

2022.06.17版

「(Rで)塩基配列解析」というウェブページを通して、RStudioの基本的な利用法を学ぶためのものです。Google Chrome(推奨)で動作確認しています。ブラウザによって若干挙動が異なります。スライドの枚数が非常に多いですが、1つ1つの手順を丁寧に解説しているだけです。

# (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ユニット  
<https://www.iu.a.u-tokyo.ac.jp/>

# Contents

- 前提条件、RStudioの起動
- 見栄えの統一、終了
- Console画面の見栄え、設定変更
- 枠の幅、電卓的な利用
- プロンプト(>)、Console画面クリア
- 計算の中断
- 指定した範囲の配列を取得
  - 項目と例題、入力ファイルの保存
  - 作業ディレクトリの変更
  - 入力ファイルの確認、FASTA形式
  - コピペ実行、出力ファイルの確認、Environmentタブ
- フィルタリング(任意のキーワードを含む行を抽出)
  - 全体像の把握、入力ファイルの保存、コピペ実行、出力ファイルの確認
  - 応用
- 警告メッセージ、ありがちなミス

# 前提条件

①の推奨手順通りにインストールができていなくても、実際にはできていなかったという事例が散見されます。ページをスクロールダウンして、

## (Rで)塩基配列解析

(last modified 2022/06/01, since 2010)



このウェブページの多くは、[インストール | についての推奨手順](#) (Windows2022.05.16版とMacintosh2022.05.17版)に従ってフリーソフトRと必要なパッケージをインストール済みであるという前提で記述しています。初心者の方は[基本的な利用法](#)(Windows2022.05.17版の[PPTX](#)と[PDF](#); Macintosh2020.03.13版の[PPTX](#)と[PDF](#))で自習してください。

[@Agribio\\_utokyo](#)さんをフォロー

、[アグリバイオ](#)、[\(Rで\)塩基配列解析のサブページ](#)、[生命情報解析研究室](#)

### What's new? (過去のお知らせはこちら)

- このページをGoogle ChromeやMicrosoft Edgeで開いても、すぐにフリーズして「応答なし」と表示されることが頻繁にあったという報告をいただきました。このような事象に遭遇した方はブラウザをFirefoxに変更すると解消されるようです。情報提供いただいた学生さんに感謝m(\_ \_)m(2022/06/01) **NEW**
- ([サブページ](#)のほうのネタではありますが...) 日本乳酸菌学会誌の[第16回](#)と[第17回](#)の原稿をこちらでも公開しました。(2022/05/29) **NEW**
- 「インストール | R本体とRStudio | 最新版 | [Mac用](#)」を更新しました。(2022/05/16)

- [はじめに](#) (last modified 2022/03/28)
- [過去のお知らせ](#) (last modified 2022/06/01) **NEW**
- [インストール | について](#) (last modified 2022/05/17)
- インストール | R本体とRStudio | 最新版 | [Win用](#) (last modified 2022/05/16)推奨
- インストール | R本体とRStudio | 最新版 | [Mac用](#) (last modified 2022/05/16)推奨

[トップページへ](#)

# 前提条件

①の推奨手順通りにインストールができていなくても、実際にはできていなかったという事例が散見されます。ページをスクロールダウンして、実際の作業は②と③ですが、インストールに失敗しているパッケージについては、④を参考に個別対応しましょう。パッケージ名や関数名のスペルミスだったというオチも散見されます。気を付けましょう。



@Agribio\_utokyoさんをフォロー、[アグリバイオ](#)、[\(Rで\)塩基配列解析のサブページ](#)、[生命情報解析研究室](#)

## What's new? (過去のお知らせはこちら)

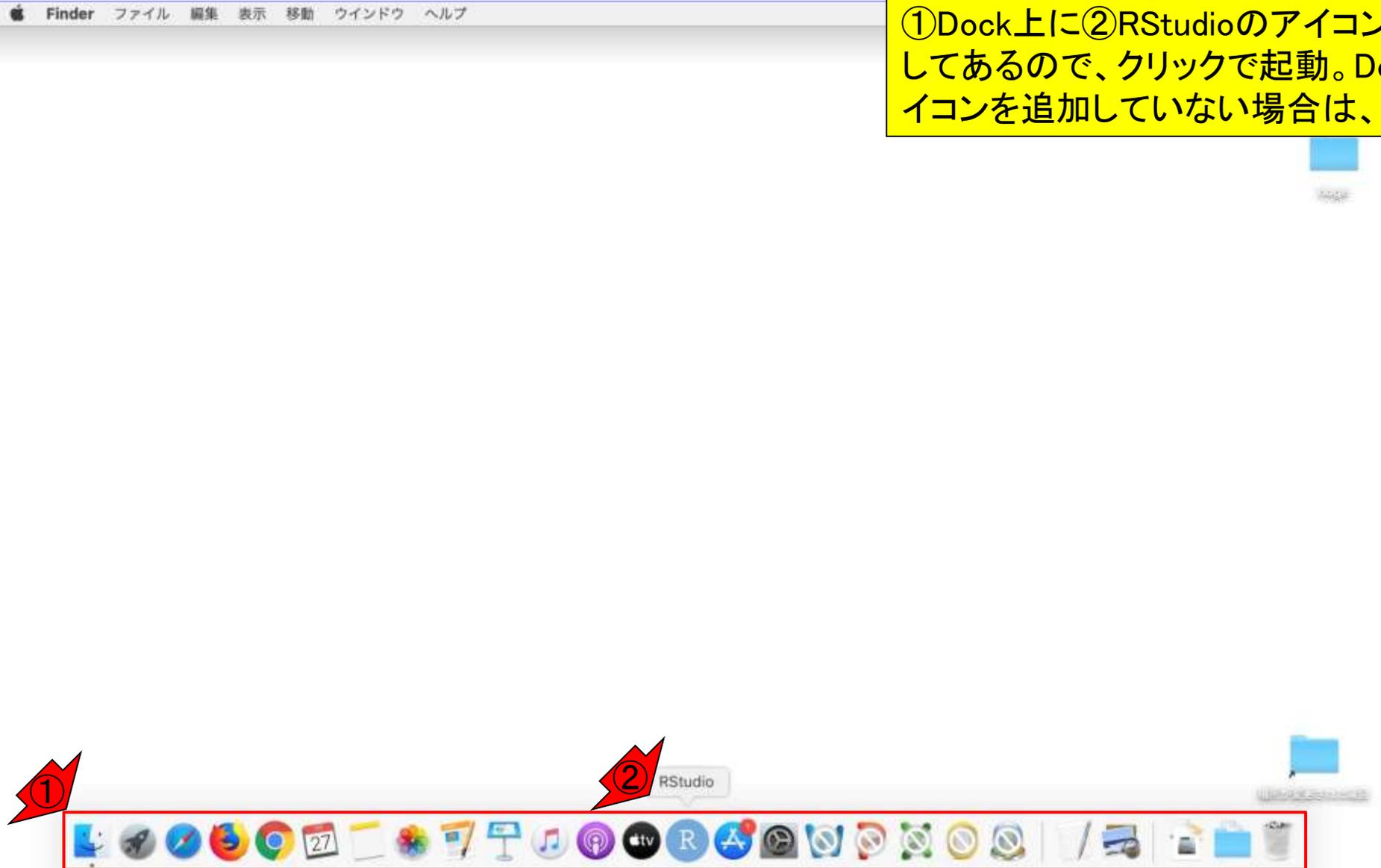
- このページをGoogle ChromeやMicrosoft Edgeで開いても、すぐにフリーズして「応答なし」と表示されるにであったという報告をいただきました。このような事象に遭遇した方はブラウザをFirefoxに変更すると解消します。情報提供いただいた学生さんに感謝m(\_ \_)m(2022/06/01) **NEW**
- ([サブページ](#)のほうのネタではありますが...) 日本乳酸菌学会誌の[第16回](#)と[第17回](#)の原稿をこちらでも公開しました。(2022/05/29) **NEW**
- 「インストール | R本体とRStudio | 最新版 | [Mac用](#)」を更新しました。(2022/05/16)

- [はじめに](#) (last modified 2022/03/28)
- [過去のお知らせ](#) (last modified 2022/06/01) **NEW**
- [インストール | について](#) (last modified 2022/05/17)
- インストール | R本体とRStudio | 最新版 | [Win用](#) (last modified 2022/05/16) 推奨
- インストール | R本体とRStudio | 最新版 | [Mac用](#) ② (last modified 2022/05/16) 推奨
- インストール | R本体 | 過去版 | [Win用](#) (last modified 2015/03/22)
- インストール | R本体 | 過去版 | [Mac用](#) (last modified 2015/03/22)
- [インストール | Rパッケージ | について](#) (last modified 2018/11/13)
- インストール | Rパッケージ | [必要最小限プラスアルファ](#) ③ (last modified 2022/05/13) 推奨
- インストール | Rパッケージ | [個別\(2018年11月以降\)](#) ④ (last modified 2022/03/31)
- インストール | Rパッケージ | [個別\(2018年11月以前\)](#) (last modified 2018/11/12)
- [基本的な利用法](#) (last modified 2022/05/17)
- [サンプルデータ](#) (last modified 2019/09/06)

[トップページへ](#)

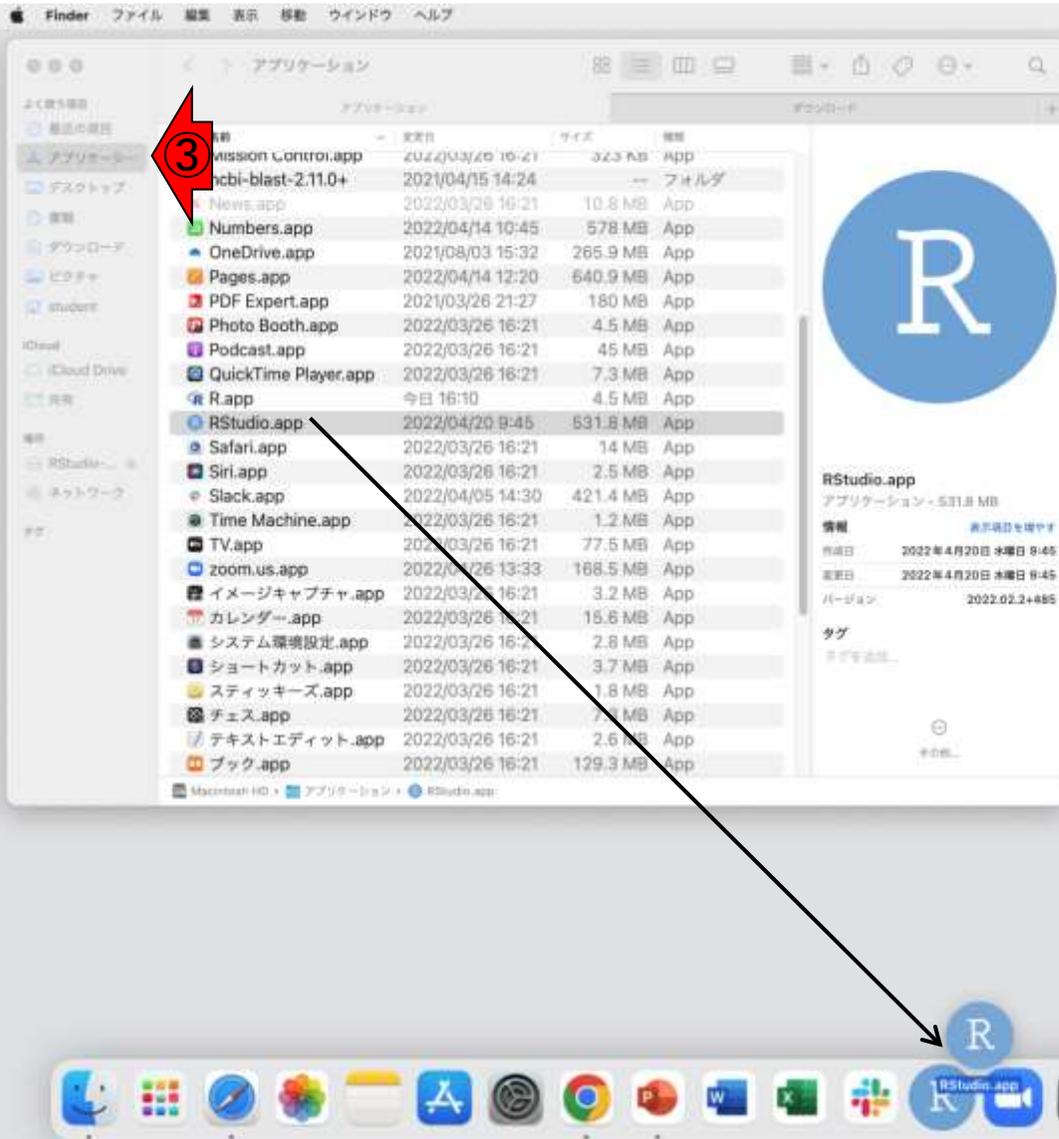
# RStudioの起動1

Macintoshのデスクトップのスクショ。インストールについてのMacintosh版の通りにインストールを行った場合は、おそらく①Dock上に②RStudioのアイコンを追加してあるので、クリックで起動。Dockにアイコンを追加していない場合は、

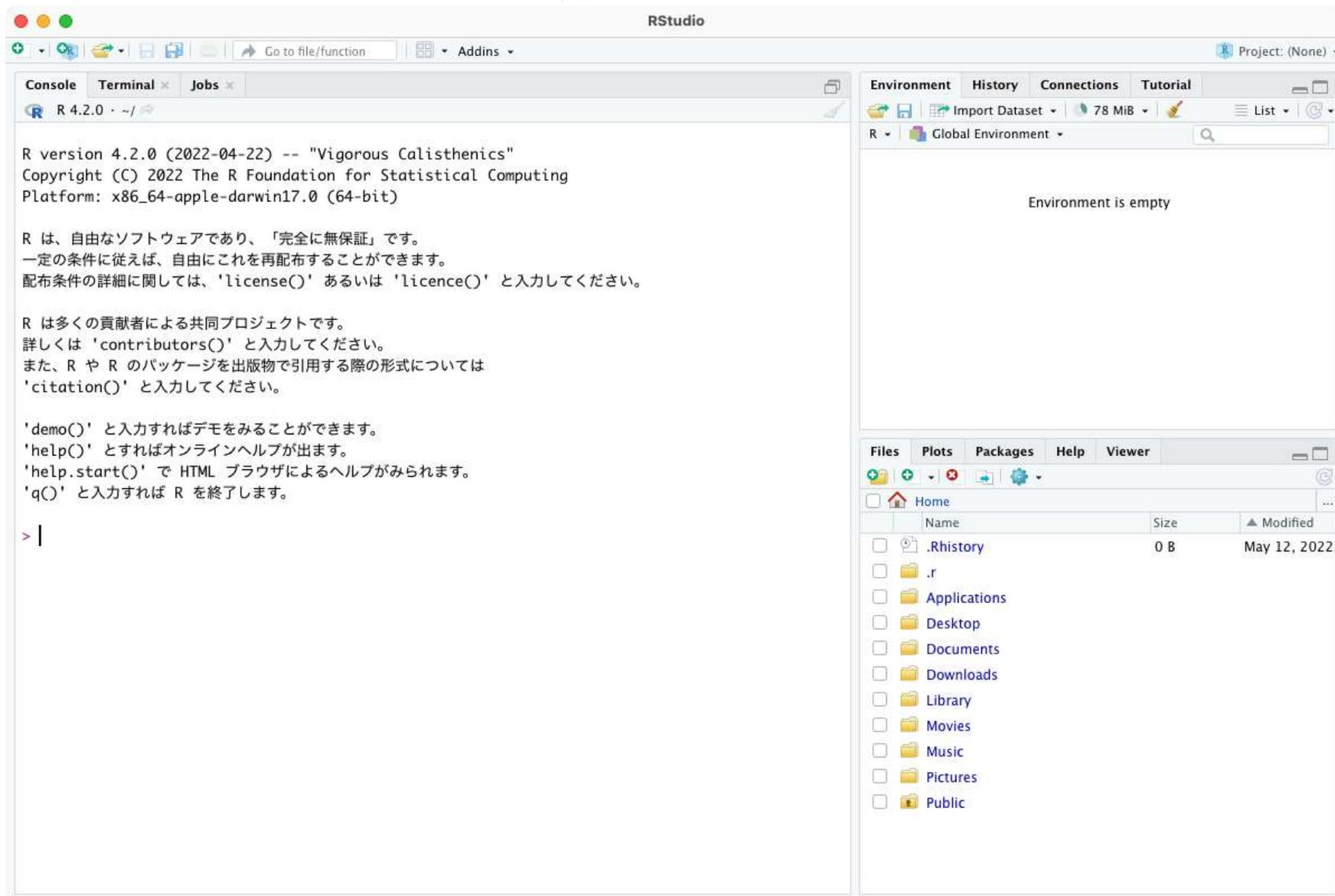


# RStudioの起動2

Macintoshのデスクトップのスクショ。インストールについてのMacintosh版の通りにインストールを行った場合は、おそらく①Dock上に②RStudioのアイコンを追加してあるので、クリックで起動。Dockにアイコンを追加していない場合は、③アプリケーションを開きRStudioのアイコンをダブルクリックして起動します。嫌でなければDockにドラッグ&ドロップして追加しておきましょう。



# RStudioの起動3



The screenshot shows the RStudio interface. The console pane on the left displays the R version information and usage instructions. The environment pane on the right shows that the environment is empty. The file explorer at the bottom shows the home directory with various folders and files.

**Console Output:**

```
R 4.2.0 · ~/
```

R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"  
Copyright (C) 2022 The R Foundation for Statistical Computing  
Platform: x86\_64-apple-darwin17.0 (64-bit)

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

R は多くの貢献者による共同プロジェクトです。  
詳しくは '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ついては  
'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す。  
'q()' と入力すれば R を終了します。

> |

**Environment Pane:**

Environment is empty

**File Explorer:**

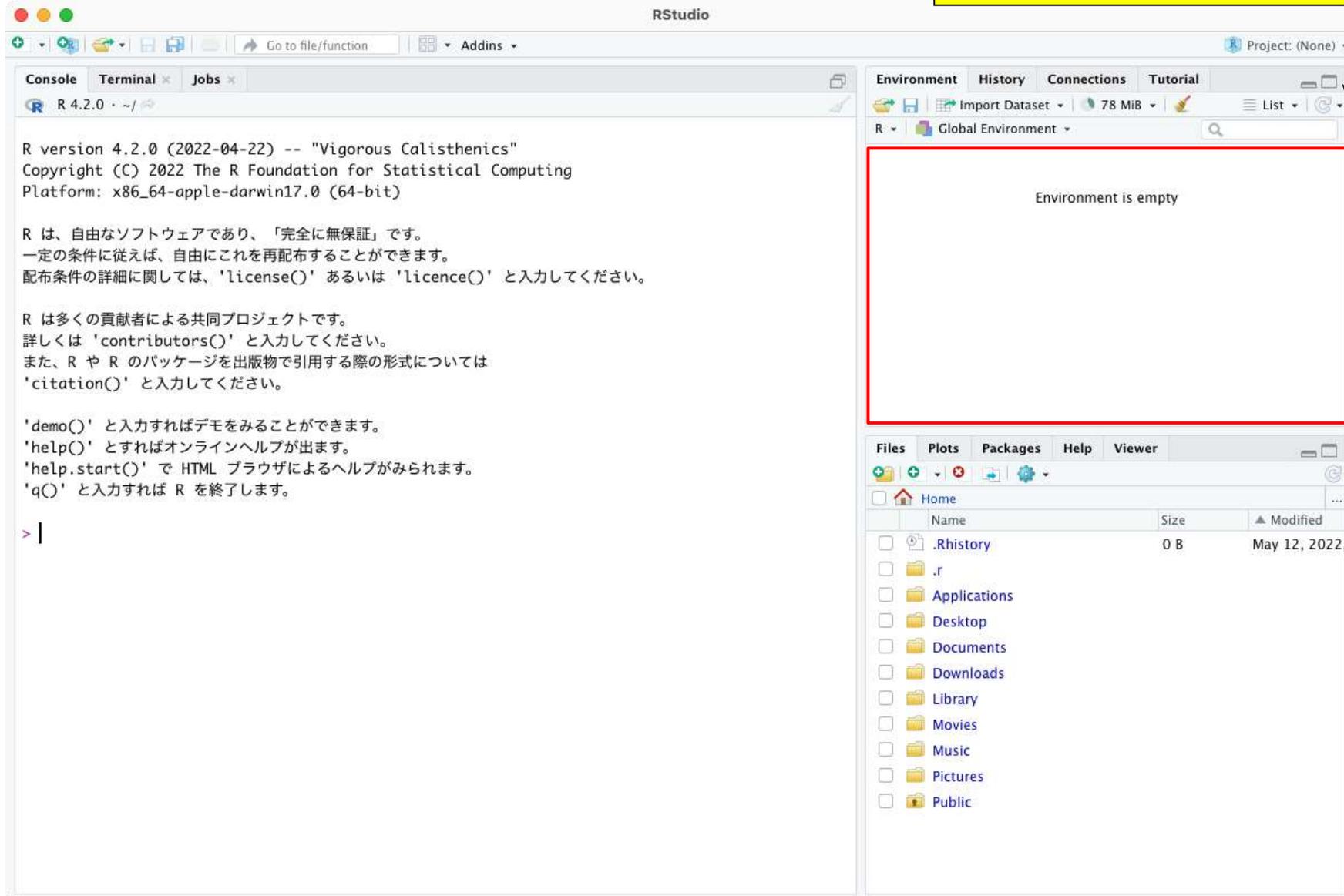
Name	Size	Modified
.Rhistory	0 B	May 12, 2022
.r		
Applications		
Desktop		
Documents		
Downloads		
Library		
Movies		
Music		
Pictures		
Public		

# Contents

- 前提条件、RStudioの起動
- 見栄えの統一、終了
- Console画面の見栄え、設定変更
- 枠の幅、電卓的な利用
- プロンプト(>)、Console画面クリア
- 計算の中断
- 指定した範囲の配列を取得
  - 項目と例題、入力ファイルの保存
  - 作業ディレクトリの変更
  - 入力ファイルの確認、FASTA形式
  - コピペ実行、出力ファイルの確認、Environmentタブ
- フィルタリング(任意のキーワードを含む行を抽出)
  - 全体像の把握、入力ファイルの保存、コピペ実行、出力ファイルの確認
  - 応用
- 警告メッセージ、ありがちなミス

# 見栄えの統一1

RStudio起動後の見え方はヒトによって異なる。赤枠内が①こんな感じになっているヒトもいれば、



# 見栄えの統一2

RStudio起動後の見え方はヒトによって異なる。赤枠内が①こんな感じになっているヒトもいれば、こんな感じでゴチャゴチャ文字が見えているヒトもいます。

The screenshot displays the RStudio interface with two side-by-side views of an R script. The left view shows a clean, consistent appearance with a red box highlighting the code. The right view shows a cluttered appearance with a red box highlighting the same code, but with additional text and formatting elements visible.

```
1 param <- "TxDb.Rnorvegicus.UCSC.rn5.refGene"#パッケージ名を指定
2
3 #前処理(BiocManagerがなければインストール)
4 if (!requireNamespace("BiocManager", quietly=T))#BiocManagerパッケージがイン
5   install.packages("BiocManager") #BiocManagerをインストールせよ
6
7 #本番
8 BiocManager::install(param, update=F) #paramで指定したパッケージのインスト
9
10 #後処理(ウェブサイト上で見えているバージョンとの比較)
11 BiocManager::valid(param) #TRUEはOK、too newでもまあOK、out-of-
12 packageVersion(param) #バージョン情報を表示
```

The console output on the left shows the following text:

```
配布条件の詳細に関しては、'license()' あるいは 'licence()' と入力してください。
R は多くの貢献者による共同プロジェクトです。
詳しくは '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ついては
'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す。
'q()' と入力すれば R を終了します。
> param <- "TxDb.Rnorvegicus.UCSC.rn5.refGene"#パッケージ名を指定
> #前処理(BiocManagerがなければインストール)
> if (!requireNamespace("BiocManager", quietly=T))#BiocManagerパッケージがインストール
   されてなければ...
+ install.packages("BiocManager") #BiocManagerをインストールせよ
> |
```

# 見栄えの統一3

①×を押すと、Saveするかどうか聞かれるので②Don't Save。この意味がわかっていて、Saveしたいヒトは自己責任で自由に。

The screenshot shows the RStudio interface with a dialog box titled "Untitled1 - Unsaved Changes". The dialog box contains the text: "The document 'Untitled1' has unsaved changes. Do you want to save these changes?". There are three buttons: "Cancel", "Don't Save", and "Save". A red arrow labeled "1" points to the "Cancel" button, and another red arrow labeled "2" points to the "Don't Save" button. The background shows R code in a script editor, including comments in Japanese and R commands for installing and validating the BiocManager package. The console output shows the results of the installation and validation process.

```
2  
3 #前処理(BiocManagerがなければインストール)  #BiocManagerをインストールせよ  
4 if (!requireNamespace("BiocManager", quietly=T))  
5   install.packages("BiocManager") #BiocManagerをインストールせよ  
6  
7 #本番  
8 BiocManager::install(param, update=F) #paramで指定したパッケージのインストール  
9  
10 #後処理(ウェブサイト上で見えているバージョンとの比較)  
11 BiocManager::valid(param) #TRUEはOK、too newでもまあOK、out-of-dateなら注意  
12 packageVersion(param) #バージョン情報を表示  
13
```

Console output:

```
*** installing help indices  
** building package indices  
** testing if installed package can be loaded from temporary location  
** testing if installed package can be loaded from final location  
** testing if installed package keeps a record of temporary installation path  
* DONE (TxDb.Rnorvegicus.UCSC.rn6.refGene)  
  
The downloaded source packages are in  
'/private/var/folders/43/lkb9vv4553b2lyjc98dryr5w0000gn/T/Rtmph2Pinb/downloaded_packages'  
>  
> #後処理(ウェブサイト上で見えているバージョンとの比較)  
> BiocManager::valid(param) #TRUEはOK、too newでもまあOK、out-of-dateなら注意  
[1] TRUE  
> packageVersion(param) #バージョン情報を表示  
[1] '3.4.6'  
>
```

## 見栄えの統一4

①×を押すと、Saveするかどうか聞かれるので②Don't Save。この意味がわかっていて、Saveしたいヒトは自己責任で自由に。ちなみにSaveにすると、③Encoding(文字コード)を聞かれることがあり、適宜選択の上OKして、④ファイル名をつけて⑤保存。

The screenshot illustrates the RStudio interface during a file save operation. The main editor window displays R code for installing and validating the BiocManager package. The console shows the output of these commands, including the installation of help indices and the successful validation of the package version (3.4.6). Two dialog boxes are overlaid on the interface: a 'Choose Encoding' dialog box and a 'Save File - Untitled1' dialog box. The 'Choose Encoding' dialog box is open, showing a list of encoding options with 'UTF-8 (System default)' selected. The 'Save File - Untitled1' dialog box is also open, showing the filename '20200313Test1.R' and the location 'kadota'. Red arrows point to the 'OK' button in the encoding dialog (3), the 'Save' button in the save dialog (5), and the 'Cancel' button in the save dialog (4).

# 見栄えの統一5

①×を押すと、Saveするかどうか聞かれるので②Don't Save。この意味がわかっていて、Saveしたいヒトは自己責任で自由に。ちなみにSaveにすると、③Encoding(文字コード)を聞かれることがあり、適宜選択の上OKして、④ファイル名をつけて⑤保存。こんな感じになります。左下の画面が上のほうに伸びて、左側全体を占拠したような感じになっていることが(直前のスライドとの比較でなんとなく)わかります。今回保存したファイルは、⑥ここ。

```
RStudio
Console Terminal Jobs
> BiocManager::install(param, update=F) #paramで指定したパッケージのインストール
Bioconductor version 3.10 (BiocManager 1.30.10), R 3.6.2 (2019-12-12)
Installing package(s) 'TxDb.Rnorvegicus.UCSC.rn6.refGene'
installing the source package 'TxDb.Rnorvegicus.UCSC.rn6.refGene'

URL 'https://bioconductor.org/packages/3.10/data/annotation/src/contrib/TxDb.Rnorvegicus.UCSC.rn6.refGene_3.4.6.tar.gz' を試しています
Content type 'application/x-gzip' length 7633538 bytes (7.3 MB)
-----
downloaded 7.3 MB

* installing *source* package 'TxDb.Rnorvegicus.UCSC.rn6.refGene' ...
** using staged installation
** R
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
** building package indices
** testing if installed package can be loaded from temporary location
** testing if installed package can be loaded from final location
** testing if installed package keeps a record of temporary installation path
* DONE (TxDb.Rnorvegicus.UCSC.rn6.refGene)

The downloaded source packages are in
'/private/var/folders/43/lkb9vv4553b21yjc98dryr5w0000gn/T/Rtmph2Pinb/downloaded_packages'
>
> #後処理(ウェブサイトで見えているバージョンとの比較)
> BiocManager::valid(param) #TRUEはOK、too newでもまあOK、out-of-dateなら注意
[1] TRUE
> packageVersion(param) #バージョン情報を表示
[1] '3.4.6'
> |
```

Environment Histor

To Conso  
packageVersion(par  
param <- "TxDb.Rno  
#前処理(BiocManager  
if (!requireNamesp  
install.packages(""  
#本番  
BiocManager::insta  
#後処理(ウェブサイト上  
BiocManager::valid  
packageVersion(par

Files Plots Packa

New Folder x D

Name	Size	Modified
.Rhistory	0 B	Mar 13, 2020, 12:07 PM
Applications		
Desktop		
Documents		
Downloads		
Library		
Movies		
Music		
Pictures		
Public		
VirtualBox VMs		
20200313Test1.R	686 B	Mar 13, 2020, 2:35 PM



## 見栄えの統一6

次に①箭のマークを押して、赤枠内にごちゃごちゃ書かれている(記載事項はヒトによって異なる)のをすっきりさせます。

The screenshot shows the RStudio interface. The console window on the left displays the output of the `BiocManager::install` function, including the download of the source package and the installation process. The Environment pane on the right shows the current environment with a red box highlighting the console output. A red arrow with the number 1 points to the 'To Console' button in the Environment pane.

```

> BiocManager::install(param, update=F) #paramで指定したパッケージのインストール
Bioconductor version 3.10 (BiocManager 1.30.10), R 3.6.2 (2019-12-12)
Installing package(s) 'TxDb.Rnorvegicus.UCSC.rn6.refGene'
installing the source package 'TxDb.Rnorvegicus.UCSC.rn6.refGene'

URL 'https://bioconductor.org/packages/3.10/data/annotation/src/contrib/TxDb.Rnorvegicus.UCSC.rn6.refGene_3.4.6.tar.gz' を試しています
Content type 'application/x-gzip' length 7633538 bytes (7.3 MB)
-----
downloaded 7.3 MB

* installing *source* package 'TxDb.Rnorvegicus.UCSC.rn6.refGene' ...
** using staged installation
** R
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
** building package indices
** testing if installed package can be loaded from temporary location
** testing if installed package can be loaded from final location
** testing if installed package keeps a record of temporary installation path
* DONE (TxDb.Rnorvegicus.UCSC.rn6.refGene)

The downloaded source packages are in
  '/private/var/folders/43/lkb9vv4553b21yjc98dryr5w0000gn/T/Rtmph2Pinb/downloaded_packages'
>
> #後処理(ウェブサイトで見えているバージョンとの比較)
> BiocManager::valid(param) #TRUEはOK、too newでもまあOK、out-of-dateなら注意
[1] TRUE
> packageVersion(param) #バージョン情報を表示
[1] '3.4.6'
>

```

## 見栄えの統一7

次に、①箇のマークを押して、赤枠内にごちゃごちゃ書かれている(記載事項はヒトによって異なる)のをすっきりさせます。本当に消して良いのか確認されるので、①Yesをクリックすると、

The screenshot shows the RStudio interface. The terminal window on the left displays the output of the `install.packages()` function, showing the installation of the `Txdb.Rnorvegicus.UCSC.rn6.refGene` package. The confirmation dialog box in the center asks "Confirm Clear History" and "Are you sure you want to clear all history entries?". The "Yes" button is highlighted with a red circle and the number 1, indicating the action to be taken.

**Terminal Output:**

```

> BiocManager::install(param, update=F) #param C
Bioconductor version 3.10 (BiocManager 1.30.10),
Installing package(s) 'TxDb.Rnorvegicus.UCSC.rn6
installing the source package 'TxDb.Rnorvegicus.UCSC.rn6.refGene

URL 'https://bioconductor.org/packages/3.10/data/annotation/src/contrib/Txdb.Rnor
vegicus.UCSC.rn6.refGene_3.4.6.tar.gz' を試しています
Content type 'application/x-gzip' length 7633538 bytes (7.3 MB)
-----
downloaded 7.3 MB

* installing *source* package 'TxDb.Rnorvegicus.UCSC.rn6.refGene' ...
** using staged installation
** R
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
** building package indices
** testing if installed package can be loaded from temporary location
** testing if installed package can be loaded from final location
** testing if installed package keeps a record of temporary installation path
* DONE (TxDb.Rnorvegicus.UCSC.rn6.refGene)

The downloaded source packages are in
  '/private/var/folders/43/1kb9vv4553b21yjc98dryr5w0000gn/T/Rtmph2Pinb/downl
oaded_packages'
>
> #後処理(ウェブサイトで見えているバージョンとの比較)
> BiocManager::valid(param) #TRUEはOK、too newでもまあOK、out-of-dateな
ら注意
[1] TRUE
> packageVersion(param) #バージョン情報を表示
[1] '3.4.6'
>

```

**Confirmation Dialog:**

Confirm Clear History  
Are you sure you want to clear all history entries?

Buttons: No, Yes (highlighted with ①)

**Files Panel:**

Name	Size	Modified
.Rhistory	0 B	Mar 13, 2020, 12:07 PM
20200313Test1.R	686 B	Mar 13, 2020, 2:35 PM

## 見栄えの統一8

次に、①のマークを押して、赤枠内にごちゃごちゃ書かれている(記載事項はヒトによって異なる)のをすっきりさせます。本当に消して良いのか確認されるので、①Yesをクリックすると、消えました。

```

RStudio
Go to file/function
Addins
Console Terminal Jobs
> BiocManager::install(param, update=F) #paramで指定したパッケージのインストール
Bioconductor version 3.10 (BiocManager 1.30.10), R 3.6.2 (2019-12-12)
Installing package(s) 'TxDb.Rnorvegicus.UCSC.rn6.refGene'
installing the source package 'TxDb.Rnorvegicus.UCSC.rn6.refGene'

URL 'https://bioconductor.org/packages/3.10/data/annotation/src/contrib/TxDb.Rnorvegicus.UCSC.rn6.refGene_3.4.6.tar.gz' を試しています
Content type 'application/x-gzip' length 7633538 bytes (7.3 MB)
-----
downloaded 7.3 MB

* installing *source* package 'TxDb.Rnorvegicus.UCSC.rn6.refGene' ...
** using staged installation
** R
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
** building package indices
** testing if installed package can be loaded from temporary location
** testing if installed package can be loaded from final location
** testing if installed package keeps a record of temporary installation path
* DONE (TxDb.Rnorvegicus.UCSC.rn6.refGene)

The downloaded source packages are in
  '/private/var/folders/43/lkb9vv4553b21yjc98dryr5w0000gn/T/Rtmph2Pinb/downloaded_packages'
>
> #後処理(ウェブサイト上で見えているバージョンとの比較)
> BiocManager::valid(param) #TRUEはOK、too newでもまあOK、out-of-dateなら注意
[1] TRUE
> packageVersion(param) #バージョン情報を表示
[1] '3.4.6'
>
  
```

Environment History

Files Plots Packages Help Viewer

New Folder Delete Rename More

Home

Name	Size	Modified
.Rhistory	0 B	Mar 13, 2020, 12:07 PM
Applications		
Desktop		
Documents		
Downloads		
Library		
Movies		
Music		
Pictures		
Public		
VirtualBox VMs		
20200313Test1.R	686 B	Mar 13, 2020, 2:35 PM

# RStudioの終了

普通のソフトウェアと同様、①で終了。もし①×を押した後に、「Save workspace image to...?」的なことを聞かれたら、② Don't Save。

The screenshot shows the RStudio interface with a 'Quit R Session' dialog box open. The dialog box asks 'Save workspace image to ~/.RData?' and has three buttons: 'Cancel', 'Don't Save', and 'Save'. A red arrow labeled '1' points to the dialog box, and another red arrow labeled '2' points to the 'Don't Save' button. The terminal window shows the installation of the 'TxDb.Rnorvegicus.UCSC.rn6.refGene' package. The terminal output includes the following text:

```
> BiocManager::install(param, update=r) #para
Bioconductor version 3.10 (BiocManager 1.30.10)
Installing package(s) 'TxDb.Rnorvegicus.UCSC.r
installing the source package 'TxDb.Rnorvegicus.UCSC.rn6.refGene'

URL 'https://bioconductor.org/packages/3.10/data/annotation/src/contrib/TxDb.Rnor
vegicus.UCSC.rn6.refGene_3.4.6.tar.gz' を試しています
Content type 'application/x-gzip' length 7633538 bytes (7.3 MB)
-----
downloaded 7.3 MB

* installing *source* package 'TxDb.Rnorvegicus.UCSC.rn6.refGene' ...
** using staged installation
** R
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
** building package indices
** testing if installed package can be loaded from temporary location
** testing if installed package can be loaded from final location
** testing if installed package keeps a record of temporary installation path
* DONE (TxDb.Rnorvegicus.UCSC.rn6.refGene)

The downloaded source packages are in
  '/private/var/folders/43/lkb9vv4553b21yjc98dryr5w0000gn/T/Rtmph2Pinb/downl
oaded_packages'
>
> #後処理(ウェブサイト上で見えているバージョンとの比較)
> BiocManager::valid(param) #TRUEはOK、too newでもまあOK、out-of-dateな
ら注意
[1] TRUE
> packageVersion(param) #バージョン情報を表示
[1] '3.4.6'
>
```

# Contents

- 前提条件、RStudioの起動
- 見栄えの統一、終了
- Console画面の見栄え、設定変更
- 枠の幅、電卓的な利用
- プロンプト(>)、Console画面クリア
- 計算の中断
- 指定した範囲の配列を取得
  - 項目と例題、入力ファイルの保存
  - 作業ディレクトリの変更
  - 入力ファイルの確認、FASTA形式
  - コピペ実行、出力ファイルの確認、Environmentタブ
- フィルタリング(任意のキーワードを含む行を抽出)
  - 全体像の把握、入力ファイルの保存、コピペ実行、出力ファイルの確認
  - 応用
- 警告メッセージ、ありがちなミス

# Console画面の見栄え1

おそらくまだ、①Console画面の②赤枠部分の文字の大きさや見栄えが異なっているヒトがいると思います。RStudioのGUI画面を③の方向で右下に広げるなどしてみてください。

The screenshot shows the RStudio interface. The Console window is highlighted with a red border. A red arrow labeled '1' points to the top-left corner of the RStudio window. Another red arrow labeled '2' points to the Console window's title bar. A third red arrow labeled '3' points to the bottom-right corner of the RStudio GUI. The Console window displays the following text:

```
R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

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

R は多くの貢献者による共同プロジェクトです。
詳しくは '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ついては
'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す。
'q()' と入力すれば R を終了します。

> |
```

The Environment window on the right shows "Environment is empty". The Files pane at the bottom shows a file explorer view of the Home directory.

# Console画面の見栄え2

おそらくまだ、①Console画面の②赤枠部分の文字の大きさや見栄えが異なっているヒトがいると思います。RStudioのGUI画面を③の方向で右下に広げるなどしてみてください。



# 設定変更1

まずは①赤枠内の文字サイズの変更方法から。

The screenshot shows the RStudio interface. The console window is highlighted with a red border and a red arrow labeled '1' pointing to it. The console displays the R version 4.2.0 startup message and Japanese text regarding the R license and help options. The environment window on the right shows an empty environment. The file browser at the bottom shows the home directory with various folders and files.

```
R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

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

R は多くの貢献者による共同プロジェクトです。
詳しくは '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ついては
'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す。
'q()' と入力すれば R を終了します。

>
```

Environment: 39 MiB, Global Environment

Files: Home

	Name	Size	Modified
<input type="checkbox"/>	.Rhistory	13.6 KB	May 31, 2022, 4:34 PM
<input type="checkbox"/>	.r		
<input type="checkbox"/>	Applications		
<input type="checkbox"/>	Desktop		
<input type="checkbox"/>	Documents		
<input type="checkbox"/>	Downloads		
<input type="checkbox"/>	Library		
<input type="checkbox"/>	Movies		
<input type="checkbox"/>	Music		
<input type="checkbox"/>	Pictures		
<input type="checkbox"/>	Public		

# 設定変更2

まずは①赤枠内の文字サイズの変更方法から。②Toolsメニューの、③Global Options。



The screenshot shows the RStudio interface. The console window is highlighted with a red border and contains the following text:

```
R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

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

R は多くの貢献者による共同プロジェクトです。
詳しくは '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ついては
'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す。
'q()' と入力すれば R を終了します。

>
```

The Tools menu is open, showing various options. The 'Global Options...' option is highlighted with a blue background and a red arrow pointing to it.

The file explorer shows the following files and folders:

Name	Size	Modified
.Rhistory	13.6 KB	May 31, 2022, 4:34 PM
.r		
Applications		
Desktop		
Documents		
Downloads		
Library		
Movies		
Music		
Pictures		
Public		

# 設定変更3

まずは①赤枠内の文字サイズの変更方法から。②Toolsメニューの、③Global Options。こんな感じになります。④これらが見えていればOK。

The screenshot shows the RStudio Options dialog box with the 'Basic' tab selected. The 'R Sessions' section has the following settings:

- Default working directory (when not in a project): ~
- Restore most recently opened project at startup
- Restore previously open source documents at startup

The 'Workspace' section has the following settings:

- Restore .RData into workspace at startup
- Save workspace to .RData on exit: Ask

The 'History' section has the following settings:

- Always save history (even when not saving .RData)
- Remove duplicate entries in history

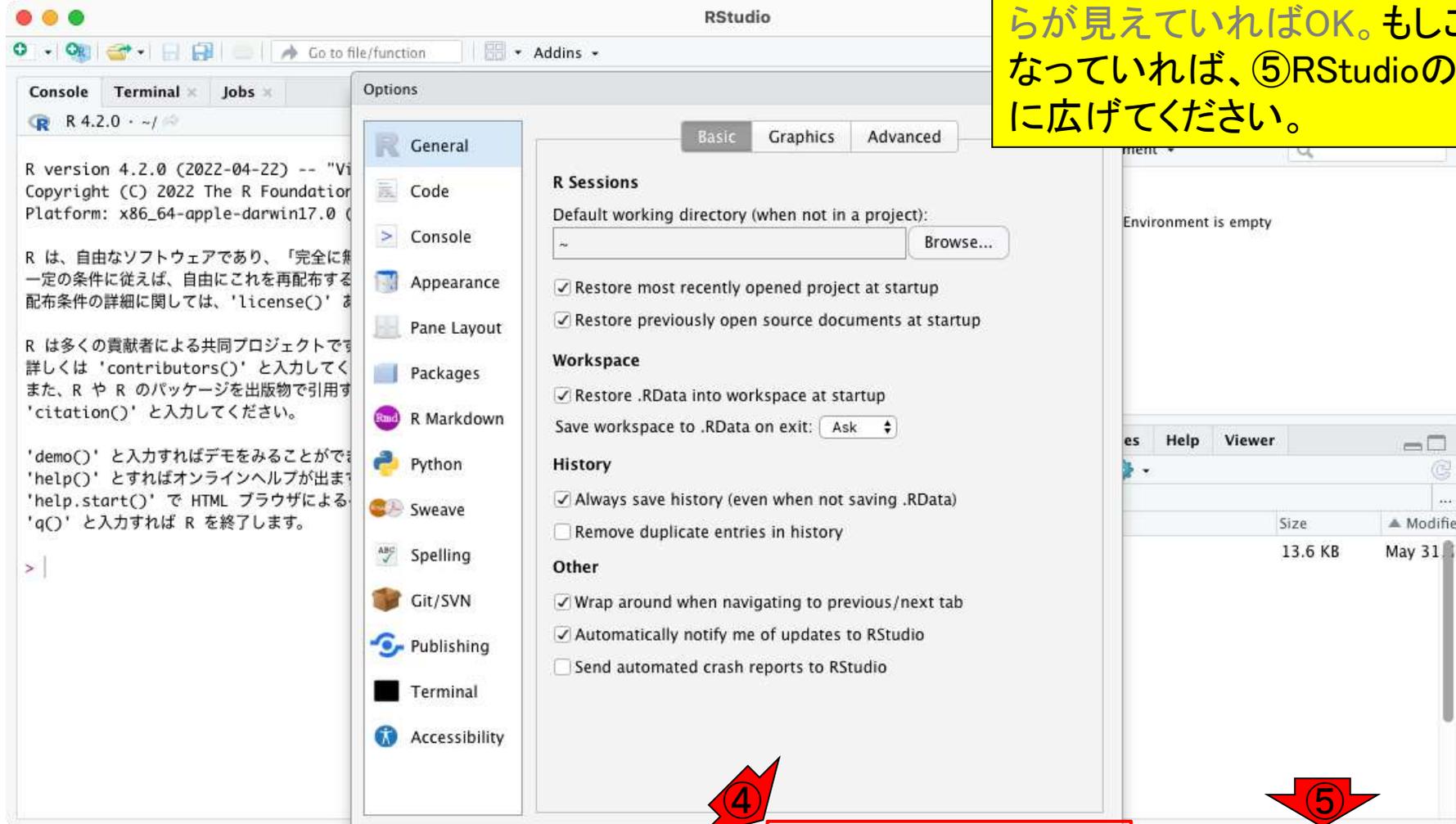
The 'Other' section has the following settings:

- Wrap around when navigating to previous/next tab
- Automatically notify me of updates to RStudio
- Send automated crash reports to RStudio

A red arrow with the number '4' points to the 'OK' button, which is also highlighted with a red box.

# 設定変更4

まずは①赤枠内の文字サイズの変更方法から。②Toolsメニューの、③Global Options。こんな感じになります。④これらが見えていればOK。もしこんな感じになっていれば、⑤RStudioのGUI画面を下に広げてください。

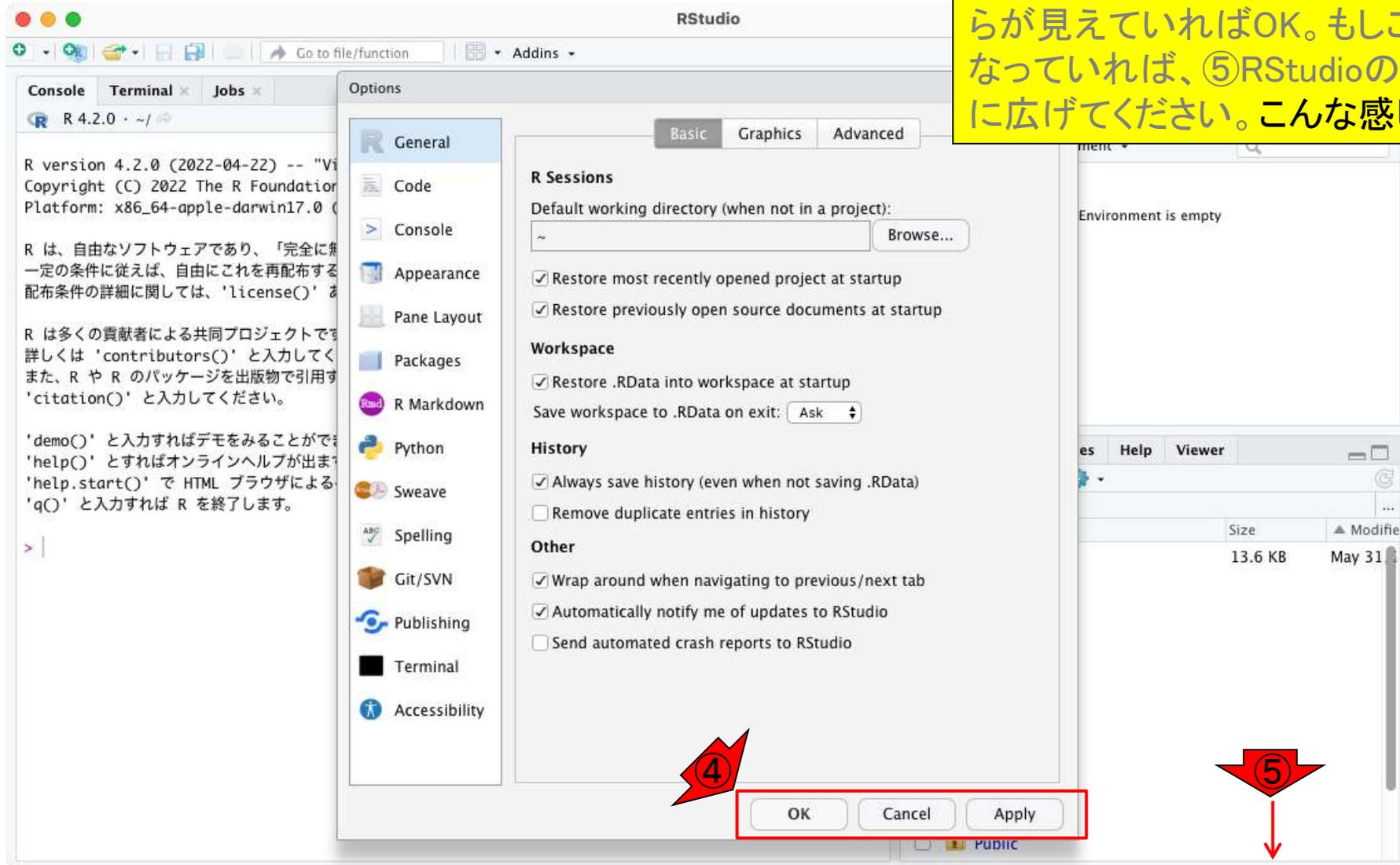


④

⑤

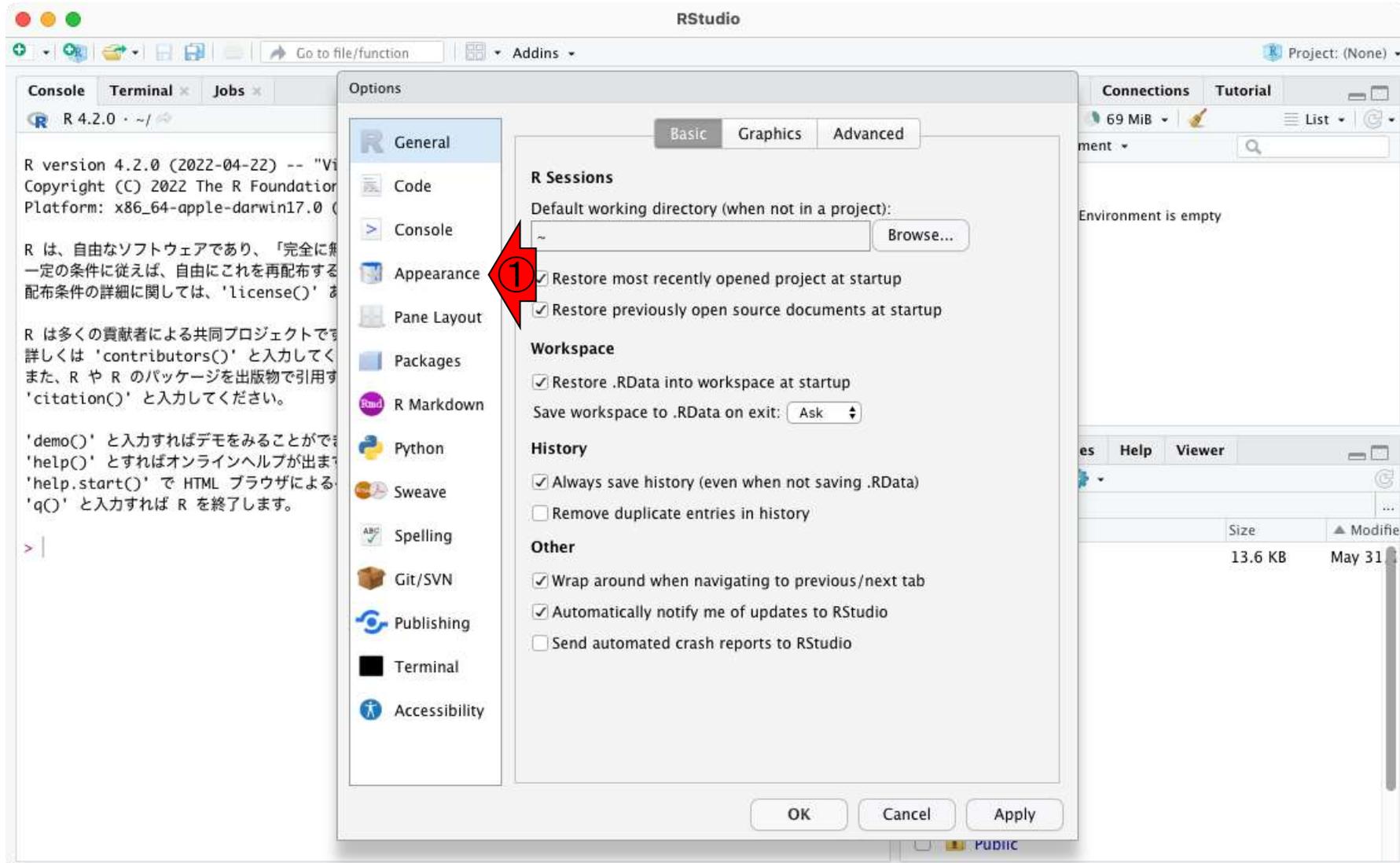
# 設定変更5

まずは①赤枠内の文字サイズの変更方法から。②Toolsメニューの、③Global Options。こんな感じになります。④これらが見えていればOK。もしこんな感じになっていれば、⑤RStudioのGUI画面を下に広げてください。こんな感じです。



# 設定変更6

文字サイズ変更は、①Appearanceのところで可能です。



# 設定変更7

本題の文字サイズ変更は、① Appearanceのところ可能です。② Editor font sizeでサイズ変更を行います。確かデフォルトは10だったと思います。

The screenshot shows the RStudio Options dialog box with the 'Appearance' tab selected. Two red arrows point to specific settings: arrow 1 points to the 'Appearance' tab, and arrow 2 points to the 'Editor font size' dropdown menu, which is currently set to 10. The 'Editor font size' is highlighted in blue. The background shows the RStudio interface with a code editor and console.

```
# plotting of R objects
plot <- function(x, y, ...)
{
  if (is.function(x) &&
      is.null(attr(x, "class")))
  {
    if (missing(y))
      y <- NULL

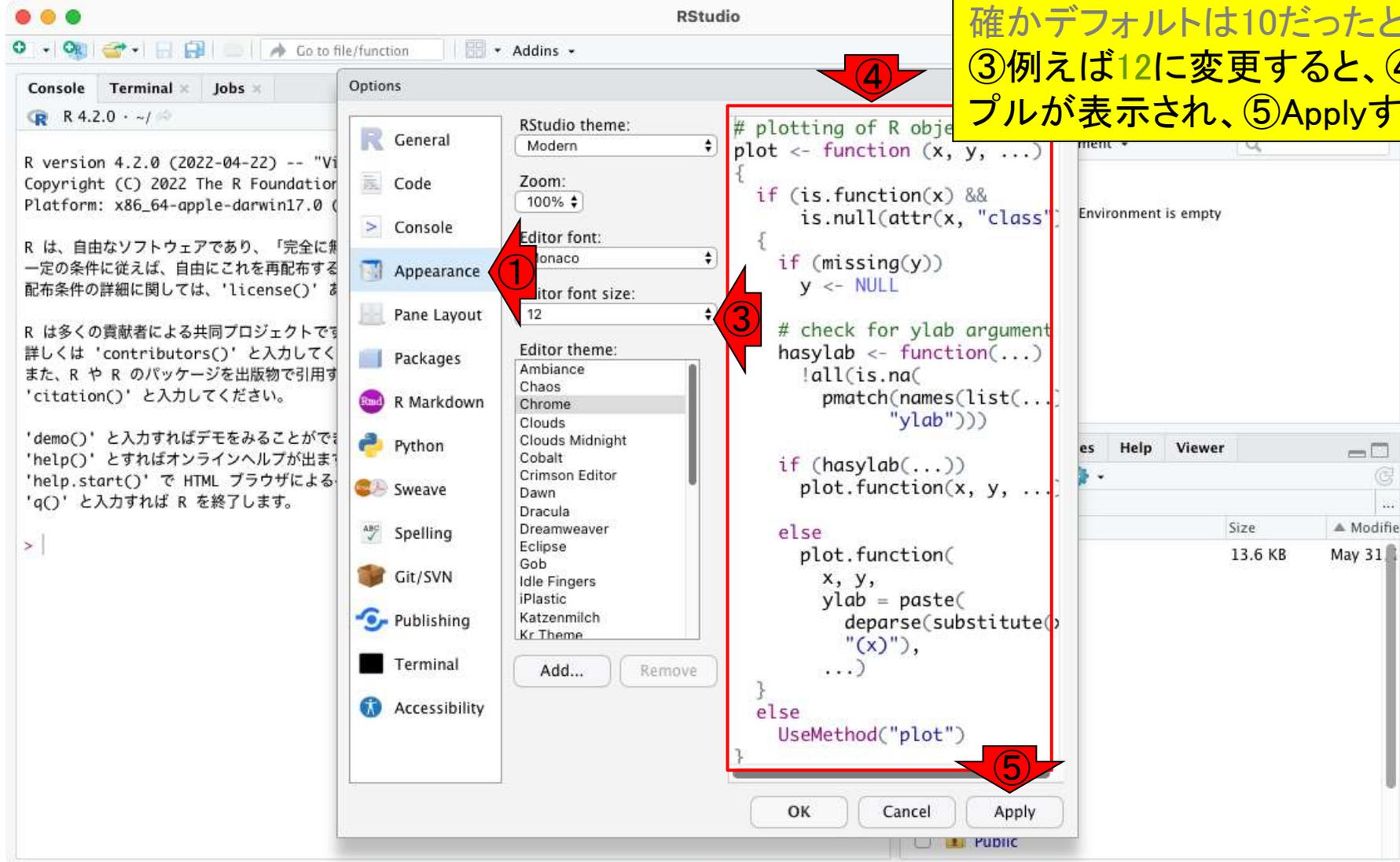
    # check for ylab argument
    hasylab <- function(...)
      !all(is.na(
        pmatch(names(list(...)),
                "ylab")))

    if (hasylab(...))
      plot.function(x, y, ...)

    else
      plot.function(
        x, y,
        ylab = paste(
          deparse(substitute(x)),
          "(x)",
          ...)
      )
  }
  else
    UseMethod("plot")
}
```

# 設定変更8

本題の文字サイズ変更は、① Appearanceのところ可能です。② Editor font sizeでサイズ変更を行います。確かデフォルトは10だったと思います。③例えば12に変更すると、④右側にサンプルが表示され、⑤Applyすると、



# 設定変更9

本題の文字サイズ変更は、① Appearanceのところ可能です。② Editor font sizeでサイズ変更を行います。確かデフォルトは10だったと思います。③例えば12に変更すると、④右側にサンプルが表示され、⑤Applyすると、⑥Console画面と呼ばれる赤枠部分の文字が大きく、変更が適用されたことがわかります。こんな感じで、見やすい大きさに変更してご利用ください。⑦OK。

The screenshot shows the RStudio Options dialog box with the 'Appearance' tab selected. The 'Editor font size' is set to 12. The 'Editor font' is set to Monaco. The 'Editor theme' is set to Chrome. The 'Console' window is visible on the left, showing R version 4.2.0 and some Japanese text. The 'Code' editor is visible on the right, showing R code. The 'Apply' button is highlighted with a red arrow labeled 5, and the 'OK' button is highlighted with a red arrow labeled 7. A red arrow labeled 6 points to the Console window, and a red arrow labeled 4 points to the Editor font size dropdown.

Console: R version 4.2.0 (2022-04-22)  
Copyright (C) 2022 The R Foundation  
Platform: x86\_64-apple-darwin20.0.0

R は、自由なソフトウェアであり、  
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、'licens

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

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

```
# plotting of R object  
plot <- function(x, y) {  
  if (is.function(x) &&  
      is.null(attr("ylab"))) {  
    if (missing(y))  
      y <- NULL  
    # check for ylab argument  
    hasylab <- function(...) {  
      !all(is.na(  
        pmatch(names(list(...)),  
                 "ylab")))  
    }  
    if (hasylab(...))  
      plot.function(x, y, ...)  
    else  
      plot.function(  
        x, y,  
        ylab = paste(  
          deparse(substitute(x)),  
          "(x)",  
          ...)  
      )  
    }  
  }  
  else  
    UseMethod("plot")  
}
```

# 設定変更10

ここまでで、文字の大きさを調整できるようになりました。本質的な部分ではありませんが、様々な年代・環境の方が利用しますので、配慮が必要な事柄です。

The screenshot shows the RStudio interface. The console pane on the left displays the R version information and usage instructions. The environment pane on the right shows that the environment is empty. The file browser at the bottom shows the file system structure.

**Console**

```
R 4.2.0 · ~/
```

R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"  
Copyright (C) 2022 The R Foundation for Statistical Computing  
Platform: x86\_64-apple-darwin17.0 (64-bit)

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

R は多くの貢献者による共同プロジェクトです。  
詳しくは '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ついては  
'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す。  
'q()' と入力すれば R を終了します。

>

**Environment** History Connections Tutorial

Import 70 MiB List

R Global Environment

Environment is empty

**Files** Plots Packages Help Viewer

Home

	Name	Size	Modifie
<input type="checkbox"/>	.Rhistory	13.6 KB	May 31
<input type="checkbox"/>	.r		
<input type="checkbox"/>	Applications		
<input type="checkbox"/>	Desktop		
<input type="checkbox"/>	Documents		
<input type="checkbox"/>	Downloads		
<input type="checkbox"/>	Library		
<input type="checkbox"/>	Movies		
<input type="checkbox"/>	Music		
<input type="checkbox"/>	Pictures		
<input type="checkbox"/>	Public		

# Contents

- 前提条件、RStudioの起動
- 見栄えの統一、終了
- Console画面の見栄え、設定変更
- 枠の幅、電卓的な利用
- プロンプト(>)、Console画面クリア
- 計算の中断
- 指定した範囲の配列を取得
  - 項目と例題、入力ファイルの保存
  - 作業ディレクトリの変更
  - 入力ファイルの確認、FASTA形式
  - コピペ実行、出力ファイルの確認、Environmentタブ
- フィルタリング(任意のキーワードを含む行を抽出)
  - 全体像の把握、入力ファイルの保存、コピペ実行、出力ファイルの確認
  - 応用
- 警告メッセージ、ありがちなミス

# 枠の幅1

RStudioは、ここで見えているだけでも画面が3分割されていることがわかります。それぞれの画面の高さや幅を変更することができます。

The screenshot shows the RStudio interface with three main panes highlighted by red boxes:

- Console (left pane):** Displays the R version (4.2.0), copyright information, and Japanese instructions for using R, including commands like `license()`, `licence()`, `contributors()`, `citation()`, `demo()`, `help()`, `help.start()`, and `q()`. The prompt `> |` is visible at the bottom.
- Environment (top right pane):** Shows "Environment is empty".
- Files (bottom right pane):** Shows a file browser view of the Home directory, listing folders like `.Rhistory`, `.r`, `Applications`, `Desktop`, `Documents`, `Downloads`, `Library`, `miniconda2`, `miniconda3`, `Movies`, and `Music`.

# 枠の幅2

例えば、①や②のところにカーソルを持っていき、右に移動させると…

The screenshot shows the RStudio interface. The console on the left displays the R version information and some introductory text. The environment pane on the right shows an empty environment. The file browser at the bottom right shows the file system structure. Two red arrows with circled numbers point to specific elements: arrow 1 points to the 'Environment' pane, and arrow 2 points to the file browser.

R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"  
Copyright (C) 2022 The R Foundation for Statistical Computing  
Platform: x86\_64-apple-darwin17.0 (64-bit)

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

R は多くの貢献者による共同プロジェクトです。  
詳しくは '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ついては  
'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す。  
'q()' と入力すれば R を終了します。

> |

Environment History Connections Tutorial  
R 75 MiB List  
Global Environment  
Environment is empty

Files Plots Packages Help Viewer  
Home  
Name Size Modif  
[ ] .Rhistory 13.6 KB May 31  
[ ] .r  
[ ] Applications  
[ ] Desktop  
[ ] Documents  
[ ] Downloads  
[ ] Library  
[ ] miniconda2  
[ ] miniconda3  
[ ] Movies  
[ ] Music  
[ ] opt

# 枠の幅3

例えば、①や②のところにカーソルを持っていき、右に移動させると…③  
Console(こんそーる)画面の幅を広くすることができます。

R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"  
Copyright (C) 2022 The R Foundation for Statistical Computing  
Platform: x86\_64-apple-darwin17.0 (64-bit)

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

R は多くの貢献者による共同プロジェクトです。  
詳しくは '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ついては  
'citation()' と入力してください。

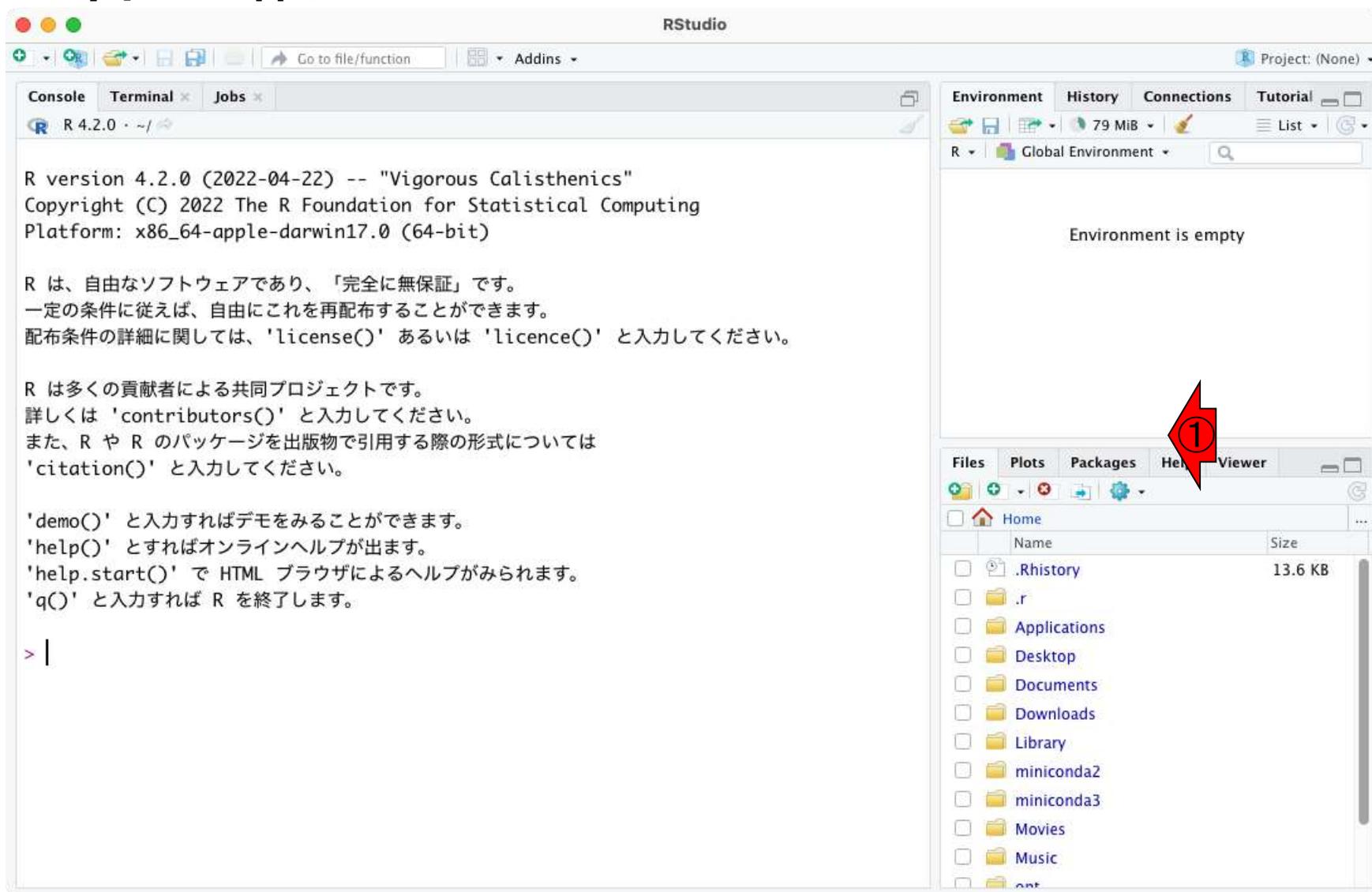
'demo()' と入力すればデモをみることができます。  
'help()' とすればオンラインヘルプが出ます。  
'help.start()' で HTML ブラウザによるヘルプがみられます。  
'q()' と入力すれば R を終了します。

> |

Environment History Connections Tutorial  
79 MiB  
Global Environment  
Environment is empty

Files Plots Packages Help Viewer  
Home  
Name Size  
.Rhistory 13.6 KB  
.r  
Applications  
Desktop  
Documents  
Downloads  
Library  
miniconda2  
miniconda3  
Movies  
Music  
ent

# 枠の幅4



The screenshot shows the RStudio interface. The console pane on the left displays the R version information and Japanese text. The environment pane on the right shows an empty environment. A red arrow with the number 1 points to the 'Environment' tab in the top right pane.

R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"  
Copyright (C) 2022 The R Foundation for Statistical Computing  
Platform: x86\_64-apple-darwin17.0 (64-bit)

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

R は多くの貢献者による共同プロジェクトです。  
詳しくは '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ついては  
'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す。  
'q()' と入力すれば R を終了します。

> |

Environment History Connections Tutorial  
79 MiB List  
Global Environment  
Environment is empty

Files Plots Packages Help Viewer  
Home  
Name Size  
[ ] .Rhistory 13.6 KB  
[ ] .r  
[ ] Applications  
[ ] Desktop  
[ ] Documents  
[ ] Downloads  
[ ] Library  
[ ] miniconda2  
[ ] miniconda3  
[ ] Movies  
[ ] Music  
[ ] .ant

# 枠の幅5

①の部分も同様に上下させることで、右側の画面の上下幅を変更することができます。状況に応じて見やすいように調整してください。

The screenshot shows the RStudio interface. The console pane on the left displays the R version information and some introductory text in Japanese. The environment pane on the right shows an empty environment. A red arrow with the number 1 points to the separator between the console and environment panes, indicating that this separator can be moved to adjust the width of the environment pane.

R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"  
Copyright (C) 2022 The R Foundation for Statistical Computing  
Platform: x86\_64-apple-darwin17.0 (64-bit)

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

R は多くの貢献者による共同プロジェクトです。  
詳しくは '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ついては  
'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す。  
'q()' と入力すれば R を終了します。

> |

Environment History Connections Tutorial  
79 MiB  
Global Environment  
Environment is empty

Files Plots Packages Help Viewer  
Home  
Name Size  
[ ] .Rhistory 13.6 KB  
[ ] .r  
[ ] Applications  
[ ] Desktop  
[ ] Documents  
[ ] Downloads  
[ ] Library  
[ ] miniconda2  
[ ] miniconda3  
[ ] Movies  
[ ] Music  
[ ] opt  
[ ] Pictures  
[ ] Public

# 電卓的な利用1

①Console画面上で、②「1+1」と打ち込んでからリターンキーを押したところ。2という結果が表示されていることがわかります。

The screenshot shows the RStudio interface. The top-left pane is the Console, which displays the R version information and instructions. The bottom-left pane is the Terminal, where the command `> 1+1` has been entered, and the output `[1] 2` is shown. A red arrow labeled '2' points to the input line. The right-hand side of the interface shows the Environment pane with the expression `1+1` and its value `2`. Below the Environment pane is the Files pane, which shows the file system structure.

```
R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

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

R は多くの貢献者による共同プロジェクトです。
詳しくは '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ついては
'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す。
'q()' と入力すれば R を終了します。

> 1+1
[1] 2
> |
```

Environment

```
1+1
```

Files

Name	Size
.Rhistory	13.5 KB
.r	
Applications	
Desktop	
Documents	
Downloads	
Library	
miniconda2	
miniconda3	
Movies	
Music	
opt	
Pictures	
Public	

# 電卓的な利用2

①Console画面上で、②「1+1」と打ち込んでからリターンキーを押したところ。2という結果が表示されていることがわかります。今度は、③「10\*5」で、50という結果が表示されていることがわかります。

The screenshot shows the RStudio interface. The console window displays the R version information and instructions. The file explorer shows the home directory with various folders and files.

```
R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

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

R は多くの貢献者による共同プロジェクトです。
詳しくは '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ついては
'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す。
'q()' と入力すれば R を終了します。

> 1+1
[1] 2
> 10*5
[1] 50
> |
```

The file explorer shows the following files and folders:

Name	Size
.Rhistory	13.5 KB
.r	
Applications	
Desktop	
Documents	
Downloads	
Library	
miniconda2	
miniconda3	
Movies	
Music	
opt	
Pictures	
Public	

# Contents

- 前提条件、RStudioの起動
- 見栄えの統一、終了
- Console画面の見栄え、設定変更
- 枠の幅、電卓的な利用
- プロンプト(>)、Console画面クリア
- 計算の中断
- 指定した範囲の配列を取得
  - 項目と例題、入力ファイルの保存
  - 作業ディレクトリの変更
  - 入力ファイルの確認、FASTA形式
  - コピペ実行、出力ファイルの確認、Environmentタブ
- フィルタリング(任意のキーワードを含む行を抽出)
  - 全体像の把握、入力ファイルの保存、コピペ実行、出力ファイルの確認
  - 応用
- 警告メッセージ、ありがちなミス

# プロンプト (>)

ここまでの作業で、コマンドを実行すると、その結果が表示されるとともに、①「>」が表示されます。これを「プロンプト」と言います。これが見えていれば、「直前のコマンド実行が完了している」と解釈してよいです。

```
R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

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

```
R は多くの貢献者による共同プロジェクトです。
詳しくは '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ついては
'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す。
'q()' と入力すれば R を終了します。
```

```
> 1+1
[1] 2
> 10*5
[1] 50
> |
```



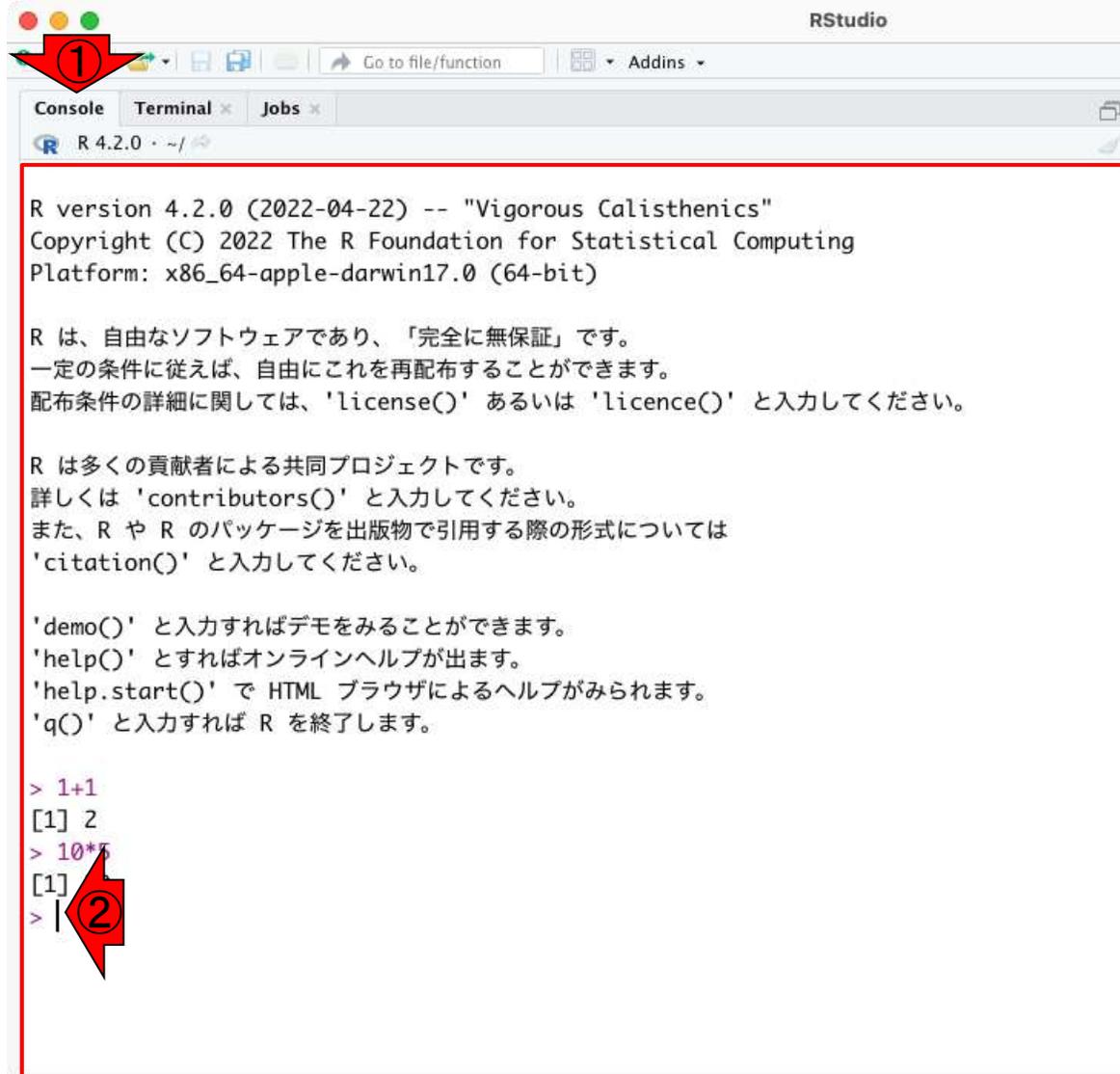
```
1+1
10*5
```

Files Plots Packages Help Viewer

Home

Name	Size
.Rhistory	13.5 KB
.r	
Applications	
Desktop	
Documents	
Downloads	
Library	
miniconda2	
miniconda3	
Movies	
Music	
opt	
Pictures	
Public	

# Console画面クリア1



R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"  
Copyright (C) 2022 The R Foundation for Statistical Computing  
Platform: x86\_64-apple-darwin17.0 (64-bit)

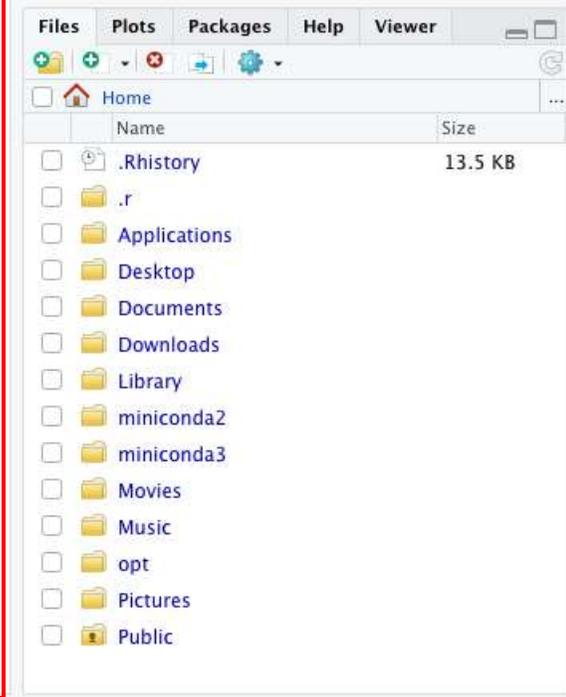
R は、自由なソフトウェアであり、「完全に無保証」です。  
一定の条件に従えば、自由にこれを再配布することができます。  
配布条件の詳細に関しては、'license()' あるいは 'licence()' と入力してください。

R は多くの貢献者による共同プロジェクトです。  
詳しくは '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ついては  
'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す。  
'q()' と入力すれば R を終了します。

```
> 1+1  
[1] 2  
> 10*5  
[1]  
> |
```

このあと、1行1行コマンドを打ち込んでいくと、常に①Console画面上の下部である、②のあたりに入力をするようになります。必要に応じて①Console画面をクリア(Clear Console)して、Console画面上部でコマンド入力できるようにするやり方を示します。



# Console画面クリア2

基本的には①を押すだけです。Editメニューから、②Clear Console。③キーボード上で「control」キーを押しながら「L」キーを押すのでもかまいません。

The screenshot shows the RStudio interface. The console window is open, displaying the R version (4.2.0) and some text. A menu is open over the console, showing options like Undo, Redo, Cut, Copy, Paste, and a highlighted 'Clear Console' option. A red box highlights the console area, and red arrows point to the 'Clear Console' button and the 'control+L' keyboard shortcut. The terminal window shows the output of the commands `1+1` and `10*5`.

R version 4.2.0  
Copyright (C) 2022  
Platform: x86\_64

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

R は多くの貢献者によって開発されています。  
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ついては  
'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す。  
'q()' と入力すれば R を終了します。

```
> 1+1  
[1] 2  
> 10*5  
[1] 50  
> |
```

# Console画面クリア3

基本的には①を押すだけです。Editメニューから、②Clear Console。③キーボード上で「control」キーを押しながら「L」キーを押すのでもかまいません。無事、④Console画面がクリアされました。

The screenshot shows the RStudio interface. The Console window is active and shows the prompt `>`. A red arrow with the number 4 points to the 'Clear Console' button in the top-left corner of the Console window. The right-hand pane shows the 'To Console' view with the output `1+1` and `10*5`. The bottom-right pane shows a file explorer view of the user's home directory.

Name	Size
.Rhistory	13.5 KB
.r	
Applications	
Desktop	
Documents	
Downloads	
Library	
miniconda2	
miniconda3	
Movies	
Music	
opt	
Pictures	
Public	

# Contents

- 前提条件、RStudioの起動
- 見栄えの統一、終了
- Console画面の見栄え、設定変更
- 枠の幅、電卓的な利用
- プロンプト(>)、Console画面クリア
- 計算の中断
- 指定した範囲の配列を取得
  - 項目と例題、入力ファイルの保存
  - 作業ディレクトリの変更
  - 入力ファイルの確認、FASTA形式
  - コピペ実行、出力ファイルの確認、Environmentタブ
- フィルタリング(任意のキーワードを含む行を抽出)
  - 全体像の把握、入力ファイルの保存、コピペ実行、出力ファイルの確認
  - 応用
- 警告メッセージ、ありがちなミス

# 計算の中断1

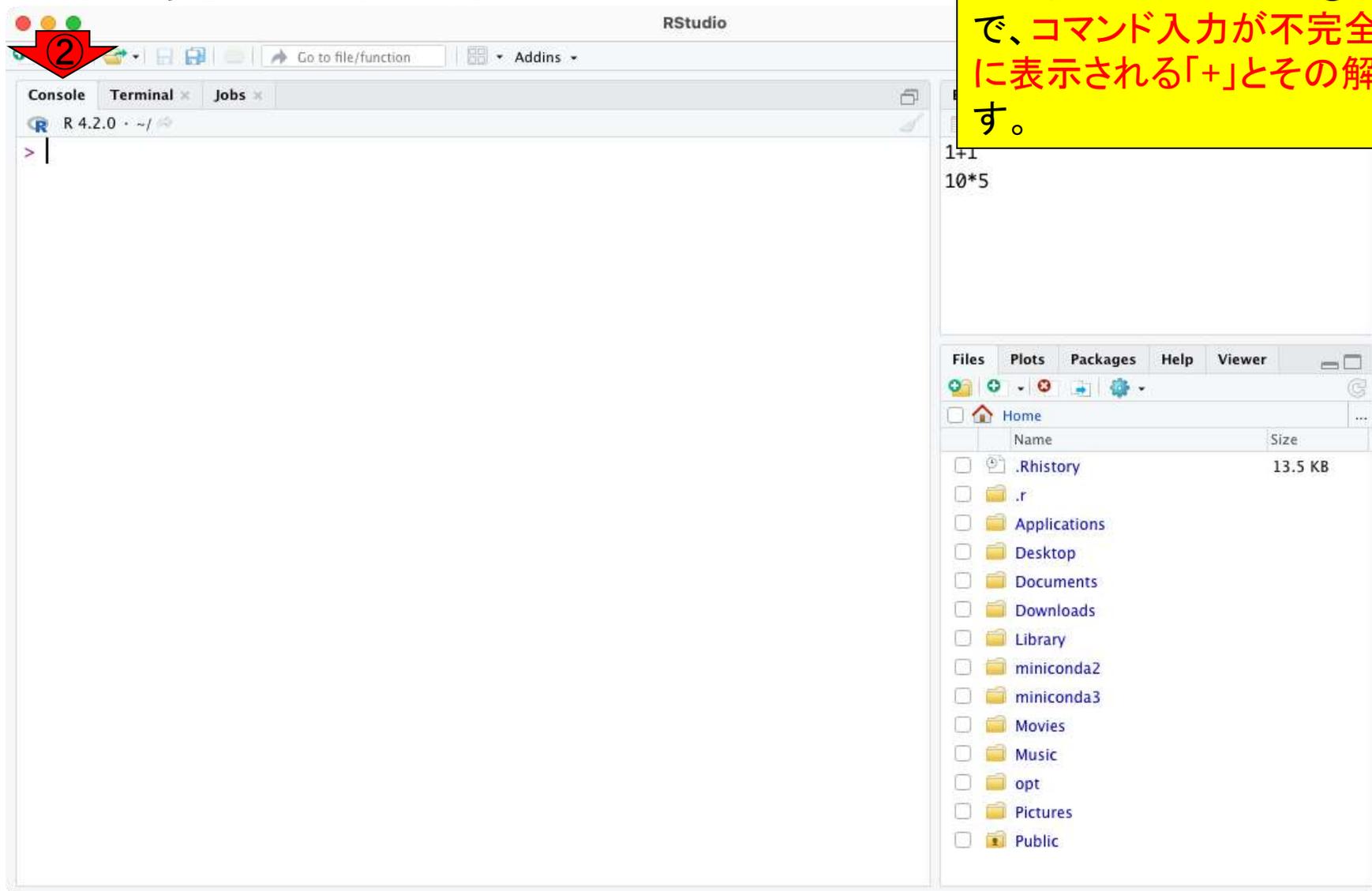
さきほどの①「control + L」は、「control キーを押しながらLキーを押す」という意味で使いました。

The screenshot shows the RStudio interface. The 'Console' tab is active, displaying the prompt '>'. A context menu is open over the console, listing various actions. The 'Clear Console' option at the bottom is highlighted in blue and has a red arrow with a circled '1' pointing to it. The 'Environment' pane on the right shows the results of the previous calculation: '1+1' and '10\*5'. The 'Files' pane at the bottom right shows the file explorer view of the user's home directory.

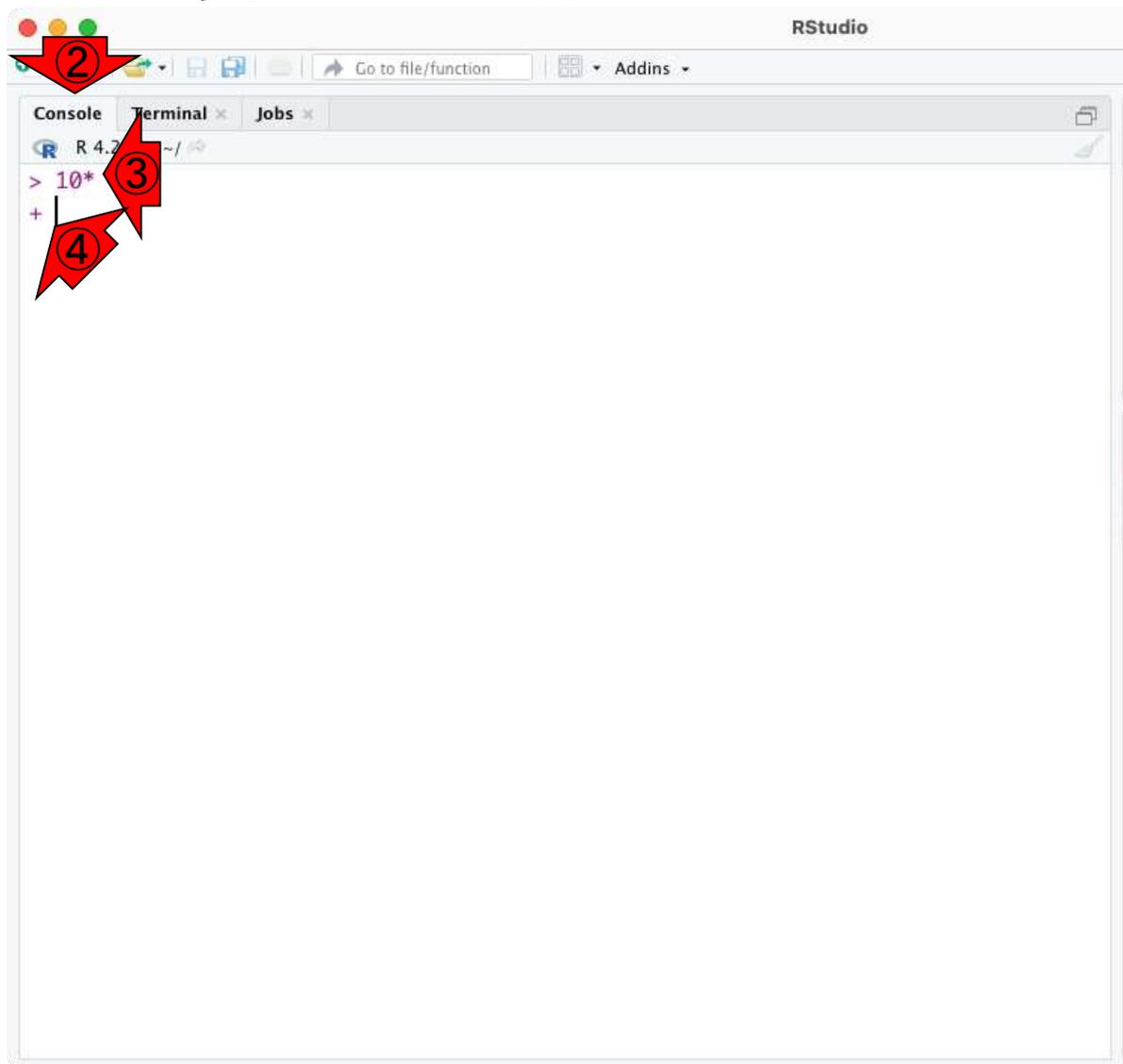
Name	Size
.Rhistory	13.5 KB
.r	
Applications	
Desktop	
Documents	
Downloads	
Library	
miniconda2	
miniconda3	
Movies	
Music	
opt	
Pictures	
Public	

# 計算の中断2

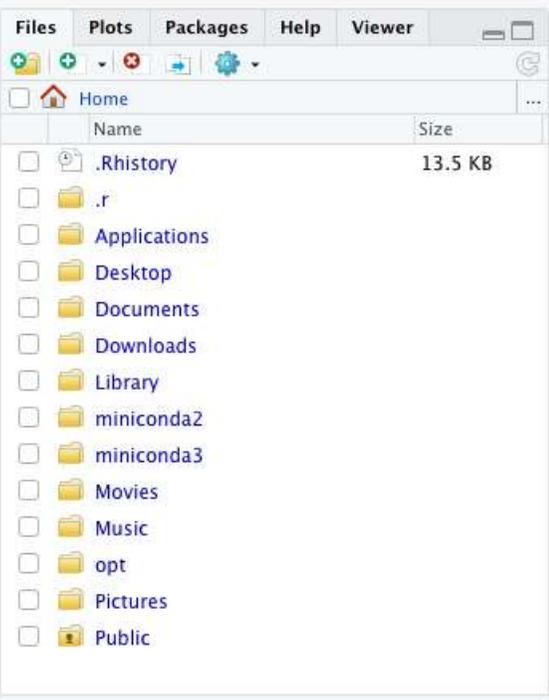
さきほどの①「control + L」は、「control キーを押しながらLキーを押す」という意味で使いました。次に②Console画面上で、コマンド入力が不完全な状態のときに表示される「+」とその解決策を示します。



# 計算の中断3



さきほどの①「control + L」は、「control キーを押しながらLキーを押す」という意味で使いました。次に②Console画面上で、コマンド入力が不完全な状態のときに表示される「+」とその解決策を示します。例として、「10\*5」と入力するつもりだったが、意図的に間違えて③「10\*」まででリターンキーを押してみましよう。④のようになればOK。



# 計算の中断4

非推奨ですが、どうにかしてこのままの状態  
で10\*5を計算したい場合は⑤「5」と  
打ち込んでリターンキーを押します。

The screenshot shows the RStudio interface. The console window on the left contains the text `> 10*` followed by a red arrow pointing to the asterisk, with a circled number 5 next to it. The Environment pane on the right shows the current state of the workspace with the following objects:

Object	Value
1+1	2
10*5	50
10*	10

The Files pane at the bottom right shows the file explorer for the user's home directory, listing various folders and files.

# 計算の中断5

非推奨ですが、どうにかしてこのままの状態  
で $10*5$ を計算したい場合は⑤「5」と  
打ち込んでリターンキーを押します。⑥  
無事「 $10*5$ 」の計算結果である50が表示  
され、⑦プロンプトが表示されていること  
がわかります。

The screenshot shows the RStudio interface with the console window open. The console displays the following text:

```
> 10*  
+ 5  
[1] 50  
> |  
>
```

Red arrows with circled numbers 6 and 7 point to the input '5' and the prompt '>' respectively. The right-hand pane shows a file explorer view of the user's home directory.

Name	Size
.Rhistory	13.5 KB
.r	
Applications	
Desktop	
Documents	
Downloads	
Library	
miniconda2	
miniconda3	
Movies	
Music	
opt	
Pictures	
Public	

# 計算の中断6

①もう一回同じ状況にします。「プロンプト(>)」ではなく、②「+」になってしまったときの推奨手順は、③「Esc」キー。または「control + C」。

The image shows the RStudio interface. The console on the left displays the following text:

```
> 10*  
+ 5  
[1] 50  
> 10*  
+
```

Red arrows labeled ①, ②, and ③ point to the prompt, the plus sign, and the Esc key on a keyboard, respectively. The keyboard also shows Japanese characters on the keys. The file explorer on the right shows the current directory with the following files and folders:

Name	Size
.Rhistory	13.5 KB
.r	
Applications	
Desktop	
Documents	
Downloads	
Library	
miniconda2	
miniconda3	
Movies	
Music	
opt	
Pictures	
Public	

# 計算の中断7

①もう一回同じ状況にする。「プロンプト(>)」ではなく、②「+」になってしまったときの推奨手順は、③「Esc」キー。または「control + C」。④「Esc」キーを押した結果。確かにプロンプト(>)が出ていますね。

The screenshot shows the RStudio interface. The console window displays the following text:

```
> 10*  
+ 5  
[1] 50  
> 10*  
+  
> | ④
```

A red lightning bolt icon with the number 4 inside is positioned next to the prompt '> |', indicating the point of interruption. The background of the console shows a list of files and folders, including .Rhistory, .r, Applications, Desktop, Documents, Downloads, Library, miniconda2, miniconda3, Movies, Music, opt, Pictures, and Public.

# 計算の中断8

The screenshot shows the RStudio interface. The console on the left contains the following text:

```
> 10*  
+ 5  
[1] 50  
> 10*  
+  
> |
```

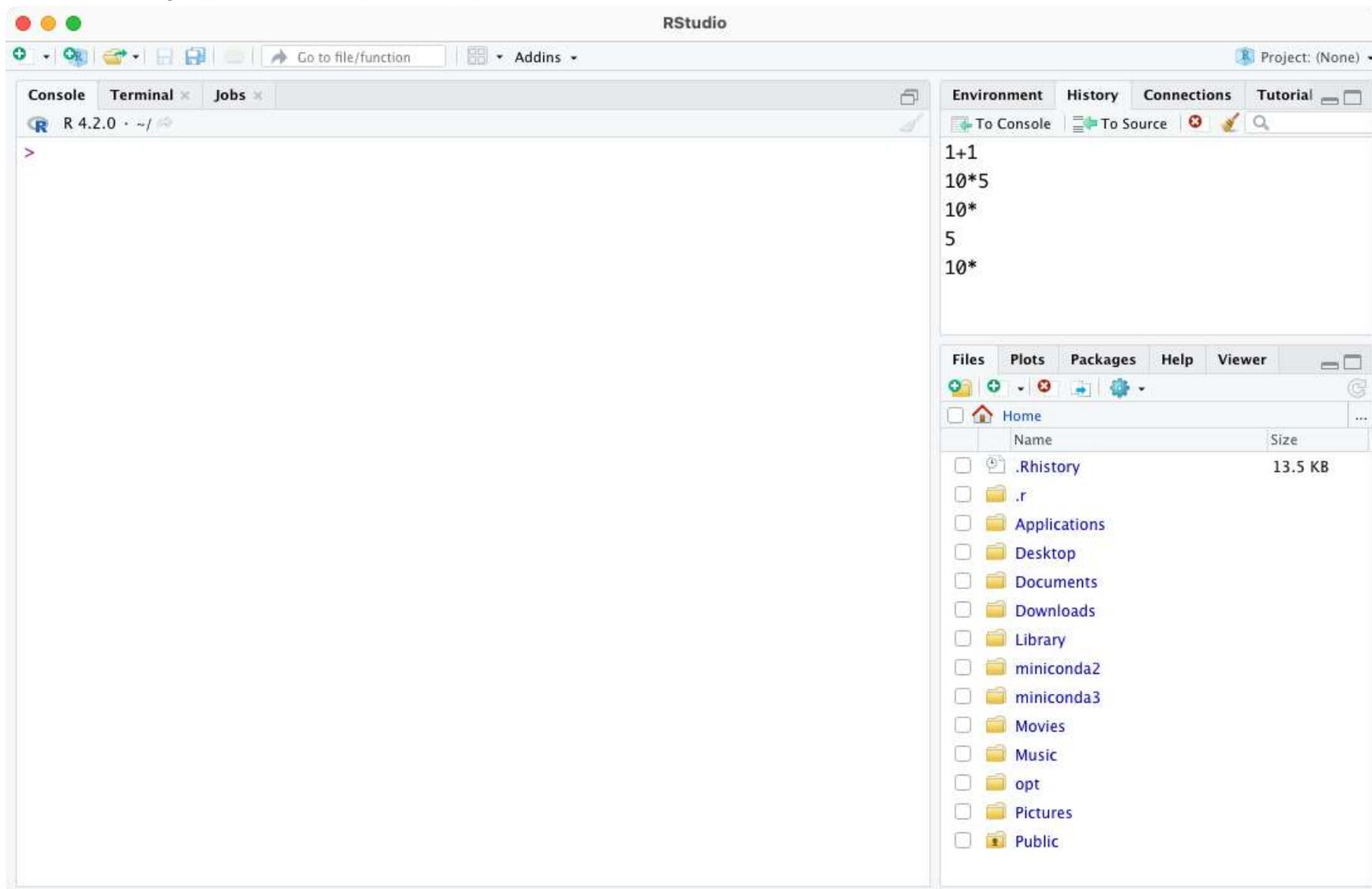
A red arrow with the number '1' inside points to the clear button (a trash can icon) in the top right corner of the console panel. The Environment pane on the right shows the following objects:

```
1+1  
10*5  
10*  
5  
10*
```

The Files pane at the bottom right shows the file explorer for the Home directory:

Name	Size
.Rhistory	13.5 KB
.r	
Applications	
Desktop	
Documents	
Downloads	
Library	
miniconda2	
miniconda3	
Movies	
Music	
opt	
Pictures	
Public	

# 計算の中断9



The screenshot shows the RStudio interface. The console on the left has a prompt `>`. The Environment panel on the right shows a list of calculations: `1+1`, `10*5`, `10*`, `5`, and `10*`. The Files panel at the bottom right shows the file explorer for the Home directory, listing various folders and files.

Name	Size
<input type="checkbox"/> .Rhistory	13.5 KB
<input type="checkbox"/> .r	
<input type="checkbox"/> Applications	
<input type="checkbox"/> Desktop	
<input type="checkbox"/> Documents	
<input type="checkbox"/> Downloads	
<input type="checkbox"/> Library	
<input type="checkbox"/> miniconda2	
<input type="checkbox"/> miniconda3	
<input type="checkbox"/> Movies	
<input type="checkbox"/> Music	
<input type="checkbox"/> opt	
<input type="checkbox"/> Pictures	
<input type="checkbox"/> Public	

# Contents

- 前提条件、RStudioの起動
- 見栄えの統一、終了
- Console画面の見栄え、設定変更
- 枠の幅、電卓的な利用
- プロンプト(>)、Console画面クリア
- 計算の中断
- 指定した範囲の配列を取得
  - 項目と例題、入力ファイルの保存
  - 作業ディレクトリの変更
  - 入力ファイルの確認、FASTA形式
  - コピペ実行、出力ファイルの確認、Environmentタブ
- フィルタリング(任意のキーワードを含む行を抽出)
  - 全体像の把握、入力ファイルの保存、コピペ実行、出力ファイルの確認
  - 応用
- 警告メッセージ、ありがちなミス

# 項目と例題1

①(Rで)塩基配列解析のウェブページは、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。項目だけのものや古いものも多数あります。



(Rで)塩基配列解析 **①**  
(last modified 2022/06/01, since 2010)

このウェブページの多くは、[インストール | についての推奨手順](#) (Windows2022.05.16版とMacintosh2022.05.17版)に従ってフリーソフトRと必要なパッケージをインストール済みであるという前提で記述しています。初心者の方は[基本的な利用法](#)(Windows2022.05.17版の[PPTX](#)と[PDF](#); Macintosh2020.03.13版の[PPTX](#)と[PDF](#))で自習してください。

[@Agribio\\_utokyo](#)さんをフォロー、[アグリバイオ](#)、[\(Rで\)塩基配列解析のサブページ](#)、[生命情報解析研究室](#)

---

**What's new? (過去のお知らせはこちら)**

- このページをGoogle ChromeやMicrosoft Edgeで開いても、すぐにフリーズして「応答なし」と表示されることが頻繁にあったという報告をいただきました。このような事象に遭遇した方はブラウザをFirefoxに変更すると解消されるようです。情報提供いただいた学生さんに感謝m(\_ \_)m(2022/06/01) **NEW**
- ([サブページ](#)のほうのネタではありますが...) 日本乳酸菌学会誌の[第16回](#)と[第17回](#)の原稿をこちらでも公開しました。(2022/05/29) **NEW**
- 「インストール | R本体とRStudio | 最新版 | [Mac用](#)」を更新しました。(2022/05/16) **NEW**

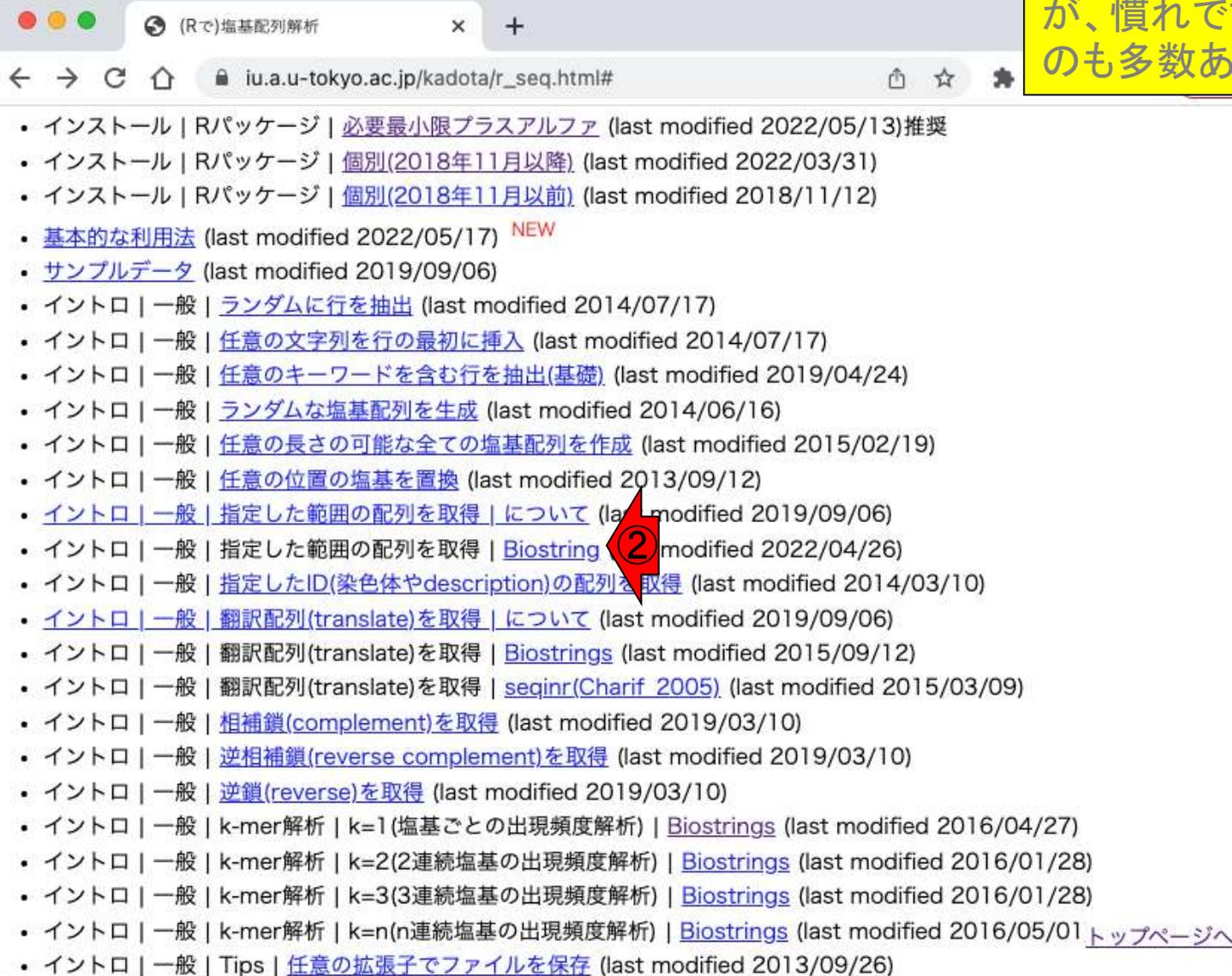
---

- [はじめに](#) (last modified 2022/03/28)
- [過去のお知らせ](#) (last modified 2022/06/01) **NEW**
- [インストール | について](#) (last modified 2022/05/17) **NEW**
- インストール | R本体とRStudio | 最新版 | [Win用](#) (last modified 2022/05/16)推奨 **NEW**

[トップページへ](#)

# 項目と例題2

①(Rで)塩基配列解析のページは、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。項目だけのものや古いものも多数あります。②をクリック。

- 
- (Rで)塩基配列解析
- iu.a.u-tokyo.ac.jp/kadota/r\_seq.html#
- ・ インストール | Rパッケージ | [必要最小限プラスアルファ](#) (last modified 2022/05/13)推奨
  - ・ インストール | Rパッケージ | [個別\(2018年11月以降\)](#) (last modified 2022/03/31)
  - ・ インストール | Rパッケージ | [個別\(2018年11月以前\)](#) (last modified 2018/11/12)
  - ・ [基本的な利用法](#) (last modified 2022/05/17) **NEW**
  - ・ [サンプルデータ](#) (last modified 2019/09/06)
  - ・ 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 modified 2014/07/17)
  - ・ イントロ | 一般 | [任意のキーワードを含む行を抽出\(基礎\)](#) (last modified 2019/04/24)
  - ・ 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 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 | について](#) (last modified 2019/09/06)
  - ・ イントロ | 一般 | 指定した範囲の配列を取得 | [Biostring](#) ② (last modified 2022/04/26)
  - ・ 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 | について](#) (last modified 2019/09/06)
  - ・ イントロ | 一般 | 翻訳配列(translate)を取得 | [Biostrings](#) (last modified 2015/09/12)
  - ・ イントロ | 一般 | 翻訳配列(translate)を取得 | [seqinr\(Charif 2005\)](#) (last modified 2015/03/09)
  - ・ イントロ | 一般 | [相補鎖\(complement\)を取得](#) (last modified 2019/03/10)
  - ・ イントロ | 一般 | [逆相補鎖\(reverse complement\)を取得](#) (last modified 2019/03/10)
  - ・ イントロ | 一般 | [逆鎖\(reverse\)を取得](#) (last modified 2019/03/10)
  - ・ イントロ | 一般 | k-mer解析 | k=1(塩基ごとの出現頻度解析) | [Biostrings](#) (last modified 2016/04/27)
  - ・ イントロ | 一般 | k-mer解析 | k=2(2連続塩基の出現頻度解析) | [Biostrings](#) (last modified 2016/01/28)
  - ・ イントロ | 一般 | k-mer解析 | k=3(3連続塩基の出現頻度解析) | [Biostrings](#) (last modified 2016/01/28)
  - ・ イントロ | 一般 | k-mer解析 | k=n(n連続塩基の出現頻度解析) | [Biostrings](#) (last modified 2016/05/01) [トップページへ](#)
  - ・ 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)

# 項目と例題3

こんな感じになります。①この項目は、塩基配列のファイルを入力として読み込んで、指定した範囲の配列を切り出して、別のファイルに出力するスクリプトです。



## イントロ | 一般 | 指定した範囲の配列を取得 | Biostrings

[Biostrings](#)パッケージ中のsubseq関数を用いて、single-FASTA形式やmulti-FASTA形式ファイルから様々な部分配列を取得するやり方を示します。この項目は、「この染色体の、ここから、ここまで」という指定の仕方になります。例えば入力ファイルがヒトゲノムだった場合に、chr3の20000から500000の座標の配列取得を行いたい場合などに利用します。したがって、chr4とchr8の配列のみ抽出といったやり方には対応していませんのでご注意ください。また、ファイルダウンロード時に、\*.fastaという拡張子が\*.txtに勝手に変更されることがありますのでご注意ください。ここでは、「ファイル」 - 「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

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

任意の範囲 (始点が3, 終点が9)の配列を抽出するやり方です。

```
in_f <- "sample1.fasta"           #入力ファイル名を指定してin_fに格納
out_f <- "hogel.fasta"           #出力ファイル名を指定してout_fに格納
param <- c(3, 9)                 #抽出したい範囲の始点と終点を指定

#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 <- subseq(fasta, param[1], param[2]) #paramで指定した始点と終点の範囲の配列を抽出した結果をfastaに
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
```

### 2. RefSeqのhuman mRNAのmulti-FASTA形式のファイル(h\_rna.fasta)の場合:

[トップページへ](#)

任意のRefSeq ID (例: [NM\\_203348.1](#))の任意の範囲 (例: 始点が2, 終点が5)の配列の抽出を行うやり方です。

# 項目と例題4

こんな感じになります。①この項目は、塩基配列のファイルを入力として読み込んで、指定した範囲の配列を切り出して、別のファイルに出力するスクリプトです。大抵の項目は、いくつかの例題を提供しています。例えば、②が例題1で、③が例題2だと解釈します。④クリックする項目を間違えて困った人は、「トップページへ」で一番上からやり直してください。



**Biostrings**パッケージ中のsubseq関数を用いて、single-FASTA形式やmulti-FASTA形式ファイルから得するやり方を示します。この項目は、「この染色体の、ここから、ここまで」という指定の仕方なファイルがヒトゲノムだった場合に、chr3の20000から500000の座標の配列取得を行いたい場合などがあって、chr4とchr8の配列のみ抽出といったやり方には対応していませんのでご注意ください。また、ど時に、\*.fastaという拡張子が\*.txtに勝手に変更されることがありますのでご注意ください。ここでは「ファイル」 - 「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

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

任意の範囲 (始点が3, 終点が9)の配列を抽出するやり方です。

```
in_f <- "sample1.fasta"           #入力ファイル名を指定してin_fに格納
out_f <- "hogel.fasta"           #出力ファイル名を指定してout_fに格納
param <- c(3, 9)                 #抽出したい範囲の始点と終点を指定

#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 <- subseq(fasta, param[1], param[2]) #paramで指定した始点と終点の範囲の配列を抽出した結果をfastaに
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
```

## 2. RefSeqのhuman mRNAのmulti-FASTA形式のファイル(h\_rna.fasta)の場合:

任意のRefSeq ID (例: NM\_203348.1)の任意の範囲 (例: 始点が2, 終点が5)の配列の抽出を行うやり方です。

[トップページへ](#)

# Contents

- 前提条件、RStudioの起動
- 見栄えの統一、終了
- Console画面の見栄え、設定変更
- 枠の幅、電卓的な利用
- プロンプト(>)、Console画面クリア
- 計算の中断
- 指定した範囲の配列を取得
  - 項目と例題、入力ファイルの保存
  - 作業ディレクトリの変更
  - 入力ファイルの確認、FASTA形式
  - コピペ実行、出力ファイルの確認、Environmentタブ
- フィルタリング(任意のキーワードを含む行を抽出)
  - 全体像の把握、入力ファイルの保存、コピペ実行、出力ファイルの確認
  - 応用
- 警告メッセージ、ありがちなミス

# 入力ファイルの保存1



## イントロ | 一般 | 指定した範囲の配列を取得 | Biostrings

[Biostrings](#)パッケージ中のsubseq関数を用いて、single-FASTA形式やmulti-FASTA形式ファイルから様々な部分配列を取得するやり方を示します。この項目は、「この染色体の、ここから、ここまで」という指定の仕方になります。例えば入力ファイルがヒトゲノムだった場合に、chr3の20000から500000の座標の配列取得を行いたい場合などに利用します。したがって、chr4とchr8の配列のみ抽出といったやり方には対応していませんのでご注意ください。また、ファイルダウンロード時に、\*.fastaという拡張子が\*.txtに勝手に変更されることがありますのでご注意ください。ここでは、「ファイル」 - 「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

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

任意の範囲 (始点が3, 終点が9)の配列を抽出するやり方です。

```
in_f <- "sample1.fasta"      #入力ファイル名を指定してin_fに格納
out_f <- "hogel.fasta"      #出力ファイル名を指定してout_fに格納
param <- c(3, 9)           #抽出したい範囲の始点と終点を指定

#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 <- subseq(fasta, param[1], param[2]) #paramで指定した始点と終点の範囲の配列を抽出した結果をfastaに
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
```

### 2. RefSeqのhuman mRNAのmulti-FASTA形式のファイル(h\_rna.fasta)の場合:

[トップページへ](#)

任意のRefSeq ID (例: [NM\\_203348.1](#))の任意の範囲 (例: 始点が2, 終点が5)の配列の抽出を行うやり方です。

# 入力ファイルの保存2

①例題1をやります。②デスクトップ上に、(名前は何でもよいですが)hogeというフォルダを作成し、そこに入力ファイルとして用いる③sample1.fastaを保存すべく、

## イントロ | 一般 | 指定した範囲の配列を取得 | Biostrings

Biostringsパッケージ中のsubseq関数を用いて、single-FASTA形式やmulti-FASTA形式ファイルから様々な部分配列を取得するやり方を示します。この項目は、「この染色体の、ここから、ここまで」という指定の仕方になります。例えば入力ファイルがヒトゲノムだった場合に、chr3の20000から500000の座標の配列取得を行いたい場合などに利用します。したがって、chr4とchr8の配列のみ抽出といったやり方には対応していませんのでご注意ください。また、ファイルダウンロード時に、\*.fastaという拡張子が\*.txtに勝手に変更されることがありますのでご注意ください。ここでは、「ファイル」 - 「ディレクトリの変更」 - 「新しいファイル」 - 「ディレクトリの変更」 - 「新しいファイル」の順に操作し、新しいファイル名を付けて保存し、その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. (single-)FASTA形式ファイル(sample1.fasta)の場合:

任意の範囲 (始点が3, 終点が9)の配列を抽出するやり方です。

```
in_f <- "sample1.fasta"
out_f <- "hoge1.fasta"
param <- c(3, 9)

#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")

#本番
fasta <- subseq(fasta, param[1], param[2])

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

よく使う項目  
最近の項目  
アプリケーション  
デスクトップ  
書類  
ダウンロード  
ピクチャ  
student

hoge

Macintosh HD > ユーザ > student > デスクトップ > hoge

### 2. RefSeqのhuman mRNAのmulti-FASTA形式のファイル(h\_rna.fasta)の場合:

[トップページへ](#)

任意のRefSeq ID (例: NM\_203348.1)の任意の範囲 (例: 始点が2, 終点が5)の配列の抽出を行うやり方です。

# 入力ファイルの保存3

①例題1をやります。②デスクトップ上に、(名前は何でもよいですが)hogeというフォルダを作成し、そこに入力ファイルとして用いる③sample1.fastaを保存すべく、右クリックで④保存。

## イントロ | 一般 | 指定した範囲の配列を取得 | Biostrings

**Biostrings**パッケージ中のsubseq関数を用いて、single-FASTA形式やmulti-FASTA形式ファイルから様々な部分配列を取得するやり方を示します。この項目は、「この染色体の、ここから、ここまで」という指定の仕方になります。例えば入力ファイルがヒトゲノムだった場合に、chr3の20000から500000の座標の配列取得を行いたい場合などに利用します。したがって、chr4とchr8の配列のみ抽出といったやり方には対応していませんのでご注意ください。また、ファイルダウンロード時に、\*.fastaという拡張子が\*.txtに勝手に変更されることがありますのでご注意ください。ここでは、「sample1.fasta」 - 「ディレクトリの変更」 - 「新しいファイル」 - 「新しいファイル」を置いてあるディレクトリに移動し以下をコピー。

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

任意の範囲 (始点が3, 終点が9)の配列を抽出

```
in_f <- "sample1.fasta"
out_f <- "hoge1.fasta"
param <- c(3, 9)

#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,
                           file.choose(),
                           as.is=T)
fasta

#本番
fasta <- subseq(fasta, param[1],
                param[2])
fasta

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

- 新しいタブで開く
- 新しいウィンドウで開く
- シークレット ウィンドウでリンクを開く
- リンク先を別名で保存...
- リンクのアドレスをコピー
- コピー
- 選択箇所へのリンクをコピー
- Google で「sample1.fasta」を検索
- 印刷...
- 検証
- スピーチ
- サービス

リンク先を別名で保存...

④

ファイルの読み込み

指定した範囲の配列を抽出した結果をfastaに

fastaの中身を指定したファイル名で保存

### 2. RefSeqのhuman mRNAのmulti-FASTA形式のファイル(h\_rna.fasta)の場合:

[トップページへ](#)

[https://www.iu.a.u-tokyo.ac.jp/kadota/R\\_seq/sample1.fasta](https://www.iu.a.u-tokyo.ac.jp/kadota/R_seq/sample1.fasta) の範囲 (例: 始点が2, 終点が5)の配列の抽出を行うやり方です。

# 入力ファイルの保存4

①例題1をやります。②デスクトップ上に、(名前は何でもよいですが)hogeというフォルダを作成し、そこに入力ファイルとして用いる③sample1.fastaを保存すべく、右クリックで④保存。⑤デスクトップ、⑥hogeを選択して、⑦保存。

Intro | 一般 | 指定した範囲の配列を取得 | Biostrings

Biostringsパッケージ中のsubseq関数を用いて、single-FASTA形式やmulti-FASTA形式ファイルから様々な部分配列を取得

よく使う項目

- アプリケーション
- デスクトップ
- 書類
- ダウンロード
- ピクチャ
- student
- iCloud
- iCloud Drive
- 共有
- 場所
- Mauve...
- ネットワーク

名前: sample1

タグ:

デスクトップ

hoge

Miniconda3-...X-x86\_64.sh

share\_Bio-Linux

share\_Bio-Linux.zip

新規フォルダ

キャンセル

保存

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

2. RefSeqのhuman mRNAのmulti-FASTA形式のファイル (h\_rna.fasta)の場合: [トップページへ](#)

任意のRefSeq ID (例: NM\_203348.1)の任意の範囲 (例: 始点が2, 終点が5)の配列の抽出を行うやり方です。

# 入力ファイルの保存5

①例題1をやります。②デスクトップ上に、(名前は何でもよいですが)hogeというフォルダを作成し、そこに入力ファイルとして用いる③sample1.fastaを保存すべく、右クリックで④保存。⑤デスクトップ、⑥hogeを選択して、⑦保存。⑧ときどき勝手に.txtが追加されることがあるので注意してください。sample1.fastaが正解です。

## イントロ | 一般 | 指定した範囲の配列を取得 | Biostrings

Biostringsパッケージ中のsubseq関数を用いて、single-FASTA形式やmulti-FASTA形式ファイルから得するやり方を示します。この項目は、「この染色体の、ここから、ここまで」という指定の仕方になファイルがヒトゲノムだった場合に、chr3の20000から500000の座標の配列取得を行いたい場合などがあって、chr4とchr8の配列のみ抽出といったやり方には対応していませんのでご注意ください。また、ド時に、\*.fastaという拡張子が\*.txtに勝手に変更されることがありますのでご注意ください。ここでは「ファイル」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

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

任意の範囲 (始点が3, 終点が9)の配列を抽出するやり方です。

```
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
param <- c(3, 9) #抽出したい範囲の始点と終点を指定

#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 <- subseq(fasta, param[1], param[2]) #paramで指定した始点と終点の範囲の配列を抽出した結果をfastaに
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
```

[トップページへ](#)

8

sample1.fasta

すべて表示

X

# 入力ファイルの保存6

①例題1をやります。②デスクトップ上に、(名前は何でもよいですが)hogeというフォルダを作成し、そこに入力ファイルとして用いる③sample1.fastaを保存すべく、右クリックで④保存。⑤デスクトップ、⑥hogeを選択して、⑦保存。⑧ときどき勝手に.txtが追加されることがあるので注意してください。sample1.fastaが正解です。⑨hogeフォルダの中身が、⑩のようになればOK。

## イントロ | 一般 | 指定した範囲の配列を取得 | Biostrings

Biostringsパッケージ中のsubseq関数を用いて、single-FASTA形式やmulti-FASTA形式ファイルから得るやり方を示します。この項目は、「この染色体の、ここから、ここまで」という指定の仕方になファイルがヒトゲノムだった場合に、chr3の20000から500000の座標の配列取得を行いたい場合などがあって、chr4とchr8の配列のみ抽出といったやり方には対応していませんのでご注意ください。また、ド時に、\*.fastaという拡張子が\*.txtに勝手に変更されることがありますのでご注意ください。ここでは「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

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

任意の範囲 (始点が3, 終点が9)の配列を抽出するやり方です。

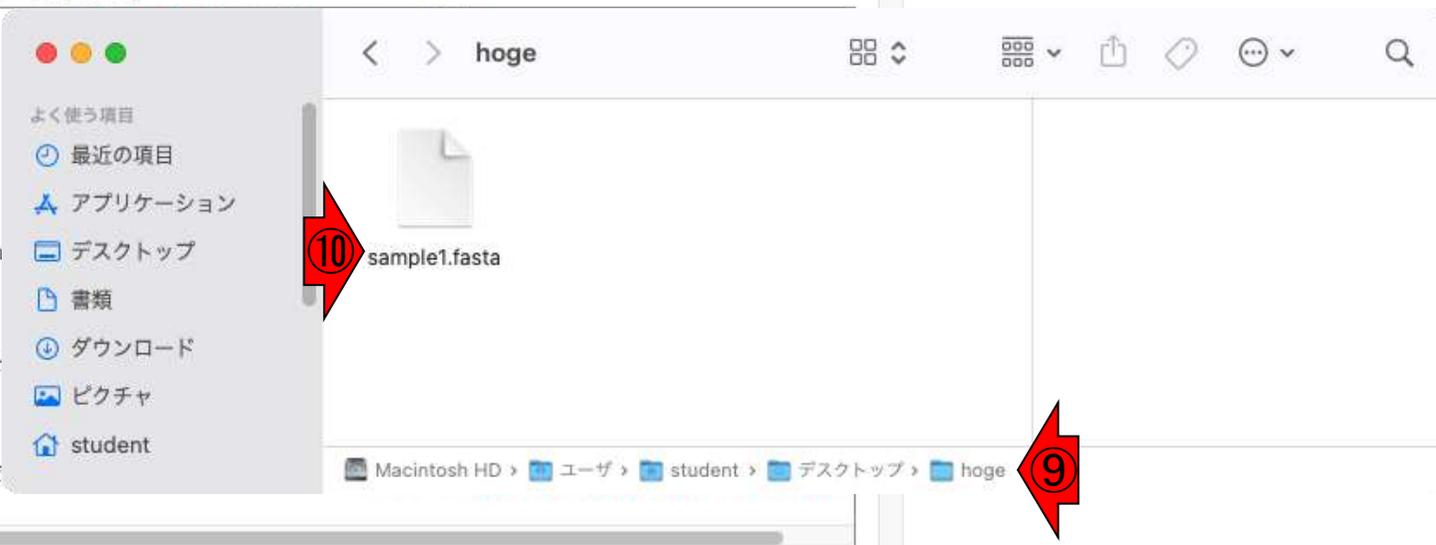
```
in_f <- "sample1.fasta"
out_f <- "hoge1.fasta"
param <- c(3, 9)

#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")

#本番
fasta <- subseq(fasta, param[1], param[2])

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



### 2. RefSeqのhuman mRNAのmulti-FASTA形式のファイル(h\_rna.fasta)の場合:

[トップページへ](#)

任意のRefSeq ID (例: NM\_203348.1)の任意の範囲 (例: 始点が2, 終点が5)の配列の抽出を行うやり方です。

# Contents

- 前提条件、RStudioの起動
- 見栄えの統一、終了
- Console画面の見栄え、設定変更
- 枠の幅、電卓的な利用
- プロンプト(>)、Console画面クリア
- 計算の中断
- 指定した範囲の配列を取得
  - 項目と例題、入力ファイルの保存
  - 作業ディレクトリの変更
  - 入力ファイルの確認、FASTA形式
  - コピペ実行、出力ファイルの確認、Environmentタブ
- フィルタリング(任意のキーワードを含む行を抽出)
  - 全体像の把握、入力ファイルの保存、コピペ実行、出力ファイルの確認
  - 応用
- 警告メッセージ、ありがちなミス

# 作業ディレクトリの変更1

①hogeフォルダ(ディレクトリ)内にある、  
②sample1.fastaが目的のファイルですが、現実には②と同じファイル名のものが別のフォルダにあるかもしれません。  
作業ディレクトリの変更という作業は、RStudio上で作業する場所(ディレクトリ or フォルダ)を宣言すること。これによって、以降の作業を③「ファイル名のみ」を与えることで済ませられるので便利です。



Biostringsパッケージ中のsubseq関数を用いて、single-FASTA形式やmulti-FASTA形式ファイルから得るやり方を示します。この項目は、「この染色体の、ここから、ここまで」という指定の仕方になファイルがヒトゲノムだった場合に、chr3の20000から500000の座標の配列取得を行いたい場合などがあって、chr4とchr8の配列のみ抽出といったやり方には対応していませんのでご注意ください。また、ド時に、\*.fastaという拡張子が\*.txtに勝手に変更されることがありますのでご注意ください。ここでは「ファイル」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

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

任意の範囲(始点が3, 終点が9)の配列を抽出するやり方です。

```
in_f <- "sample1.fasta"
out_f <- "hoge1.fasta"
param <- c(3, 9)

#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")

#本番
fasta <- subseq(fasta, param[1], param[2])

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



## 2. RefSeqのhuman mRNAのmulti-FASTA形式のファイル(h\_rna.fasta)の場合:

[トップページへ](#)

任意のRefSeq ID (例: NM\_203348.1)の任意の範囲(例: 始点が2, 終点が5)の配列の抽出を行うやり方です。

# 作業ディレクトリの変更2

この環境では、①の場所を作業ディレクトリに設定することになります。正確には、「/Users/student/Desktop/hoge」という①の全ての情報(フルパス、といいます)を指定することで、作業ディレクトリがどこであっても②を読み込むことはできます。そういうやり方を教えるヒトもいると思います。

## イントロ | 一般 | 指定した範囲の配列を取得 | Biostrings

Biostringsパッケージ中のsubseq関数を用いて、single-FASTA形式やmulti-FASTA形式ファイルから得するやり方を示します。この項目は、「この染色体の、ここから、ここまで」という指定の仕方になファイルがヒトゲノムだった場合に、chr3の20000から500000の座標の配列取得を行いたい場合などがあって、chr4とchr8の配列のみ抽出といったやり方には対応していませんのでご注意ください。また、ファイルダウンロード時に、\*.fastaという拡張子が\*.txtに勝手に変更されることがありますのでご注意ください。ここでは、「ファイル」 - 「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

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

任意の範囲 (始点が3, 終点が9)の配列を抽出するやり方です。

```
in_f <- "sample1.fasta"
out_f <- "hoge1.fasta"
param <- c(3, 9)

#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")

#本番
fasta <- subseq(fasta, param[1], param[2])

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



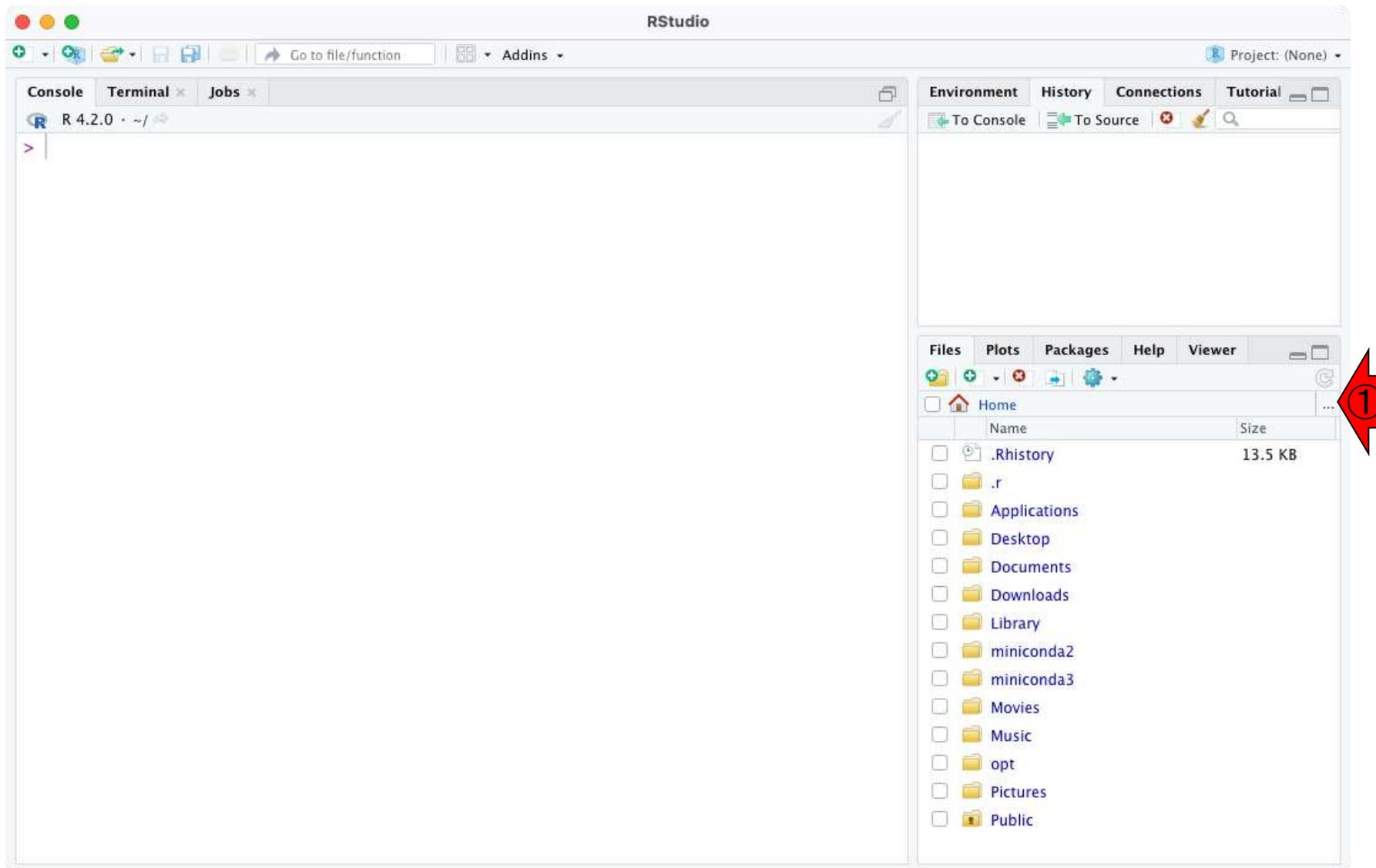
### 2. RefSeqのhuman mRNAのmulti-FASTA形式のファイル(h\_rna.fasta)の場合:

[トップページへ](#)

任意のRefSeq ID (例: NM\_203348.1)の任意の範囲 (例: 始点が2, 終点が5)の配列の抽出を行うやり方です。

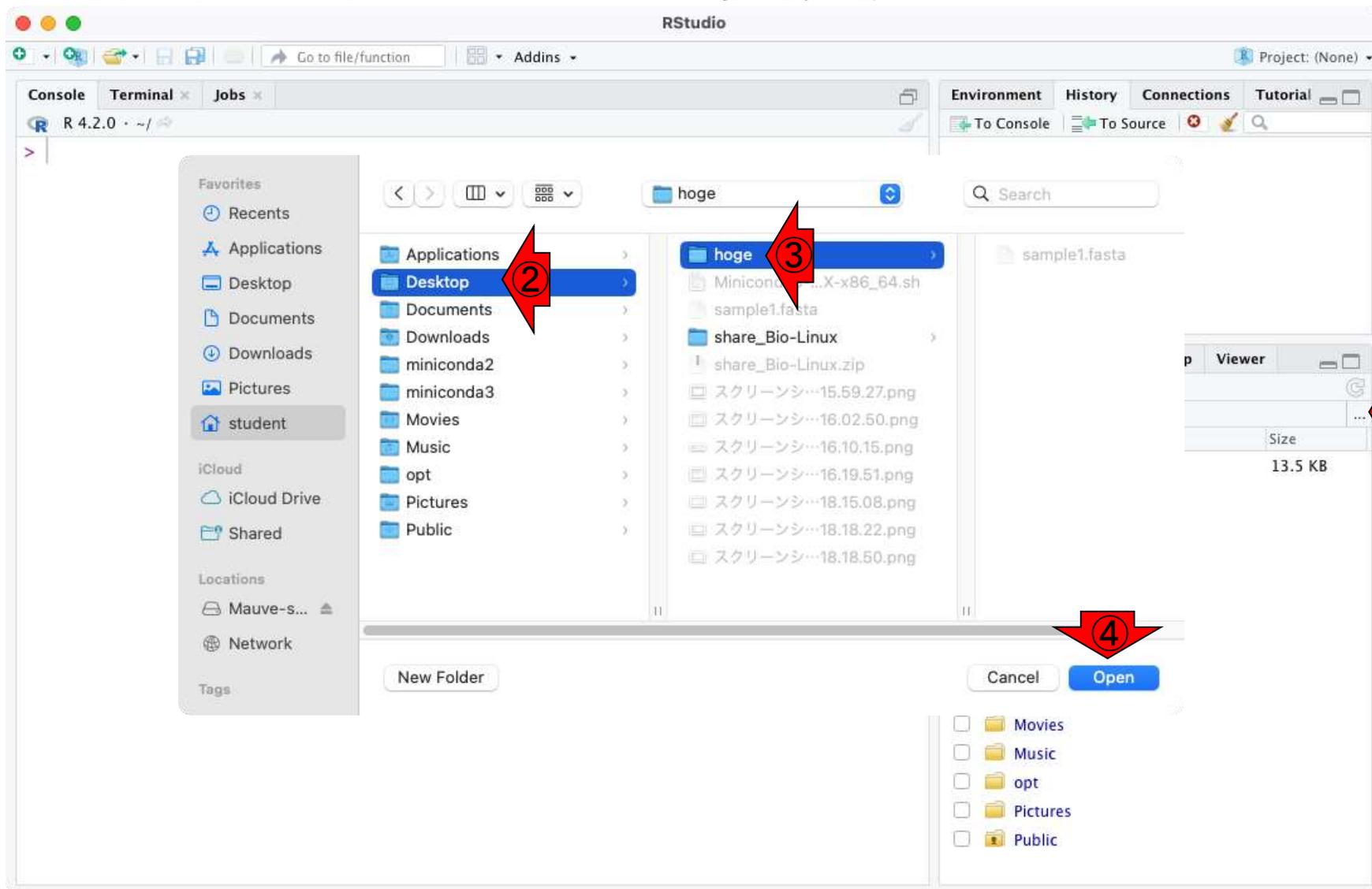
①をクリック。

# 作業ディレクトリの変更3



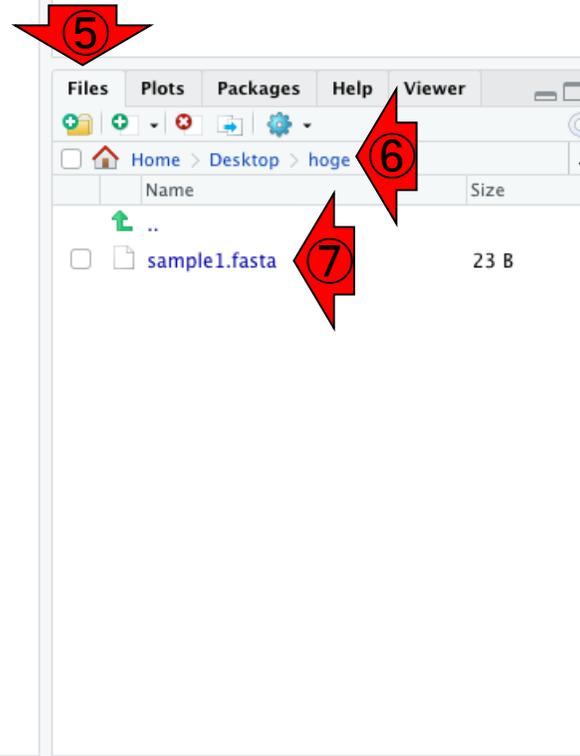
# 作業ディレクトリの変更4

①をクリック。先ほど作成した②デスクトップの③hogeを選択して、④Open。



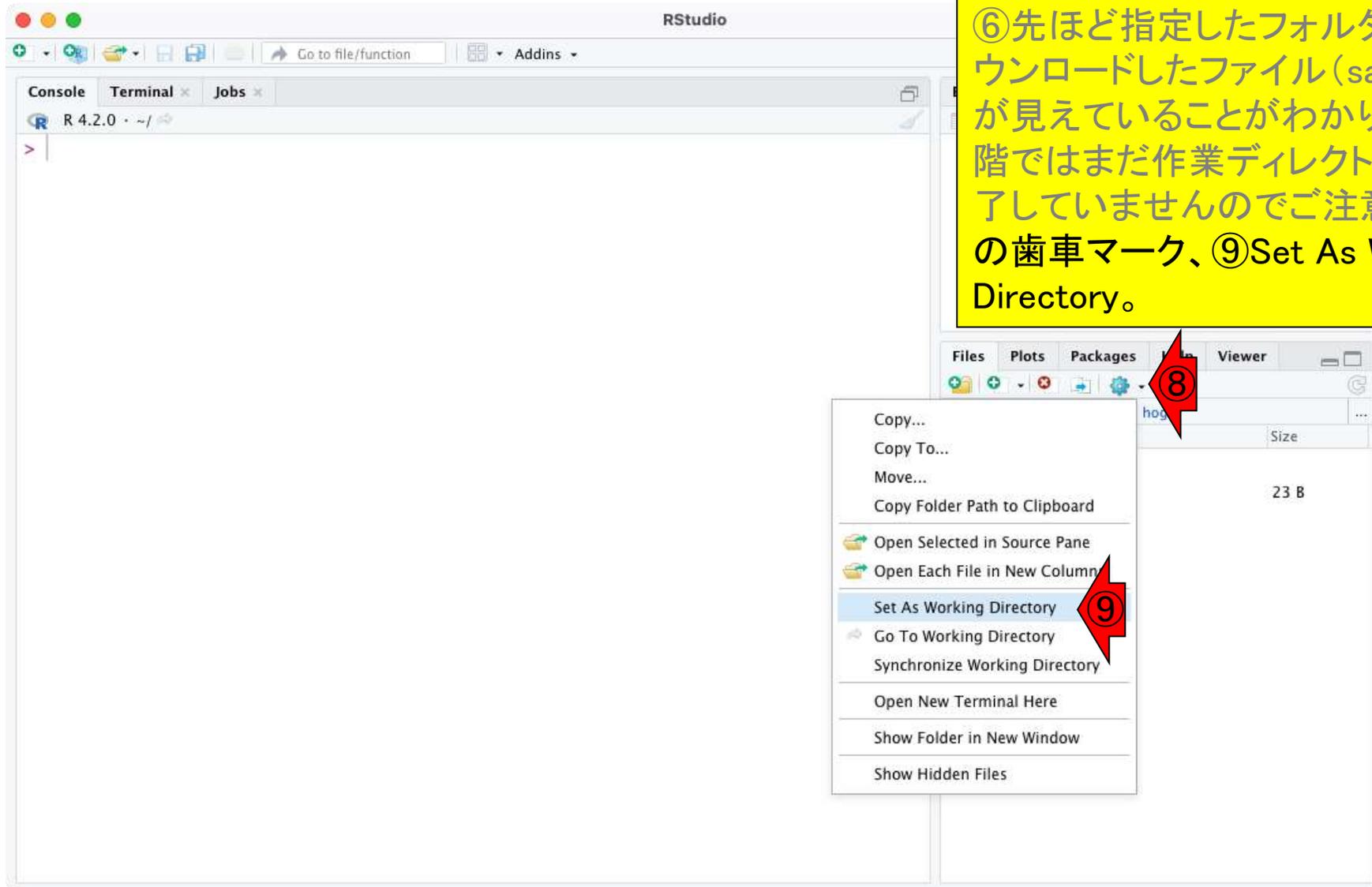
# 作業ディレクトリの変更5

①をクリック。先ほど作成した②デスクトップの③hogeを選択して、④Open。こんな感じになって、⑤Filesタブの部分が⑥先ほど指定したフォルダになり、⑦ダウンロードしたファイル(sample1.fasta)が見えていることがわかります。この段階ではまだ作業ディレクトリの変更が完了していませんのでご注意ください。



# 作業ディレクトリの変更6

①をクリック。先ほど作成した②デスクトップの③hogeを選択して、④Open。こんな感じになって、⑤Filesタブの部分が⑥先ほど指定したフォルダになり、⑦ダウンロードしたファイル(sample1.fasta)が見えていることがわかります。この段階ではまだ作業ディレクトリの変更が完了していませんのでご注意ください。⑧の歯車マーク、⑨Set As Working Directory。



# 作業ディレクトリの変更7

実行後の状態。⑩のコマンドが自動的に実行されます。ここまで行ってきたGUI上での作業が、⑩に相当することが理解できます。

The screenshot shows the RStudio interface. The console window on the left displays the command `setwd("~/Desktop/hoge")` being entered, with a red arrow and the number 10 pointing to it. The file explorer on the right shows the current directory as `~/Desktop/hoge` and lists a file named `sample1.fasta` with a size of 23 B.

```
R 4.2.0 · ~/Desktop/hoge/
> setwd("~/Desktop/hoge")
> |
```

Name	Size
..	
sample1.fasta	23 B

# 作業ディレクトリの変更8

実行後の状態。⑩のコマンドが自動的に実行されます。ここまで行ってきたGUI上での作業が、⑩に相当することが理解できます。逆に言えば、⑩のコマンドを実行することは、⑪の場所で作業すると設定することであり、ここまでGUIのマウスクリックで行って来た設定作業と同義だということです。

The screenshot shows the RStudio interface. The top panel has tabs for Console, Terminal, and Jobs. The Terminal window shows the command `setwd("~/Desktop/hoge")` entered, with a red arrow labeled ⑩ pointing to the command and another red arrow labeled ⑪ pointing to the prompt. The bottom panel shows the Files view with the directory `~/Desktop/hoge` selected, containing a file named `sample1.fasta` with a size of 23 B.

# 作業ディレクトリの変更9

Macでは、①の部分が具体的に表示されずHomeとなっています。Consoleでは、この部分を②〜(チルダ)で表しています。〜(チルダ)は、Linuxなどでは一般的にホームディレクトリを意味し、ここでは /Users/studentということになります。

The screenshot displays the RStudio interface. The top-left pane is the Console, showing the command `setwd("~/Desktop/hoge")` and a prompt `> |`. A red arrow labeled '2' points to the tilde character in the prompt. The bottom-right pane is the Files view, showing the path `Home > Desktop > hoge` and a file `sample1.fasta` with a size of `23 B`. A red arrow labeled '1' points to the `Home` icon in the file browser.

# Contents

- 前提条件、RStudioの起動
- 見栄えの統一、終了
- Console画面の見栄え、設定変更
- 枠の幅、電卓的な利用
- プロンプト(>)、Console画面クリア
- 計算の中断
- 指定した範囲の配列を取得
  - 項目と例題、入力ファイルの保存
  - 作業ディレクトリの変更
  - 入力ファイルの確認、FASTA形式
  - コピペ実行、出力ファイルの確認、Environmentタブ
- フィルタリング(任意のキーワードを含む行を抽出)
  - 全体像の把握、入力ファイルの保存、コピペ実行、出力ファイルの確認
  - 応用
- 警告メッセージ、ありがちなミス

# 入力ファイルの確認1

①hogeフォルダ内の、②sample1.fastaをクリックしてみましょう。

The screenshot shows the RStudio interface. The top-left pane is the Console, showing the command `setwd("~/Desktop/hoge")` and the prompt `>`. The top-right pane is the Environment pane, showing the same command. The bottom-right pane is the Files pane, showing the file explorer for the `~/Desktop/hoge` directory. The file `sample1.fasta` is listed with a size of 23 B. Red arrows with circled numbers 1 and 2 point to the `hoge` folder and the `sample1.fasta` file, respectively.

Environment History Connections Tutorial

To Console To Source

```
setwd("~/Desktop/hoge")
```

Files Plots Packages Help Viewer

Home > Desktop > hoge

Name	Size
..	
sample1.fasta	23 B

# 入力ファイルの確認2

①hogeフォルダ内の、②sample1.fastaをクリックしてみましょう。こんな感じになります。

The screenshot shows the RStudio interface. The top-left pane displays the contents of 'sample1.fasta':

```
1 >kadota
2 AGTGACGGTCTT
3
```

The bottom-left pane (Console) shows the command `setwd("~/Desktop/hoge")` being executed in an R 4.2.0 session.

The bottom-right pane (Files) shows the file explorer view of the `~/Desktop/hoge` directory. The file `sample1.fasta` is highlighted, and a red arrow labeled '2' points to it. Another red arrow labeled '1' points to the `hoge` directory path in the breadcrumb navigation.

# 入力ファイルの確認3

①hogeフォルダ内の、②sample1.fastaをクリックしてみましょう。こんな感じになります。赤枠が、③sample1.fastaの中身に相当します。

The screenshot shows the RStudio interface with the following components:

- File Editor:** The file `sample1.fasta` is open, showing the following content:

```
1 >kadota
2 AGTGACGGTCTT
3
```

A red rectangle highlights the content of the file.
- Console:** The terminal shows the command `setwd("~/Desktop/hoge")` being executed.

```
R 4.2.0 ~~/Desktop/hoge/
> setwd("~/Desktop/hoge")
>
```
- File Browser:** The file browser shows the directory `~/Desktop/hoge` with a file named `sample1.fasta` (23 B). Red arrows point to the file name (labeled ②) and the directory path (labeled ①).

# 入力ファイルの確認4

①hogeフォルダ内の、②sample1.fastaをクリックしてみましょう。こんな感じになります。赤枠が、③sample1.fastaの中身に相当します。④行番号情報です。1行が長いデータの場合に、④を眺めることで行の区切りを把握できます。

The screenshot shows the RStudio interface. The top-left pane displays the contents of the file 'sample1.fasta' with line numbers 1, 2, and 3. The first line is '>kadota' and the second line is 'AGTGACGGTCTT'. The third line is empty. A red box highlights the content of the file. The bottom-left pane shows the console with the command '> setwd("~/Desktop/hoge")' and the prompt '>'. The bottom-right pane shows a file explorer window with the path 'Home > Desktop > hoge'. The file 'sample1.fasta' is selected, and its size is shown as '23 B'. Red arrows with numbers 1, 2, 3, and 4 point to the folder, the file, the editor content, and the line numbers respectively.

# 入力ファイルの確認5

今はカーソルが①の位置(1行目の行頭)にあります。

The screenshot shows the RStudio interface with a text editor window open to a file named 'sample1.fasta'. The file content is as follows:

```
1 |kadota
2 |AGTGACGGTCTT
3 |
```

A red arrow with the number 1 points to the cursor at the beginning of the first line. The console window shows the command `setwd("~/Desktop/hoge")` being executed.

The Files pane on the right shows the directory structure: Home > Desktop > hoge, with a file named 'sample1.fasta' (23 B) listed.

# 入力ファイルの確認6

今はカーソルが①の位置(1行目の行頭)にあります。②は2行目の行末の位置に変更しただけ。

The screenshot shows the RStudio interface. The top-left pane displays a text editor with the following content:

```
1 >kadota
2 AGTGACGGTCTT|
3
```

Red arrows labeled ① and ② point to the start of line 1 and the end of line 2, respectively. The bottom-left pane shows the terminal window with the command `setwd("~/Desktop/hoge")` executed. The bottom-right pane shows the file browser with the directory `~/Desktop/hoge` selected, containing a file named `sample1.fasta` with a size of 23 B.

# 入力ファイルの確認7

今はカーソルが①の位置(1行目の行頭)にあります。②は2行目の行末の位置に変更しただけ。③の拡大図が示すように、①のカーソル位置「1:1」から②のカーソル位置「2:13」に変更されていることも分かります。そこそこのエディタには普通についている機能ではありますが、ゲノム情報を解析する場合は、例えば②の位置が何番目の塩基に相当するのかを知りたい場合があるので便利。

The screenshot shows the RStudio interface with a text editor window open to a file named 'sample1.fasta'. The editor contains the following text:

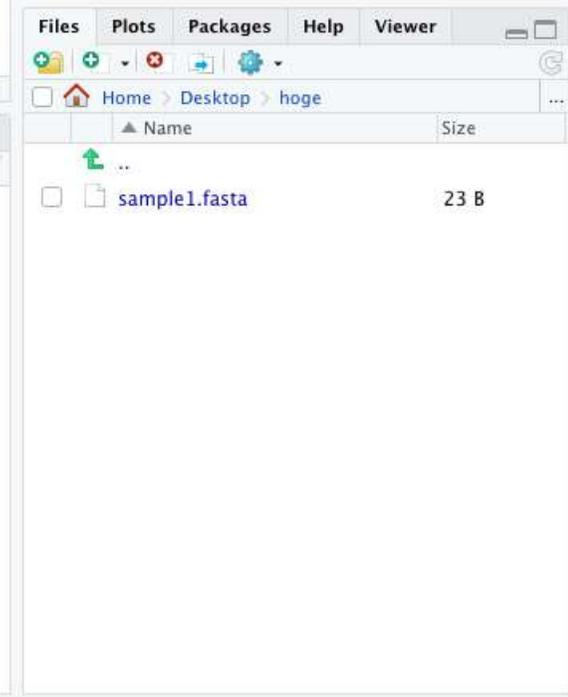
```
1 >kadota
2 AGTGACGGTCTT|
3
```

Three red arrows with circled numbers indicate cursor positions:

- ①: Points to the start of the first line (column 1, row 1).
- ②: Points to the end of the second line (column 13, row 2).
- ③: Points to a zoomed-in view of the cursor position '2:13' in the status bar.

The console window at the bottom shows the following commands:

```
R 4.2.0 ~./Desktop/hoge/
> setwd("~/Desktop/hoge")
>
```



# Contents

- 前提条件、RStudioの起動
- 見栄えの統一、終了
- Console画面の見栄え、設定変更
- 枠の幅、電卓的な利用
- プロンプト(>)、Console画面クリア
- 計算の中断
- 指定した範囲の配列を取得
  - 項目と例題、入力ファイルの保存
  - 作業ディレクトリの変更
  - 入力ファイルの確認、FASTA形式
  - コピペ実行、出力ファイルの確認、Environmentタブ
- フィルタリング(任意のキーワードを含む行を抽出)
  - 全体像の把握、入力ファイルの保存、コピペ実行、出力ファイルの確認
  - 応用
- 警告メッセージ、ありがちなミス

# FASTA形式1

①赤枠で見えているのが「FASTA形式」です。

The screenshot shows the RStudio interface with a text editor window open to 'sample1.fasta'. The editor contains the following text:

```
1 >kadota
2 AGTGACGGTCTT
3
```

A red box highlights the first two lines, and a red arrow with the number 1 points to it. The console window at the bottom shows the command:

```
> setwd("~/Desktop/hoge")
>
```

The Files pane on the right shows the directory structure:

Name	Size
..	
sample1.fasta	23 B

# FASTA形式2

①赤枠で見えているのが「FASTA形式」です。最初に②「大なり記号(>)」、その後③任意の記述内容(この場合はkadota)からなる1行の「description行」があり、その次の行に④塩基 or アミノ酸配列がある(複数行になってもよい)という形式です。

The screenshot shows the RStudio interface. The top-left pane displays a text file named 'sample1.fasta' with the following content:

```
1 >kadota
2 AGTGACGGTCTT
3
```

Red arrows with circled numbers point to the components: ② points to the greater-than sign (>), ③ points to the text 'kadota', and ④ points to the sequence 'AGTGACGGTCTT'. The bottom-left pane shows the R console with the command:

```
> setwd("~/Desktop/hoge")
>
```

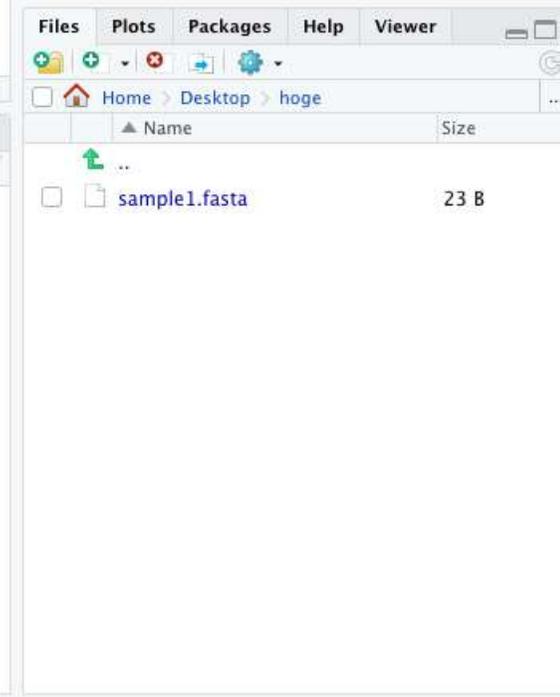
The bottom-right pane shows a file explorer view of the directory ~/Desktop/hoge, listing the file 'sample1.fasta' with a size of 23 B.

# FASTA形式3

①赤枠で見えているのが「FASTA形式」です。最初に②「大なり記号(>)」、その後③任意の記述内容(この場合はkadota)からなる1行の「description行」があり、その次の行に④塩基 or アミノ酸配列がある(複数行になってもよい)という形式です。⑤の部分は任意に変更可能なので、例えばdescription1にしてみると、⑥何か変更したことを意味するアスタリスク(\*)が付加されます。

```
sample1.fasta
1 >description1
2 AGTGACGGTCTT
3
```

```
R 4.2.0 ~./Desktop/hoge/
> setwd("~/Desktop/hoge")
>
```



# FASTA形式4

①赤枠で見えているのが「FASTA形式」です。最初に②「大なり記号(>)」、その後③任意の記述内容(この場合はkadota)からなる1行の「description行」があり、その次の行に④塩基 or アミノ酸配列がある(複数行になってもよい)という形式です。⑤の部分は任意に変更可能なので、例えばdescription1にしてみると、⑥何か変更したことを意味するアスタリスク(\*)が付加されます。⑦を押して保存します。

The screenshot shows the RStudio interface. The top pane displays a text editor for 'sample1.fasta' with the following content:

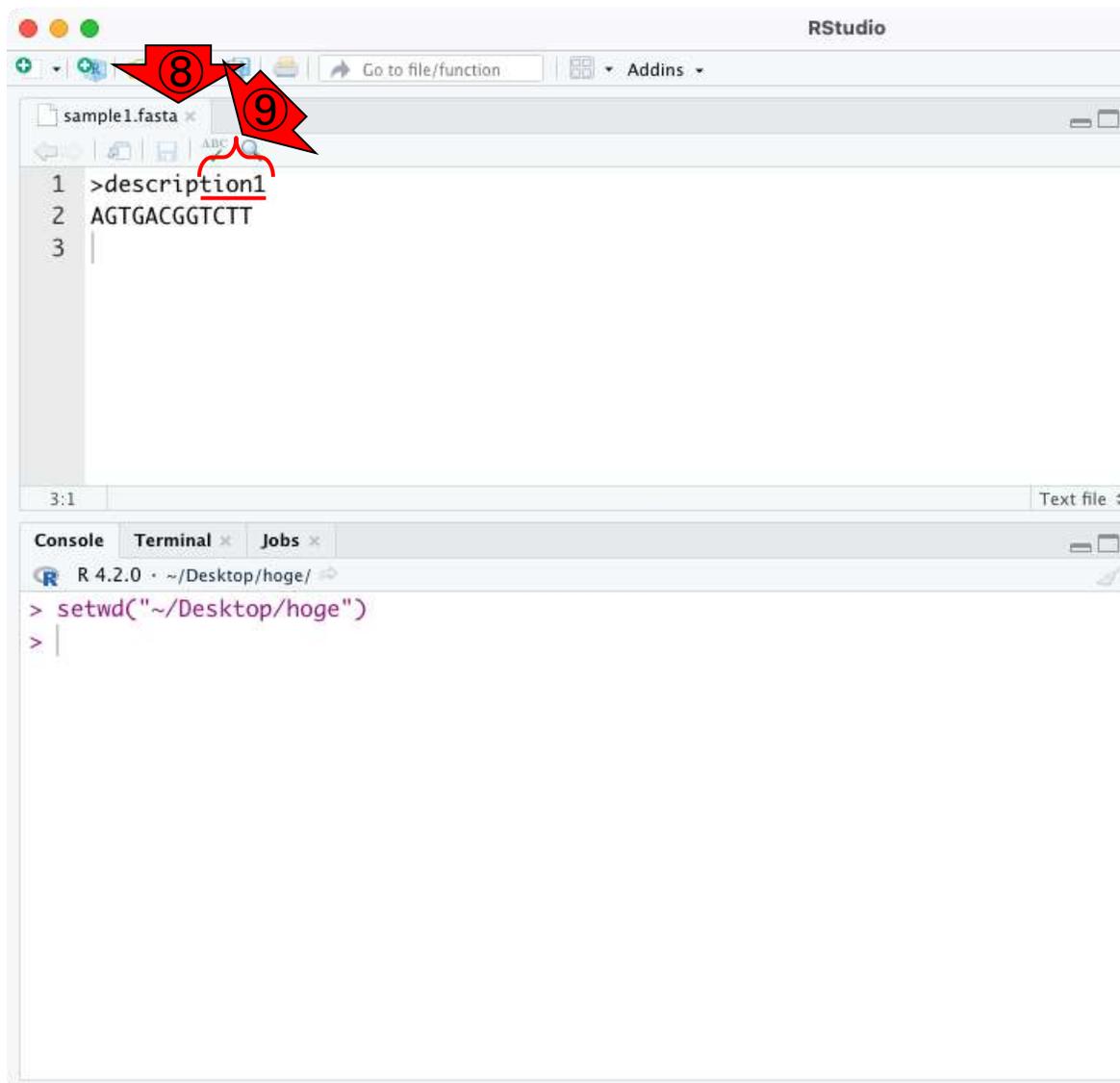
```
1 >description1
2 AGTTCCT
3
```

A red arrow points to the asterisk in the first line, which is circled in red. The bottom pane shows the R console with the following commands:

```
> setwd("~/Desktop/hoge")
>
```

The right pane shows a file explorer view of the 'hoge' directory on the desktop, containing a file named 'sample1.fasta' with a size of 23 B.

# FASTA形式5



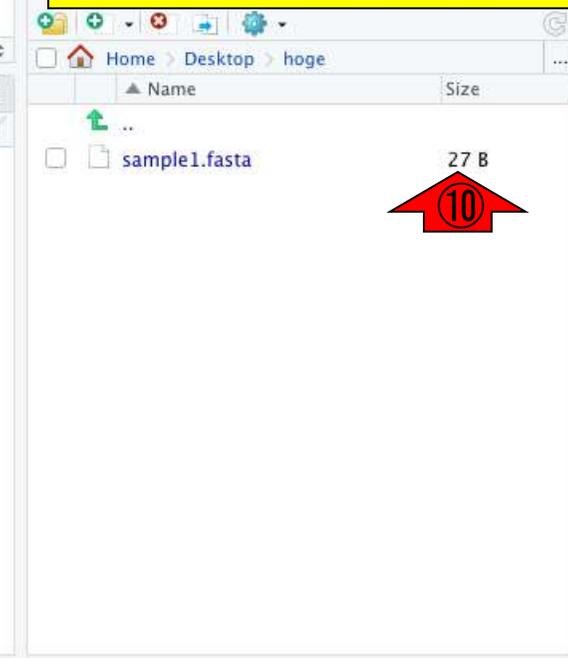
The screenshot shows the RStudio interface. The top pane displays the contents of 'sample1.fasta':

```
1 >description1
2 AGTGACGGTCTT
3 |
```

Red arrows labeled 8 and 9 point to the asterisk in '>description1' and the '1' in '1 >description1' respectively. The bottom pane shows a terminal window with the following commands:

```
R 4.2.0 ~./Desktop/hoge/
> setwd("~/Desktop/hoge")
> |
```

⑧\*が消え、⑨kadotaの6文字から description1 の12文字に増え、⑩ファイルサイズが大きくなっている(23 → 27 bytes)ことが分かります。ここで、1文字1byteなので6bytes増えるはずなのに4bytesしか増えていないことに気づくかも知れませんが、その理由は、元のファイルがWindowsで作られ、改行1行あたりに2bytes使われていたものが、Macintoshでは改行に1byteしか使われないためです(23 + 6 - 1 - 1)。



# FASTA形式6

```

sample1.fasta x
>kadota
AGTGACGGTCTT
3
  
```

```

R 4.2.0 ~./Desktop/hoge/
> setwd("~/Desktop/hoge")
>
  
```

①description行を元と同じkadotaとして上書き保存すると、ファイルサイズは元の23bytesではなく②21bytesとなります。これは改行コードがMacintosh仕様の1改行あたり1byteで保存されたためです(7 + 12 + 1 + 1)。

ゲノム情報の場合は、③の塩基配列情報のほうが圧倒的に①description行の情報よりも多いので、②「ファイルサイズ ≒ ゲノムサイズ」と解釈する場合があります。実際、ヒトゲノムは30億塩基対と言われますが、そのファイルサイズは約3GBです。

sample1.fasta

21 B

②

# FASTA形式7

赤枠がRのエディタ画面。①×を押して編集を終了させると、

The screenshot shows the RStudio interface. The top-left pane displays a text editor for 'sample1.fasta' with the following content:

```
1 >description1
2 AGTGACGGTCTT
3 |
```

A red arrow with the number '1' points to the close button (X) in the top-left corner of the editor window. The bottom-left pane shows the R console with the command:

```
> setwd("~/Desktop/hoge")
> |
```

The bottom-right pane shows a file browser view of the directory ~/Desktop/hoge, containing a file named 'sample1.fasta' with a size of 27 B.

# FASTA形式8

赤枠がRのエディタ画面。①×を押して編集を終了させると、こんな感じになります。

The screenshot shows the RStudio interface. The console on the left displays the command `setwd("~/Desktop/hoge")` and the prompt `>`. The file browser on the right shows the directory `~/Desktop/hoge` containing a file named `sample1.fasta` with a size of 27 B. The Environment pane is empty, and the History pane shows the executed command.

```
R 4.2.0 · ~/Desktop/hoge/
> setwd("~/Desktop/hoge")
> |
```

Name	Size
..	
sample1.fasta	27 B

# Contents

- 前提条件、RStudioの起動
- 見栄えの統一、終了
- Console画面の見栄え、設定変更
- 枠の幅、電卓的な利用
- プロンプト(>)、Console画面クリア
- 計算の中断
- 指定した範囲の配列を取得
  - 項目と例題、入力ファイルの保存
  - 作業ディレクトリの変更
  - 入力ファイルの確認、FASTA形式
  - コピペ実行、出力ファイルの確認、Environmentタブ
- フィルタリング(任意のキーワードを含む行を抽出)
  - 全体像の把握、入力ファイルの保存、コピペ実行、出力ファイルの確認
  - 応用
- 警告メッセージ、ありがちなミス

# コピペ実行1

おさらい。①この項目は、塩基配列のファイルを入力として読み込んで、指定した範囲の配列を切り出して、別のファイルに出力するスクリプトです。②例題1は、先ほどdescription行部分を変更した③sample1.fastaを入力として、



**Biostrings**パッケージ中のsubseq関数を用いて、single-FASTA形式やmulti-FASTA形式ファイルから様々な部分配列を取得するやり方を示します。この項目は、「この染色体の、ここから、ここまで」という指定の仕方になります。例えば入力ファイルがヒトゲノムだった場合に、chr3の20000から500000の座標の配列取得を行いたい場合などに利用します。したがって、chr4とchr8の配列のみ抽出といったやり方には対応していませんのでご注意ください。また、ファイルダウンロード時に、\*.fastaという拡張子が\*.txtに勝手に変更されることがありますのでご注意ください。ここでは、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. (single-)FASTA形式ファイル(sample1.fasta)の場合:

任意の範囲 (始点が3, 終点が9)の配列を抽出するやり方です。

```
in_f <- "sample1.fasta" #入力ファイル名を指定してin_fに格納
out_f <- "hogel.fasta" #出力ファイル名を指定してout_fに格納
param <- c(3, 9) #抽出したい範囲の始点と終点を指定

#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 <- subseq(fasta, param[1], param[2]) #paramで指定した始点と終点の範囲の配列を抽出した結果を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
```

## 2. RefSeqのhuman mRNAのmulti-FASTA形式のファイル(h\_rna.fasta)の場合:

[トップページへ](#)

任意のRefSeq ID (例: NM\_203348.1)の任意の範囲 (例: 始点が2, 終点が5)の配列の抽出を行うやり方です。

# コピペ実行2

おさらい。①この項目は、塩基配列のファイルを入力として読み込んで、指定した範囲の配列を切り出して、別のファイルに出力するスクリプトです。②例題1は、先ほどdescription行部分を変更した③sample1.fastaを入力として、④3-9番目の塩基を切り出して、⑤hoge1.fastaというファイル名で保存するスクリプトです。

## イントロ | 一般 | 指定した範囲の配列を取得 | Biostrings

**Biostrings**パッケージ中のsubseq関数を用いて、single-FASTA形式やmulti-FASTA形式ファイルから様々な部分を得るやり方を示します。この項目は、「この染色体の、ここから、ここまで」という指定の仕方になります。ファイルがヒトゲノムだった場合に、chr3の20000から500000の座標の配列取得を行いたい場合などに利用し、chr4とchr8の配列のみ抽出といったやり方には対応していませんのでご注意ください。また、ファイルド時に、\*.fastaという拡張子が\*.txtに勝手に変更されることがありますのでご注意ください。ここでは、「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. (single-)FASTA形式ファイル(sample1.fasta)の場合:

任意の範囲 (始点が3, 終点が9)の配列を抽出するやり方です。

```
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
param <- c(3, 9) #抽出したい範囲の始点と終点を指定

#必要なパッケージを #パッケージの読み込み
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
fasta #確認してるだけです

#本番
fasta <- subseq(fasta, param[1], param[2]) #paramで指定した始点と終点の範囲の配列を抽出した結果をfastaに
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
```

### 2. RefSeqのhuman mRNAのmulti-FASTA形式のファイル(h\_rna.fasta)の場合:

[トップページへ](#)

任意のRefSeq ID (例: NM\_203348.1)の任意の範囲 (例: 始点が2, 終点が5)の配列の抽出を行うやり方です。

# コピペ実行3

おさらい。①この項目は、塩基配列のファイルを入力として読み込んで、指定した範囲の配列を切り出して、別のファイルに出力するスクリプトです。②例題1は、先ほどdescription行部分を変更した③sample1.fastaを入力として、④3-9番目の塩基を切り出して、⑤hoge1.fastaというファイル名で保存するスクリプトです。赤枠内を全選択して(反転させて)、⑥右クリックしてコピー。

## イントロ | 一般 | 指定した範囲の配列を取得 | Biostrings

Biostringsパッケージ中のsubseq関数を用いて、single-FASTA形式やmulti-FASTA形式ファイルから様々な部分を得るやり方を示します。この項目は、「この染色体の、ここから、ここまで」という指定の仕方になります。ファイルがヒトゲノムだった場合に、chr3の20000から500000の座標の配列取得を行いたい場合などに利用が、chr4とchr8の配列のみ抽出といったやり方には対応していませんのでご注意ください。また、ファイルド時に、\*.fastaという拡張子が\*.txtに勝手に変更されることがありますのでご注意ください。ここでは、「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. (single-)FASTA形式ファイル(sample1.fasta)の場合:

任意の範囲(始点が3, 終点が9)の配列を抽出するやり方です。

```
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
param <- c(3, 9)

#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")

#本番
fasta <- subseq(fasta, param[1], param[2])

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

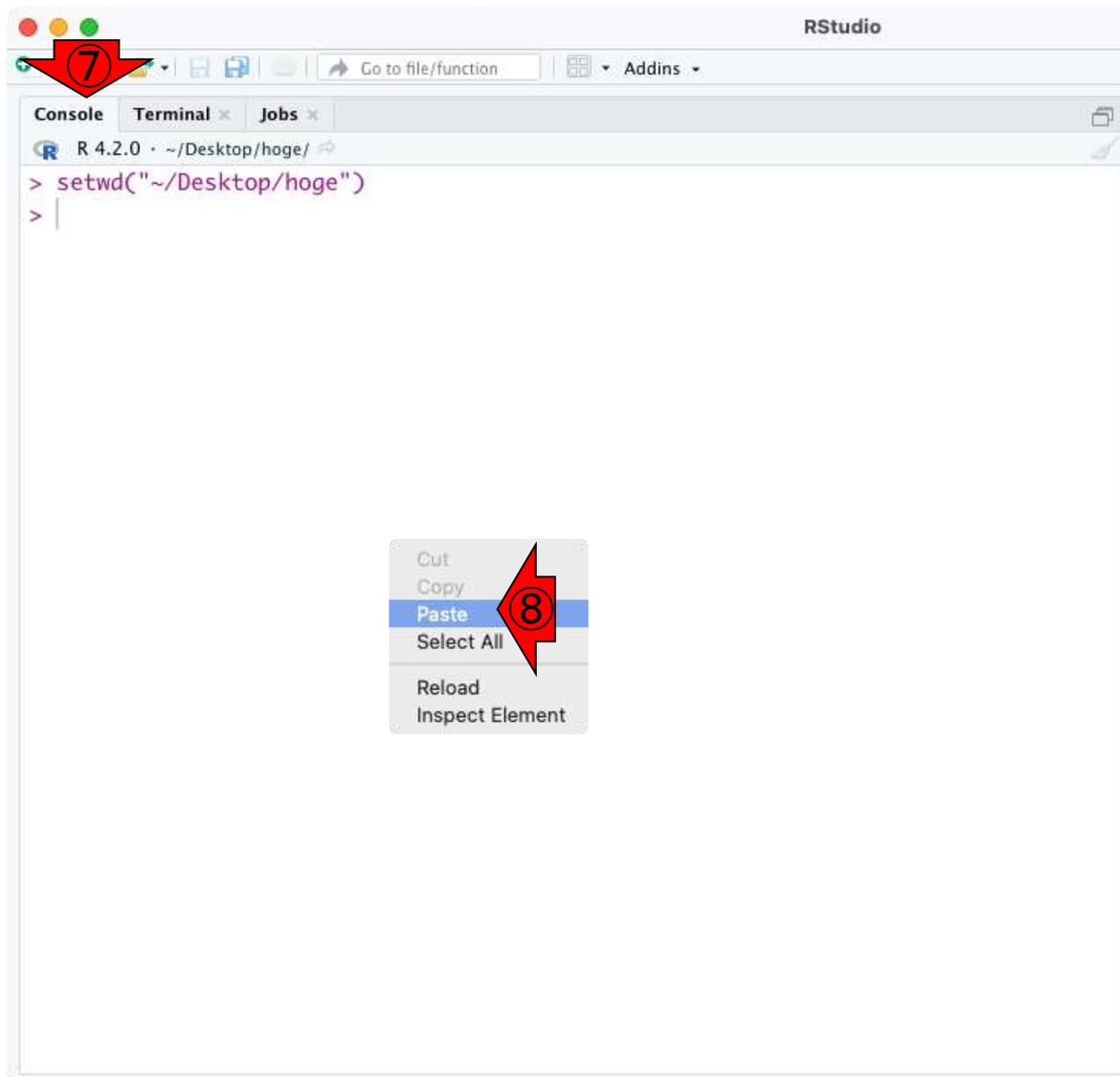


### 2. RefSeqのhuman mRNAのmulti-FASTA形式のファイル(h\_rna.fasta)の場合:

[トップページへ](#)

任意のRefSeq ID (例: NM\_203348.1)の任意の範囲 (例: 始点が2, 終点が5)の配列の抽出を行うやり方です。

# コピペ実行4



おさらい。①この項目は、塩基配列のファイルを入力として読み込んで、指定した範囲の配列を切り出して、別のファイルに出力するスクリプトです。②例題1は、先ほどdescription行部分を変更した③sample1.fastaを入力として、④3-9番目の塩基を切り出して、⑤hoge1.fastaというファイル名で保存するスクリプトです。赤枠内を全選択して(反転させて)、⑥右クリックしてコピー。⑦Consoleがアクティブな状態であることを確認して、⑧右クリックでペースト。(うまくいかないヒトは「⌘ + V」やEditメニューからPasteを選択するといった方法もトライしてください。)

# コピペ実行5

```
RStudio

Console Terminal Jobs
R 4.2.0 · ~/Desktop/hoge/
> setwd("~/Desktop/hoge/")
> 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
  param <- c(3, 9)                #抽出したい範囲の始点と終点を指定

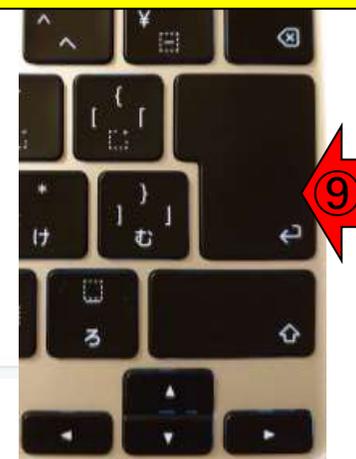
#必要なパッケージをロード
library(Biostrings)              #パッケージの読み込み

#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 <- subseq(fasta, param[1], param[2])
)#paramで指定した始点と終点の範囲の配列を抽出した結果を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
|
```

おさらい。①この項目は、塩基配列のファイルを入力として読み込んで、指定した範囲の配列を切り出して、別のファイルに出力するスクリプトです。②例題1は、先ほどdescription行部分を変更した③sample1.fastaを入力として、④3-9番目の塩基を切り出して、⑤hoge1.fastaというファイル名で保存するスクリプトです。赤枠内を全選択して(反転させて)、⑥右クリックしてコピー。⑦Consoleがアクティブな状態であることを確認して、⑧右クリックでペースト。(うまくいかないヒトは「⌘ + V」やEditメニューからPasteを選択するといった方法もトライしてください。)こんな感じになればOK。⑨リターンキーを押す。



# コピペ実行6

実行完了後の状態。①赤枠内がこんな感じになればOK。出力ファイルである、②hoge1.fastaが作成されていることも分かります。もし①の画面になっていて、②が見られない場合は、③リロードしてみてください。

```
RStudio

Console Terminal Jobs
R 4.2.0 ~/Desktop/hoge/

次のパッケージを付け加えます: 'Biostrings'

以下のオブジェクトは 'package:base' からマスクされています:

  strsplit

>
> #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
DNASTringSet object of length 1:
  width seq                      names
[1]   12 AGTGACGGTCTT            description1
>
> #本番
> fasta <- subseq(fasta, param[1], param[2])#paramで指定した始点と終点の範囲の配列を抽出した結果をfastaに格納
> fasta                                     #確認してるだけです
DNASTringSet object of length 1:
  width seq                      names
[1]    7 TGACGGT                description1
>
> #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
>
> |
```

①

```
fasta <- readDNASTringSet(in_f, format="fasta")
fasta #確認してるだけです
#本番
fasta <- subseq(fasta, param[1], param[2])
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)
```

Files Plots Packages Help Viewer

Home Desktop hoge

Name	Size
..	
sample1.fasta	27 B
hoge1.fasta	22 B

②

③

# コピペ実行7

実行完了後の状態。①赤枠内がこんな感じになればOK。出力ファイルである、②hoge1.fastaが作成されていることも分かります。もし①の画面になっていて、②が見られない場合は、③リロードしてみてください。④のような感じでエラーが出たヒトは、前提条件を満たしていない(この場合はBiostringsパッケージのインストールができてない)ことを意味します。しかるべき作業を行ってください。

```

RStudio

Console Terminal Jobs
R 4.2.0 ~/

>
> #必要なパッケージをロード
> library(Biostrings)
library(Biostrings) でエラー:
  'Biostrings' という名前のパッケージはありません
>
> #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
readDNASTringSet(in_f, format = "fasta") でエラー:
  関数 "readDNASTringSet" を見つけることができませんでした
> fasta
エラー: オブジェクト 'fasta' がありません
>
> #本番
> fasta <- subseq(fasta, param[1], param[2])#paramで指定した始点と終点の範囲の配列を抽出した結果をfastaに格納
subseq(fasta, param[1], param[2]) でエラー:
  関数 "subseq" を見つけることができませんでした
> fasta
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)#fastaの中身を指定したファイル名で保存
writeXStringSet(fasta, file = out_f, format = "fasta", width = 50) でエラー:
  関数 "writeXStringSet" を見つけることができませんでした
>
>

```

```

Files Plots Packages Help Viewer
Home > Desktop > hoge
Name Size
..
sample1.fasta 27 B

```

# Contents

- 前提条件、RStudioの起動
- 見栄えの統一、終了
- Console画面の見栄え、設定変更
- 枠の幅、電卓的な利用
- プロンプト(>)、Console画面クリア
- 計算の中断
- 指定した範囲の配列を取得
  - 項目と例題、入力ファイルの保存
  - 作業ディレクトリの変更
  - 入力ファイルの確認、FASTA形式
  - コピペ実行、出力ファイルの確認、Environmentタブ
- フィルタリング(任意のキーワードを含む行を抽出)
  - 全体像の把握、入力ファイルの保存、コピペ実行、出力ファイルの確認
  - 応用
- 警告メッセージ、ありがちなミス

# 出力ファイルの確認1

①が入力ファイルの情報、②が出力ファイルの情報に対応します。③が抽出したい領域の範囲に相当します。①が抽出元で、②が抽出結果に相当します。

The screenshot displays the RStudio interface with the following components:

- Console:** Shows R commands and their outputs. A red box highlights the output of `readDNASTringSet`, and another red box highlights the output of `subseq`. A red arrow labeled '3' points to the `param` assignment.
- Environment:** Shows the objects created in the environment, including `fasta`.
- Files:** Shows the file explorer with `sample1.fasta` and `hoge1.fasta` files. Red arrows labeled '1' and '2' point to these files.

```
> 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
> param <- c(3, 9) #抽出したい範囲の始点と終点を指定
>
> #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
DNASTringSet object of length 1:
  width seq          names
[1]   12 AGTGACGGTCTT description1
>
> #本番
> fasta <- subseq(fasta, param[1], param[2])#paramで指定した始点と終点の範囲の配列を抽出した結果をfastaに格納
> fasta #確認してるだけです
DNASTringSet object of length 1:
  width seq          names
[1]    7 TGACGGT      description1
>
> #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
>
>
> |
```

# 出力ファイルの確認2

①が入力ファイルの情報、②が出力ファイルの情報に対応します。③が抽出したい領域の範囲に相当します。①が抽出元で、②が抽出結果に相当します。④が抽出領域に相当し、⑤と同じです。

```
R 4.2.0 · ~/Desktop/hoge/

> 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
> param <- c(3, 9)                 #抽出したい範囲の始点と終点を指定
>
> #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
DNASTringSet object of length 1:
  width seq                      names
[1]   12 AGTGACGGTCTT             description1
>
> #本番
> fasta <- subseq(fasta, param[1], param[2])#paramで指定した始点と終点の範囲の配列を抽出した結果をfastaに格納
> fasta                           #確認してるだけです
DNASTringSet object of length 1:
  width seq                      names
[1]    7 TGACGGT                 description1
>
> #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
>
>
> |
```

Files | Plots | Packages | Help | Viewer

Home > Desktop > hoge

Name	Size
..	
sample1.fasta	27 B
hoge1.fasta	22 B

# 出力ファイルの確認3

①が入力ファイルの情報、②が出力ファイルの情報に対応します。③が抽出したい領域の範囲に相当します。①が抽出元で、②が抽出結果に相当します。④が抽出領域に相当し、⑤と同じです。⑥をクリックして、エディタを開いた状態です。⑦切り出された配列は、確かに④と同じ。

The screenshot displays the RStudio interface with the following components:

- Editor:** Shows a FASTA file named 'hoge1.fasta' with the following content:

```
1 |description1
2 |TGACGGT
3 |
```

A red bracket underlines the sequence 'TGACGGT' on line 2, and a red arrow labeled '7' points to it.
- Console:** Shows the R code and its output:

```
> 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
> param <- c(3, 9)                 #抽出したい範囲の始点と終点を指定
>
> #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
DNASTringSet object of length 1:
  width seq          names
[1]   12 AGTGACGGTCTT description1
>
> #本番
```

A red bracket underlines the sequence 'AGTGACGGTCTT' in the output, and a red arrow labeled '4' points to it.
- Files Panel:** Shows the file explorer for the 'hoge' directory, listing 'sample1.fasta' (27 B) and 'hoge1.fasta' (22 B). A red arrow labeled '6' points to 'hoge1.fasta'.

# 出力ファイルの確認4

①が入力ファイルの情報、②が出力ファイルの情報に対応します。③が抽出したい領域の範囲に相当します。①が抽出元で、②が抽出結果に相当します。④が抽出領域に相当し、⑤と同じです。⑥をクリックして、エディタを開いた状態です。⑦切り出された配列は、確かに④と同じ。⑧入力ファイルの方もクリックした状態。⑨のような感じでタブが増えていくだけです。

The screenshot shows the RStudio interface. At the top, there are two tabs: 'hoge1.fasta' and 'sample1.fasta'. A red arrow labeled '9' points to the 'hoge1.fasta' tab. The text editor shows the following content:

```
1 >description1
2 AGTGACGGTCTT
3
```

The console shows the following R code and output:

```
> 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
> param <- c(3, 9)           #抽出したい範囲の始点と終点を指定
>
> #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
DNASTringSet object of length 1:
  width seq                names
[1]   12 AGTGACGGTCTT      description1
>
> #本番
```

The screenshot shows a file explorer window with the following table of files:

Name	Size
..	
sample1.fasta	27 B
hoge1.fasta	22 B

A red arrow labeled '8' points to the 'sample1.fasta' file.

# Contents

- 前提条件、RStudioの起動
- 見栄えの統一、終了
- Console画面の見栄え、設定変更
- 枠の幅、電卓的な利用
- プロンプト(>)、Console画面クリア
- 計算の中断
- 指定した範囲の配列を取得
  - 項目と例題、入力ファイルの保存
  - 作業ディレクトリの変更
  - 入力ファイルの確認、FASTA形式
  - コピペ実行、出力ファイルの確認、Environmentタブ
- フィルタリング(任意のキーワードを含む行を抽出)
  - 全体像の把握、入力ファイルの保存、コピペ実行、出力ファイルの確認
  - 応用
- 警告メッセージ、ありがちなミス

# Environmentタブ1

①Environmentタブをアクティブにした状態です。現時点では、こういうタブもあるのだという程度で良いですが、慣れてくると非常に便利です。

①

The screenshot shows the RStudio interface. The top-left pane displays a text file with the following content:

```
1 >description1
2 AGTGACGGTCTT
3
```

The bottom-left pane shows the R console with the following code and output:

```
> 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
> param <- c(3, 9)                 #抽出したい範囲の始点と終点を指定
>
> #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
DNASTringSet object of length 1:
  width seq                      names
[1]   12 AGTGACGGTCTT             description1
>
> #本番
```

The right-hand side of the RStudio interface shows the Environment tab, which is active. It displays the following information:

- Environment: Global Environment
- Data: fasta (Formal class DNASTri...)
- Values:

in_f	"sample1.fasta"
out_f	"hoge1.fasta"
param	num [1:2] 3 9

The Files pane at the bottom right shows the directory structure:

```
Home > Desktop > hoge
Name      Size
..
sample1.fasta 27 B
hoge1.fasta  22 B
```

# Environmentタブ2

①Environmentタブをアクティブにした状態です。現時点では、こういうタブもあるのだという程度で良いですが、慣れてくると非常に便利です。ここは、シンプルに言えば、「現在利用可能なオブジェクト」がリストアップされている部分です。

The screenshot shows the RStudio interface. The top-left pane displays a text file named 'sample1.fasta' with the following content:

```
1 >description1
2 AGTGACGGTCTT
3
```

The bottom-left pane shows the R console with the following code and output:

```
> 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
> param <- c(3, 9) #抽出したい範囲の始点と終点を指定
>
> #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
DNASTringSet object of length 1:
  width seq          names
[1]  12 AGTGACGGTCTT description1
>
> #本番
```

The right-hand pane shows the Environment tab with the following data:

Data	
fasta	Formal class DNASTri...
Values	
in_f	"sample1.fasta"
out_f	"hoge1.fasta"
param	num [1:2] 3 9

The bottom-right pane shows the Files pane with the following content:

Name	Size
..	
sample1.fasta	27 B
hoge1.fasta	22 B

# Environmentタブ3

例えば、②は切り出したい領域の始点と終点を、2つの要素からなる数値ベクトルとして表現し、それをparamという名前で利用できるようにするコマンドです。それぞれの独立した整数を連結 (concatenate) すべく、③cという関数で連結してスカラーをベクトルにしています。

The screenshot shows the RStudio interface. The top editor pane displays a FASTA file with the following content:

```
1 >description1
2 AGTGACGGTCTT
3
```

The bottom console pane shows the following R code and output:

```
> 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
> param <- c(3, 9) #抽出したい範囲の始点と終点を指定
>
> #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
DNASTringSet object of length 1:
  width seq          names
[1]   12 AGTGACGGTCTT description1
>
> #本番
```

The environment browser on the right shows the following table:

fasta	Formal class DNASTri...
Values	
in_f	"sample1.fasta"
out_f	"hoge1.fasta"
param	num [1:2] 3 9

The file browser at the bottom right shows the directory structure:

```
Home > Desktop > hoge
├── ..
├── sample1.fasta 27 B
└── hoge1.fasta 22 B
```

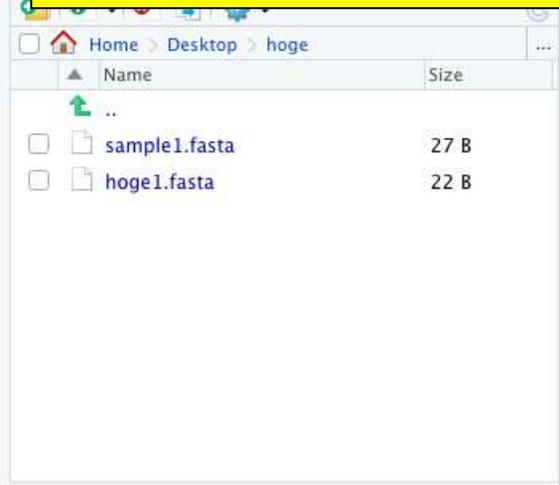
# Environmentタブ4

例えば、②は切り出したい領域の始点と終点を、2つの要素からなる数値ベクトルとして表現し、それをparamという名前で見られるようにするコマンドです。それぞれの独立した整数を連結 (concatenate) するべく、③cという関数で連結してスカラーをベクトルにしています。④が左向きの矢印っぽくなっていることからある程度予想できますが、「右側の情報を左側のものに代入せよ」という意味です。プログラミングの世界では、通常この向きで物事を表現しますので、「<-」の代わりに「=」が用いられることもあります。

```
hoge1.fasta x sample1.fasta x
1 >description1
2 AGTGACGGTCTT
3

3:1 Text file

Console Terminal Jobs
R 4.2.0 ~/Desktop/hoge/
> 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
> param <- c(3, 9) #抽出したい範囲の始点と終点を指定
> #必要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
DNASTringSet object of length 1:
  width seq          names
[1]  12 AGTGACGGTCTT description1
> #本番
```



# Environmentタブ5

それゆえ、⑤paramというオブジェクトの中身は、2つの要素からなる数値ベクトルです。1番目の要素が3で、2番目の要素が9です。それを踏まえると、⑥の意味がよくわかると思います。

The screenshot shows the RStudio interface with the following components:

- Editor:** Contains two files: `hoge1.fasta` and `sample1.fasta`. The content of `hoge1.fasta` is:

```
1 >description1
2 AGTGACGGTCTT
3
```
- Console:** Shows the execution of R code:

```
R 4.2.0 ~./Desktop/hoge/
> 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
> param <- c(3, 9) #抽出したい範囲の始点と終点を指定
>
> #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
DNASTringSet object of length 1:
  width seq          names
[1]  12 AGTGACGGTCTT description1
>
> #本番
> fasta <- subseq(fasta, param[1], param[2])#paramで指定した始点と終点の範囲の配列を抽出した結果をfastaに格納
> fasta #確認してるだけです
DNASTringSet object of length 1:
  width seq          names
[1]   7 TGACGGT      description1
```
- Environment:** Shows the `Global Environment` with the following variables:

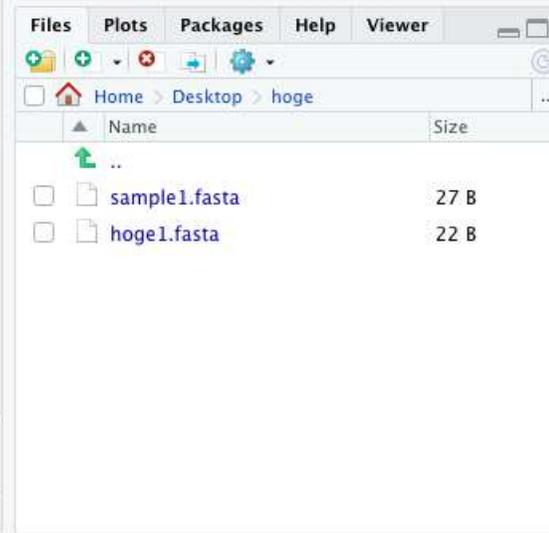
Variable	Value
<code>fasta</code>	Formal class DNASTri...
<code>in_f</code>	"sample1.fasta"
<code>out_f</code>	"hoge1.fasta"
<code>param</code>	num [1:2] 3 9
- Files:** Shows the file explorer for the `hoge` directory on the desktop, containing `sample1.fasta` (27 B) and `hoge1.fasta` (22 B).

Red arrows labeled ⑤ and ⑥ point to the `param` variable in the console and the `param` row in the environment window, respectively.

# Environmentタブ6

それゆえ、⑤paramというオブジェクトの中身は、2つの要素からなる数値ベクトルです。1番目の要素が3で、2番目の要素が9です。それを踏まえると、⑥の意味がよくわかると思います。⑦paramオブジェクトの1番目の要素はparam[1]、⑧2番目の要素はparam[2]として表現できます。⑨は部分配列(subsequence)の抽出を行うsubseq関数を用いて、⑩の情報を、⑪で入力配列、⑦で始点、⑧で終点を与えて目的の部分配列を抽出していると読み解けば良いです。

```
RStudio
Go to file/function | Addins
hoge1.fasta x sample1.fasta x
1 >description1
2 AGTGACGGTCTT
3
3:1 | Text file
Console Terminal x Jobs x
R 4.2.0 ~ ~/Desktop/hoge/
> 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
> param <- c(3, 9) #抽出したい範囲の始点と終点を指定
>
> #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
DNASTringSet object of length 1:
  width seq names
[1] 12 AGTGACGGTCTT description1
>
> #本番
> fasta <- subseq(fasta, param[1], param[2])#paramで指定した始点と終点の範囲の配列を抽出した結果をfastaに格納
> fasta #確認してるだけです
DNASTringSet object of length 1:
  width seq names
[1] 7 TGACGGT description1
```



# Environmentタブ7

```
RStudio
Go to file/function | Addins
Console Terminal Jobs
R 4.2.0 · ~/Desktop/hoge/
> 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
> param <- c(3, 9) #抽出したい範囲の始点と終点を指定
>
> #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
DNASTringSet object of length 1:
  width seq names
[1] 12 AGTGACGGTCTT description1
>
> #本番
> fasta <- subseq(fasta, param[1], param[2])#paramで指定した始点と終点の範囲の配列を抽出した結果をfastaに格納
> fasta #確認してるだけです
DNASTringSet object of length 1:
  width seq names
[1] 7 TGACGGT description1
>
> #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
>
>
>
```

それゆえ、⑤paramというオブジェクトの中身は、2つの要素からなる数値ベクトルです。1番目の要素が3で、2番目の要素が9です。それを踏まえると、⑥の意味がよくわかると思います。⑦paramオブジェクトの1番目の要素はparam[1]、⑧2番目の要素はparam[2]として表現できます。⑨は部分配列(subsequence)の抽出を行うsubseq関数を用いて、⑩の情報を、⑪で入力配列、⑦で始点、⑧で終点を与えて目的の部分配列を抽出していると読み解けば良いです。Rのエディタ部分を削除した状態。⑩と⑫は同じオブジェクト名であるにも関わらず、中身が異なるのは、⑨subseq関数の実行結果を同じオブジェクト名(fasta)でほぞんしているからだと理解すれば良いです。

# Environmentタブ8

①をクリックするとfastaオブジェクトの中身が見られますが、話がややこしくなるのと、まだ難解な事柄であるため無視して良いです。②は、対応関係が理解しやすいと思います。

The screenshot shows the RStudio interface with the following components:

- Console:** Displays R code and its output. The code defines variables for file names and parameters, reads a FASTA file, and writes a subset to a new file. The output shows the contents of the FASTA file and the subset.
- Environment:** Shows the current environment with variables `in_f`, `out_f`, and `param`. A red arrow labeled ① points to the `fasta` object in the Data pane. A red arrow labeled ② points to the values of `in_f` and `out_f` in the Values pane.
- Files:** Shows the file explorer with `sample1.fasta` and `hoge1.fasta` files.

```
> 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
> param <- c(3, 9) #抽出したい範囲の始点と終点を指定
>
> #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
DNASTringSet object of length 1:
  width seq          names
[1]   12 AGTGACGGTCTT description1
>
> #本番
> fasta <- subseq(fasta, param[1], param[2])#paramで指定した始点と終点の範囲の配列を抽出した結果をfastaに格納
> fasta #確認してるだけです
DNASTringSet object of length 1:
  width seq          names
[1]    7 TGACGGT      description1
>
> #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
>
>
>
```

# Environmentタブ9

①をクリックするとfastaオブジェクトの中身が見られますが、話がややこしくなるのと、まだ難解な事柄であるため無視して良いです。②は、対応関係が理解しやすいと思います。理由は、③で自分が指定している内容だからです。



## イントロ | 一般 | 指定した範囲の配列を取得 | Biostrings

**Biostrings**パッケージ中のsubseq関数を用いて、single-FASTA形式やmulti-FASTA形式ファイルから様々な部分配列を取得するやり方を示します。この項目は、「この染色体の、ここから、ここまで」という指定の仕方になります。例えば入力ファイルがヒトゲノムだった場合に、chr3の20000から500000の座標の配列取得を行いたい場合などに利用します。したがって、chr4とchr8の配列のみ抽出といったやり方には対応していませんのでご注意ください。また、ファイルダウンロード時に、\*.fastaという拡張子が\*.txtに勝手に変更されることがありますのでご注意ください。ここでは、「ファイル」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

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

任意の範囲 (始点が3, 終点が9)の配列を抽出するやり方です。

```
in_f <- "sample1.fasta" }
out_f <- "hogel.fasta" }
param <- c(3, 9) }

#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
fasta

#本番
fasta <- subseq(fasta, param[1], param[2]) #paramで指定した始点と終点の範囲の配列を抽出した結果をfastaに
fasta

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



### 2. RefSeqのhuman mRNAのmulti-FASTA形式のファイル(h\_rna.fasta)の場合:

[トップページへ](#)

任意のRefSeq ID (例: NM\_203348.1)の任意の範囲 (例: 始点が2, 終点が5)の配列の抽出を行うやり方です。

# Contents

- 前提条件、RStudioの起動
- 見栄えの統一、終了
- Console画面の見栄え、設定変更
- 枠の幅、電卓的な利用
- プロンプト(>)、Console画面クリア
- 計算の中断
- 指定した範囲の配列を取得
  - 項目と例題、入力ファイルの保存
  - 作業ディレクトリの変更
  - 入力ファイルの確認、FASTA形式
  - コピペ実行、出力ファイルの確認、Environmentタブ
- フィルタリング(任意のキーワードを含む行を抽出)
  - 全体像の把握、入力ファイルの保存、コピペ実行、出力ファイルの確認
  - 応用
- 警告メッセージ、ありがちなミス

①で行う内容自体は非常に原始的なものであり、実際には同じ目的を果たす、より美しい手段が存在します。しかし、最初のうちは原始的な方法の方が理解しやすいと思います。②のあたりもしっかり理解し、Rを使う他の先生の講義の中身に集中できるようにしておきましょう。



# 全体像の把握1

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	gene name	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

出力:hoge1.txt



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

入力:リストファイル(genelist1.txt)

	A
1	gene1
2	gene7
3	gene9

# 全体像の把握2

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。①を押して一番上に移動。

## イントロ | 一般 | 指定した範囲の配列を取得 | Biostrings

**Biostrings**パッケージ中のsubseq関数を用いて、single-FASTA形式やmulti-FASTA形式ファイルから様々な部分配列を取得するやり方を示します。この項目は、「この染色体の、ここから、ここまで」という指定の仕方になります。例えば入力ファイルがヒトゲノムだった場合に、chr3の20000から500000の座標の配列取得を行いたい場合などに利用します。したがって、chr4とchr8の配列のみ抽出といったやり方には対応していませんのでご注意ください。また、ファイルダウンロード時に、\*.fastaという拡張子が\*.txtに勝手に変更されることがありますのでご注意ください。ここでは、「ファイル」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

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

任意の範囲(始点が3, 終点が9)の配列を抽出するやり方です。

```
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
param <- c(3, 9)                 #抽出したい範囲の始点と終点を指定

#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 <- subseq(fasta, param[1], param[2]) #paramで指定した始点と終点の範囲の配列を抽出した結果をfastaに
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
```

### 2. RefSeqのhuman mRNAのmulti-FASTA形式のファイル(h\_rna.fasta)の場合:

[トップページへ](#)

任意のRefSeq ID (例: NM\_203348.1)の任意の範囲(例: 始点が2, 終点が5)の配列の抽出を行うやり方です。

# 全体像の把握3

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。①を押して一番上に移動。ここから②下矢印キーを押して、、、

## (Rで)塩基配列解析

(last modified 2022/06/01, since 2010)

このウェブページの多くは、[インストール | についての推奨手順](#) (Windows2022.05.16版とMacintosh2022.05.17版)に従ってフリーソフトRと必要なパッケージをインストール済みであるという前提で記述しています。初心者の方は[基本的な利用法](#)(Windows2022.05.17版のPPTXとPDF；Macintosh2020.03.13版のPPTXとPDF)で自習してください。

[@Agriblio\\_utokyo](#)さんをフォロー

[アグリバイオ](#)、[\(Rで\)塩基配列解析のサブページ](#)、[生命情報解析研究室](#)

### What's new? ([過去のお知らせはこちら](#))

- このページをGoogle ChromeやMicrosoft Edgeで開いても、すぐにフリーズして「応答なし」と表示されることが頻繁にあったという報告をいただきました。このような事象に遭遇した方はブラウザをFirefoxに変更すると解消されるようです。情報提供いただいた学生さんに感謝m(\_ \_)m(2022/06/01) **NEW**
- [\(サブページ\)のほうのネタではありますが...](#) 日本乳酸菌学会誌の[第16回](#)と[第17回](#)の原稿をこちらでも公開しました。(2022/05/29) **NEW**
- 「インストール | R本体とRStudio | 最新版 | [Mac用](#)」を更新しました。(2022/05/16)

- [はじめに](#) (last modified 2022/03/28)
- [過去のお知らせ](#) (last modified 2022/06/01) **NEW**
- [インストール | について](#) (last modified 2022/05/17)
- インストール | R本体とRStudio | 最新版 | [Win用](#) (last modified 2022/05/16) 推奨
- インストール | R本体とRStudio | 最新版 | [Mac用](#) (last modified 2022/05/16) 推奨

[トップページへ](#)



# 全体像の把握4

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。①を押して一番上に移動。ここから②下矢印キーを押して、③さきほど行った項目の上部にある、④をクリック。

(Rで)塩基配列解析

iu.a.u-tokyo.ac.jp/kadota/r\_seq.html#

- インストール | Rパッケージ | [個別\(2018年11月以前\)](#) (last modified 2018/11/12)
- [基本的な利用法](#) (last modified 2022/05/17)
- [サンプルデータ](#) (last modified 2019/09/06)
- 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 | [任意の文字列を最初の行に挿入](#) (last modified 2014/07/17)
- イントロ | 一般 | [任意のキーワードを含む行を抽出\(基礎\)](#) (last modified 2019/04/24) **④**
- 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 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 | について](#) (last modified 2019/09/06)
- イントロ | 一般 | 指定した範囲の配列を取得 | [Biostring](#) (last modified 2022/04/26)
- 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 | について](#) (last modified 2019/09/06)
- イントロ | 一般 | 翻訳配列(translate)を取得 | [Biostrings](#) **③** (last modified 2015/09/12)
- イントロ | 一般 | 翻訳配列(translate)を取得 | [seqinr\(Chantal 2005\)](#) (last modified 2015/03/09)
- イントロ | 一般 | [相補鎖\(complement\)を取得](#) (last modified 2019/03/10)
- イントロ | 一般 | [逆相補鎖\(reverse complement\)を取得](#) (last modified 2019/03/10)
- イントロ | 一般 | [逆鎖\(reverse\)を取得](#) (last modified 2019/03/10)
- イントロ | 一般 | k-mer解析 | k=1(塩基ごとの出現頻度解析) | [Biostrings](#) (last modified 2016/04/27)
- イントロ | 一般 | k-mer解析 | k=2(2連続塩基の出現頻度解析) | [Biostrings](#) (last modified 2016/01/28)
- イントロ | 一般 | k-mer解析 | k=3(3連続塩基の出現頻度解析) | [Biostrings](#) (last modified 2016/01/28)
- イントロ | 一般 | k-mer解析 | k=n(n連続塩基の出現頻度解析) | [Biostrings](#) (last modified 2016/05/01)
- 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 2017/04/11)

[トップページへ](#)

# 全体像の把握5

こんな感じになります。①例題1が一番上になるように、5行分ほど下にスクロール。



## 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。

① **ファイル** - 「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

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

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

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

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

2. 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([genelist2.txt](#))中のものが含まれる行全体を出力したい場合：

```
in_f1 <- "annotation.txt" #入力ファイル名を指定してin_f1に格納(アノテーションファイル)
in_f2 <- "genelist2.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)
```

# 全体像の把握6

こんな感じになります。①例題1が一番上になるように、5行分ほど下にスクロール。例題1が表示部分の一番上になりました。

(Rで)塩基配列解析

iu.a.u-tokyo.ac.jp/kadota/r\_seq.html#intro\_general\_grep\_kiso

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

```
in_f1 <- "annotation.txt"      #入力ファイル名を指定してin_f1に格納(アノテーションファイル)
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", quote="") #in_f1で指定したファイルの読み込み
keywords <- readLines(in_f2) #in_f2で指定したファイルの読み込み
dim(data)                  #オブジェクトdataの行数と列数を表示

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

2. 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([genelist2.txt](#))中のものが含まれる行全体を出力したい場合：

```
in_f1 <- "annotation.txt"      #入力ファイル名を指定してin_f1に格納(アノテーションファイル)
in_f2 <- "genelist2.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 <- "hoge2.txt"          #出力ファイル名を指定してout_fに格納
param <- 1                    #アノテーションファイル中の検索したい列番号を指定

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

# Contents

- 前提条件、RStudioの起動
- 見栄えの統一、終了
- Console画面の見栄え、設定変更
- 枠の幅、電卓的な利用
- プロンプト(>)、Console画面クリア
- 計算の中断
- 指定した範囲の配列を取得
  - 項目と例題、入力ファイルの保存
  - 作業ディレクトリの変更
  - 入力ファイルの確認、FASTA形式
  - コピペ実行、出力ファイルの確認、Environmentタブ
- フィルタリング(任意のキーワードを含む行を抽出)
  - 全体像の把握、入力ファイルの保存、コピペ実行、出力ファイルの確認
  - 応用
- 警告メッセージ、ありがちなミス

# 入力ファイルの保存1

こんな感じになります。①例題1が一番上になるように、5行分ほど下にスクロール。例題1が表示部分の一番上になりました。②が入力ファイルであり、これらは③と④からダウンロード可能なので、右クリックで「デスクトップ上にあるhogeフォルダ」に保存。

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

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

2. 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([genelist2.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 <- "genelist2.txt"
out_f <- "hoge2.txt"
param <- 1

#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
```

# 入力ファイルの保存2

正しく保存できれば、RStudio上の① hogeフォルダは、②のように見えていると思います。③必要に応じてリロードも行ってください。④Historyタブをクリック、

The screenshot shows the RStudio interface with the following components:

- Console:** R 4.2.0 · ~/Desktop/hoge/
- Code Editor:** Contains R code for reading and saving FASTA files. The code defines variables for input/output files and a range, reads a FASTA file, and saves a subsequence.
- Environment/History:** Shows the 'fasta' object as a 'Formal class DNASTri...' and the 'param' object as a 'num [1:2] 3 9'. A red arrow labeled '4' points to the History tab.
- Files Panel:** Shows the file explorer for the 'hoge' directory. It lists files: 'sample1.fasta' (27 B), 'hoge1.fasta' (22 B), 'annotation.txt' (383 B), and 'genelist1.txt' (18 B). A red arrow labeled '1' points to the 'hoge' folder, and a red arrow labeled '2' points to the list of files.
- Viewer Panel:** A red arrow labeled '3' points to the Viewer tab.

```
> 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
> param <- c(3, 9) #抽出したい範囲の始点と終点を指定
>
> #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
DNASTringSet object of length 1:
  width seq          names
[1]   12 AGTGACGGTCTT description1
>
> #本番
> fasta <- subseq(fasta, param[1], param[2])#paramで指定した始点と終点の範囲の配列を抽出した結果をfastaに格納
> fasta #確認してるだけです
DNASTringSet object of length 1:
  width seq          names
[1]    7 TGACGGT      description1
>
> #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
>
>
>
```



# 入力ファイルの保存4

正しく保存できれば、RStudio上の① hogeフォルダは、②のように見えていると思います。③必要に応じてリロードも行ってください。④Historyタブをクリック、⑤をクリック、⑥Historyをクリアします。⑦をクリックして、Console画面もクリアした状態です。

The screenshot shows the RStudio interface with the following elements:

- File Explorer:** Shows the path `Home > Desktop > hoge`. The file list is as follows:

Name	Size
..	
sample1.fasta	27 B
hoge1.fasta	22 B
annotation.txt	383 B
genelist1.txt	18 B

Red arrows indicate the following steps:

- ①: Points to the `hoge` folder in the file explorer.
- ②: Points to the list of files in the file explorer.
- ③: Points to the `Viewer` tab in the top right.
- ⑦: Points to the `Clear` button in the console area.

# 入力ファイルの保存5

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、という目的を達成する上で必要な①2つの入力ファイルが作業ディレクトリ上に存在するのでready-to-analyze。

入力:アノテーションファイル(annotation.txt)

	A	B	C	D
1	gene name	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

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

# Contents

- 前提条件、RStudioの起動
- 見栄えの統一、終了
- Console画面の見栄え、設定変更
- 枠の幅、電卓的な利用
- プロンプト(>)、Console画面クリア
- 計算の中断
- 指定した範囲の配列を取得
  - 項目と例題、入力ファイルの保存
  - 作業ディレクトリの変更
  - 入力ファイルの確認、FASTA形式
  - コピペ実行、出力ファイルの確認、Environmentタブ
- フィルタリング(任意のキーワードを含む行を抽出)
  - 全体像の把握、入力ファイルの保存、コピペ実行、出力ファイルの確認
  - 応用
- 警告メッセージ、ありがちなミス

# コピー実行1



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

```

in_f1 <- "annotation.txt"           #入力ファイル名を指定してin_f1に格納(アノテーションファイル)
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", quote="") #in_f1で指定したファイルの読み込み
keywords <- readLines(in_f2)       #in_f2で指定したファイルの読み込み
dim(data)                          #オブジェクトdataの行数と列数を表示

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

```

2. 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([genelist2.txt](#))中のものが含まれる行全体を出力したい場合：

```

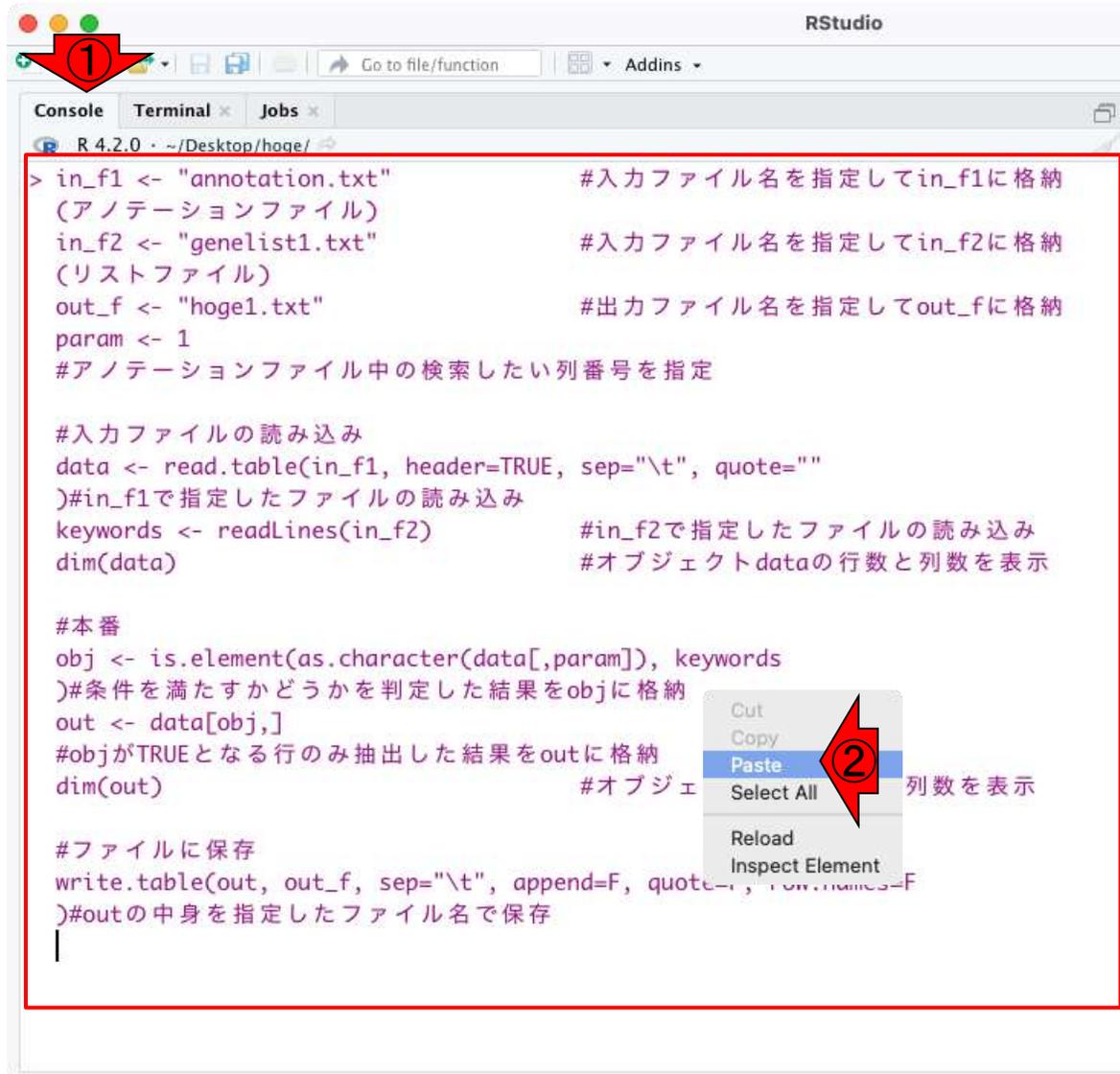
in_f1 <- "annotation.txt"           #入力ファイル名を指定してin_f1に格納(アノテーションファイル)
in_f2 <- "genelist2.txt"           #入力ファイル名を指定してin_f2に格納(リストファイル)
out_f <- "hoge2.txt"               #出力ファイル名を指定してout_fに格納
param <- 1                          #アノテーションファイル中の検索したい列番号を指定

#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トップページへ
keywords <- readLines(in_f2)       #in_f2で指定したファイルの読み込み

```

# コピペ実行2

赤枠内をコピー実行。①RStudioのConsole画面上で、②(右クリック、Pasteでは動作せず、Editメニューから)Paste直後の状態。③リターンキーを押す。

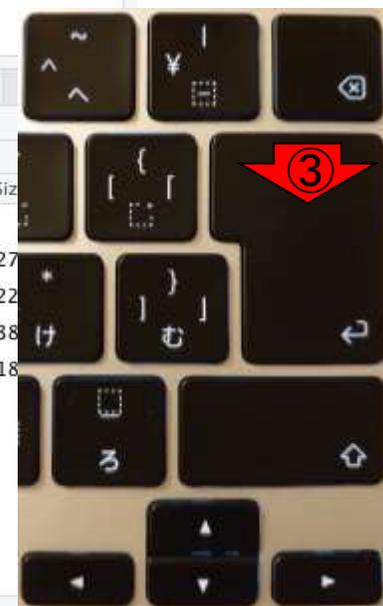
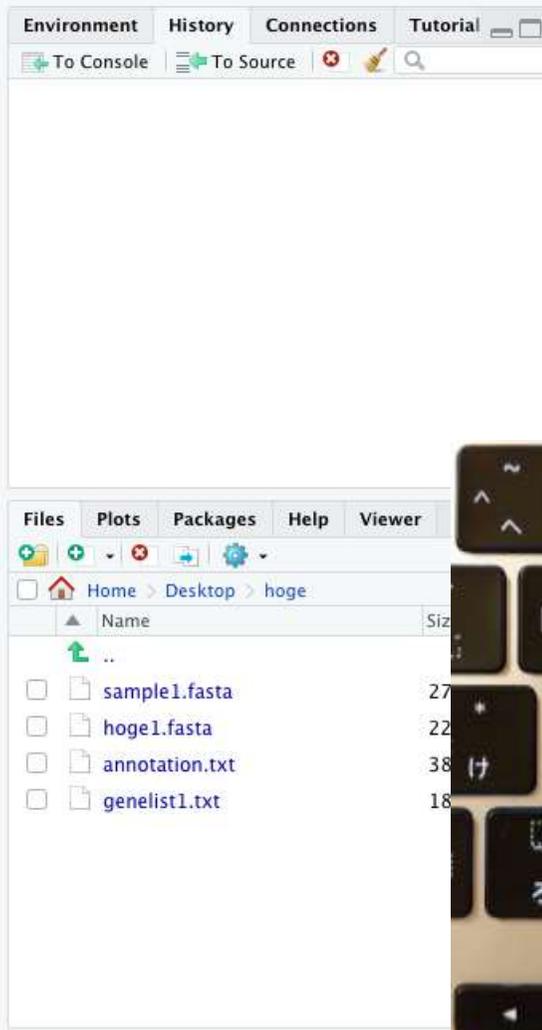


```
> in_f1 <- "annotation.txt"      #入力ファイル名を指定してin_f1に格納
(アノテーションファイル)
in_f2 <- "genelist1.txt"       #入力ファイル名を指定してin_f2に格納
(リストファイル)
out_f <- "hoge1.txt"          #出力ファイル名を指定してout_fに格納
param <- 1
#アノテーションファイル中の検索したい列番号を指定

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

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

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



# コピペ実行3

赤枠内をコピー実行。①RStudioのConsole画面上で、②(右クリック、Pasteでは動作せず、Editメニューから)Paste直後の状態。③リターンキーを押す。こんな感じになって、④指定した出力ファイル(hoge1.txt)が作成されれば基本成功です。④をクリック。

```
RStudio
Go to file/function
Addins
Console Terminal Jobs
R 4.2.0 · ~/Desktop/hoge/
> 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", quote="")#in_f1で指定したファイルの読み込み
> keywords <- readLines(in_f2) #in_f2で指定したファイルの読み込み
> dim(data) #オブジェクトdataの行数と列数を表示
[1] 11 4
>
> #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
>
> |
```

```
data <- read.table(in_f1, header=TRUE, sep="\t", quote="")
keywords <- readLines(in_f2) #in_f2で指定したファイルの読み込み
dim(data) #オブジェクトdataの行数と列数を表示
#本番
obj <- is.element(as.character(data[,param]), keywords)#条件を満たすかどうかを判定した結果を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
#ファイルに保存
write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F)
```

Files Plots Packages Help Viewer

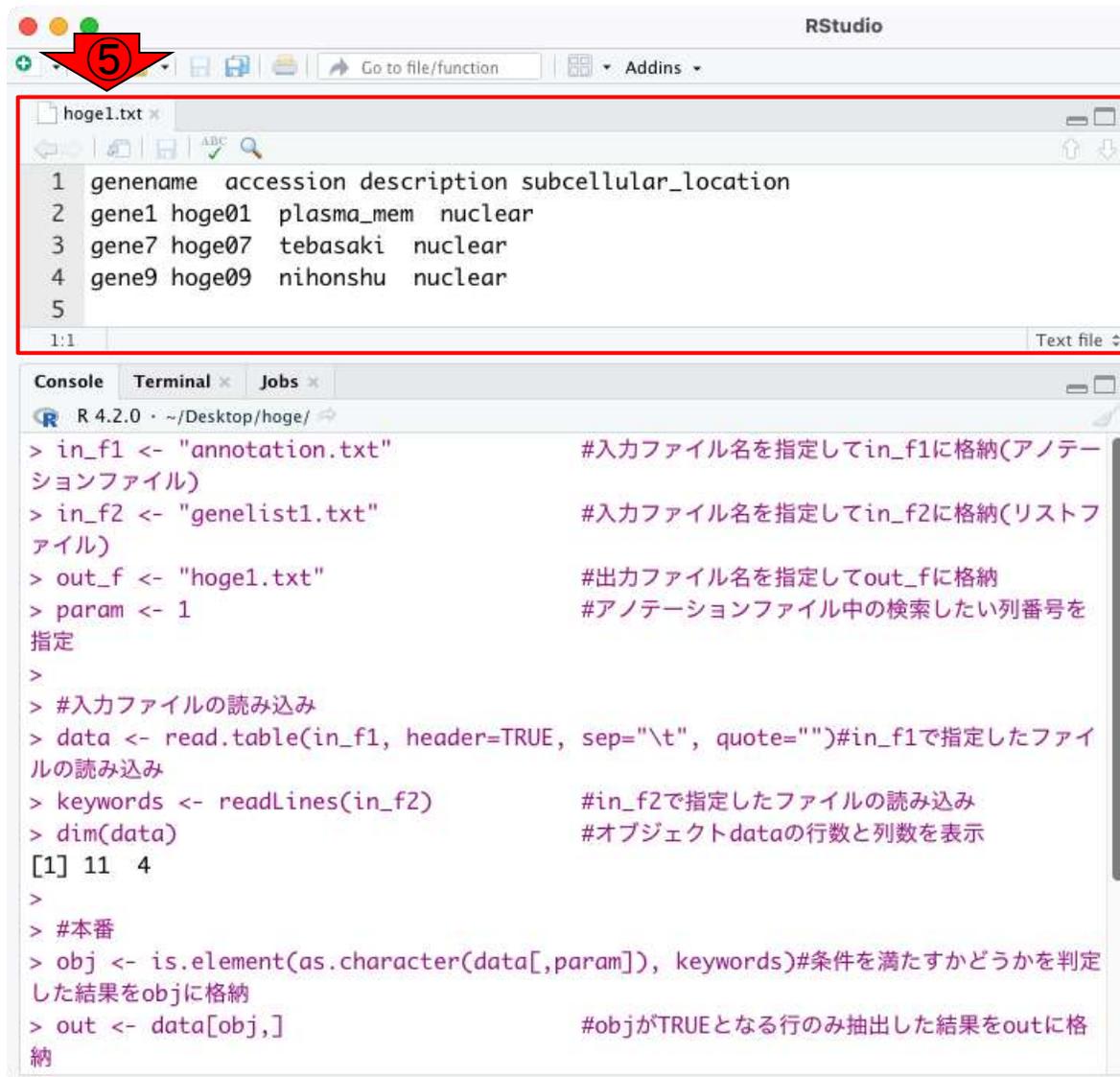
Home > Desktop > hoge

Name	Size
..	
sample1.fasta	27 B
hoge1.fasta	22 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	144 B



# 出力ファイルの確認1

赤枠内をコピー実行。①RStudioのConsole画面上で、②(右クリック、Pasteでは動作せず、Editメニューから)Paste直後の状態。③リターンキーを押す。こんな感じになって、④指定した出力ファイル(hoge1.txt)が作成されれば基本成功です。④をクリック。⑤のような感じで出力ファイルの中身が見られます。

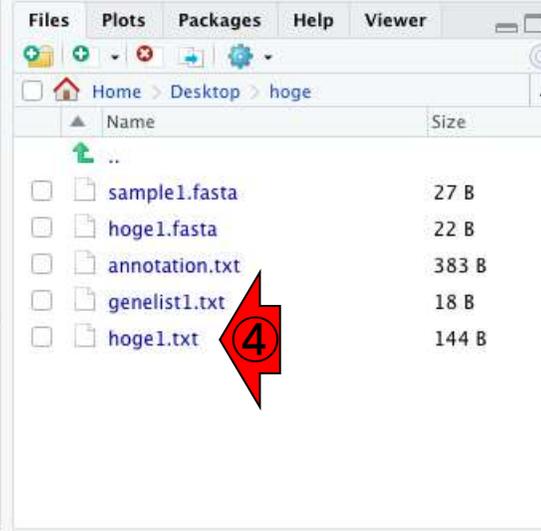


```
hoge1.txt
1  geneaccessiondescriptionsubcellular_location
2  gene1  hoge01  plasma_mem  nuclear
3  gene7  hoge07  tebasaki    nuclear
4  gene9  hoge09  nihonshu    nuclear
5

1:1 Text file

Console Terminal Jobs
R 4.2.0 ~/Desktop/hoge/
> 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", quote="")#in_f1で指定したファイルの読み込み
> keywords <- readLines(in_f2) #in_f2で指定したファイルの読み込み
> dim(data) #オブジェクトdataの行数と列数を表示
[1] 11 4
>
> #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
```

```
keywords <- readLines(in_f2) #in_f2...
dim(data) #オブジェクトdataの行数と列数...
#本番
obj <- is.element(as.character(data[...
out <- data[obj,] #objがTRUEとなる行の...
dim(out) #オブジェクトoutの行数と列数を...
#ファイルに保存
write.table(out, out_f, sep="\t", ap...
```



Name	Size
..	
sample1.fasta	27 B
hoge1.fasta	22 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	144 B

# 出力ファイルの確認2

入力: アノテーションファイル( `annotation.txt` )

	A	B	C	D
1	gene name	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

赤枠内をコピー実行。①RStudioのConsole画面上で、②(右クリック、Pasteでは動作せず、Editメニューから)Paste直後の状態。③リターンキーを押す。こんな感じになって、④指定した出力ファイル(`hoge1.txt`)が作成できれば基本成功です。④をクリック。⑤のような感じで出力ファイルの中身が見られます。最初に見せたものと同じ結果が得られていますね。

出力: `hoge1.txt`

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

# Contents

- 前提条件、RStudioの起動
- 見栄えの統一、終了
- Console画面の見栄え、設定変更
- 枠の幅、電卓的な利用
- プロンプト(>)、Console画面クリア
- 計算の中断
- 指定した範囲の配列を取得
  - 項目と例題、入力ファイルの保存
  - 作業ディレクトリの変更
  - 入力ファイルの確認、FASTA形式
  - コピペ実行、出力ファイルの確認、Environmentタブ
- フィルタリング(任意のキーワードを含む行を抽出)
  - 全体像の把握、入力ファイルの保存、コピペ実行、出力ファイルの確認
  - 応用
- 警告メッセージ、ありがちなミス

# 応用1

①例題1は、②の中身をもつ、③リストファイル(hoge1.txt)を読み込んで、④1列目で一致する行を抽出するスクリプトでした。

① (Rで)塩基配列解析

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

```
in_f1 <- "annotation.txt" #入力ファイル名を指定してin_f1に格納(アノテーションファイル)
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 #検索したい列番号を指定

#入力ファイルを読み込み
data <- read.table(in_f1, header=TRUE, as.is=T) #in_f1で指定したファイルの読み込み
keywords <- readLines(in_f2) #リストファイルの読み込み
dim(data) #行数と列数を表示

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

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

gene1 ↓  
gene7 ↓  
gene9 ↓  
|

2. 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(genelist2.txt)中のものが含まれる行全体を出力したい場合：

```
in_f1 <- "annotation.txt" #入力ファイル名を指定してin_f1に格納(アノテーションファイル)
in_f2 <- "genelist2.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 <- "hoge2.txt" #出力ファイル名を指定してout_fに格納
param <- 1 #検索したい列番号を指定

#入力ファイルを読み込み
data <- read.table(in_f1, header=TRUE, as.is=T) #in_f1で指定したファイルの読み込み
keywords <- readLines(in_f2) #リストファイルの読み込み
dim(data) #行数と列数を表示

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

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

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

# 応用2

①例題1は、②の中身をもつ、③リストファイル(hoge1.txt)を読み込んで、④1列目で一致する行を抽出するスクリプトでした。応用として、⑤新たに作成したリストファイル(list.txt)を読み込んで、⑥4列目で一致する行の抽出を行います。

① 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(genelist)の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 <- "genelist.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 #検索したい列番号を指定

#入力ファイルを読み込み
data <- read.table(in_f1, header=TRUE, as.is=T) #in_f1で指定したファイルの読み込み
keywords <- readLines(in_f2) #リストファイルの読み込み
dim(data) #行数と列数を表示

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

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

gene1 ↓  
gene7 ↓  
gene9 ↓  
| ↓

nuclear ↓  
membrane ↓  
| ↓

2. 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(genelist)の含まれる行全体を出力したい場合:

```

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

#入力ファイルを読み込み
data <- read.table(in_f1, header=TRUE, as.is=T)
keywords <- readLines(in_f2)
dim(data)

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

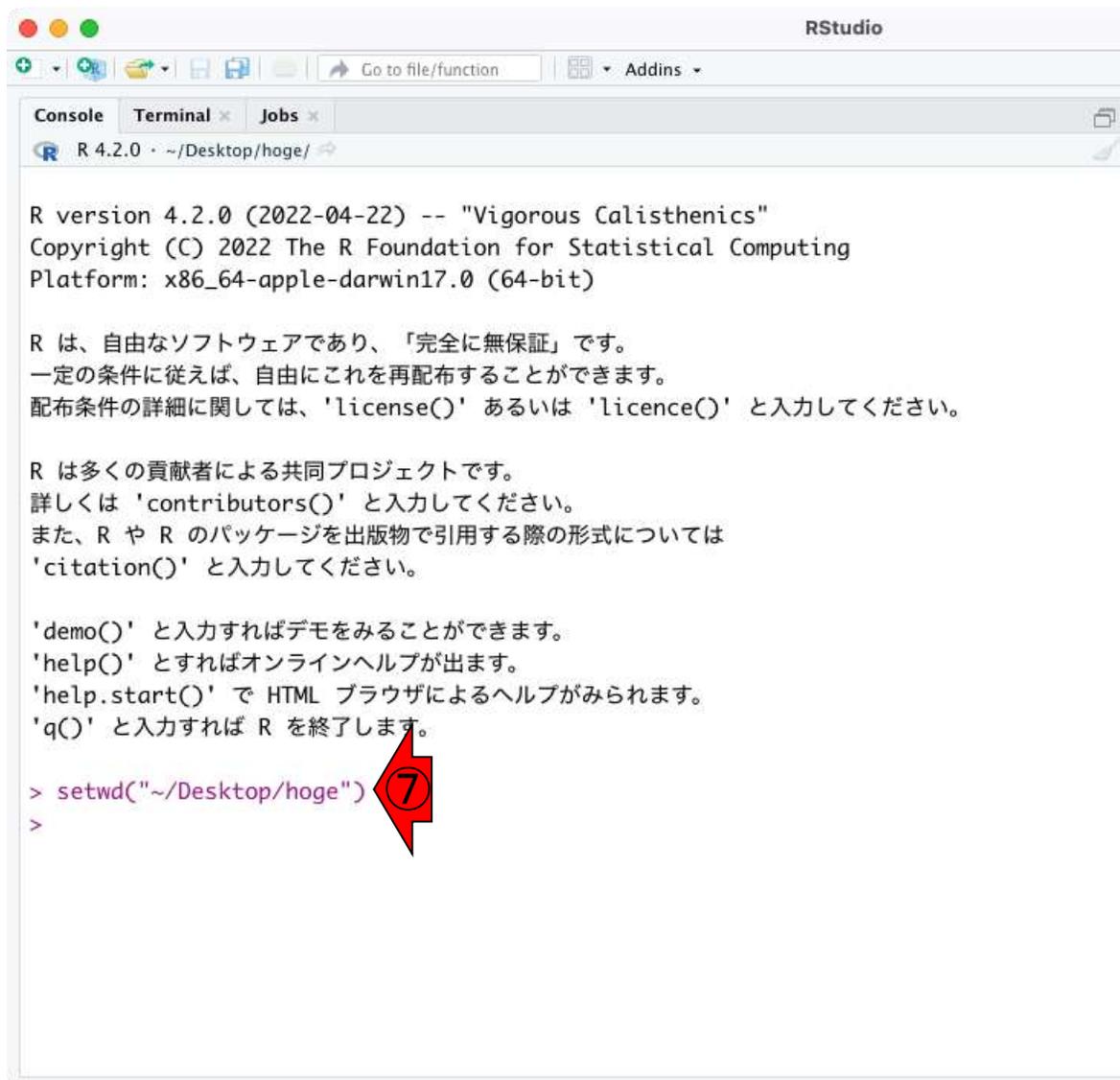
#ファイルに保存
write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F)
  
```

	A	B	C	D
1	gene name	accession	description	subcellular_location
2	gene1	hoge01	plasma_mem	nuclear
3	gene2	hoge02	ho hinu	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	gene name	accession	description	subcellular_location
2	gene1	hoge01	plasma_mem	nuclear
3	gene2	hoge02	ho hinu	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

# 応用3

①例題1は、②の中身をもつ、③リストファイル(hoge1.txt)を読み込んで、④1列目で一致する行を抽出するスクリプトでした。応用として、⑤新たに作成したりストファイル(list.txt)を読み込んで、⑥4列目で一致する行の抽出を行います。これはRStudioを再起動して、⑦作業ディレクトリの変更まで終了した状態。



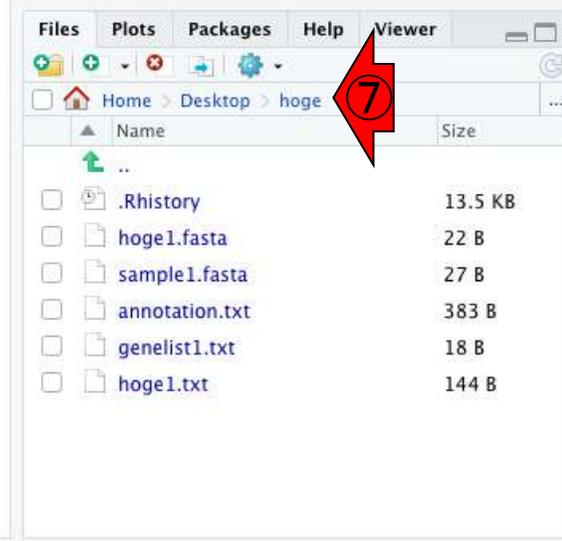
```
R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

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

R は多くの貢献者による共同プロジェクトです。
詳しくは '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ついては
'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す。
'q()' と入力すれば R を終了します。

> setwd("~/Desktop/hoge")
>
```



Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	144 B

# 応用4

まずは①リストファイル(list.txt)を作成すべく、RStudio Fileメニューの②New File、③Text File。

The screenshot shows the RStudio interface. The 'New File' menu is open, with 'Text File' selected. A red arrow labeled '2' points to the 'New File' menu, and another red arrow labeled '3' points to the 'Text File' option. In the background, a file explorer window shows a directory named 'hoge' containing several files, including 'hoge1.txt'. A red arrow labeled '1' points to a text box containing the text 'nuclear' and 'membrane' with blue arrows pointing to the right, indicating the content of the 'hoge1.txt' file.

```
setwd("~/Desktop/hoge")
```

Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	144 B

nuclear  
membrane

# 応用5

まずは①リストファイル(list.txt)を作成すべく、RStudio Fileメニューの②New File、③Text File。こんな感じになります。



The screenshot shows the RStudio interface. The main editor window displays a text file with the following content:

```
1  
nuclear  
membrane
```

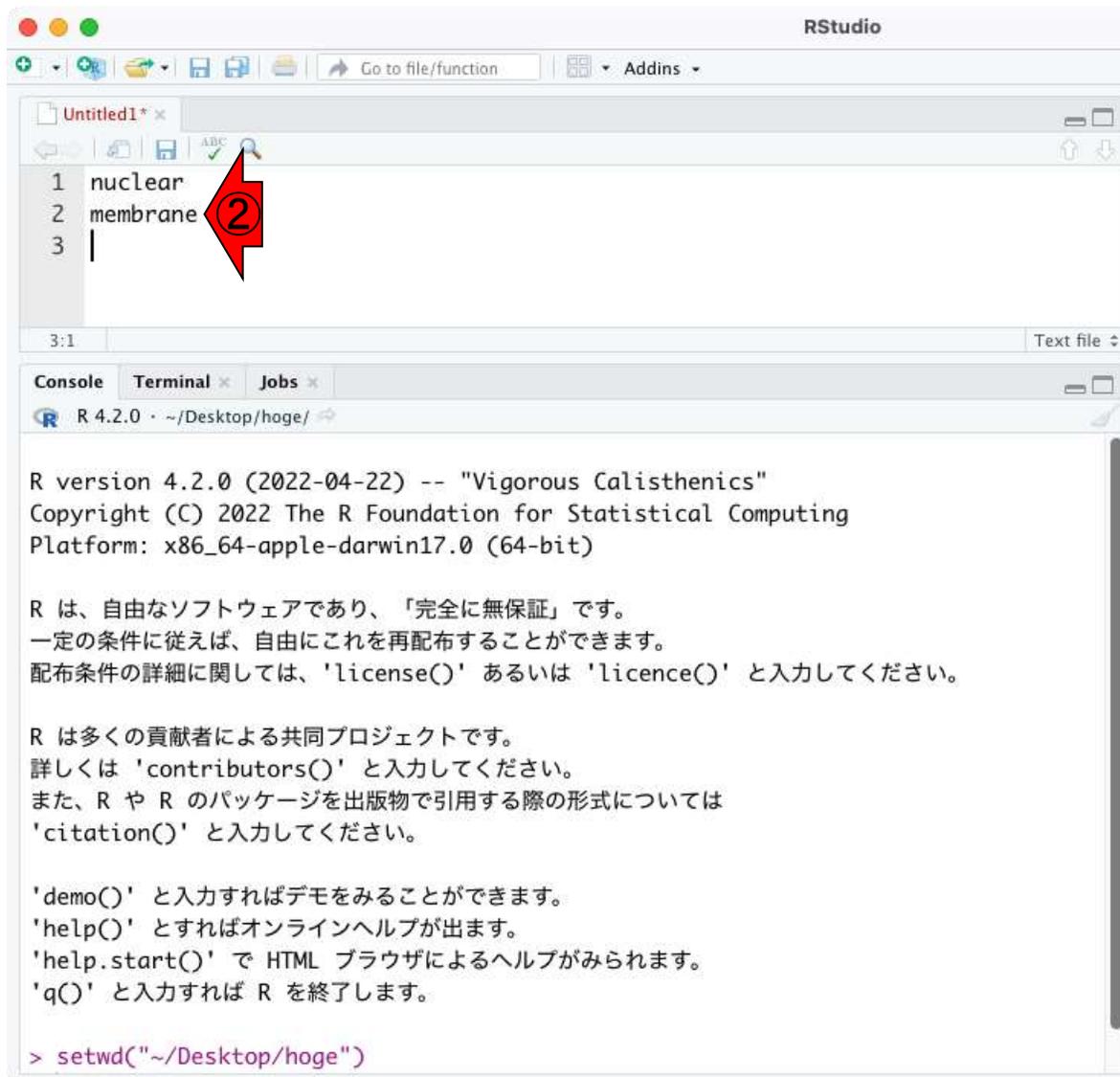
The console window shows the R version and the command `setwd("~/Desktop/hoge")` being executed. The file explorer shows the directory structure:

Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	144 B

A red arrow labeled '1' points to the text in the editor. A red box highlights the text 'nuclear' and 'membrane' with blue arrows pointing to the right. A blue arrow points to the left margin of the text.

# 応用6

まずは①リストファイル(list.txt)を作成すべく、RStudio Fileメニューの②New File、③Text File。こんな感じになります。①と同じように抽出したいキーワードを入力。この際、②の部分で最後にリターンキーを「押すか押さないか」でコピー実行時に「警告が出ないか出るか」の違いがでますがそれも含めて楽しんでください。



RStudio interface showing a text editor with the following content:

```
1 nuclear
2 membrane
3 |
```

The console shows the R version and environment information:

```
R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)
```

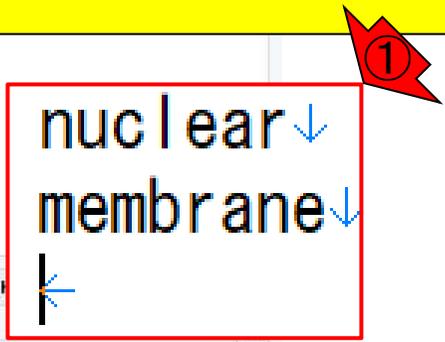
Below the console, there is Japanese text providing information about R:

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

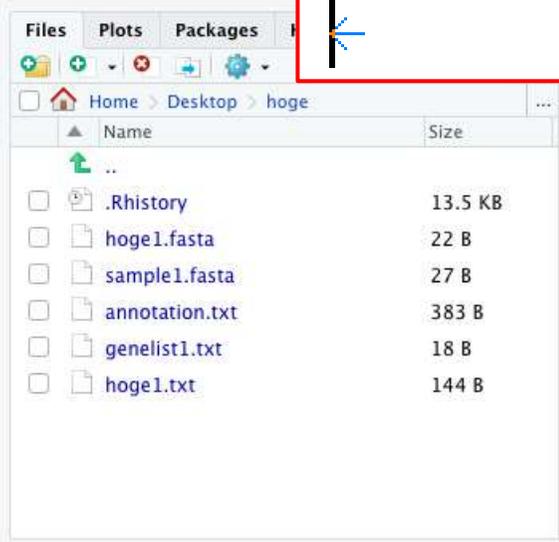
R は多くの貢献者による共同プロジェクトです。詳しくは 'contributors()' と入力してください。また、R や R のパッケージを出版物で引用する際の形式については 'citation()' と入力してください。

'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す。'q()' と入力すれば R を終了します。

> setwd("~/Desktop/hoge")



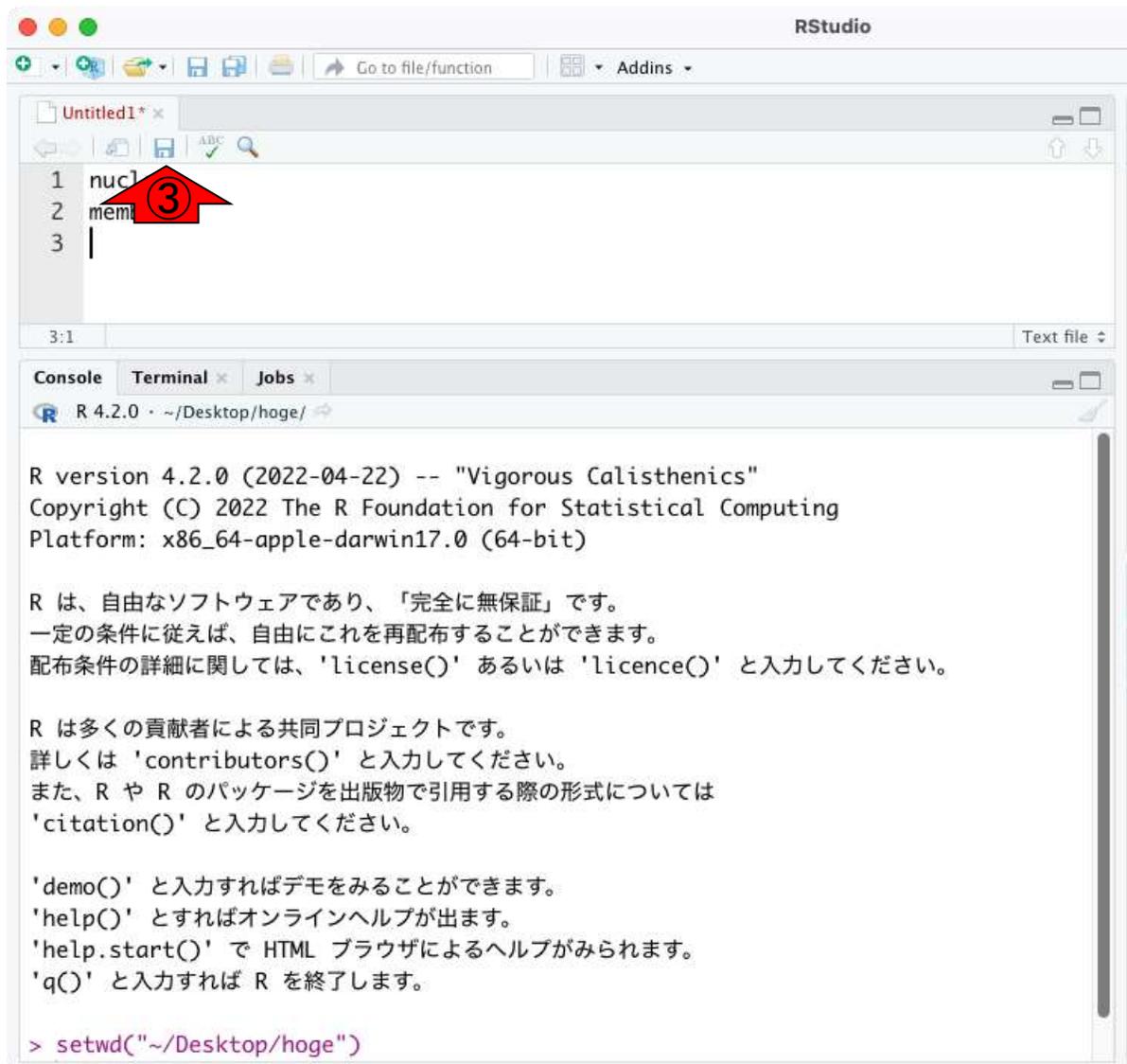
nuclear ↓  
membrane ↓



Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	144 B

# 応用7

まずは①リストファイル(list.txt)を作成すべく、②File、③New File、④Test File。こんな感じになります。①と同じように抽出したいキーワードを入力。この際、②の部分で最後にリターンキーを「押すか押さないか」でコピー実行時に「警告が出ないか出るか」の違いがでますがそれも含めて楽しんでください。③を押して保存。



```
1 nucl
2 mem
3 |
```

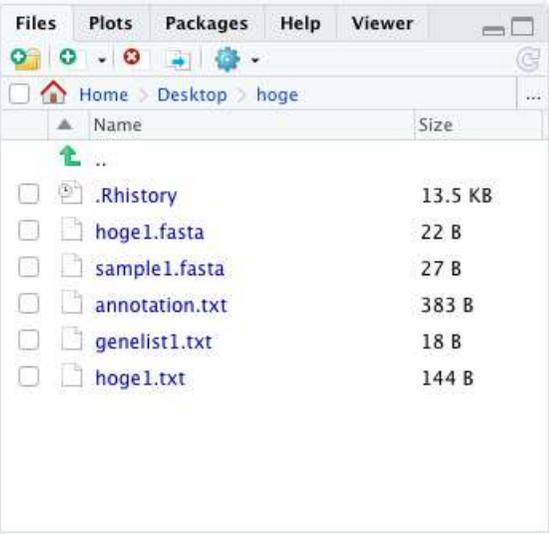
R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"  
Copyright (C) 2022 The R Foundation for Statistical Computing  
Platform: x86\_64-apple-darwin17.0 (64-bit)

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

R は多くの貢献者による共同プロジェクトです。  
詳しくは '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ついては  
'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す。  
'q()' と入力すれば R を終了します。

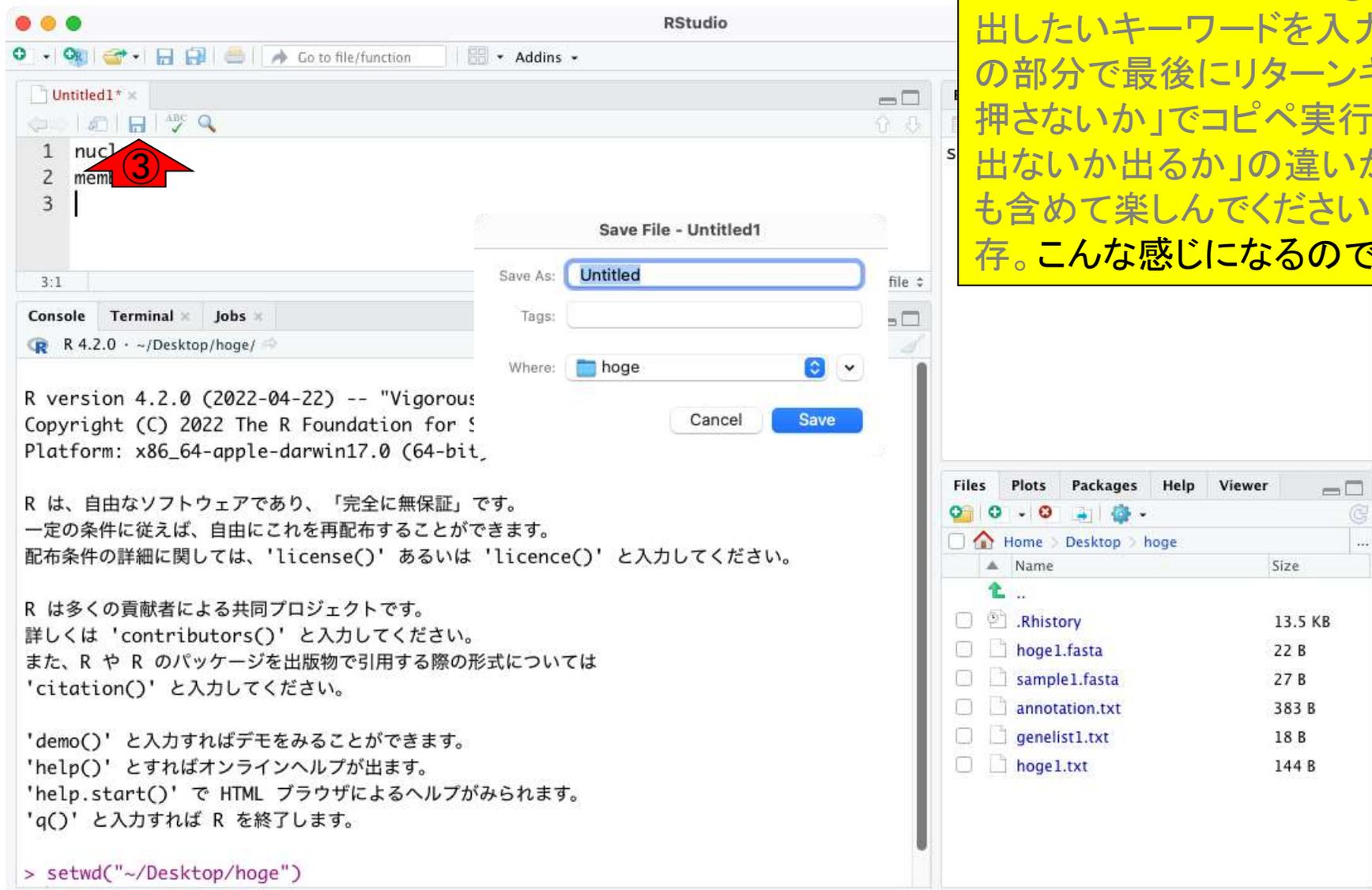
```
> setwd("~/Desktop/hoge")
```



Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	144 B

# 応用8

まずは①リストファイル(list.txt)を作成すべく、②File、③New File、④Test File。こんな感じになります。①と同じように抽出したいキーワードを入力。この際、②の部分で最後にリターンキーを「押すか押さないか」でコピー実行時に「警告が出ないか出るか」の違いがでますがそれも含めて楽しんでください。③を押して保存。こんな感じになるので、



R version 4.2.0 (2022-04-22) -- "Vigorous"  
Copyright (C) 2022 The R Foundation for Statistical Computing  
Platform: x86\_64-apple-darwin17.0 (64-bit)

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

R は多くの貢献者による共同プロジェクトです。  
詳しくは '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ついては  
'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す。  
'q()' と入力すれば R を終了します。

```
> setwd("~/Desktop/hoge")
```

Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	144 B

# 応用9

まずは①リストファイル(list.txt)を作成すべく、②File、③New File、④Test File。こんな感じになります。①と同じように抽出したいキーワードを入力。この際、②の部分で最後にリターンキーを「押すか押さないか」でコピー実行時に「警告が出ないか出るか」の違いがでますがそれも含めて楽しんでください。③を押して保存。こんな感じになるので、④list.txtと打ち込んでから、⑤Save。

Save File - Untitled1

Save As: list.txt

Tags:

Where: hoge

Cancel Save

R version 4.2.0 (2022-04-22) -- "Vigorous"  
Copyright (C) 2022 The R Foundation for Statistical Computing  
Platform: x86\_64-apple-darwin17.0 (64-bit)

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

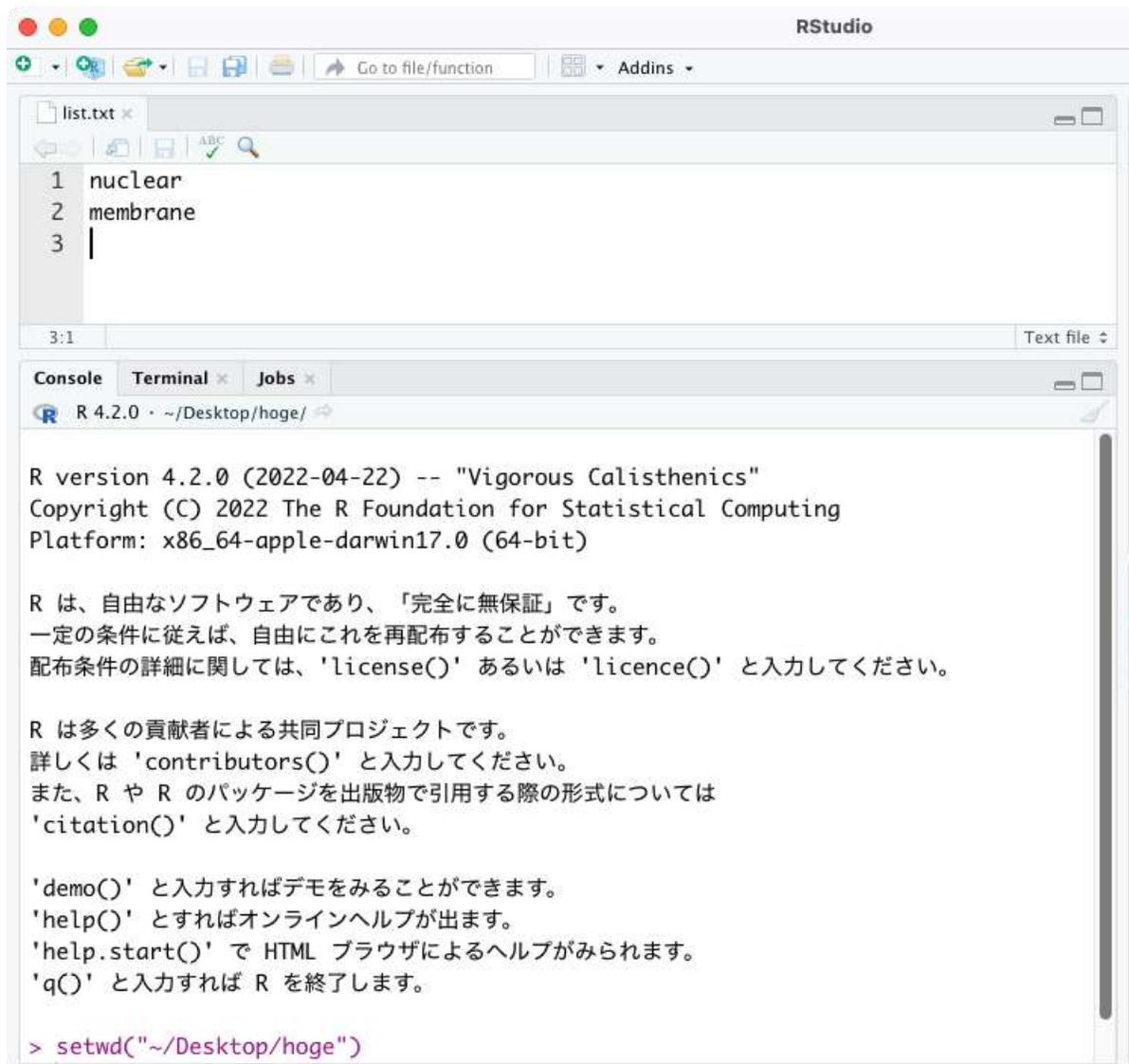
R は多くの貢献者による共同プロジェクトです。  
詳しくは 'contributors()' と入力してください。  
また、R や R のパッケージを出版物で引用する際の形式については  
'citation()' と入力してください。

'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す。  
'q()' と入力すれば R を終了します。

```
> setwd("~/Desktop/hoge")
```

Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	144 B

# 応用10



```
list.txt
1 nuclear
2 membrane
3 |

R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

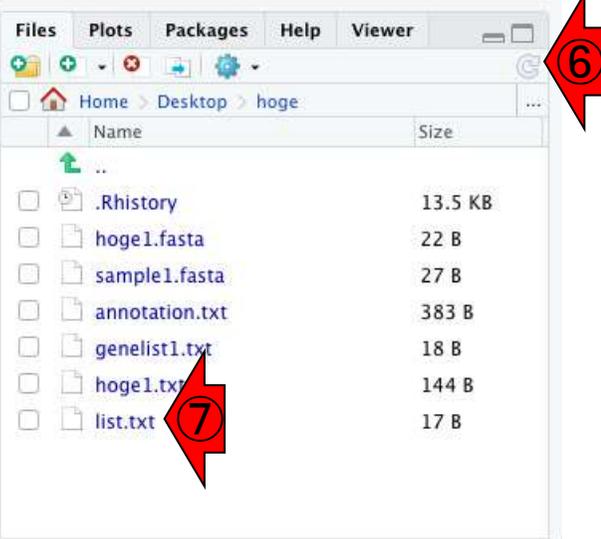
R は、自由なソフトウェアであり、「完全に無保証」です。
一定の条件に従えば、自由にこれを再配布することができます。
配布条件の詳細に関しては、'license()' あるいは 'licence()' と入力してください。

R は多くの貢献者による共同プロジェクトです。
詳しくは '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ついては
'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す。
'q()' と入力すれば R を終了します。

> setwd("~/Desktop/hoge")
```

まずは①リストファイル(list.txt)を作成すべく、②File、③New File、④Test File。こんな感じになります。①と同じように抽出したいキーワードを入力。この際、②の部分で最後にリターンキーを「押すか押さないか」でコピー実行時に「警告が出ないか出るか」の違いがでますがそれも含めて楽しんでください。③を押して保存。こんな感じになるので、④list.txtと打ち込んでから、⑤Save。表示が更新されない場合は⑥リロードすると、⑦list.txtが見えました。



# 応用11

次は、①例題1のスクリプト(or コード)を  
テンプレートとして利用したいので…



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

```
in_f1 <- "annotation.txt" #入力ファイル名を指定してin_f1に格納(アノテーションファイル)
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", quote="") #in_f1で指定したファイルの読み込み
keywords <- readLines(in_f2) #in_f2で指定したファイルの読み込み
dim(data) #オブジェクトdataの行数と列数を表示

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

2. 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([genelist2.txt](#))中のものが含まれる行全体を出力したい場合：

```
in_f1 <- "annotation.txt" #入力ファイル名を指定してin_f1に格納(アノテーションファイル)
in_f2 <- "genelist2.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 <- "hoge2.txt" #出力ファイル名を指定してout_fに格納
param <- 1 #アノテーションファイル中の検索したい列番号を指定

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

# 応用12

次は、①例題1のスクリプト(or コード)を  
テンプレートとして利用したいので、②赤  
枠内のコード全体をコピーしておく。



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

```
in_f1 <- "annotation.txt" #入力ファイル名を指定してin_f1に格納(アノテーションファイル)
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", quote="")
keywords <- readLines(in_f2)
dim(data)

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

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

「in\_f1 <- "annotation.txt" #入力ファイル名を...」を検索

コピー

選択箇所をリンクをコピー

Googleで "in\_f1 <- "annotation.txt" #入力ファイル名を..."を検索

印刷...

格納

検証

スピーチ >

サービス > 名で保

2. 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([genelist2.txt](#))中のものが含まれる行全体を出力したい場合：

```
in_f1 <- "annotation.txt" #入力ファイル名を指定してin_f1に格納(アノテーションファイル)
in_f2 <- "genelist2.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 <- "hoge2.txt" #出力ファイル名を指定してout_fに格納
param <- 1 #アノテーションファイル中の検索したい列番号を指定

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

# 応用13

次は、①例題1のスクリプト(or コード)をテンプレートとして利用したいので、②赤枠内のコード全体をコピーしておく。  
RStudio Fileメニューの③New File、④R Scriptとして、「新規のRスクリプト」を開く。

The screenshot shows the RStudio interface. The 'File' menu is open, with 'New File' (marked with a red circle 3) and 'R Script' (marked with a red circle 4) highlighted. The console shows the following R code:

```
> setwd("~/Desktop/hoge")
```

The file explorer at the bottom shows the directory structure:

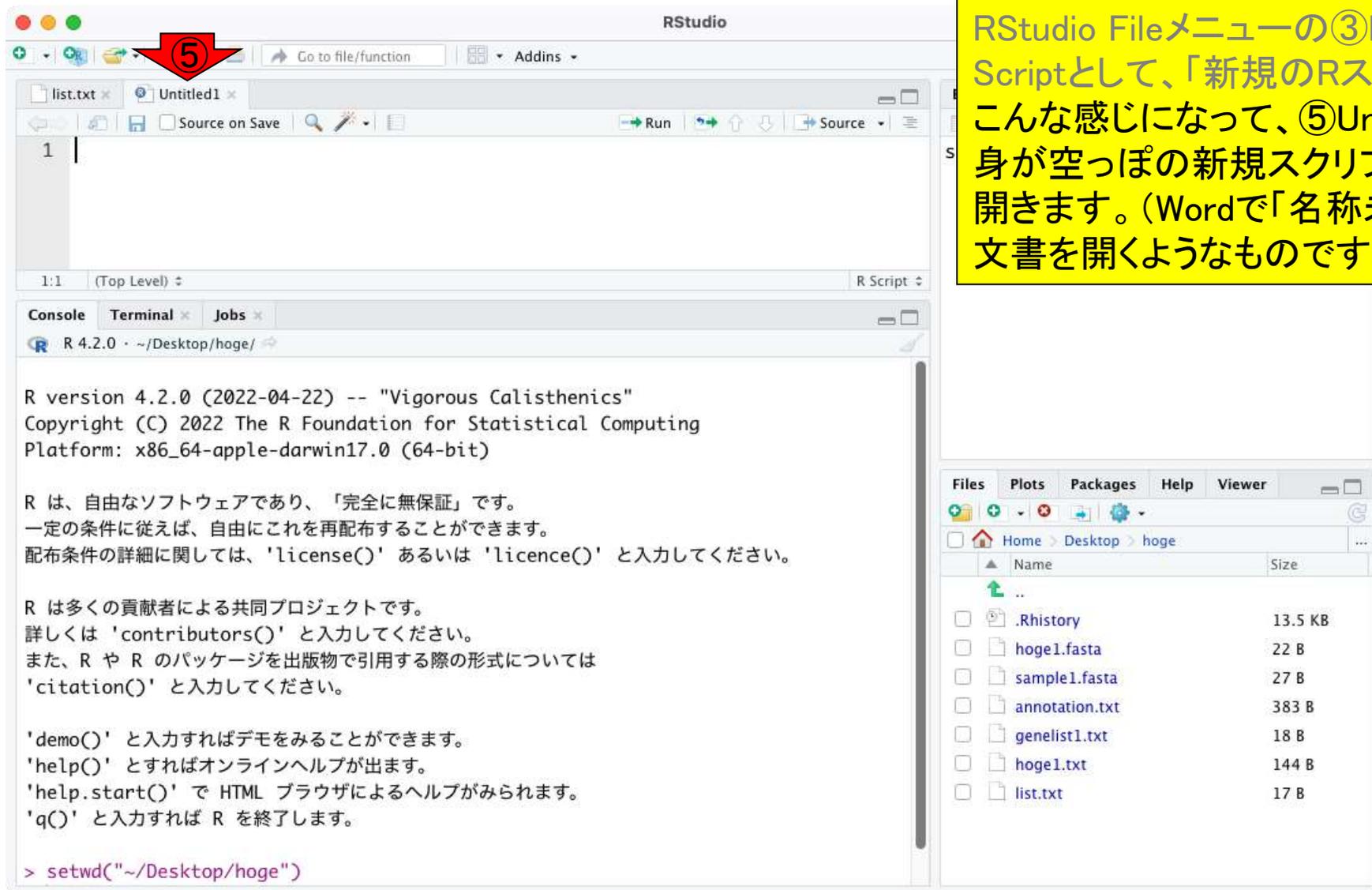
Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	144 B
list.txt	17 B

R は、自由な  
一定の条件に  
配布条件の詳  
R は多くの貢  
詳しくは 'co  
また、R や R  
'citation()'  
> setwd("~/Desktop/hoge")

'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す。  
'q()' と入力すれば R を終了します。

# 応用14

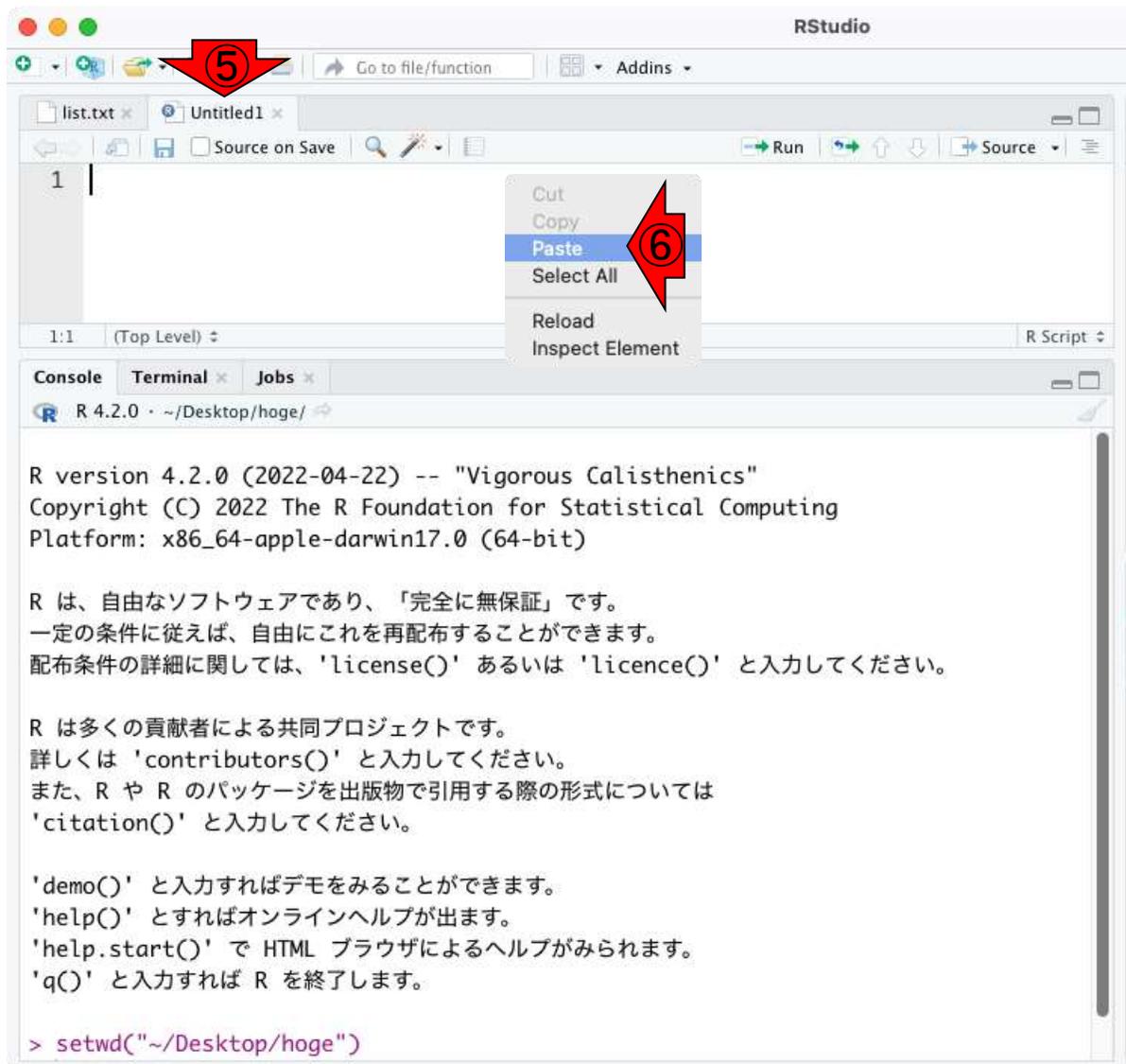
次は、①例題1のスクリプト(or コード)をテンプレートとして利用したいので、②赤枠内のコード全体をコピーしておく。  
RStudio Fileメニューの③New File、④R Scriptとして、「新規のRスクリプト」を開く。こんな感じになって、⑤Untitled1という中身が空っぽの新規スクリプトファイルが開きます。(Wordで「名称未設定」の新規文書を開くようなものですね。)



The screenshot shows the RStudio interface. The main editor window displays a new R script file named 'Untitled1' with a single line of code: `> setwd("~/Desktop/hoge")`. A red arrow with the number '5' points to the 'Untitled1' tab. The console window shows the R version 4.2.0 (2022-04-22) and the current directory is `~/Desktop/hoge/`. The file explorer window shows the directory structure:

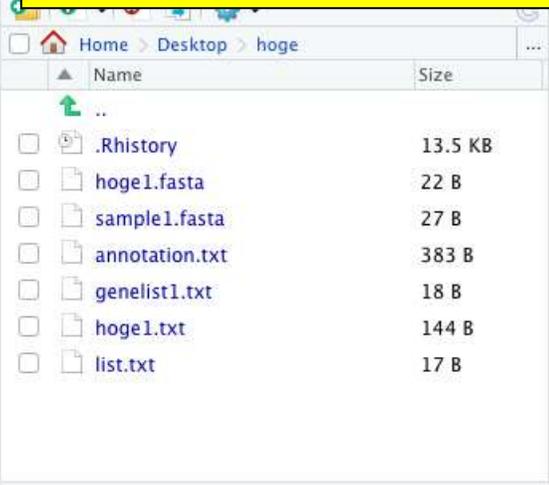
Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	144 B
list.txt	17 B

# 応用15



次は、①例題1のスクリプト(or コード)をテンプレートとして利用したいので、②赤枠内のコード全体をコピーしておく。

RStudio Fileメニューの③New File、④R Scriptとして、「新規のRスクリプト」を開く。こんな感じになって、⑤Untitled1という中身が空っぽの新規スクリプトファイルが開きます。(Wordで「名称未設定」の新規文書を開くようなものですね。)さきほどクリップボードにコピーしておいたテンプレートスクリプトを、⑥右クリックでPaste。(うまくいかない場合は「⌘ + V」やEditメニューからPasteの選択なども試してください。)



# 応用16

貼り付けた全体が見えるようにした状態。こんな感じになればOK。①エディタ画面と②Console画面の縦幅を調節する場合、その境界にカーソルを合わせると③矢印が表示されるので、調節してください。①エディタ画面を縦に最大限広く使いたい場合は、④を押します。

The screenshot shows the RStudio interface with the following components and callouts:

- Code Editor:** Contains R code for reading files and processing data. A red box highlights the code area, and a red arrow labeled ① points to the right edge of the editor.
- Console:** Shows the R version and platform information. A red box highlights the console area, and a red arrow labeled ② points to the left edge of the console.
- File Explorer:** Shows a directory listing for 'hoge' on the desktop. A red arrow labeled ④ points to the top edge of the file explorer.
- Vertical Resizer:** A red arrow labeled ③ points to the vertical double-headed arrow on the border between the code editor and the console.

```
1 in_f1 <- "annotation.txt"      #入力ファイル名を指定してin_f1に格納(
2 in_f2 <- "genelist1.txt"      #入力ファイル名を指定してin_f2に格納(
3 out_f <- "hoge1.txt"         #出力ファイル名を指定してout_fに格納
4 param <- 1                   #アノテーションファイル中の検索したい
5
6 #入力ファイルの読み込み
7 data <- read.table(in_f1, header=TRUE, sep="\t", quote="")#in_f1で指定したフ
8 keywords <- readLines(in_f2) #in_f2で指定したファイルの読み込み
9 dim(data)                    #オブジェクトdataの行数と列数を表示
10
11 #本番
12 obj <- is.element(as.character(data[,param]), keywords)#条件を満たすかどうか
13 out <- data[obj,]           #objがTRUEとなる行のみ抽出した結果をo
14 dim(out)                   #オブジェクトoutの行数と列数を表示
15
16 #ファイルに保存
17 write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F)#outの中身を
18 |
```

R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"  
Copyright (C) 2022 The R Foundation for Statistical Computing  
Platform: x86\_64-apple-darwin17.0 (64-bit)

R は、自由なソフトウェアであり、「完全に無保証」です。

Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	144 B
list.txt	17 B

# 応用17

貼り付けた全体が見えるようにした状態。こんな感じになればOK。①エディタ画面と②Console画面の縦幅を調節する場合、その境界にカーソルを合わせると③矢印が表示されるので、調節してください。①エディタ画面を縦に最大限広く使いたい場合は、④を押します。こんな感じになります。もう一度④を押すと元に戻せる。

The screenshot shows the RStudio interface. The editor window contains R code for reading and processing text files. The console window is at the bottom. A file explorer on the right shows the contents of a directory named 'hoge'.

```
1 in_f1 <- "annotation.txt"      #入力ファイル名を指定してin_f1に格納(
2 in_f2 <- "genelist1.txt"      #入力ファイル名を指定してin_f2に格納(
3 out_f <- "hoge1.txt"         #出力ファイル名を指定してout_fに格納
4 param <- 1                   #アノテーションファイル中の検索したい
5
6 #入力ファイルの読み込み
7 data <- read.table(in_f1, header=TRUE, sep="\t", quote="")#in_f1で指定したフ
8 keywords <- readLines(in_f2)  #in_f2で指定したファイルの読み込み
9 dim(data)                    #オブジェクトdataの行数と列数を表示
10
11 #本番
12 obj <- is.element(as.character(data[,param]), keywords)#条件を満たすかどうか
13 out <- data[obj,]           #objがTRUEとなる行のみ抽出した結果をo
14 dim(out)                    #オブジェクトoutの行数と列数を表示
15
16 #ファイルに保存
17 write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F)#outの中身を
18
```

Files | Plots | Packages | Help | Viewer

Home > Desktop > hoge

Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	144 B
list.txt	17 B

18:1 (Top Level) | R Script

Console

# 応用18

元に戻りました。⑤はエディタ画面の縦幅を最小にする操作であり、⑥Console画面をを最大にする操作と同じです。以降は特に明記しませんが、見やすいように適宜いじってください。

The screenshot shows the RStudio interface with three main panels:

- Source Editor:** Contains R code for reading files and filtering data. A red arrow labeled ⑤ points to the window control buttons (minimize, maximize, close) in the top right corner of this panel.
- Console:** Shows the R version (4.2.0) and system information. A red arrow labeled ⑥ points to the maximize button in the top right corner of this panel.
- Files Panel:** Shows a file explorer view of the Desktop/hoge directory, listing files like .Rhistory, hoge1.fasta, sample1.fasta, annotation.txt, genelist1.txt, hoge1.txt, and list.txt.

```
1 in_f1 <- "annotation.txt"      #入力ファイル名を指定してin_f1に格納(
2 in_f2 <- "genelist1.txt"      #入力ファイル名を指定してin_f2に格納(
3 out_f <- "hoge1.txt"         #出力ファイル名を指定してout_fに格納
4 param <- 1                   #アノテーションファイル中の検索したい
5
6 #入力ファイルの読み込み
7 data <- read.table(in_f1, header=TRUE, sep="\t", quote="")#in_f1で指定したフ
8 keywords <- readLines(in_f2)  #in_f2で指定したファイルの読み込み
9 dim(data)                    #オブジェクトdataの行数と列数を表示
10
11 #本番
12 obj <- is.element(as.character(data[,param]), keywords)#条件を満たすかどうか
13 out <- data[obj,]           #objがTRUEとなる行のみ抽出した結果をo
14 dim(out)                    #オブジェクトoutの行数と列数を表示
15
16 #ファイルに保存
17 write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F)#outの中身を
18 |
```

R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"  
Copyright (C) 2022 The R Foundation for Statistical Computing  
Platform: x86\_64-apple-darwin17.0 (64-bit)

R は、自由なソフトウェアであり、「完全に無保証」です。

# 応用19

①エディタ画面の最上部を表示している状態。

The screenshot shows the RStudio interface with the following components:

- Script Editor:** Contains R code for reading files and filtering data. A red arrow labeled '1' points to the top of the script.
- Environment Pane:** Shows the current working directory set to `~/Desktop/hoge`.
- Files Pane:** Shows the directory contents, including files like `.Rhistory`, `hoge1.fasta`, `sample1.fasta`, `annotation.txt`, `genelist1.txt`, `hoge1.txt`, and `list.txt`.
- Console:** Shows the R version and platform information.

```
1 in_f1 <- "annotation.txt"      #入力ファイル名を指定してin_f1に格納(
2 in_f2 <- "genelist1.txt"      #入力ファイル名を指定してin_f2に格納
3 out_f <- "hoge1.txt"         #出力ファイル名を指定してout_fに格納
4 param <- 1                   #アノテーションファイル中の検索したい
5
6 #入力ファイルの読み込み
7 data <- read.table(in_f1, header=TRUE, sep="\t", quote="")#in_f1で指定したフ
8 keywords <- readLines(in_f2)  #in_f2で指定したファイルの読み込み
9 dim(data)                    #オブジェクトdataの行数と列数を表示
10
11 #本番
12 obj <- is.element(as.character(data[,param]), keywords)#条件を満たすかどうか
13 out <- data[obj,]           #objがTRUEとなる行のみ抽出した結果をo
14 dim(out)                   #オブジェクトoutの行数と列数を表示
15
16 #ファイルに保存
17 write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F)#outの中身を
18
```

R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"  
Copyright (C) 2022 The R Foundation for Statistical Computing  
Platform: x86\_64-apple-darwin17.0 (64-bit)

R は、自由なソフトウェアであり、「完全に無保証」です。

# 応用20

①エディタ画面の最上部を表示している状態。今は、②の部分のファイル名を、③list.txtに置き換えたい。

The screenshot shows the RStudio interface. The main editor displays an R script with the following code:

```
1 in_f1 <- "annotation.txt"
2 in_f2 <- "genelist1.txt"
3 out_f <- "hoge1.txt"
4 param <- 1
5
6 #入力ファイルの読み込み
7 data <- read.table(in_f1, header=TRUE, sep="\t", quote="")#in_f1で指定したフ
8 keywords <- readLines(in_f2)
9 dim(data)
10
11 #本番
12 obj <- is.element(as.character(data[,param]), keywords)#条件を満たすかどうか
13 out <- data[obj,]
14 dim(out)
15
16 #ファイルに保存
17 write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F)#outの中身を
18
```

The console shows the R version and platform information:

```
R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R は、自由なソフトウェアであり、「完全に無保証」です。
```

The file explorer shows the following files:

Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	144 B
list.txt	17 B

# 応用21

①エディタ画面の上部に移動したところ。今は、②の部分のファイル名を、③ list.txtに置き換えたい。また、④1列目ではなく、⑤4列目で一致する行を抽出したいので、

```
1 in_f1 <- "annotation.txt" #入力ファイル名を指定してin_f1に格納(
2 in_f2 <- "genelist1.txt" #入力ファイル名を指定してin_f2に格納
3 out_f <- "hoge1.txt" #出力ファイル名を指定してout_fに格納
4 param <- 1 #アノテーションファイル中の検索したい
5
6 #入力ファイルの読み込み
7 data <- read.table(in_f1, header=TRUE, sep="\t", quote="")#in_f1で指定したフ
8 keywords <- readLines(in_f2) #in_f2で指定したファイルの読み込み
9 dim(data) #オブジェクトdataの行数と列数を表示
10
11 #本番
12 obj <- is.element(as.character(data[,param]), keywords)#条件を満たすかどうか
13 out <- data[obj,] #objがTRUEとなる行のみ抽出した結果をo
14 dim(out) #オブジェクトoutの行数と列数を表示
15
16 #ファイルに保存
17 write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F)#outの中
18 |
```

setwd("~/Desktop/hoge")

	A	B	C	D
1	gene name	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

R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"  
Copyright (C) 2022 The R Foundation for Statistical Computing  
Platform: x86\_64-apple-darwin17.0 (64-bit)

R は、自由なソフトウェアであり、「完全に無保証」です。

# 応用22

①エディタ画面の上部に移動したところ。今は、②の部分のファイル名を、③ list.txtに置き換えたい。また、④1列目ではなく、⑤4列目で一致する行を抽出したいので、⑥こんな感じで書き替えます。

The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains an R script with Japanese comments. Red arrows labeled '⑥' point to the file names 'list.txt' in lines 2 and 3, and the column index '4' in line 4.
- Console:** Shows the execution of `setwd("~/Desktop/hoge")`.
- Files Panel:** Displays a directory listing for the 'hoge' folder on the Desktop.

```
1 in_f1 <- "annotation.txt" #入力ファイル名を指定してin_f1に格納(
2 in_f2 <- "list.txt" #入力ファイル名を指定してin_f2に格納(リスト
3 out_f <- "hoge.txt" #出力ファイル名を指定してout_fに格納
4 param <- 4 #アノテーションファイル中の検索したい
5
6 #入力ファイルの読み込み
7 data <- read.table(in_f1, header=TRUE, sep="\t", quote="")#in_f1で指定したフ
8 keywords <- readLines(in_f2) #in_f2で指定したファイルの読み込み
9 dim(data) #オブジェクトdataの行数と列数を表示
10
11 #本番
12 obj <- is.element(as.character(data[,param]), keywords)#条件を満たすかどうか
13 out <- data[obj,] #objがTRUEとなる行のみ抽出した結果をo
14 dim(out) #オブジェクトoutの行数と列数を表示
15
16 #ファイルに保存
17 write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F)#outの中身を
18
```

Console output:

```
> setwd("~/Desktop/hoge")
>
```

Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	144 B
list.txt	17 B

# 応用23

①エディタ画面の上部に移動したところ。今は、②の部分のファイル名を、③ list.txtに置き換えたい。また、④1列目ではなく、⑤4列目で一致する行を抽出したいので、⑥こんな感じで書き替えます。⑦を押して、(.txtにしても特に問題はありませんが)Rスクリプトであることを明示した拡張子.Rをつけたファイル名⑧ hoge.Rにして、⑨Save。

The screenshot shows the RStudio interface. The script editor contains R code for reading a file, filtering rows based on a parameter, and saving the result. A 'Save File - Untitled1' dialog box is open, with the filename 'hoge.R' and the location 'hoge'. A file explorer window shows the contents of the 'hoge' directory, including files like 'hoge1.fasta', 'sample1.fasta', 'annotation.txt', 'genelist1.txt', 'hoge1.txt', and 'list.txt'. Red arrows with circled numbers 1 through 9 point to specific elements in the image corresponding to the instructions in the yellow box.

```
1 in_f1 <- "annotation.txt" #入力ファイル名を指定してin_f1に格納(リスト)
2 in_f2 <- "list.txt" #入力ファイル名を指定してin_f2に格納(リスト)
3 out_f <- "hoge1.txt" #出力ファイル名を指定してout_fに格納
4 param <- 4 #出力ファイル名を指定してout_fに格納
5
6 #入力ファイルの読み込み
7 data <- read.table(in_f1, header=T, as.is=T) #in_f1で指定したファイルの読み込み
8 keywords <- readLines(in_f2) #in_f2で指定したファイルの読み込み
9 dim(data) #行数と列数を表示
10
11 #本番
12 obj <- is.element(as.character(data[,param]), keywords) #objがTRUEとなる行のみ抽出した結果をオブジェクトoutに格納
13 out <- data[obj,] #オブジェクトoutの行数と列数を表示
14 dim(out)
15
16 #ファイルに保存
17 write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F)#outの中身をout_fに保存
18
```

Save File - Untitled1

Save As: hoge.R

Tags:

Where: hoge

Cancel Save

Files Plots Packages Help Viewer

Home > Desktop > hoge

Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	144 B
list.txt	17 B

Console Terminal Jobs

R 4.2.0 ~~/Desktop/hoge/

'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す。  
'q()' と入力すれば R を終了します。

```
> setwd("~/Desktop/hoge")
>
```

# 応用24

①エディタ画面の上部に移動したところ。今は、②の部分のファイル名を、③ list.txtに置き換えたい。また、④1列目ではなく、⑤4列目で一致する行を抽出したいので、⑥こんな感じで書き替えます。⑦を押して、(.txtにしても特に問題はありませんが)Rスクリプトであることを明示した拡張子.Rをつけたファイル名⑧ hoge.Rにして、⑨Save。表示が更新されない場合は、⑩リロードすると、⑪hoge.Rが見えました。

```
RStudio

list.txt x  hoge.R x

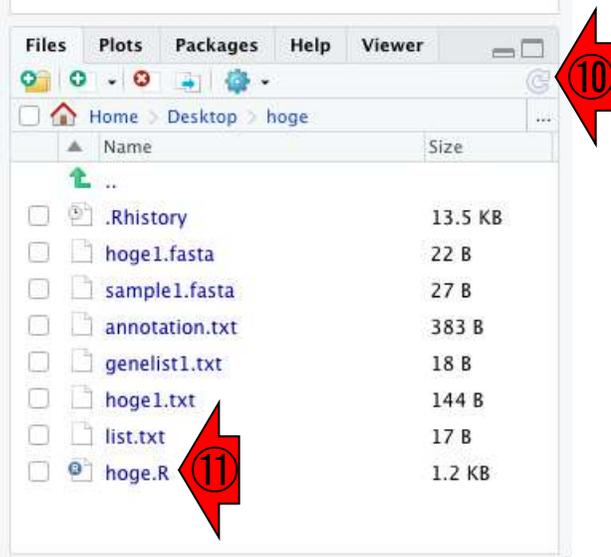
Source on Save  Run  Source

1 in_f1 <- "annotation.txt"      #入力ファイル名を指定してin_f1に格納(
2 in_f2 <- "list.txt"           #入力ファイル名を指定してin_f2に格納(リスト
3 out_f <- "hoge1.txt"         #出力ファイル名を指定してout_fに格納
4 param <- 4                    #アノテーションファイル中の検索したい
5
6 #入力ファイルの読み込み
7 data <- read.table(in_f1, header=TRUE, sep="\t", quote="")#in_f1で指定したフ
8 keywords <- readLines(in_f2)  #in_f2で指定したファイルの読み込み
9 dim(data)                     #オブジェクトdataの行数と列数を表示
10
11 #本番
12 obj <- is.element(as.character(data[,param]), keywords)#条件を満たすかどうか
13 out <- data[obj,]            #objがTRUEとなる行のみ抽出した結果をo
14 dim(out)                     #オブジェクトoutの行数と列数を表示
15
16 #ファイルに保存
17 write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F)#outの中身を
18

4:11 (Top Level) R Script
```

```
Console Terminal Jobs
R 4.2.0 ~/Desktop/hoge/
'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す。
'q()' と入力すれば R を終了します。

> setwd("~/Desktop/hoge/")
>
```



# 応用25

いよいよ変更後のスクリプトの実行。①  
赤枠内のコード全体を選択。

The screenshot shows the RStudio interface with a script editor containing R code. A red rectangular box highlights the code from line 6 to line 18. A red arrow with the number '1' points to the first line of the highlighted code. The console at the bottom shows the execution of the script, including the command `setwd("~/Desktop/hoge")`.

```
1 in_f1 <- "annotation.txt"           #入力ファイル名を指定してin_f1に格納(
2 in_f2 <- "list.txt"                 #入力ファイル名を指定してin_f2に格納(リスト
3 out_f <- "hoge1.txt"                #出力ファイル名を指定してout_fに格納
4 param <- 4                          #アノテーションファイル中の検索したい
5
6 #入力ファイルの読み込み
7 data <- read.table(in_f1, header=TRUE, sep="\t", quote="")#in_f1で指定したフ
8 keywords <- readLines(in_f2)        #in_f2で指定したファイルの読み込み
9 dim(data)                           #オブジェクトdataの行数と列数を表示
10
11 #本番
12 obj <- is.element(as.character(data[,param]), keywords)#条件を満たすかどうか
13 out <- data[obj,]                  #objがTRUEとなる行のみ抽出した結果をo
14 dim(out)                           #オブジェクトoutの行数と列数を表示
15
16 #ファイルに保存
17 write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F)#outの中身を
18
```

Console output:

```
R 4.2.0 ~./Desktop/hoge/
'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す。
'q()' と入力すれば R を終了します。

> setwd("~/Desktop/hoge")
>
```

Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	144 B
list.txt	17 B
hoge.R	1.2 KB

# 応用26

いよいよ変更後のスクリプトの実行。①  
赤枠内のコード全体を選択。こんな感じ。

The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains R code for file processing. A red box highlights lines 6 through 18. A red arrow points to the Run button (a play icon) in the toolbar above the editor.
- Environment/Console:** Shows the command `setwd("~/Desktop/hoge")` has been executed.
- Files Panel:** Displays the file structure of the `hoge` directory on the Desktop.
- Console:** Shows the R prompt and the execution of `setwd("~/Desktop/hoge")`.

```
1 in_f1 <- "annotation.txt"      #入力ファイル名を指定してin_f1に格納(リスト)
2 in_f2 <- "list.txt"           #入力ファイル名を指定してin_f2に格納(リスト)
3 out_f <- "hoge1.txt"         #出力ファイル名を指定してout_fに格納
4 param <- 4                   #アノテーションファイル中の検索したい行数
5
6 #入力ファイルの読み込み
7 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
8 keywords <- readLines(in_f2) #in_f2で指定したファイルの読み込み
9 dim(data)                    #オブジェクトdataの行数と列数を表示
10
11 #本番
12 obj <- is.element(as.character(data[,param]), keywords)#条件を満たすかどうか
13 out <- data[obj,]            #objがTRUEとなる行のみ抽出した結果をオブジェクトoutに格納
14 dim(out)                    #オブジェクトoutの行数と列数を表示
15
16 #ファイルに保存
17 write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F)#outの中身をout_fに保存
18
```

Files Panel:

Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	144 B
list.txt	17 B
hoge.R	1.2 KB

Console:

```
> setwd("~/Desktop/hoge")
>
```

# 応用27

いよいよ変更後のスクリプトの実行。①  
赤枠内のコード全体を選択。こんな感じ。  
この状態で②Runを押すと、

The screenshot shows the RStudio interface with the following components:

- Script Editor:** Contains R code for reading files, processing data, and saving it. A red box highlights the code from line 1 to 18.
- Run Button:** A red arrow labeled '2' points to the 'Run' button in the toolbar.
- Environment Panel:** A red arrow labeled '1' points to the 'Environment' panel, which shows the current working directory as `setwd("~/Desktop/hoge")`.
- Files Panel:** Shows the file structure of the 'hoge' directory on the Desktop, including files like `hoge1.fasta`, `annotation.txt`, and `hoge.R`.
- Console:** Shows the output of the R script execution, including the command `setwd("~/Desktop/hoge")`.

```
1 in_f1 <- "annotation.txt"           #入力ファイル名を指定してin_f1に格納(リスト)
2 in_f2 <- "list.txt"                 #入力ファイル名を指定してin_f2に格納(リスト)
3 out_f <- "hoge1.txt"                #出力ファイル名を指定してout_fに格納
4 param <- 4                           #アノテーションファイル中の検索したい行数
5
6 #入力ファイルの読み込み
7 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
8 keywords <- readLines(in_f2)         #in_f2で指定したファイルの読み込み
9 dim(data)                             #オブジェクトdataの行数と列数を表示
10
11 #本番
12 obj <- is.element(as.character(data[,param]), keywords)#条件を満たすかどうか
13 out <- data[obj,]                    #objがTRUEとなる行のみ抽出した結果をオブジェクトoutに格納
14 dim(out)                              #オブジェクトoutの行数と列数を表示
15
16 #ファイルに保存
17 write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F)#outの中身をout_fに保存
18
```

```
> setwd("~/Desktop/hoge")
>
```

# 応用28

いよいよ変更後のスクリプトの実行。①赤枠内のコード全体を選択。こんな感じ。この状態で②Runを押すと、最後まで実行されたようです。(③表示領域のサイズや④スクロールバーを動かして確認。)

The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains an R script with Japanese comments. A red box highlights the code from line 6 to line 12.
- Console:** Shows the output of the script execution, including the result of the `is.element` function and the dimensions of the output object.
- File Explorer:** Shows the file system structure, including the `hoge.R` file and the output files `hoge1.fasta` and `hoge1.txt`.

```
1 in_f1 <- "annotation.txt" #入力ファイル名を指定してin_f1に格納
2 in_f2 <- "list.txt" #入力ファイル名を指定してin_f2に格納(リスト)
3 out_f <- "hoge1.txt" #出力ファイル名を指定してout_fに格納
4 param <- 4 #アノテーションファイル中の検索した単語の本番
5
6 #入力ファイルの読み込み
7 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
8 keywords <- readLines(in_f2) #in_f2で指定したファイルの読み込み
9 dim(data) #オブジェクトdataの行数と列数を表示
10
11 #本番
12 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
13
14 out <- data[obj,] #objがTRUEとなる行のみ抽出した結果をoutに格納
15 dim(out) #オブジェクトoutの行数と列数を表示
16
17 #ファイルに保存
18 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
```

```
> #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] 7 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
>
```

Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	257 B
list.txt	17 B
hoge.R	1.2 KB

# 応用29

いよいよ変更後のスクリプトの実行。①赤枠内のコード全体を選択。こんな感じ。この状態で②Runを押すと、最後まで実行されたようです。(③表示領域のサイズや④スクロールバーを動かして確認。) Macintoshでも同様の現象が起きるかは不明ですが、**出力予定ファイル(hoge.1.txt)をExcelで開いている状態で実行したWinユーザは③のような感じでエラーが出ます。** Macintoshでも遭遇した場合は、Excelファイルを閉じて再実行してください。

```

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

list.txt x hoge.R x

Source on Save Run Source

12 obj <- list.element(as.character(data[,param]), ke
13 out <- data[obj,] #objがTRUE
14 dim(out) #オブジェクトの次元
15
16 #ファイルに保存
17 write.table(out, out_f, sep="\t", append=F, quot
18 |
  
```

Console Terminal x Jobs x

C:/Users/kadota/Desktop/hoge/ ↗

row.names=F)#outの中身を指定したファイル名で保存  
file(file, ifelse(append, "a", "w")) でエラー:  
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:  
file(file, ifelse(append, "a", "w")) で:  
ファイル 'hoge1.txt' を開くことができません: Permission  
denied

> |



Name	Size
..	
.Rhistory	1.2 KB
annotation.txt	383 B
genelist1.txt	18 B
hoge.R	1.2 KB
hoge1.fasta	21 B
hoge1.txt	148 B
list.txt	19 B
sample1.fasta	29 B

# 応用30

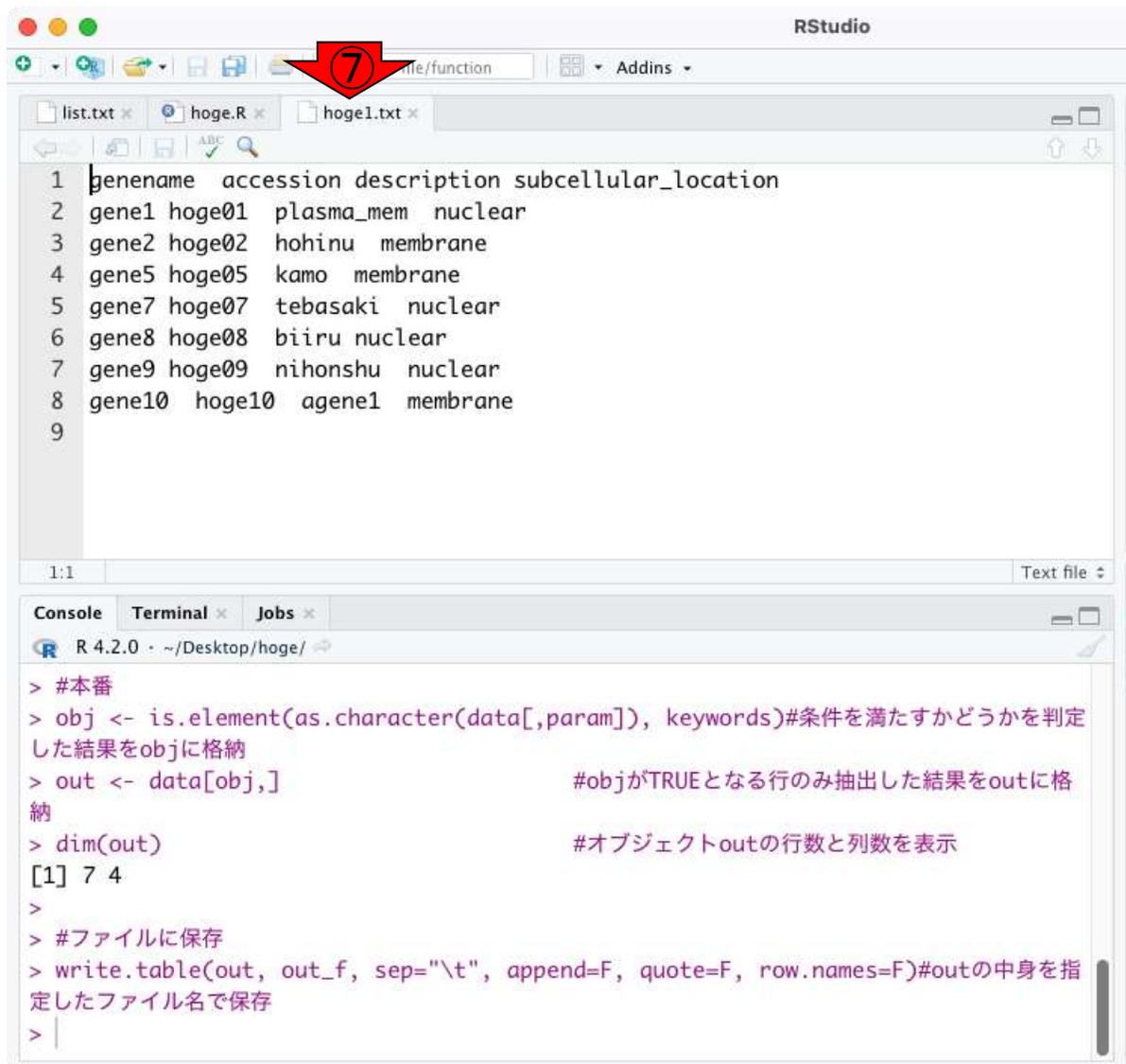
いよいよ変更後のスクリプトの実行。①赤枠内のコード全体を選択。こんな感じ。この状態で②Runを押すと、最後まで実行されたようです。(③表示領域のサイズや④スクロールバーを動かして確認。)Macintoshでも同様の現象が起きるかは不明ですが、出力予定ファイル(hoge.1.txt)をExcelで開いている状態で実行したWinユーザは③のような感じでエラーが出ます。Macintoshでも遭遇した場合は、Excelファイルを閉じて再実行してください。実行結果。⑥をクリックして中身をエディタで表示させてみましょう。

```
1 in_f1 <- "annotation.txt" #入力ファイル名を指定してin_f1に格納
2 in_f2 <- "list.txt" #入力ファイル名を指定してin_f2に格納(リスト)
3 out_f <- "hoge1.txt" #出力ファイル名を指定してout_fに格納
4 param <- 4 #アノテーションファイル中の検索した行数
5
6 #入力ファイルの読み込み
7 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
8 keywords <- readLines(in_f2) #in_f2で指定したファイルの読み込み
9 dim(data) #オブジェクトdataの行数と列数を表示
10
11 #本番
12 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
13
14 #objがTRUEとなる行のみ抽出した結果をoutに格納
15 out <- data[obj,]
16
17 #オブジェクトoutの行数と列数を表示
18 dim(out)
19 [1] 7 4
20
21 #ファイルに保存
22 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
23
```

```
> #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] 7 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
>
```

Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge.1.txt	257 B
list.txt	17 B
hoge.R	1.2 KB

# 応用31



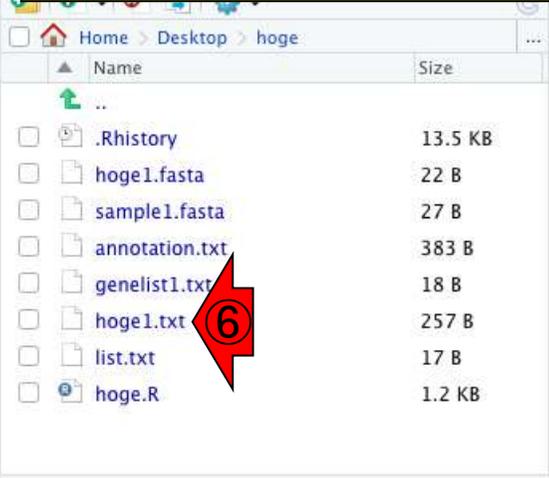
The screenshot shows the RStudio interface. The top pane displays a script with the following content:

```
1 | gene | name | accession | description | subcellular_location |
2 | gene1 | hoge01 | plasma_mem | nuclear |
3 | gene2 | hoge02 | hohinu | membrane |
4 | gene5 | hoge05 | kamo | membrane |
5 | gene7 | hoge07 | tebasaki | nuclear |
6 | gene8 | hoge08 | biiru | nuclear |
7 | gene9 | hoge09 | nihonshu | nuclear |
8 | gene10 | hoge10 | agene1 | membrane |
9 |
```

The bottom pane shows the R console output:

```
> #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] 7 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
> |
```

いよいよ変更後のスクリプトの実行。①赤枠内のコード全体を選択。こんな感じ。この状態で②Runを押すと、最後まで実行されたようです。(③表示領域のサイズや④スクロールバーを動かして確認。)Macintoshでも同様の現象が起きるかは不明ですが、出力予定ファイル(hoge.1.txt)をExcelで開いている状態で実行したWinユーザは③のような感じでエラーが出ます。Macintoshでも遭遇した場合は、Excelファイルを閉じて再実行してください。実行結果。⑥をクリックして中身をエディタで表示させてみましょう。⑦表示結果。



The screenshot shows a file explorer window for the directory ~/Desktop/hoge. The files and their sizes are as follows:

Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	257 B
list.txt	17 B
hoge.R	1.2 KB

A red arrow labeled ⑥ points to the file hoge1.txt.

# 応用32

The screenshot shows the RStudio interface. The top pane displays a script with the following content:

```
1 | gene | name | accession | description | s | lular_location
2 | gene1 | hoge01 | plasma_mem | nuclear
3 | gene2 | hoge02 | hohinu | membrane
4 | gene5 | hoge05 | kamo | membrane
5 | gene7 | hoge07 | tebasaki | nuclear
6 | gene8 | hoge08 | biiru | nuclear
7 | gene9 | hoge09 | nihonshu | nuclear
8 | gene10 | hoge10 | agene1 | membrane
9
```

The bottom pane shows the console output:

```
> #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] 7 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
>
```

Red arrows labeled ⑦ and ⑧ point to the script and the console output, respectively.

いよいよ変更後のスクリプトの実行。①赤枠内のコード全体を選択。こんな感じ。この状態で②Runを押すと、最後まで実行されたようです。(③表示領域のサイズや④スクロールバーを動かして確認。)Macintoshでも同様の現象が起きるかは不明ですが、出力予定ファイル(hoge.1.txt)をExcelで開いている状態で実行したWinユーザは③のような感じでエラーが出ます。Macintoshでも遭遇した場合は、Excelファイルを閉じて再実行してください。実行結果。⑥をクリックして中身をエディタで表示させてみましょう。⑦表示結果。確かに意図通り⑧4列目が nuclear or membraneとなっている行の抽出ができていますね。

The screenshot shows a file explorer window with the following files and sizes:

File Name	Size
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	257 B
list.txt	17 B
hoge.R	1.2 KB

A red arrow labeled ⑥ points to the file 'hoge1.txt'.

# Contents

- 前提条件、RStudioの起動
- 見栄えの統一、終了
- Console画面の見栄え、設定変更
- 枠の幅、電卓的な利用
- プロンプト(>)、Console画面クリア
- 計算の中断
- 指定した範囲の配列を取得
  - 項目と例題、入力ファイルの保存
  - 作業ディレクトリの変更
  - 入力ファイルの確認、FASTA形式
  - コピペ実行、出力ファイルの確認、Environmentタブ
- フィルタリング(任意のキーワードを含む行を抽出)
  - 全体像の把握、入力ファイルの保存、コピペ実行、出力ファイルの確認
  - 応用
- 警告メッセージ、ありがちなミス

## 警告メッセージ1

The screenshot shows the RStudio interface with a warning message in the console. The warning message is:

```
Warning: file 'in_f1' does not exist.
In: read.table(in_f1, header=TRUE, as.is=TRUE, sep=";", as.data.frame(...))
```

Red arrows indicate the warning icon (1) and the console window (2).

The source editor shows the following R code:

```
1 genename accession description subcellular_location
2 gene1 hoge01 plasma_mem nuclear
3 gene2 hoge02 hohinu membrane
4 gene5 hoge05 kamo membrane
5 gene7 hoge07 tebasaki nuclear
6 gene8 hoge08 biiru nuclear
7 gene9 hoge09 nihonshu nuclear
8 gene10 hoge10 agene1 membrane
9
```

The console shows the following R code and output:

```
> #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] 7 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
>
```

The Files pane shows the following files and sizes:

Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	257 B
list.txt	17 B
hoge.R	1.2 KB

# 警告メッセージ2

①を押して、②Console画面を最大化。こんな感じになります。③ページ上部に移動して、コピペ実行したコードの最初のほうが見えるようにします。

The screenshot shows the RStudio interface with the following components:

- Source:** Contains the R script code being executed.
- Console:** Shows the output of the R code, including file paths and dimensions of data objects.
- Environment:** Shows the current environment with variables like 'data', 'keywords', and 'out'.
- Files:** A file explorer showing the directory structure on the desktop, including files like 'hoge1.fasta', 'sample1.fasta', 'annotation.txt', 'genelist1.txt', 'hoge1.txt', 'list.txt', and 'hoge.R'.

```
> 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 <- "list.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 <- 4                          #アノテーションファイル中の検索したい列番号を指定
>
> #入力ファイルの読み込み
> data <- read.table(in_f1, header=TRUE, sep="\t", quote="")#in_f1で指定したファイルの読み込み
> keywords <- readLines(in_f2)        #in_f2で指定したファイルの読み込み
> dim(data)                           #オブジェクトdataの行数と列数を表示
[1] 11 4
>
> #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] 7 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
>
```

The Files panel shows the following table of files:

Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	257 B
list.txt	17 B
hoge.R	1.2 KB

③

# 警告メッセージ3

①を押して、②Console画面を最大化。こんな感じになります。③ページ上部に移動して、コピペ実行したコードの最初のほうが見えるようにします。これは、④の部分で改行を入れた、⑤list.txtを、⑥の部分で読み込ませたもの。特に何のメッセージも表示されていません。

```
RStudio

Source
Console Terminal Jobs
R 4.2.0 - ~/Desktop/hoge/

> 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 <- "list.txt"      ⑤ #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 <- 4              #アノテーションファイル中の検索したい列番号を指定
>
> #入力ファイルの読み込み
> data <- read.table(in_f1, header=TRUE, sep="\t", quote="")#in_f1で指定したファイルの読み込み
> keywords <- readLines(in_f2) ⑥ #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
[1] 11 4
>
> #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] 7 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
>
```

```
#
data <- read.table(in_f1, header=TRUE, sep="\t", quote="")
keywords <- readLines(in_f2) #in_f2で指定したファイルの読み込み
dim(data) #オブジェクトdataの行数と列数を表示
#本番
obj <- is.element(as.character(data[,param]), keywords)#条件を満たすかどうかを判定した結果を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
```

nuclear ↓  
membrane ← ④

Files Plots Packages

Home > Desktop > hoge

Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	257 B
list.txt	17 B
hoge.R	1.2 KB

# 警告メッセージ4

既存の⑤hoge1.txtがある場合は削除しておき、④の部分で改行を入れていない⑥list.txtで実行すると、⑦当該ファイルの読み込み部分で、⑧警告メッセージがでます。

```
RStudio

Source
Console Terminal Jobs
R 4.2.0 ~./Desktop/hoge/

> 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 <- "list.txt" ⑥ #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 <- 4 #アノテーションファイル中の検索したい列番号を指定
>
> #入力ファイルの読み込み
> data <- read.table(in_f1, header=TRUE, sep="\t", quote="")#in_f1で指定したファイルの読み込み
> keywords <- readLines(in_f2) ⑦ #in_f2で指定したファイルの読み込み
警告メッセージ:
readLines(in_f2) で: 'list.txt' で不完全な最終行が見つかりました
> dim(data) #オブジェクトdataの行数と列数を表示
[1] 11 4
>
> #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] 7 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
```

```
To Console To Source
param <- 4 #アノテーションファイル中の検索...
#入力ファイルの読み込み
data <- read.table(in_f1, header=TRUE...
keywords <- readLines(in_f2) #in_f2...
dim(data) #オブジェクトdataの行数と列数...
#本番
obj <- is.element(as.character(data[,param]), keywords)
out <- data[obj,] #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)

Files Plots Packages
Home > Desktop > hoge
Name Size
..
.Rhistory 13.5 KB
hoge1.fasta 22 B
sample1.fasta 27 B
annotation.txt 383 B
genelist1.txt 18 B
hoge1.txt ⑤ 257 B
list.txt 16 B
hoge.R 1.2 KB
```

```
nuclear ↓
membrane ← ④
```

# 警告メッセージ5

既存の⑤hoge1.txtがある場合は削除しておき、④の部分で改行を入れていない⑥list.txtで実行すると、⑦当該ファイルの読み込み部分で、⑧警告メッセージがでます。この場合は、新たにできた⑨出力ファイルを眺めることで、結果には影響しないことがわかります。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。

```
RStudio

Source
Console Terminal Jobs
R 4.2.0 - ~/Desktop/hoge/

> 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 <- "list.txt" ⑥ #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 <- 4 #アノテーションファイル中の検索したい列番号を指定
>
> #入力ファイルの読み込み
> data <- read.table(in_f1, header=TRUE, sep="\t", quote="")#in_f1で指定したファイルの読み込み
> keywords <- readLines(in_f2) ⑦ #in_f2で指定したファイルの読み込み
警告メッセージ:
readLines(in_f2) で: 'list.txt' で不完全な最終行が見つかりました
> dim(data) #オブジェクトdataの行数と列数を表示
[1] 11 4
>
> #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] 7 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
```

```
#本番
obj <- is.element(as.character(data[,param]), keywords)
out <- data[obj,] #オブジェクトoutの抽出
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)
```

```
nuclear ↓
membrane ← ④
```

Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
hoge1.txt	257 B
list.txt	16 B
hoge.R	1.2 KB

# Contents

- 前提条件、RStudioの起動
- 見栄えの統一、終了
- Console画面の見栄え、設定変更
- 枠の幅、電卓的な利用
- プロンプト(>)、Console画面クリア
- 計算の中断
- 指定した範囲の配列を取得
  - 項目と例題、入力ファイルの保存
  - 作業ディレクトリの変更
  - 入力ファイルの確認、FASTA形式
  - コピペ実行、出力ファイルの確認、Environmentタブ
- フィルタリング(任意のキーワードを含む行を抽出)
  - 全体像の把握、入力ファイルの保存、コピペ実行、出力ファイルの確認
  - 応用
- 警告メッセージ、ありがちなミス

# ありがちなミス0

この後エラーを意図的に出すので、まずはうまくいく例を必要最小限のコードで示します。

The screenshot shows the RStudio interface. The console on the left displays the execution of an R script. The environment pane on the right shows the objects created during execution.

```
> 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, 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] 3 4
>
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F)#
>
>
>
```

The Environment pane shows the following objects:

```
param <- 1 #
#入力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) #
#ファイルに保存
write.table(out, out_f, sep="\t", ap...
```

The Files pane shows the directory structure:

Name	Size
..	
.Rhistory	13.5 KB
hoge1.fasta	22 B
sample1.fasta	27 B
annotation.txt	383 B
genelist1.txt	18 B
list.txt	16 B
hoge.R	1.2 KB
hoge1.txt	144 B

# ありがちなミス1

①作業ディレクトリの変更(Set As Working Directory)を忘れているため、  
②in\_f1で指定した最初のファイルの読み込み段階で、③エラーが出ます。作業ディレクトリ内には入力ファイルは存在しないので当然。

The screenshot shows the RStudio interface. The console window displays the following R code and its output:

```
> 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, header=TRUE, sep="\t", quote="")#
file(file, "rt") でエラー: 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:
file(file, "rt") で:
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) #
file(con, "r") でエラー: 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:
file(con, "r") で:
ファイル 'genelist1.txt' を開くことができません: No such file or directory
> dim(data) #
[1] 11 4
>
> #本番
> obj <- is.element(as.character(data[,param]), keywords)#
> out <- data[obj,] #
> dim(out) #
[1] 3 4
>
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F)#
```

Red arrows point to the file paths in the code: ② points to "annotation.txt" and "genelist1.txt", and ③ points to the error messages. The Files pane on the right shows the Home directory with a red arrow ① pointing to the Home icon, indicating that the working directory is not set to the project directory.

# ありがちなミス2

①作業ディレクトリの変更(Set As Working Directory)はできているが、②in\_f2で指定した最初のファイルが③存在しないので、④その読み込み段階でエラーが出ます。

```
RStudio

Source
Console Terminal Jobs
R 4.2.0 - ~/Desktop/hoge/
> 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, header=TRUE, sep="\t", quote="")#
> keywords <- readLines(in_f2) #
file(con, "r") でエラー: 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:
file(con, "r") で:
ファイル 'genelist1.txt' を開くことができません: No such file or directory
> dim(data) #
[1] 11 4
>
> #本番
> obj <- is.element(as.character(data[,param]), keywords)#
> out <- data[obj,] #
> dim(out) #
[1] 3 4
>
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F)#
>
>
>
```

```
To Console To Source
param <- 1 #
#入力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) #
#ファイルに保存
write.table(out, out_f, sep="\t", ap...

Files Plots Packages Help Viewer
Home > Desktop > hoge
Name Size
..
annotation.txt 383 B
```

