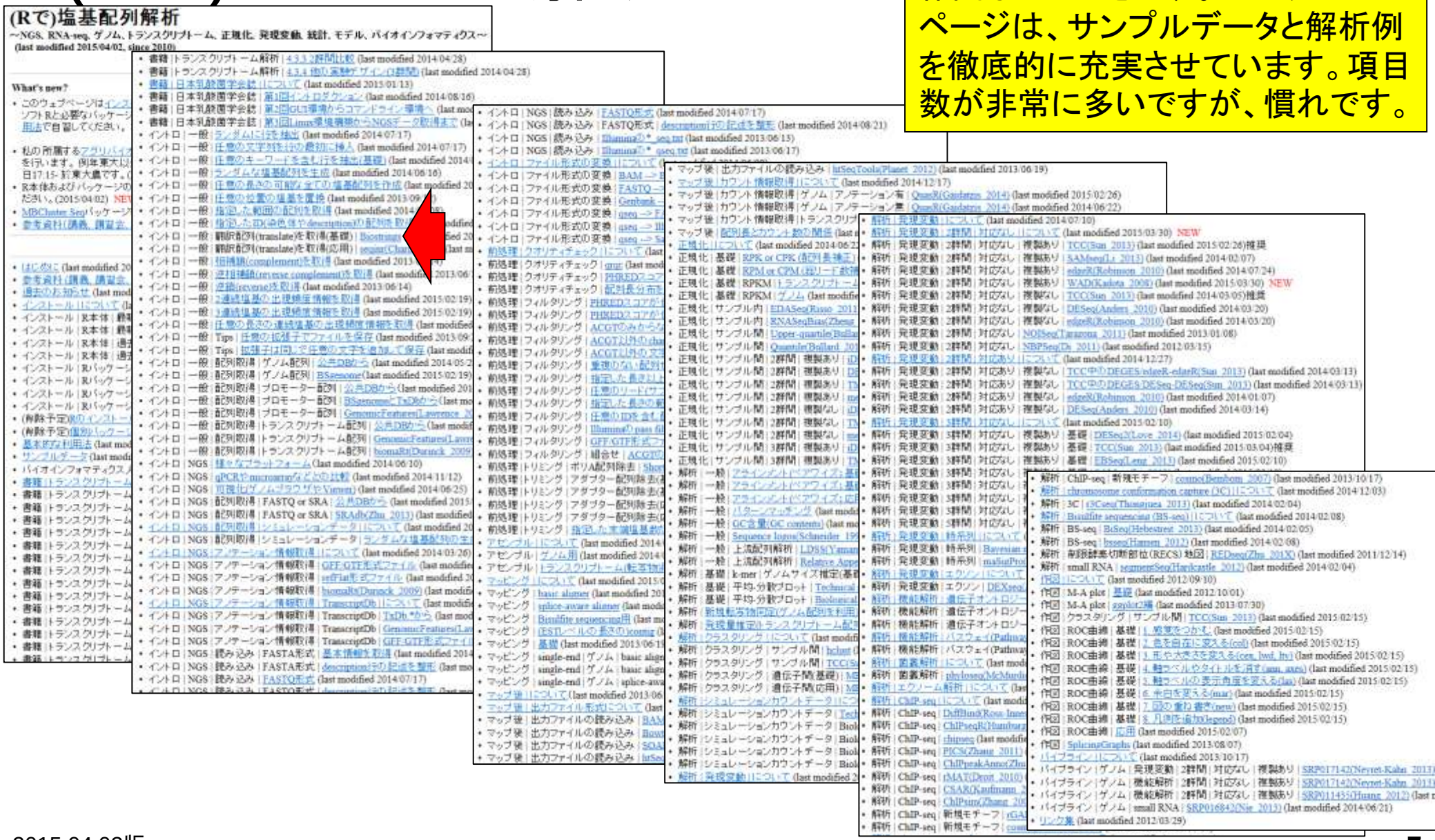
Rの終了



「ワークスペースのイメージファイルを保存しますか?」というダイアログが出るが、最初のうちは「保存しない」でよい。(間違って「保存」を押してしまっても.Rapp.historyというドットから始まる隠しファイルが作成されるだけなので特に問題はない。ユーザ名kadotaの環境では、/Users/kadota/.Rapp.historyにイメージファイルが自動作成される。尚、隠しファイルは通常は表示されない。

(Rで)塩基配列解析

基本的な塩基配列解析から、NGSデータ取得、マッピング、統計解析、作図などができます。このウェブページは、サンプルデータと解析例を徹底的に充実させています。項目数が非常に多いですが、慣れです。



解析基礎1：翻訳配列取得

塩基配列を入力として、その翻訳されたアミノ酸配列を取得することができます

イントロ | 一般 | 翻訳配列(translate)を取得(基礎) | Biostrings NEW

Biostringsパッケージを用いて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。翻訳のための遺伝コード(genetic code)は、Standard Genetic Codeだそうです。もちろん生物種?!によって多少違い(variants)があるようで、"Standard", "SGC0", "Vertebrate Mitochondrial", "SGC1"などいろいろ選べるようです。
「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。

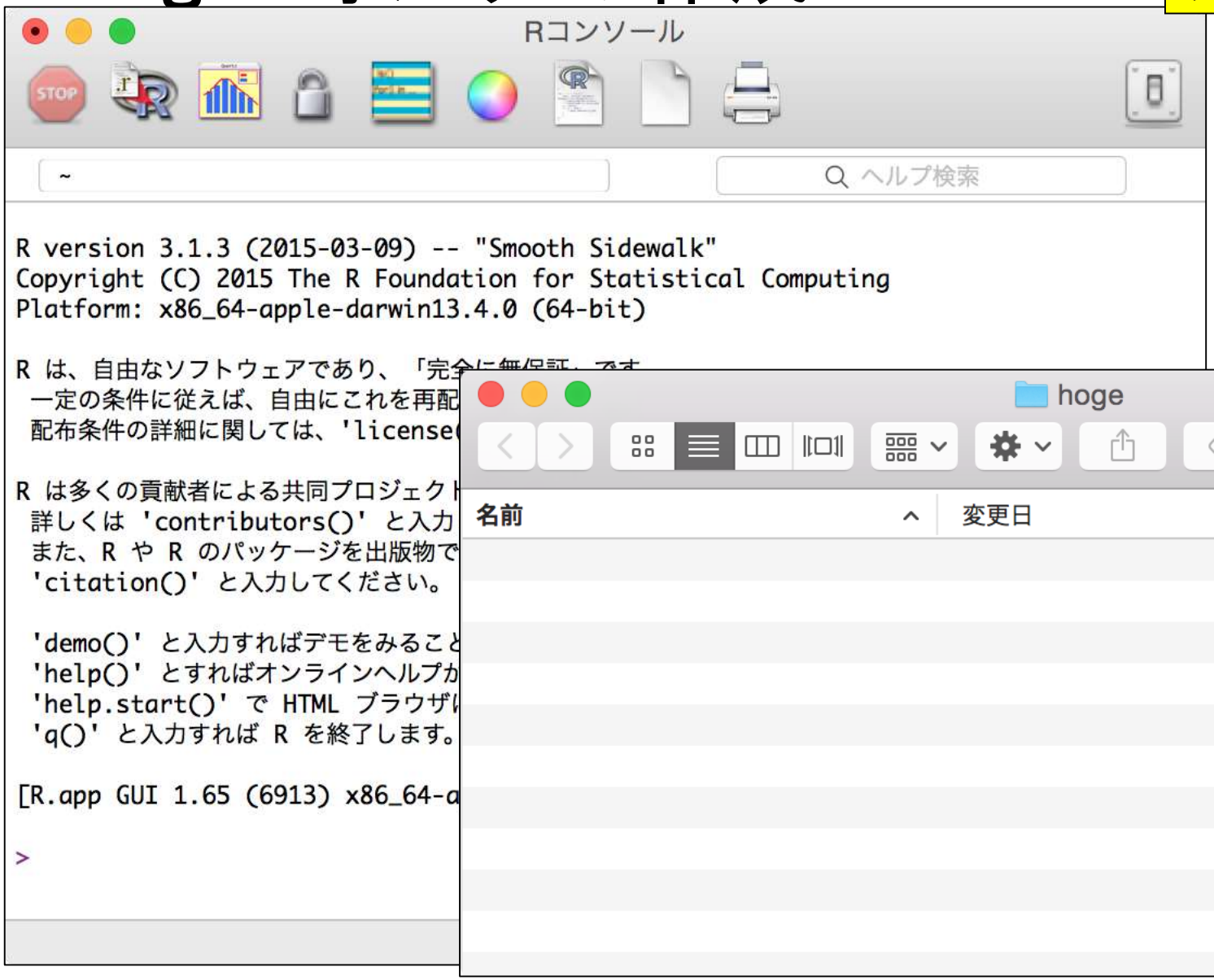
1. FASTA形式ファイル(sample1.fasta)の場合:

multi-FASTAではないsingle-FASTA形式ファイルです。

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

hogeフォルダの作成

デスクトップにあるhogeフォルダ中のファイルを解析するやり方として説明します



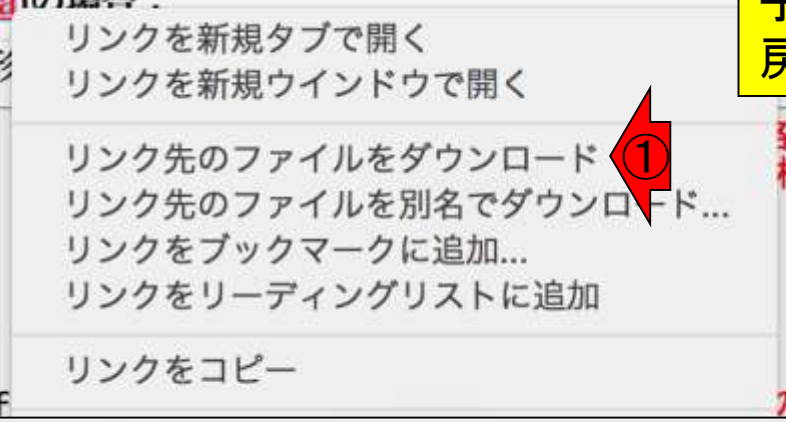
ファイルの保存

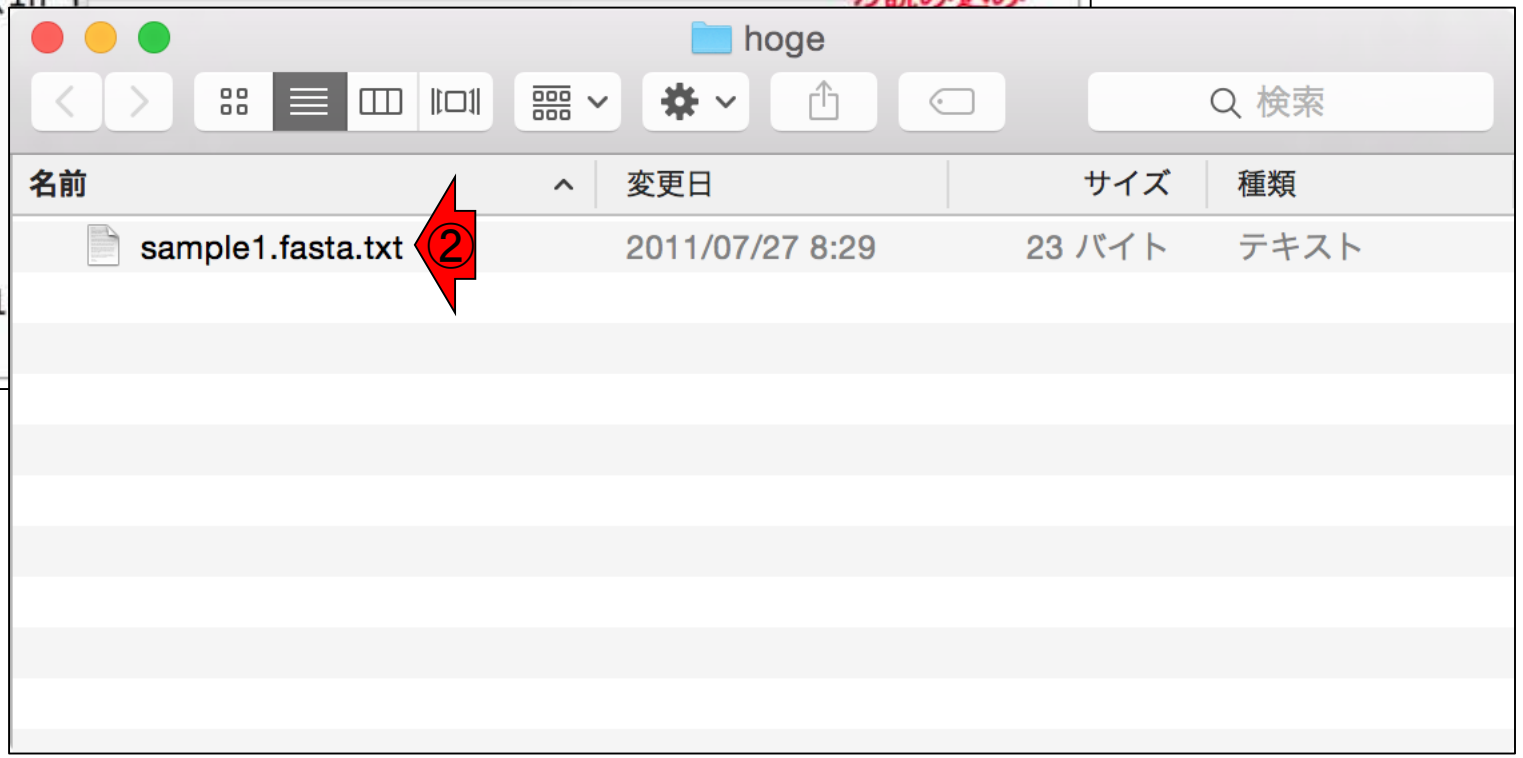
①解析したいファイルsample1.fastaをhogeフォルダ中に保存。②勝手に.txtという拡張子が追加されてしまいますので、戻しておきましょう。

1. FASTA形式ファイル(sample1.fasta)の場合：

multi-FASTAではないsingle-FASTA形

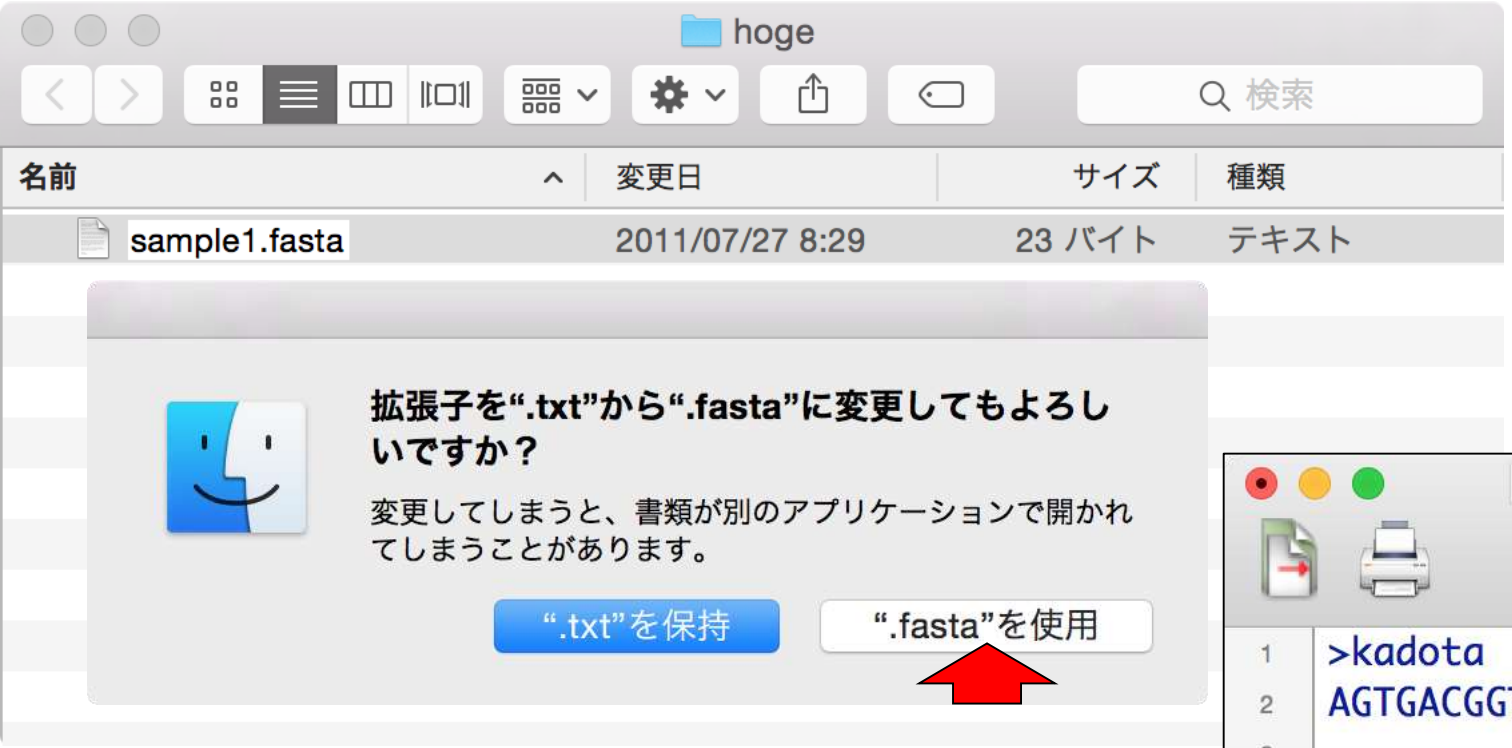
```
in_f <- "sample1.fasta"
out_f <- "hoge1.fasta"

#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
fasta

#本番
fasta <- translate(fasta)
fasta

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

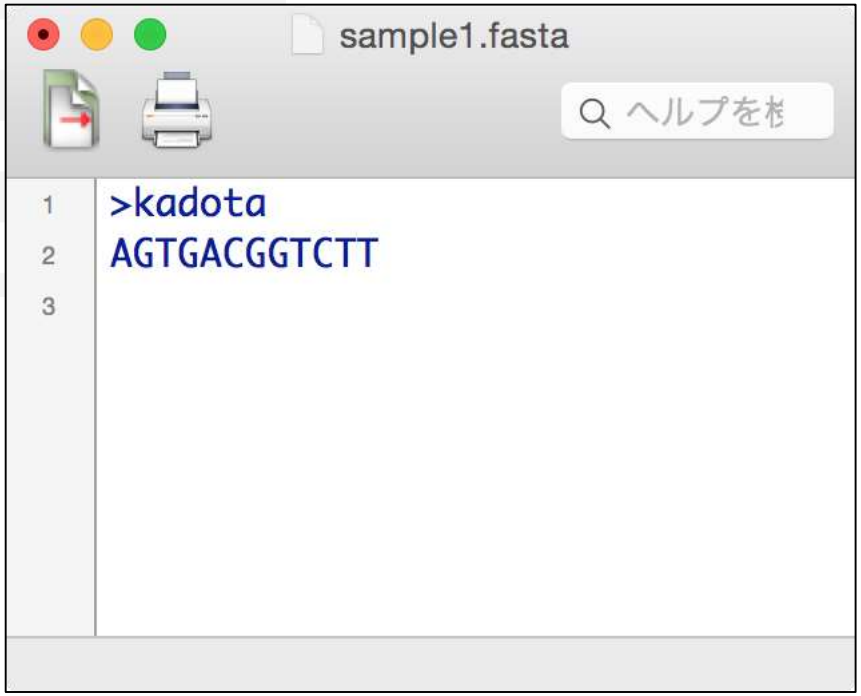




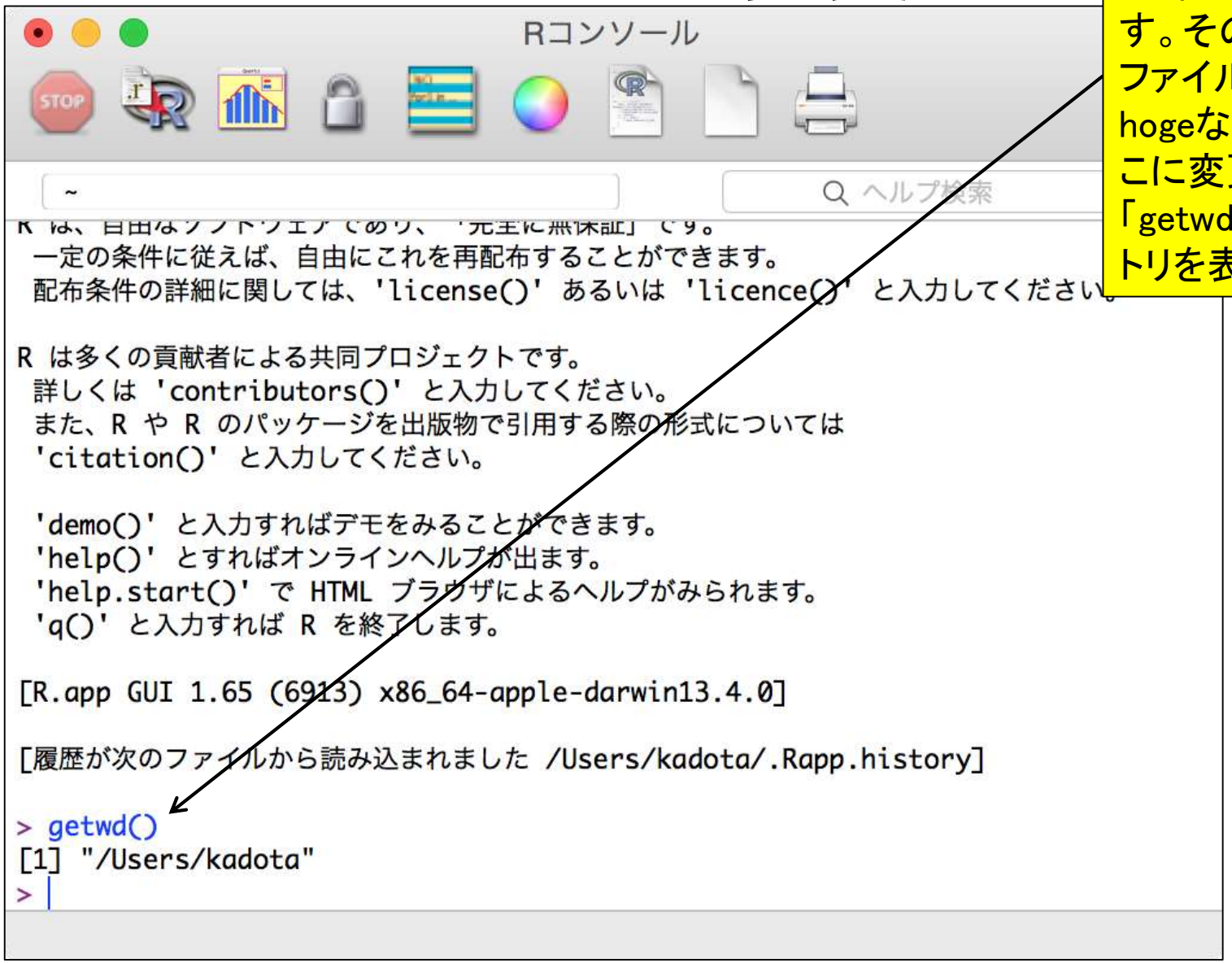
ファイルの保存

基本Rで取り扱うので、エディタもRのものを利用したほうが無難です。





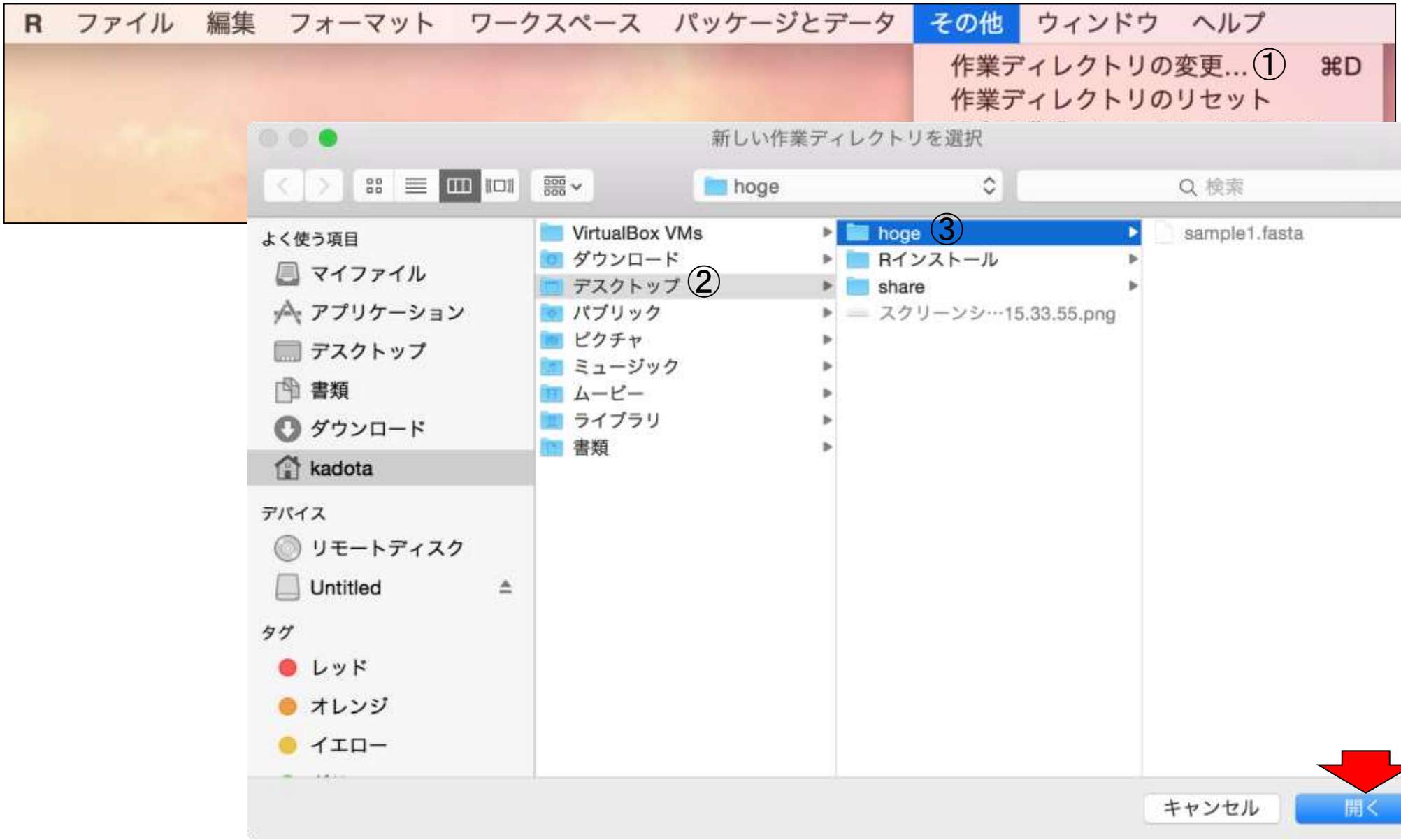
作業ディレクトリの変更



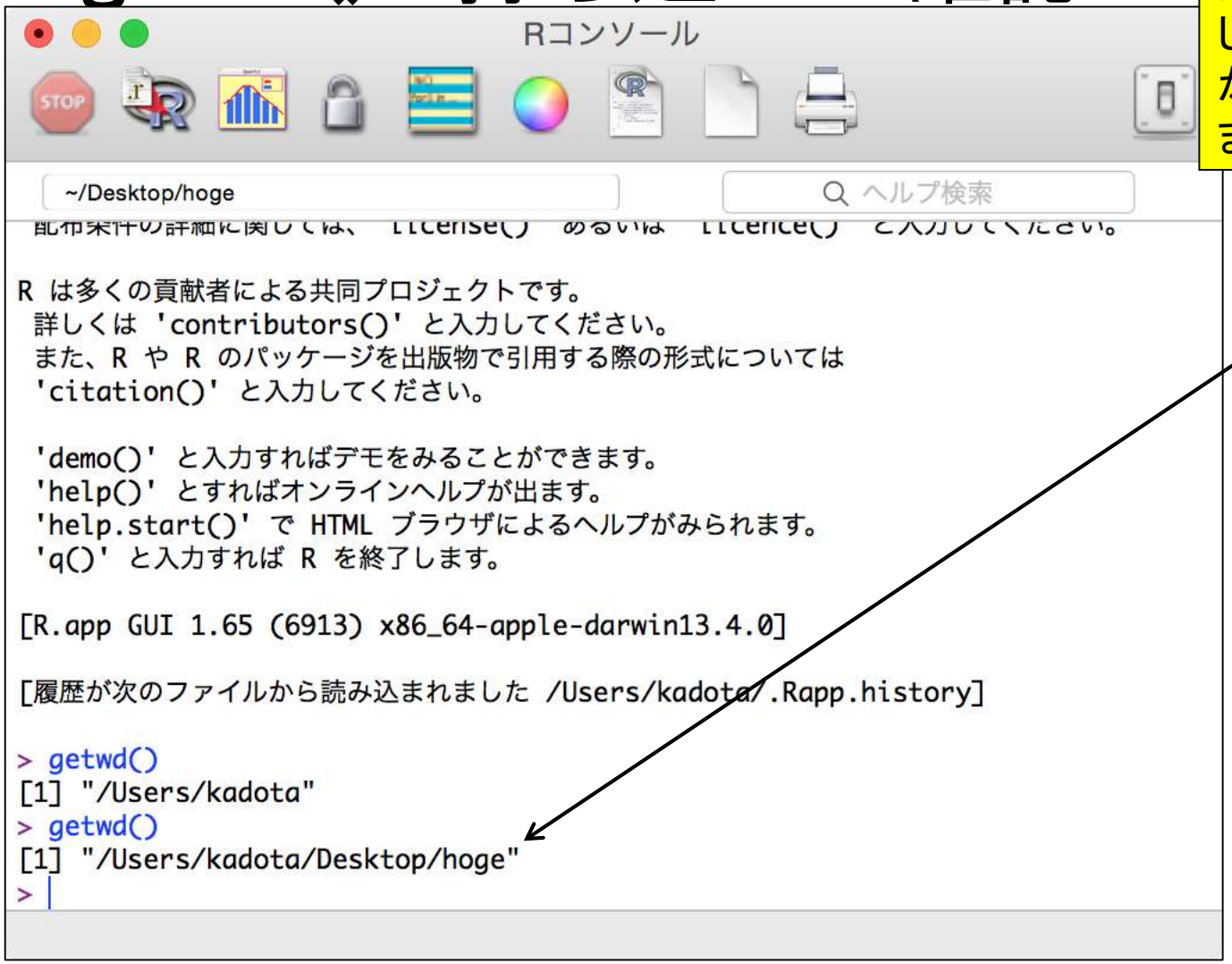
R起動直後のデフォルトの作業ディレクトリは、ユーザ名kadotaの環境では、「/Users/kadota」です。その一方で、今解析したいファイルはデスクトップ上にあるhogeなので、作業ディレクトリをそこに変更する必要があります。「getwd()」は、現在の作業ディレクトリを表示させるコマンドです。

作業ディレクトリの変更

デスクトップのhogeを指定して「開く」を押す。



getwd()と打ち込んで確認

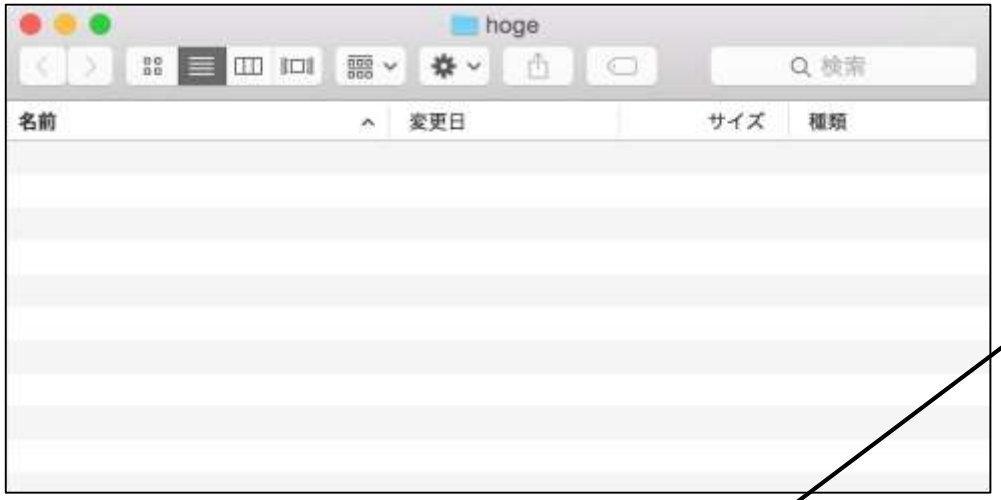


当たり前ですが、解析したいディレクトリ（またはフォルダ）を正しく指定できていなければエラーに遭遇します。また、解析したいファイルが存在しない状態でもエラーが出ます。

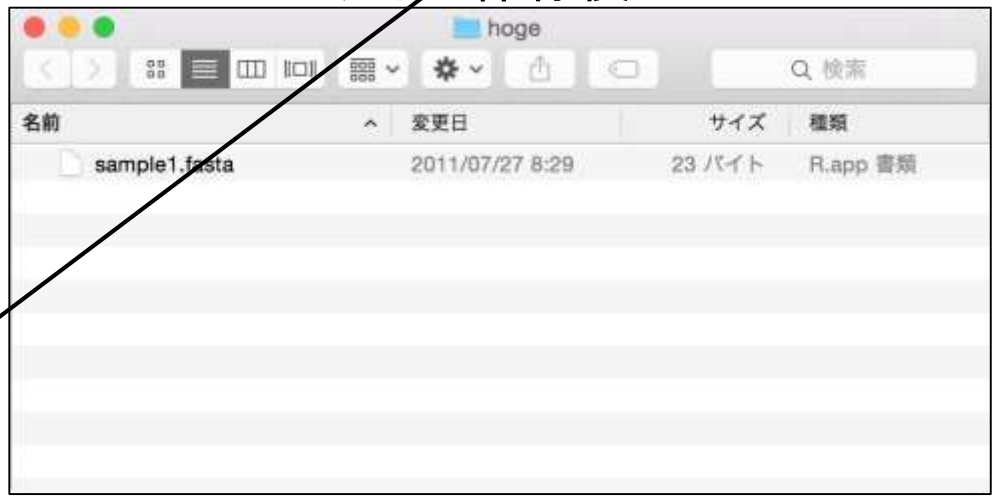
character(0)は何もないという意味

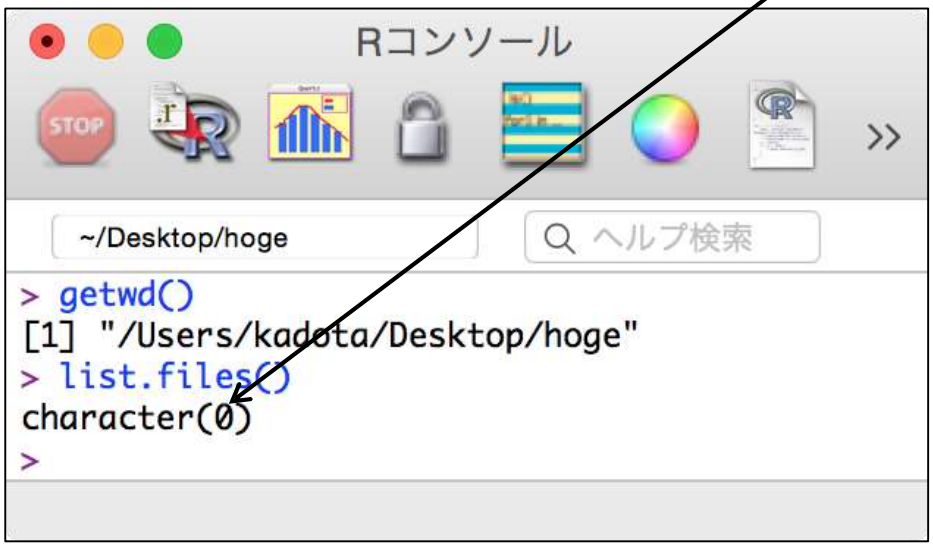
実際のhogeフォルダとR操作画面の関係

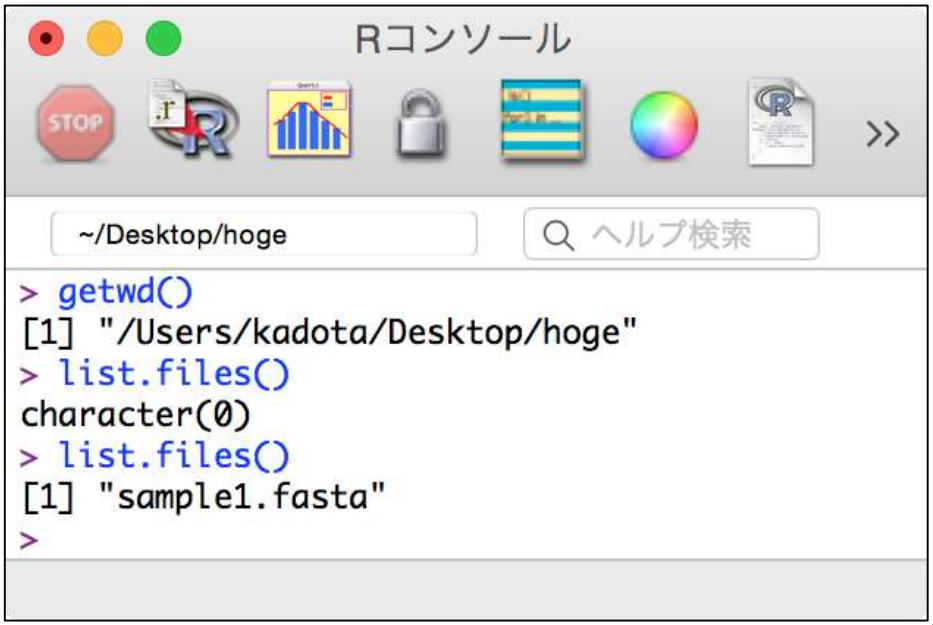
ファイル保存前



ファイル保存後







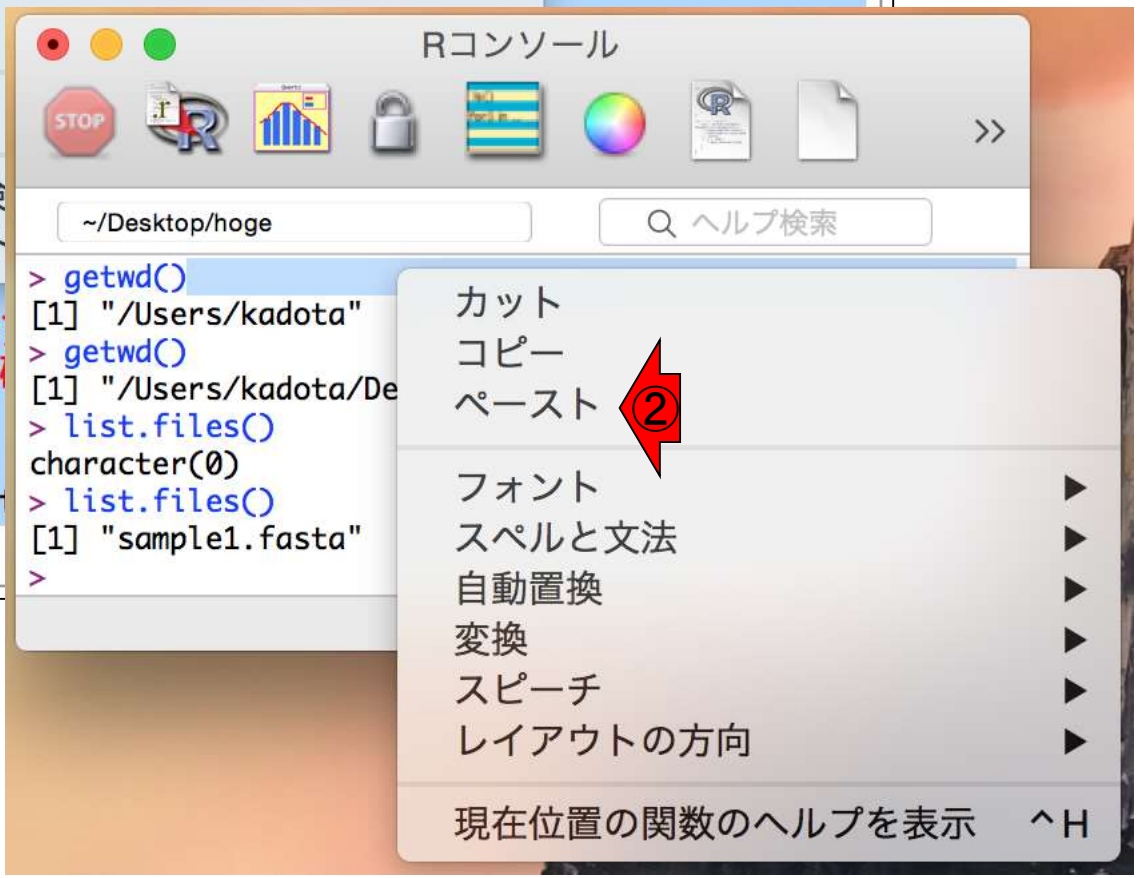
基本はコピペ

①一連のコマンド群をコピーして
②R Console画面上でペースト。

1. FASTA形式ファイル(sample1.fasta)の場合：

multi-FASTAではないsingle-FASTA形式ファイルです。

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

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

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



基本はコピペ

ペースト直後の状態。「リターンキー」を押す。

Rコンソール

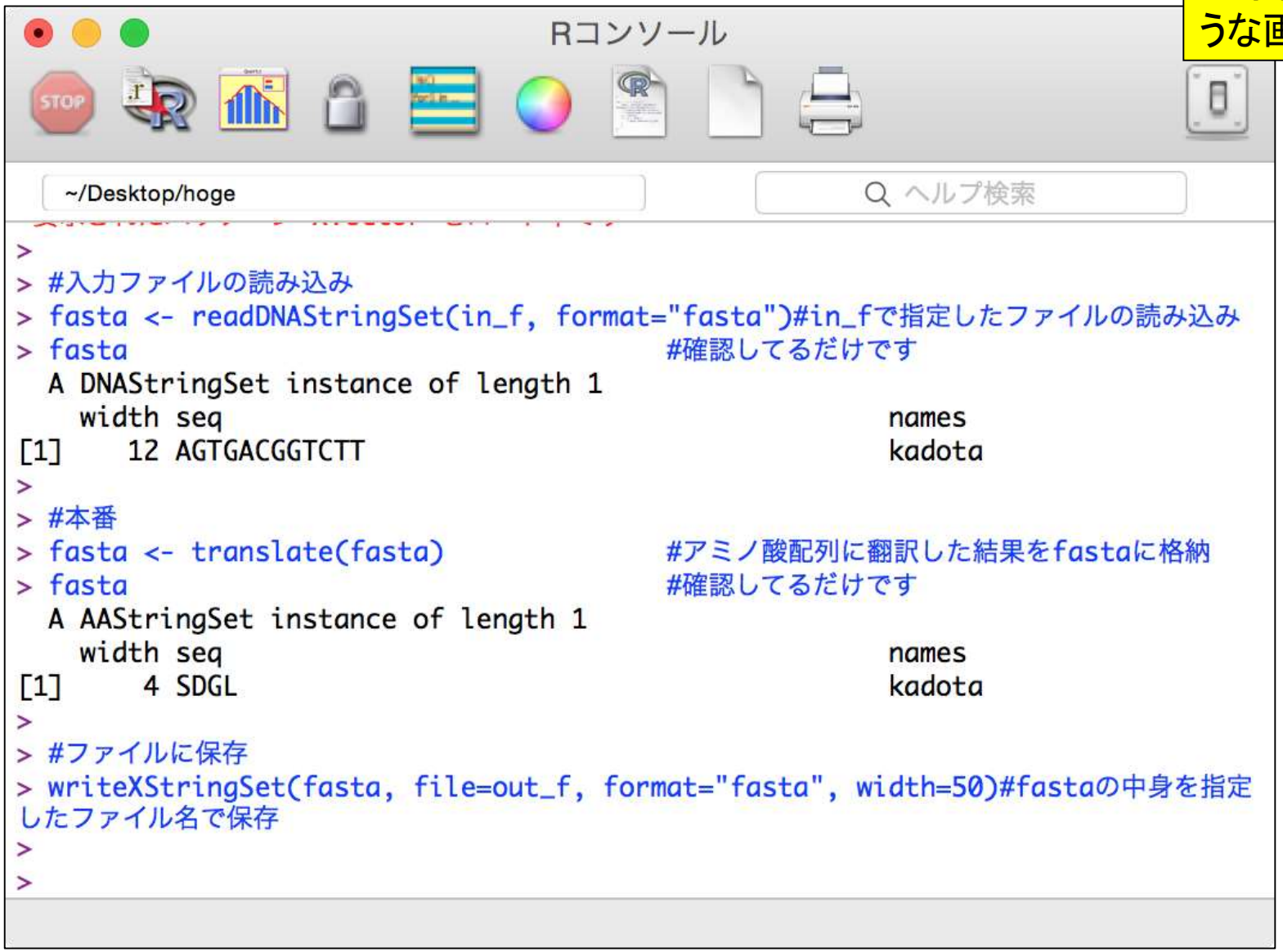
~/Desktop/hoge

ヘルプ検索

```
> list.files()
[1] "sample1.fasta"
> 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 <- translate(fasta)                   #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ル名で保存
```

基本はコピペ

「リターンキー」を押すとコピペしたコードが実行される。無事実行が終わると、このような画面になる。



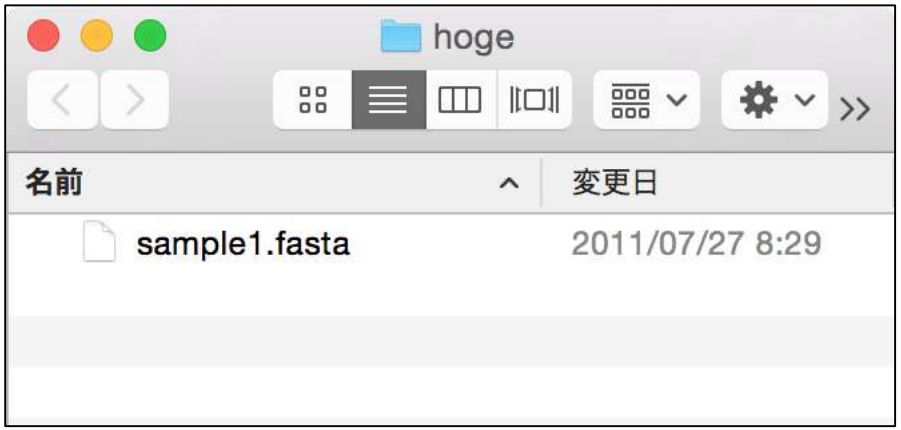
実行結果

出力ファイル名として指定したhoge1.fastaが生成されていることが分かります

Rコンソール

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

The following objects are masked from 'package:parallel':

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

The following object is masked from 'package:stats':

    xtabs

The following objects are masked from 'package:base':

    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, unsplit

 要求されたパッケージ S4Vectors をロード中です
 要求されたパッケージ stats4 をロード中です
 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
>
> #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
  A DNAStringSet instance of length 1
    width seq                                               names
[1]    12 AGTGACGGTCTT                                      kadota
>
> #本番
> fasta <- translate(fasta)             #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
  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)#fastaの中身を指定
したファイル名で保存
>
```

実行前のhogeフォルダ



実行後のhogeフォルダ

実行結果

「list.files()で表示される結果」と「実行後のhogeフォルダの中身」は当然同じです

Rコンソール

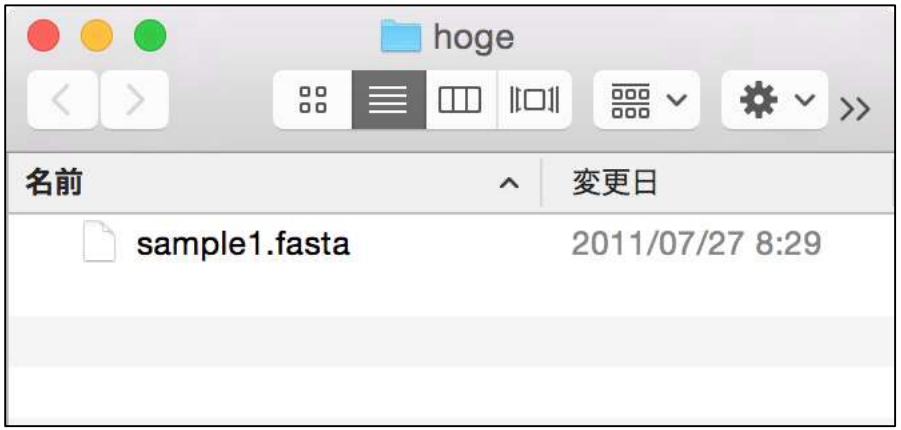
~/Desktop/hoge

ヘルプ検索

結果を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)#fastaの中身を指定したファイル名で保存
>
> list.files()
[1] "hoge1.fasta"    "sample1.fasta"
>
```

実行前のhogeフォルダ

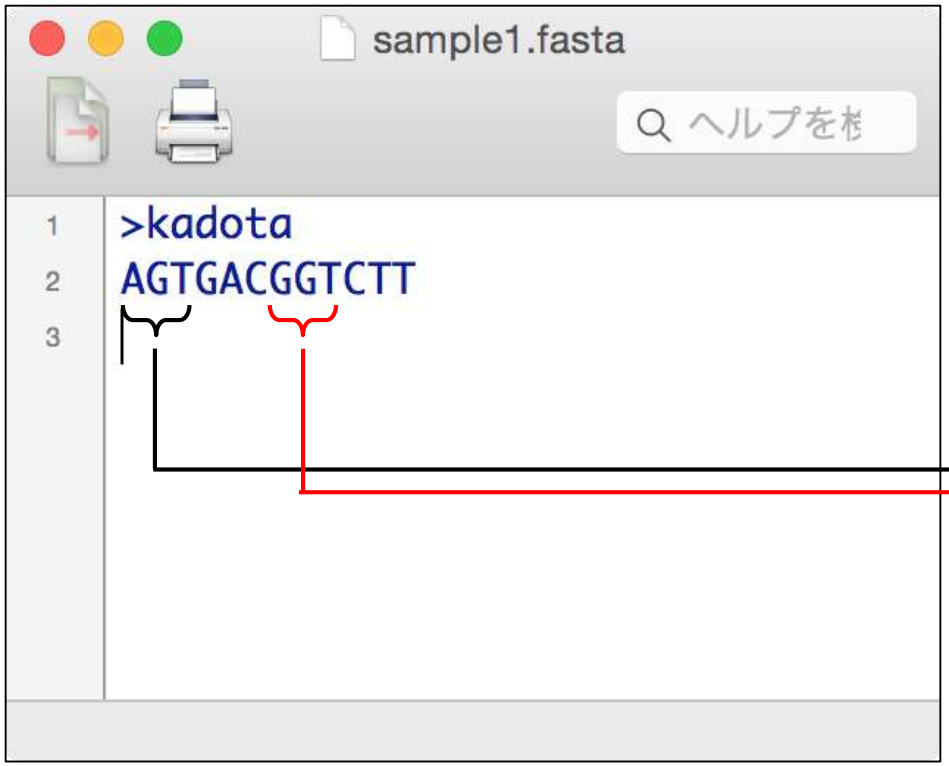


実行後のhogeフォルダ

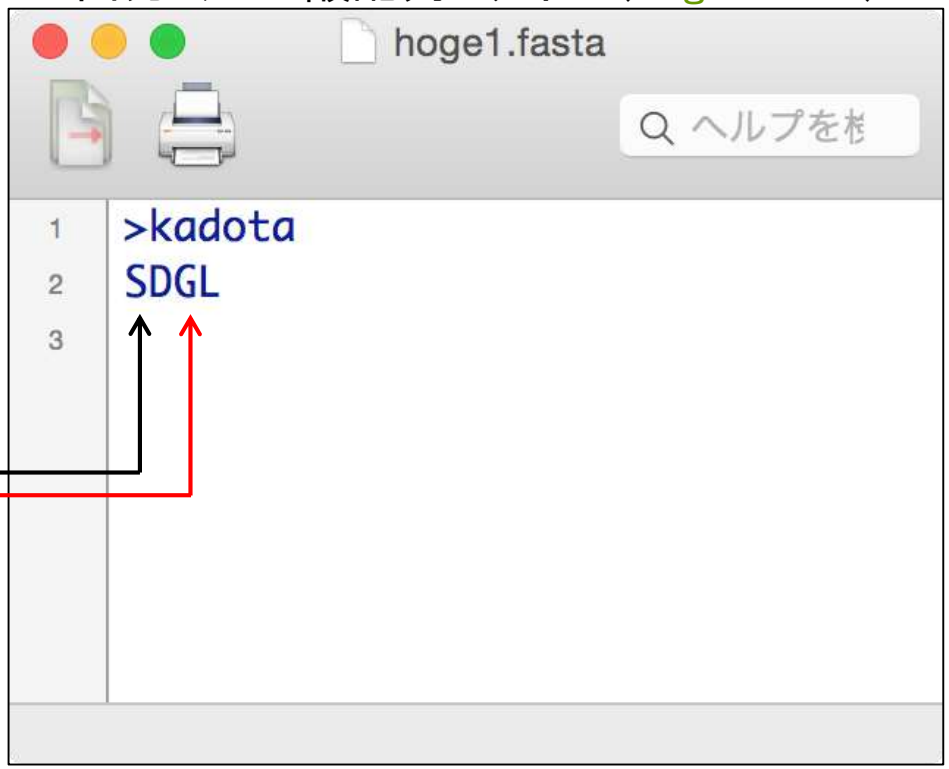
実行結果

入力ファイル中の塩基配列は、3の倍数の12塩基長、ACGTのみからなるので何のエラーも出ない

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）



解析基礎2

目的:アノテーションファイル（annotation.txt）中の第1列目に対して、リストファイル（genelist1.txt）中の文字列と一致する行を抜き出して、hoge1.txtというファイル名で出力したい

入力:アノテーションファイル（annotation.txt）

	A	B	C	D
1	genename	accession	description	subcellular_location
2	gene1	hoge01	plasma_mem	nuclear
3	gene2	hoge02	hohinu	membrane
4	gene3	hoge03	agribio	endoplasmic
5	gene4	hoge04	genesis	endoplasmic
6	gene5	hoge05	kamo	membrane
7	gene6	hoge06	netteba	humei
8	gene7	hoge07	tebasaki	nuclear
9	gene8	hoge08	biiru	nuclear
10	gene9	hoge09	nihonshu	nuclear
11	gene10	hoge10	agene1	membrane
12	gene11	hoge11	iyaaaa	endoplasmic

入力:リストファイル（genelist1.txt）

	A
1	gene1
2	gene7
3	gene9

出力:hoge1.txt

hoge1.txt

	A	B	C	D
1	gene1	hoge01	plasma_mem	nuclear
2	gene7	hoge07	tebasaki	nuclear
3	gene9	hoge09	nihonshu	nuclear

解析基礎2

目的:アノテーションファイル（annotation.txt）中の第1列目に対して、リストファイル（genelist1.txt）中の文字列と一致する行を抜き出して、hoge1.txtというファイル名で出力したい

- イントロ | 一般 | ランダムに行を抽出 (last modified 2014/07/17)
- イントロ | 一般 | 任意の文字列を行の最初に挿入 (last mod d 2014/07/17)
- イントロ | 一般 | 任意のキーワードを含む行を抽出(基礎) modified 2014/04/11)
- イントロ | 一般 | ランダムな塩基配列を生成 (last modified 4/06/16)
- 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 2015/02/19)
- 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
- イントロ | 一般 | 指定したID(染色体やdescription
- イントロ | 一般 | 翻訳配列(translate)を取得(基礎)
- イントロ | 一般 | 翻訳配列(translate)を取得(応用)

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み
keywords <- readLines(in_f2)             #in_f2で指定したファイルの読み込み
dim(data)                                #オブジェクトdataの行数と列数を表示

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

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

解析基礎2

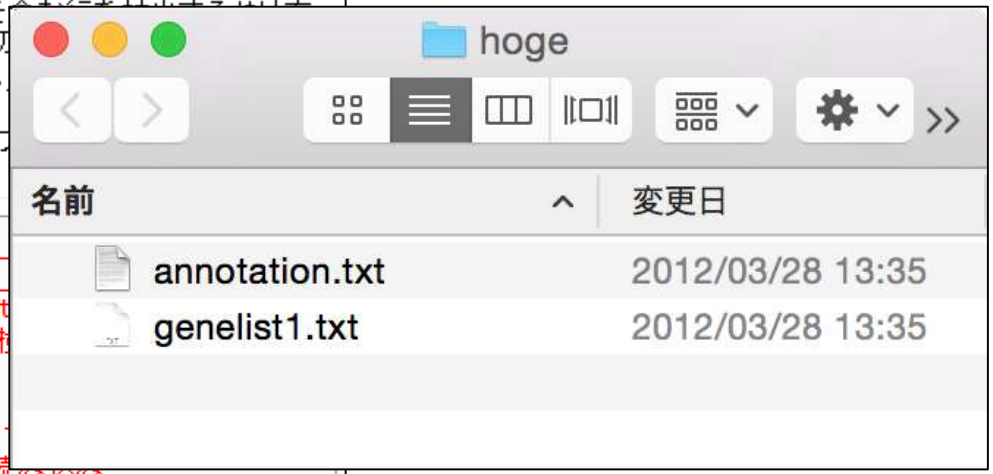
作業ディレクトリは「デスクトップ – hoge」。hogeフォルダ中にannotation.txtとgenelist1.txtが存在するという前提。

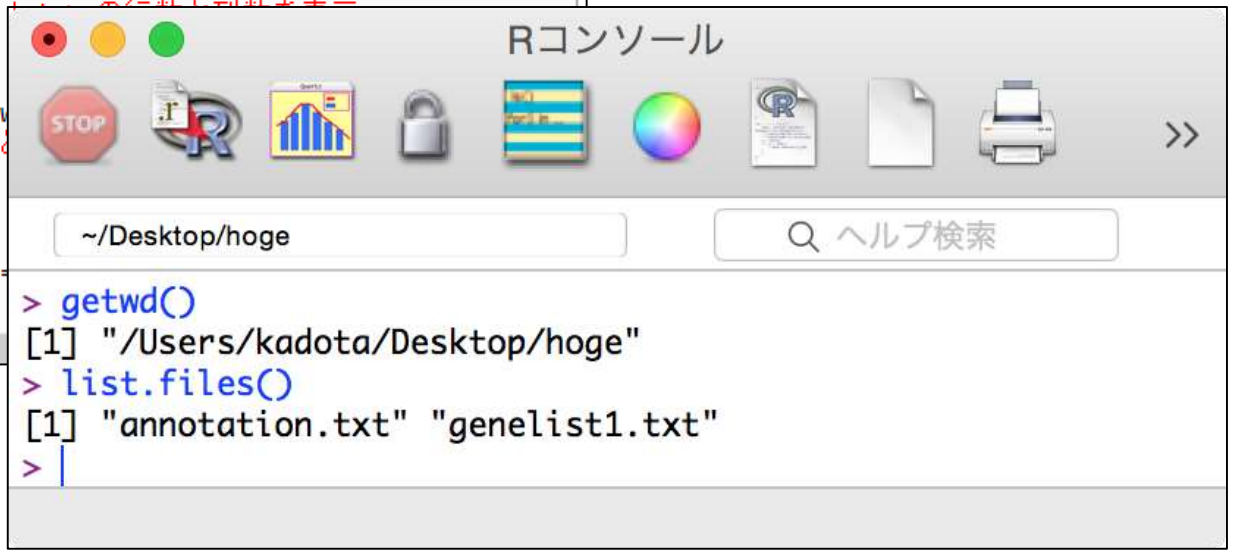
イントロ | 一般 | 任意のキーワードを含む行を抽出(基礎)

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
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
「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

**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
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
in_f2 <- "genelist1.txt"                 #入力ファイル名を指定してin
out_f <- "hoge1.txt"                     #出力ファイル名を指定してout
param <- 1                               #アノテーションファイル中の

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

#本番
obj <- is.element(as.character(data[,param]), keyw
out <- data[obj,]                        #objがTRUE
dim(out)                                 #オブジェク

#ファイルに保存
write.table(out, out_f, sep="\t", append=F, quote=
```





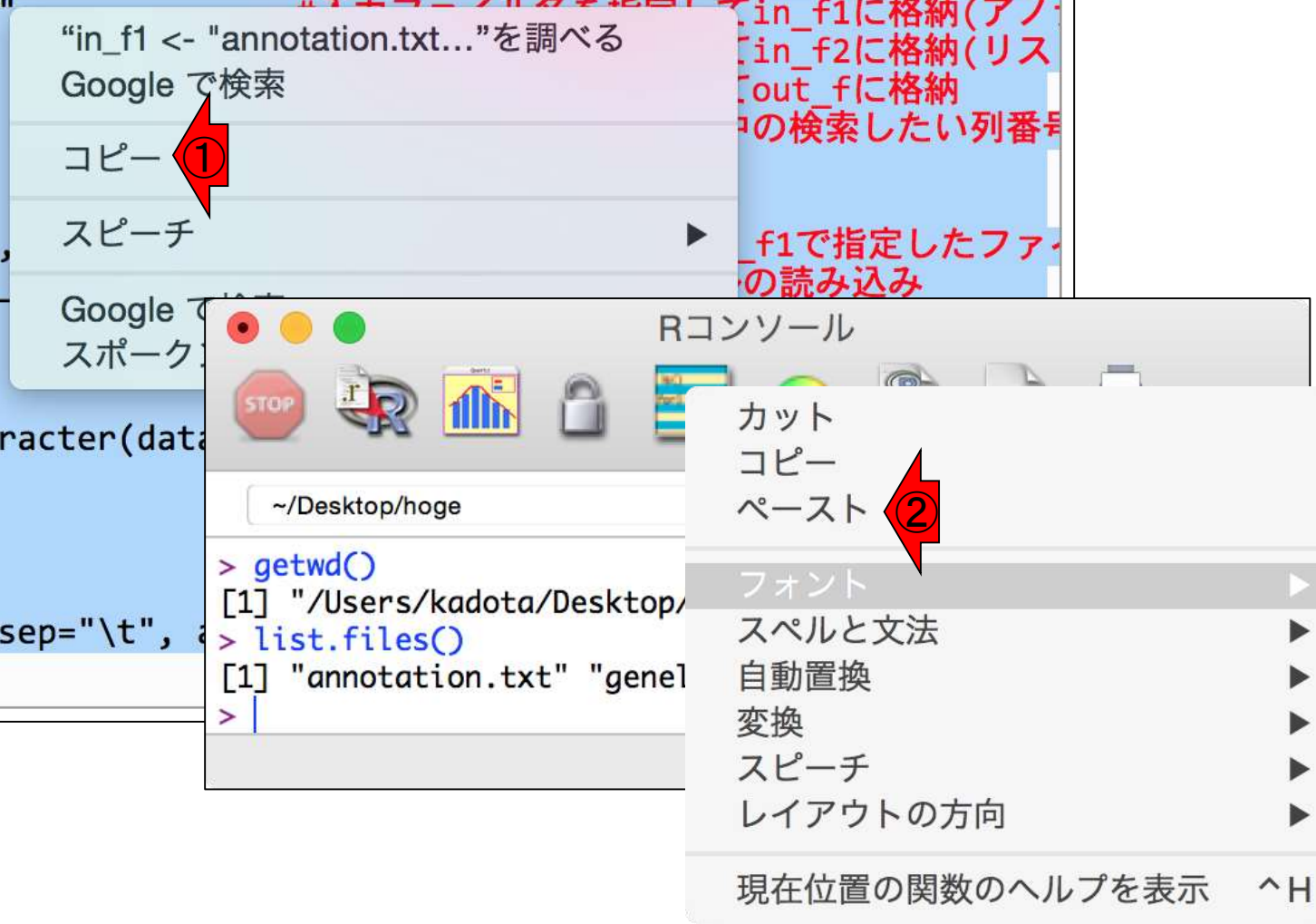
基本はコピペ

①一連のコマンド群をコピーして
②R Console画面上でペースト。
③「リターンキー」を押す

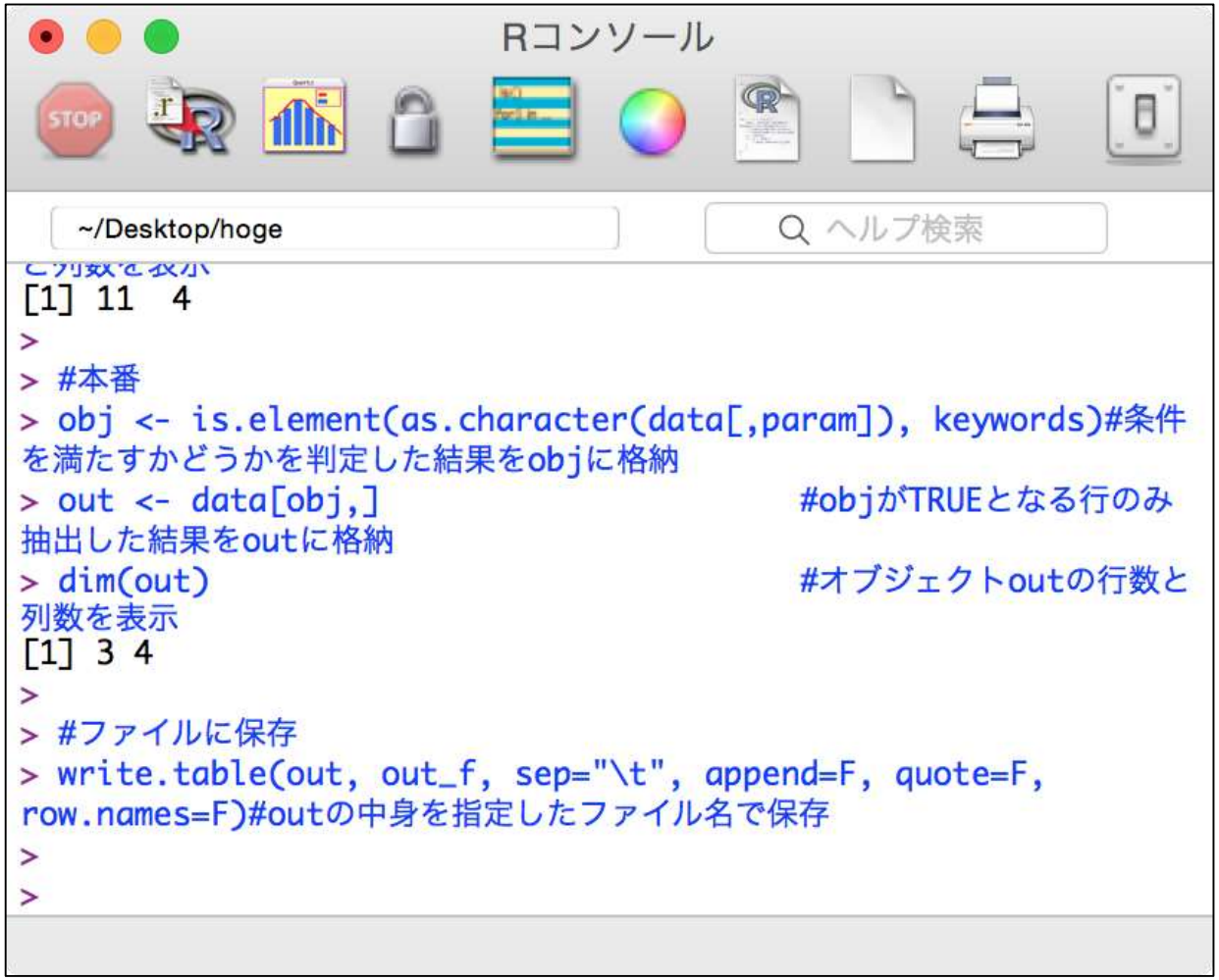
1. 目的のタブ区切りテキストファイル(annotation.txt)中の第1列目をキーとして、リストファイル(genelist1.txt)中のものが含まれる行全体を出力したい場合：

```
in_f1 <- "annotation.txt"
in_f2 <- "genelist1.txt"
out_f <- "hoge1.txt"
param <- 1

#入力ファイルの読み込み
data <- read.table(in_f1,
keywords <- readLines(in_
dim(data)

#本番
obj <- is.element(as.character(data
out <- data[obj,]
dim(out)

#ファイルに保存
write.table(out, out_f, sep="\t", a
```

```
#in_f1に格納(アノ
in_f2に格納(リス
out_fに格納
の検索したい列番号
f1で指定したファ
の読み込み
```



基本はコピペ



「リターンキー」を押すとコピペしたコードが実行される。無事実行が終わると、このような画面になる。

実行結果

「list.files()で表示される結果」と「実行後のhogeフォルダの中身」は当然同じです

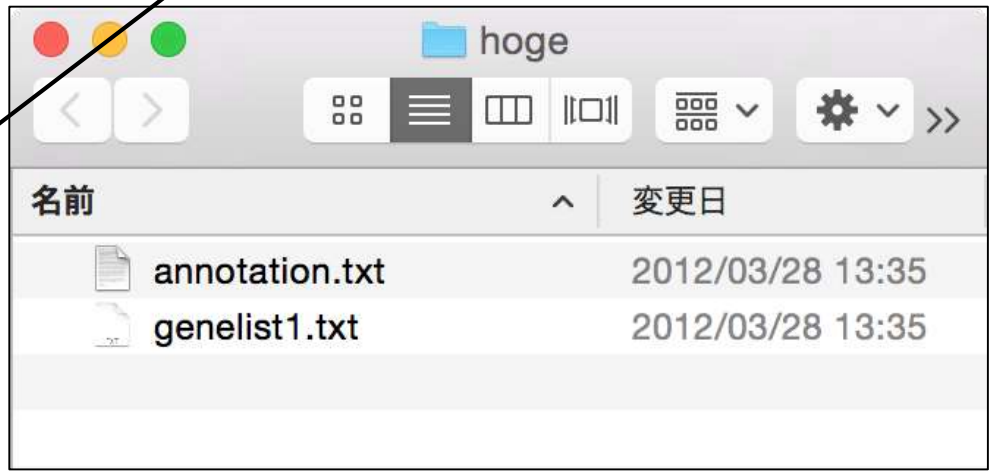
Rコンソール

~/Desktop/hoge

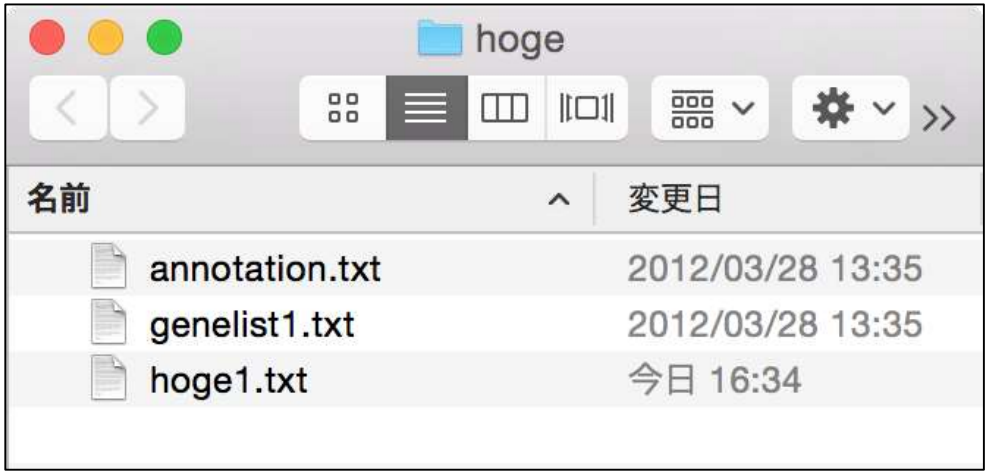
ヘルプ検索

```
> #本番
> obj <- is.element(as.character(data[,param]),
keywords)#条件を満たすかどうかを判定した結果をobjに格納
> 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
[1] 3 4
>
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F,
quote=F, row.names=F)#outの中身を指定したファイル名で保存
>
> list.files()
[1] "annotation.txt" "genelist1.txt"  "hoge1.txt"
>
```

実行前のhogeフォルダ



実行後のhogeフォルダ



実行結果

outというオブジェクトの中身をwrite.tableという関数でファイルに出力しています。それゆえ、出力ファイル(hoge1.txt)の中身は、Rコンソール画面中で「out」と打ち込むことでも見られます。

Rコンソール

~/Desktop/hoge

ヘルプ検索

```
> write.table(out,"out_f, sep="\t", append=F, quote=F, row.names=F)#out
の中身を指定したファイル名で保存
>
> list.files()
[1] "annotation.txt" "genelist1.txt"  "hoge1.txt"
> out
  genename accession description subcellular_location
1    gene1    hoge01  plasma_mem               nuclear
7    gene7    hoge07    tebasaki               nuclear
9    gene9    hoge09    nihonshu               nuclear
>
```

hoge1.txt

	A	B	C	D
1	genename	accession	description	subcellular_location
2	gene1	hoge01	plasma_mem	nuclear
3	gene7	hoge07	tebasaki	nuclear
4	gene9	hoge09	nihonshu	nuclear

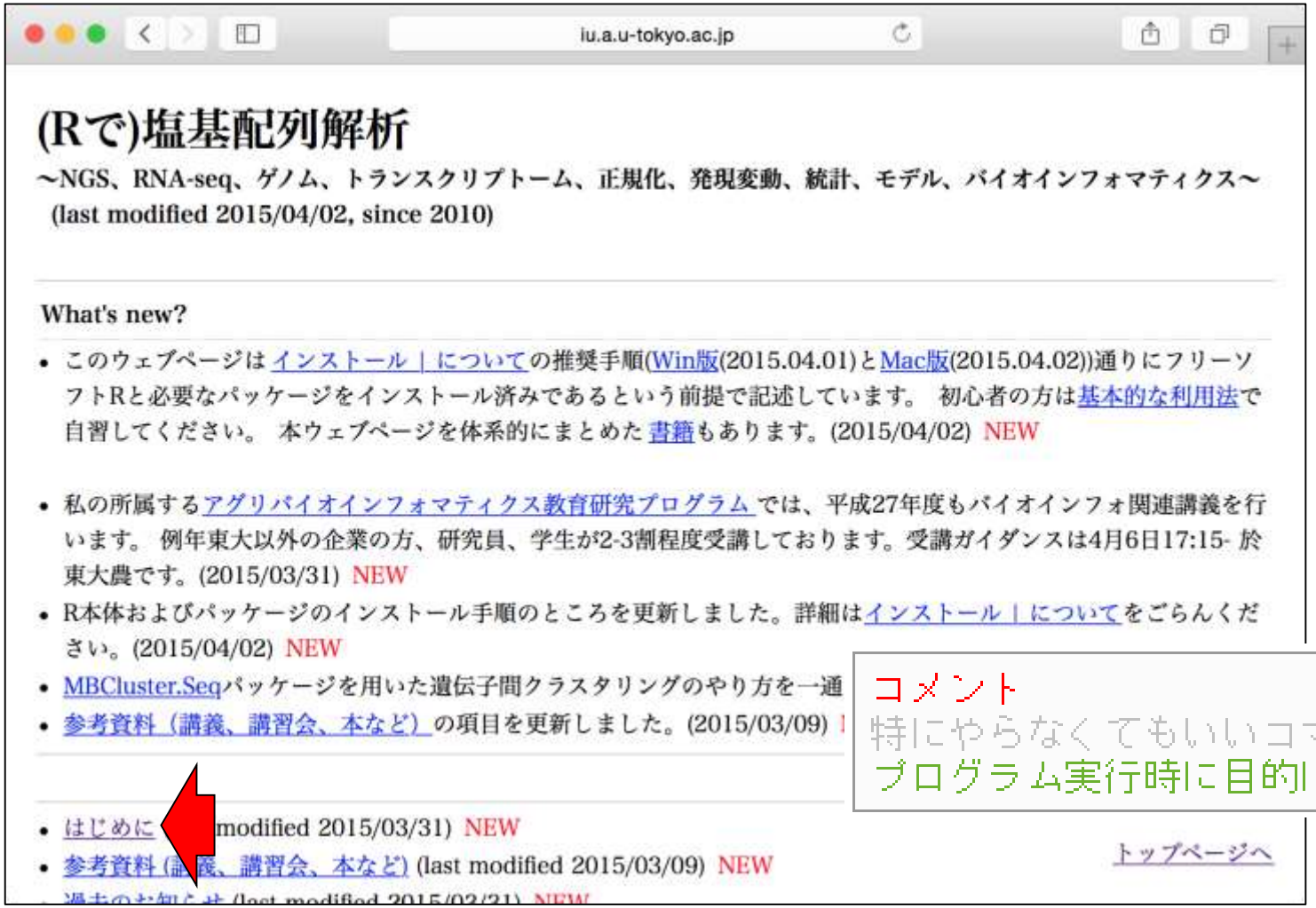
実行後のhogeフォルダ

hoge

名前	変更日
annotation.txt	2012/03/28 13:35
genelist1.txt	2012/03/28 13:35
hoge1.txt	今日 16:34

色の説明

Rコード中の色の使い分けについて説明します。



コメント
特にやらなくてもいいコマンド
プログラム実行時に目的に応じて変更すべき箇所

応用

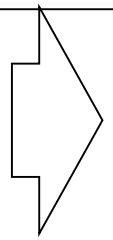
このサンプルコードは1列目でキーワード検索する場合。別のリストファイルを読み込んで4列目で検索したい場合のやり方を示します。

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(g
のが含まれる行全体を出力したい場合：

```
in_f1 <- "annotation.txt"                    #入力ファイル名を指定してin_f1に格納(アノテーションファイル)
in_f2 <- "genelist1.txt"                     #入力ファイル名を指定してin_f2に格納(リストファイル)
out_f <- "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", qu
keywords <- readLines(in_f2)                 #in_f2で指定
dim(data)                                    #オブジェクト

#本番
obj <- is.element(as.character(data[,param]), keywords)#条件を満たすかどうかを判定した結果をobjに格納
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
write.t                                      row.name
```

コメント
特にやらなくてもいいコマンド
プログラム実行時に目的に応じて変更すべき箇所

	A	B	C	D
1	genename	accession	description	subcellular_location
2	gene1	hoge01	plasma_mem	nuclear
3	gene2	hoge02	hohinu	membrane
4	gene3	hoge03	agribio	endoplasmic
5	gene4	hoge04	genesis	endoplasmic
6	gene5	hoge05	kamo	membrane
7	gene6	hoge06	netteba	humei
8	gene7	hoge07	tebasaki	nuclear
9	gene8	hoge08	biiru	nuclear
10	gene9	hoge09	nihonshu	nuclear
11	gene10	hoge10	agene1	membrane
12	gene11	hoge11	iyaaaa	endoplasmic

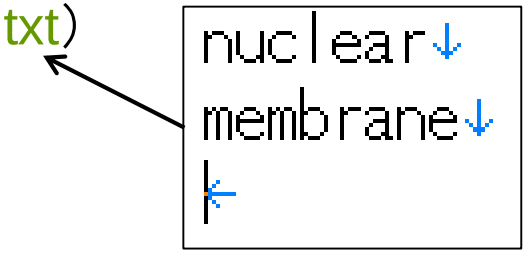


	A	B	C	D
1	genename	accession	description	subcellular_location
2	gene1	hoge01	plasma_mem	nuclear
3	gene2	hoge02	hohinu	membrane
4	gene3	hoge03	agribio	endoplasmic
5	gene4	hoge04	genesis	endoplasmic
6	gene5	hoge05	kamo	membrane
7	gene6	hoge06	netteba	humei
8	gene7	hoge07	tebasaki	nuclear
9	gene8	hoge08	biiru	nuclear
10	gene9	hoge09	nihonshu	nuclear
11	gene10	hoge10	agene1	membrane
12	gene11	hoge11	iyaaaa	endoplasmic

解答例

一連の作業手順を記述した**スクリプト**を1つのファイルとして保存することをお勧め

1. 目的のキーワードリストを含むファイルを作成し(例:list.txt)
2. 該当箇所を変更し、Rコンソール画面上でコピペ



run1.txt - メモ帳

ファイル(F) 編集(E) 書式(O) 表

```
in_f1 <- "annotation.txt"
in_f2 <- "genelist1.txt"
out_f <- "hoge1.txt"
param <- 1

#ファイルの読み込み
data <- read.table(in_f1,
keywords <- readLines(in_f
dim(data)

#本番
obj <- is.element(as.chara
out <- data[obj,]
dim(out)
write.table(out, out_f, se
```

run1.txt - メモ帳

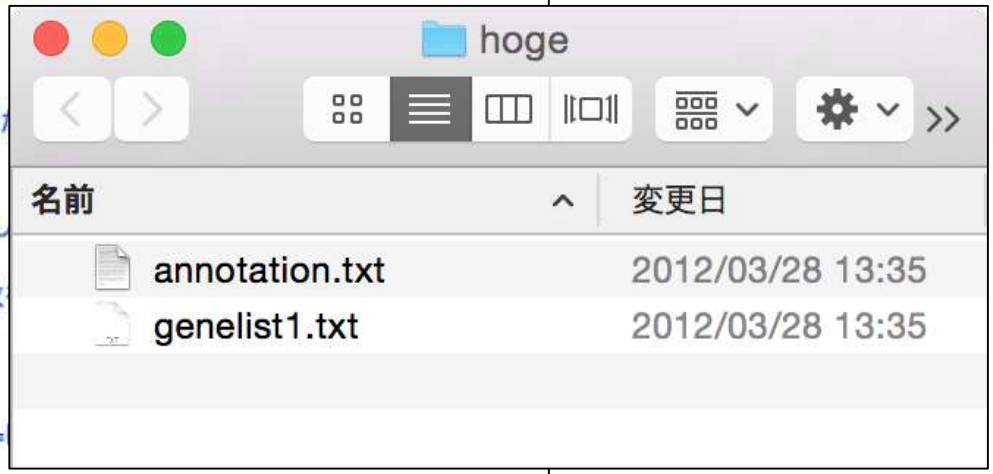
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

```
in_f1 <- "annotation.txt"
in_f2 <- "list.txt"
out_f <- "hoge1.txt"
param <- 4

#ファイルの読み込み
data <- read.table(in_f1, header=TRUE, sep="¥t", quote="")
keywords <- readLines(in_f2)
dim(data)

#本番
obj <- is.element(as.character(data[,param]), keywords)
out <- data[obj,]
dim(out)
write.table(out, out_f, sep="¥t", append=F, quote=F, row.names=
```

ありがちなミス1

```
> in_f1 <- "annotation.txt"                    #入力ファイル名を指定してin_f1に格納(アノテーションフ
> in_f2 <- "genelist1.txt"                     #入力ファイル名を指定してin_f2に格納(リストファイル)
> out_f <- "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み込
 以下にエラー file(file, "rt") :  コネクションを開くことができません
 追加情報:  警告メッセージ:
In file(file, "rt") :
   ファイル 'annotation.txt' を開くことができません: No such file or directory
> keywords <- readLines(in_f2)                 #in_f2で指定したファイルの読み込み
 以下にエラー file(con, "r") :  コネクションを開くことができません
 追加情報:  警告メッセージ:
In file(con, "r") :
   ファイル 'genelist1.txt' を開くことができません: No such file or directory
> dim(data)                                    #オブジェクトdataの行数と列数を表示
NULL
>
> #本番
> obj <- is.element(as.character(data[,param]), keywords)#条件を満
 以下にエラー data[, param] :
   'closure' 型のオブジェクトは部分代入可能ではありません
> out <- data[obj,]                            #objがTRUEとなる行のみ抽出し
 エラー:  オブジェクト 'obj' がありません
> dim(out)                                     #オブジェクトoutの行数と列数
 エラー:  オブジェクト 'out' がありません
>
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F, quote=F, row.names=
 以下にエラー is.data.frame(x) :  オブジェクト 'out' がありません
>
> getwd()
[1] "/Users/kadota"
>
```

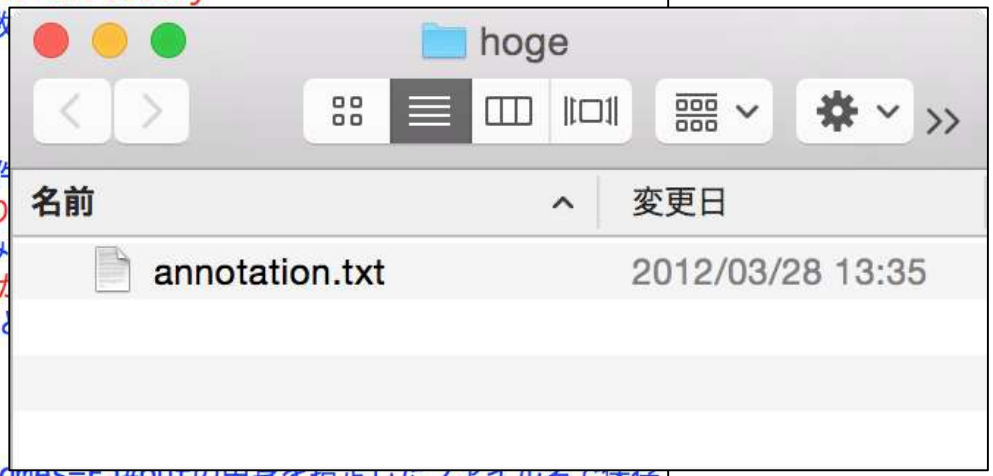
作業ディレクトリの変更を忘れているため、in_f1で指定した最初のファイルの読み込み段階でエラーが出る。つまり、実際に行ったフォルダ中にはannotation.txtというファイルは存在しないということ。



ありがちなミス2

```
> getwd()
[1] "/Users/kadota/Desktop/hoge"
> list.files()
[1] "annotation.txt"
> 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
> 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
> 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
>
> #入力ファイルの読み込み
> data <- read.table(in_f1, header=TRUE, sep="\t", quote="")#in_f1で指定したファイルの読み込み
> keywords <- readLines(in_f2)                 #in_f2で指定したファイルの読み込み
 以下にエラー file(con, "r") :  コネクションを開くことができません
 追加情報:  警告メッセージ:
In file(con, "r") :
   ファイル 'genelist1.txt' を開くことができません: No such file or directory
> dim(data)                                    #オブジェクトdataの行数
[1] 11  4
>
> #本番
> obj <- is.element(as.character(data[,param]), keywords)#条件
 以下にエラー match(el, set, 0L) :  オブジェクト 'keywords' があり
> out <- data[obj,]                            #objがTRUEとなる行のみ
 以下にエラー `[.data.frame`(data, obj, ) :  オブジェクト 'obj' が
> dim(out)                                     #オブジェクトoutの行数と
 エラー:  オブジェクト 'out' がありません
>
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F)#outの中身を指定したファイル名で保存
 以下にエラー is.data.frame(x) :  オブジェクト 'out' がありません
>
```

必要な入力ファイルが作業ディレクトリ中に存在しない。この場合、in_f2で指定したgenelist1.txtが存在しないため、それの読み込み段階でエラーが出ている。それゆえ、その情報を用いているコマンド部分でエラーが出ている。



ありがちなミス3

出力予定のファイル名と同じものをエクセルなど別のプログラムで開いているため、最後のwrite.table関数のところでエラーが出る。対処法は、出力ファイル名を変更するか、開いている別のプログラムを閉じる。これはWindowsの例。Macの場合、Excelで開いていようがおかまいなしに上書き保存されてしまうが、他のアプリケーションで開いている場合にこのようなエラーに遭遇する可能性があるため、念のため掲載。

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウインドウ ヘルプ

R Console

```
> in_f1 <- "annotation.txt"
> in_f2 <- "list.txt"                                    #入
> out_f <- "hoge1.txt"
> param <- 4
>
> #ファイルの読み込み
> data <- read.table(in_f1, header=TRUE, sep="\t", quote="")
> keywords <- readLines(in_f2)
> dim(data)
[1] 11  4
>
> #本番
> obj <- is.element(as.character(data[,param]), keywords)
> out <- data[obj,]
> dim(out)
[1] 7 4
> write.table(out, out_f, sep="\t", append=F, quote=F, row.names=
 以下にエラー file(file, ifelse(append, "a", "w")) :
   コネクションを開くことができません
 追加情報:    警告メッセージ:
In file(file, ifelse(append, "a", "w")) :
   ファイル 'hoge1.txt' を開くことができません: Permission denied
>
```

	genename	accession	description	subcellular_location
1				
2	gene1	hoge01	plasma_mer	nuclear
3	gene7	hoge07	tebasaki	nuclear
4	gene9	hoge09	nihonshu	nuclear
5				
6				

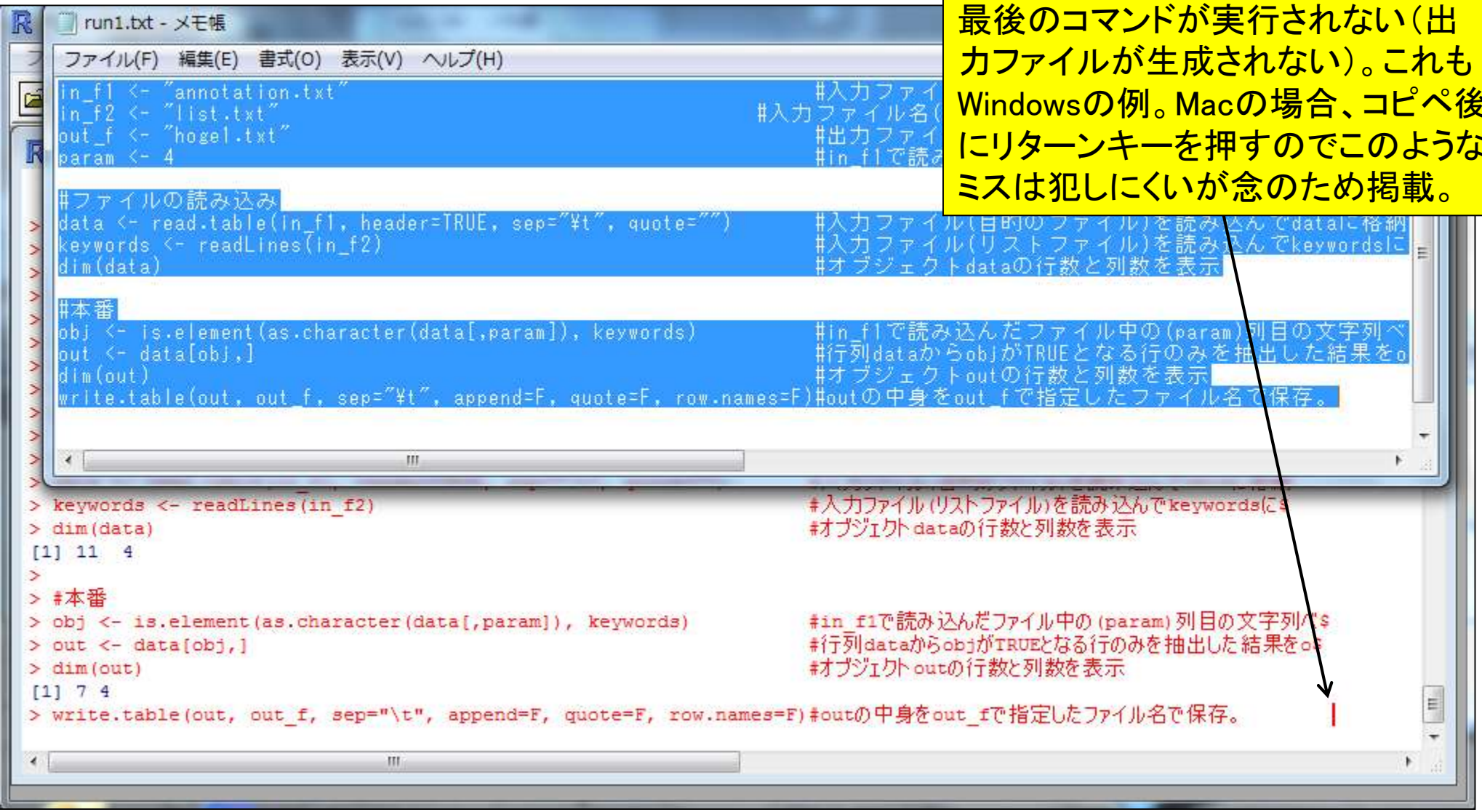
run1.txt - メモ帳

ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

```
in_f1 <- "annotation.txt"
in_f2 <- "list.txt"
out_f <- "hoge1.txt"
param <- 4

#ファイルの読み込み
data <- read.table(in_f1, header=TRUE, sep="¥t", quote=""
keywords <- readLines(in_f2)
dim(data)

#本番
obj <- is.element(as.character(data[,param]), keywords)
out <- data[obj,]
dim(out)
write.table(out, out_f, sep="¥t", append=F, quote=F, row.n
```

ありがちなミス4

実行スクリプトをコピーする際、最後の行のところで改行を含ませずにR Console画面上でペーストしたため、最後のコマンドが実行されない（出力ファイルが生成されない）。これもWindowsの例。Macの場合、コピペ後にリターンキーを押すのでこのようなミスは犯しにくいが念のため掲載。



警告メッセージ

list.txtファイル作成時に、membraneと打った後に改行を入れた場合（左）と入れない場合（右）の挙動の違いを把握し、後学のために警告メッセージの意味を理解しておくとよい。この場合は結果には影響していないことがわかる。Rは警告メッセージ後の記述内容が比較的分かりやすいのでよく読むべし。

```
> in_f1 <- "annotation.txt"
> in_f2 <- "list.txt"
> out_f <- "hoge1.txt"
> param <- 4
>
> #入力ファイルの読み込み
> data <- read.table(in_f1, header=TRUE, sep="\t"
> keywords <- readLines(in_f2)
> dim(data)
[1] 11  4
>
> #本番
> obj <- is.element(as.character(data[,param]), k
> out <- data[obj,]
> dim(out)
[1] 7 4
>
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F, quo
>
```

nuclear↓
membrane↓
←

```
> in_f1 <- "annotation.txt"
> in_f2 <- "list.txt"
> out_f <- "hoge1.txt"
> param <- 4
>
> #入力ファイルの読み込み
> data <- read.table(in_f1, header=TRUE, sep="\t", quote="")#in_f1
> keywords <- readLines(in_f2)
 警告メッセージ:
In readLines(in_f2) :  'list.txt' で不完全な最終行が見つかりました
> dim(data)
[1] 11  4
>
> #本番
> obj <- is.element(as.character(data[,param]), keywords)#条件を満た
> out <- data[obj,]
> dim(out)
[1] 7 4
>
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F
>
```

nuclear↓
membrane←