

ゲノム情報解析基礎：第2回

¹大学院農学生命科学研究科
アグリバイオインフォマティクス教育研究プログラム
²微生物科学イノベーション連携研究機構
門田幸二(かどた こうじ)
kadota@iu.a.u-tokyo.ac.jp
<http://www.iu.a.u-tokyo.ac.jp/~kadota/>

各講義科目へのアクセス

①教育プログラム、②各講義のページ、③「ゲノム情報解析基礎」の場合。ブラウザは、Google Chromeを推奨。



東京大学大学院農学系研究センター
アグリバイオ
Agricultural Bioinformatics

ようこそ!!
アグリバイオ
教育研究センター

- + ホーム
- + 本ユニットについて
- + メンバー
- + 教育プログラム **①**
- + 研究フォーラム
- + イベント
- + お問い合わせ
- + リンク

東京大学
THE UNIVERSITY OF TOKYO

教育プログラム

▼ プログラム概要 ▼ 講義について ▼ 受講について ▼ 各講義のページ **②**

▼ スケジュール

プログラム概要

本プログラムで開講する講義科目は()に分けられます。カテゴリーと

カテゴリー	目的
基礎	主にバイオインフォマティクス。生命科学のためのツールやソフトウェアを利用した様々なツ
方法論	「基礎」の科目を土台として、解析手法、質量分析法など)やモデル選択、分子シミュレーション(加)について解説し

お知らせ

- ▶ 2019年度の受講希望者は2019年4月5日までに事務局までお越しください。
- ▶ 2019年度受講生募集要項は [こちら](#) (PDF) です。 **NEW!!**
- ▶ 受講に関する質問はまずこちらをごらんください。
Q & A集(本学の大学院生の方)
Q & A集(本学の大学院生以外の方)
- ▶ 成績証明書の発行を希望される方は申込用紙「Word形式、PDF形式」を事務局までご連絡ください。

各講義のページ

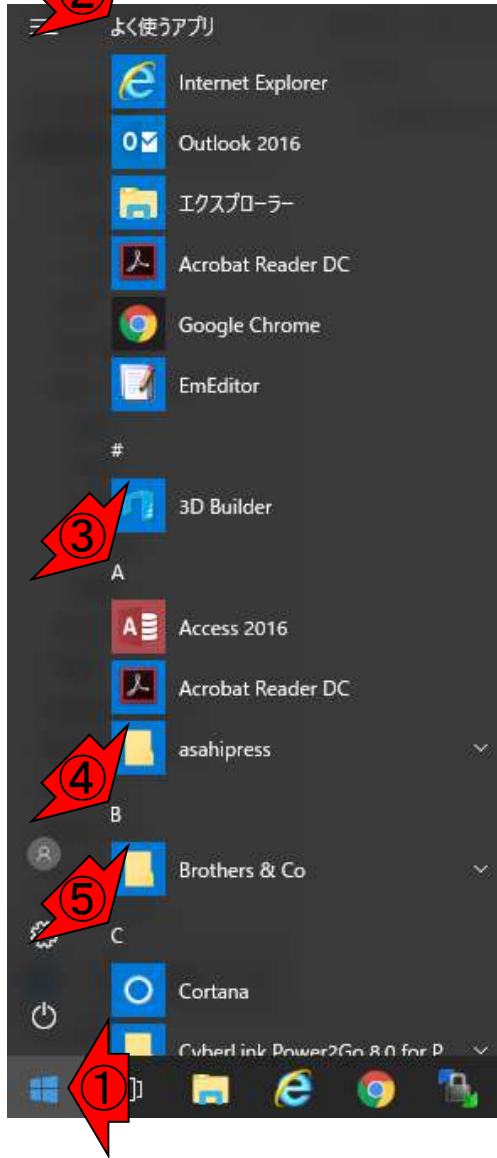
(科目名をクリックすると各講義のページに移動します)

先端トピックス セミナー・討論形式 研究指導	農学生命情報科学特別演習			
	農学生命情報科学特論 I	農学生命情報科学特論 II	農学生命情報科学特論 III	農学生命情報科学特論 IV
方法論 講義・実習を 一体化	フィールドインフォマティクス			
	ゲノム情報解析基礎	構造バイオインフォマティクス基礎	生物配列解析基礎	バイオスタティスティクス基礎論
基礎 講義・実習を 一体化	ゲノム情報解析基礎			

③

Excelの場所

Windows10の画面左下のあたりを表示。②のあたりに着目して下のほうをずっと眺めていくと、③A、④B、⑤Cのようにアルファベット順になっていることがわかる。EからはじまるところにExcelがあるはずです。今回は行列データファイルを取り扱うので、Excelで眺めると分かりやすいかもしれません。



Contents

- 任意のキーワードを含む行を抽出(行列データの取り扱い)
- 色の説明と応用
- ありがちなミスや警告メッセージ
- コードの解説
 - 最初の4行分、read.table関数、quoteオプション(昔話)
 - Tips(上下左右の矢印キーを有効活用)、ヘッダ一行の有無
 - dim関数、nrow関数とncol関数、?関数名、class関数、dim関数マニュアルの続き
 - dim関数マニュアルの続き、dimnames、rownames、colnames、length
 - readLines、Tips(タブ補完)

位置づけと心構え

①(Rで)塩基配列解析の、②基本的な利用法(③Win版 or ④Mac版)の続きとして、Rを用いた行列形式ファイルの取り扱いからスタートします。ゲノム情報解析と関係性が低いと思われるかもしれませんが、このあたりの基礎が多くの発展的な解析の基本スキルとして重要。

(Rで)塩基配列解析

x +

保護されていない通信 | www.iu.a.u-tokyo.ac.jp/~kadota/

(Rで)塩基配列解析 ①

(last modified 2019/03/11, since 2010)

このウェブページのR関連部分は、[インストール |](#) についての推奨手順 (Windows2018.11.15版とMacintosh2018.11.27版) についてフリーソフト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](#)(Windows2018.12.23版と Macintosh2019.01.15版)で自習してください。2018年7月に(Rで)塩基配列解析の一部(講習会・書籍③ 会誌など)を切り分け、[ページ](#)に移行しました。(2018/07/18)

What's new? (過去のお知らせはこちら)

- 2019年度も[アグリバイオインフォマティクス教育研究プログラム](#)を実施します。例年東大以外の企業の方、研究員、大学院生が2割程度受講しております。受講ガイダンスは、2019年4月5日(Fri.)17:15より東大農学部2号館2階化学第一講義室で開催します。(2019/03/11) **NEW**
- 細かいところの修正はここに明記してなくても随時行っています。(2019/03/11) **NEW**
- 「インストール | Rパッケージ | [必要最小限プラスアルファ](#)」を更新しました。(2019/03/08) **NEW**
- 「[生命科学データ解析を支える情報技術](#) (監修: 坊農秀雅)」が出版されています。最先端のネタを含むかなり広範な内容を含んでいますので、一通り目次を眺めてみるとよいと思います。Bioconda, Homebrew, Docker, GitHub, EC2, AWSなど聞いたことがある有用そうなものの全体像がわかるというメリットがあると思います。(2019/02/06)

- [はじめに](#) (last modified 2018/08/04)

[トップページへ](#)

- [過去のお知らせ](#) (last modified 2019/03/02) **NEW**


全体像

目的: タブ区切りテキストファイル(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	gene1	hoge01	plasma_mem	nuclear
2	gene2	hoge02	hohinu	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

出力: hoge1.txt



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

入力: リストファイル(genelist1.txt)

	A
1	gene1
2	gene7
3	gene9

任意のキーワード

目的: タブ区切りテキストファイル(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の②例題1をやりませう。

- インストール | Rパッケージ | [個別\(2018年11月以前\)](#) (last modified 2015/04/03)
- [基本的な利用法](#) (last modified 2015/04/03)
- [サンプルデータ](#) (last modified 2018/06/09)
- 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\(基礎\)](#) ① (last modified 2016/04/20)
- 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 2015/02/19)
- イントロ | 一般 | [指定した範囲の配列を取得](#) (last modified 2015/02/19)
- イントロ | 一般 | [指定したID\(染色体やdescription\)の配列を取得](#) (last modified 2015/02/19)
- イントロ | 一般 | [翻訳配列\(translate\)を取得\(基礎\)](#) ② (last modified 2015/02/19)
- イントロ | 一般 | [翻訳配列\(translate\)を取得\(応用\)](#) (last modified 2015/02/19)

イントロ | 一般 | 任意のキーワードを含む行を抽出(基礎)

例えばタブ区切りテキストファイルが手元があり、この中からリストファイル中の文字列を含む行を抽出するやり方を示します。Linux (UNIX)のgrepコマンドのようなものであり、perlのハッシュのようなものです。「ファイル」 - 「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

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

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

作業ディレクトリは①「デスクトップ - hoge」。hogeフォルダ中に②annotation.txtと③genelist1.txtが存在するという前提。

任意のキーワード

イントロ | 一般 | 任意のキーワードを含む行を抽出(基礎)

例えばタブ区切りテキストファイルが手元があり、この中からリストファイルを示します。Linux (UNIX)のgrepコマンドのようなものであり、perlのパーラー「ファイル」 - 「ディレクトリの変更」で解析したいファイルを用意してある

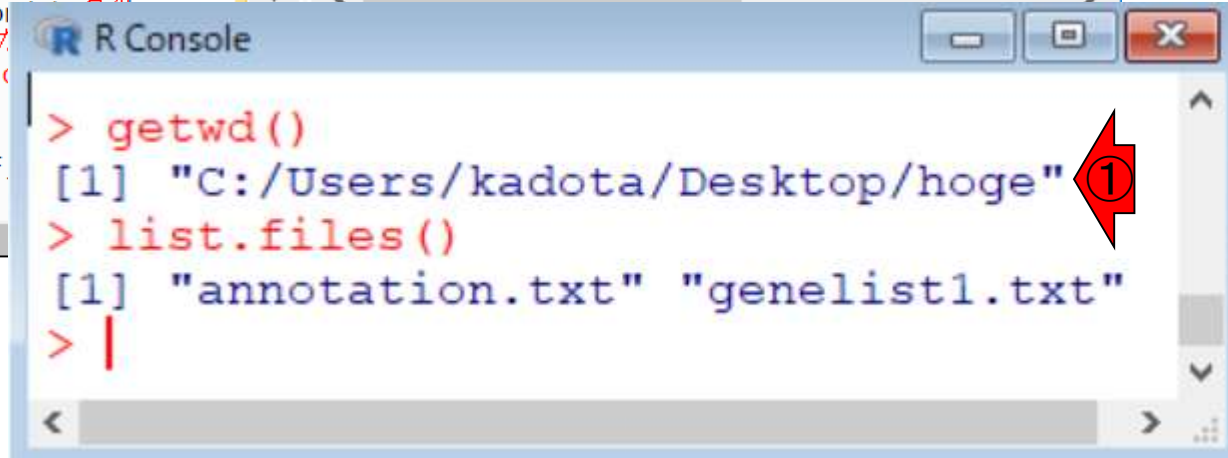
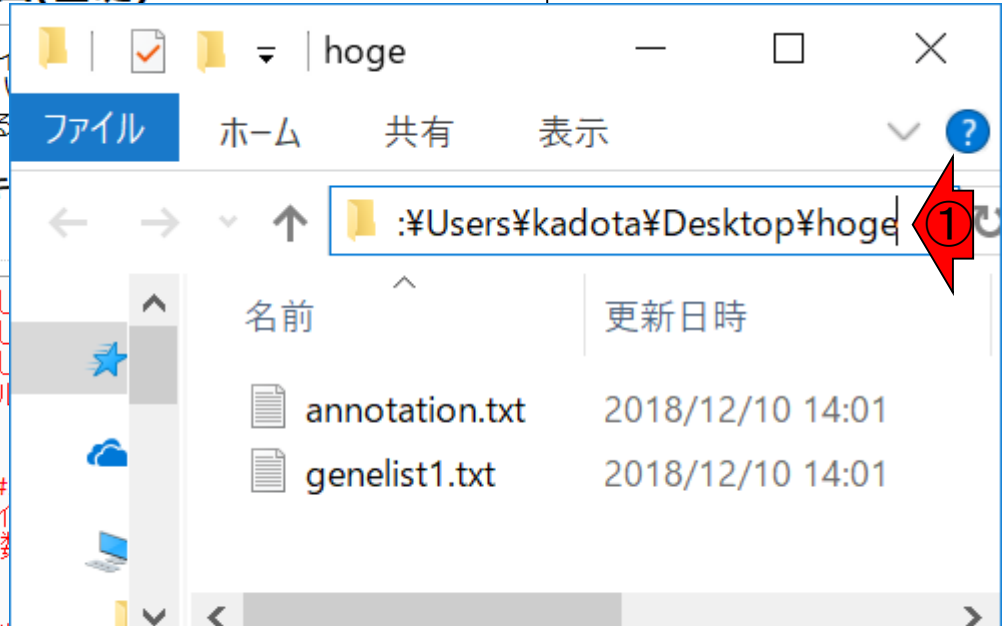
1. 目的のタブ区切りテキストファイル(②annotation.txt)中の第1列目をキーワード(③genelist1.txt)中のものが含まれる行全体を出力したい場合:

```
in_f1 <- "annotation.txt" #入力ファイル名を指定し
in_f2 <- "genelist1.txt" #入力ファイル名を指定し
out_f <- "hoge1.txt" #出力ファイル名を指定し
param <- 1 #アンテーションファイル

#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) #in_f2で指定したファイルの各行を読み込み
dim(data) #オブジェクトdataの行数と列数を取得

#本番
obj <- is.element(as.character(data[,param]), keywords)
out <- data[obj,] #objがTRUEとならばその行を抽出
dim(out) #オブジェクトoutの行数と列数を取得

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



実行結果

①list.files()で表示される結果と、②実行後のhogeフォルダの中身は当然同じ

```
R Console
> #本番
> obj <- is.element(as.character(data[,p$
> out <- data[obj,]
> dim(out)
[1] 3 4
>
> #ファイルに保存
> write.table(out, out_f, sep="\t", appe$
> list.files()
[1] "annotation.txt" "genelist1.txt"
[3] "hoge1.txt"
> |
```

実行前のhogeフォルダ

名前	更新日時
annotation.txt	2018/12/10 14:01
genelist1.txt	2018/12/10 14:01

実行後のhogeフォルダ

名前	更新日時
annotation.txt	2018/12/10 14:01
genelist1.txt	2018/12/10 14:01
hoge1.txt	2018/12/10 14:07

R上で確認できるように

①出力ファイル(hoge1.txt)の中身。
これは②outというオブジェクトの中
身を、③write.tableという関数を用
いてファイルに出力したもの。

```
R Console  
> write.table(out, out_f, sep="\t", append=F, quote=F, $  
> list.files()  
[1] "annotation.txt" "genelist1.txt" "hoge1.txt"  
> out  
  gene_name accession description subcellular_location  
1   gene1     hoge01  plasma_mem      nuclear  
7   gene7     hoge07   tebasaki        nuclear  
9   gene9     hoge09   nihonshu        nuclear  
> |
```

	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



実行後のhogeフォルダ

名前	更新日時
annotation.txt	2018/12/10 14:01
genelist1.txt	2018/12/10 14:01
hoge1.txt	2018/12/10 14:07

R上で確認できるように

それゆえ、①出力ファイル(hoge1.txt)の②中身は、Rコンソール画面中で③outと打ち込むことで確認できる。早く慣れてRコンソール画面のみ動作確認できるようになっていこう。

```
R Console  
> write.table(out, out_f, sep="\t", append=F, quote=F, $  
> list.files()  
[1] "annotation.txt" "genelist1.txt" "hoge1.txt"  
> out  
  gene_name accession description subcellular_location  
1   gene1     hoge01  plasma_mem      nuclear  
7   gene7     hoge07   tebasaki       nuclear  
9   gene9     hoge09   nihonshu       nuclear  
> |
```

hoge1.txt ①

	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

②

実行後のhogeフォルダ

名前	更新日時
annotation.txt	2018/12/10 14:01
genelist1.txt	2018/12/10 14:01
hoge1.txt	2018/12/10 14:07

Contents

- 任意のキーワードを含む行を抽出(行列データの取り扱い)
- 色の説明と応用
- ありがちなミスや警告メッセージ
- コードの解説
 - 最初の4行分、read.table関数、quoteオプション(昔話)
 - Tips(上下左右の矢印キーを有効活用)、ヘッダ一行の有無
 - dim関数、nrow関数とncol関数、?関数名、class関数、dim関数マニュアルの続き
 - dim関数マニュアルの続き、dimnames、rownames、colnames、length
 - readLines、Tips(タブ補完)

色の説明

(Rで)塩基配列解析

(last modified 2018/11/25, since 2010)

このウェブページのR関連部分は、[インストール](#) | についての推奨手順 (Windows2018.11.15版とMacintosh2015.04.03版; Mac版は古い)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](#) (Windows2015.04.03版とMacintosh2015.04.03版)を習ってください。2018年7月に[\(Rで\)塩基配列解析](#)の一部 (講習会・書籍・学会誌など) を切り分けて[サブページ](#)に移行

What's new? (過去のお知らせはこちら)

- 2018年11月に「よくわかるバイオインフォマティクス入門 (藤博幸 編)」という本
- [TCCのオンラインGUI版 \(のベータ版\)](#) を公開しました。(2018/10/15)

- [はじめに](#) (last modified 2018/08/04)
- [過去のお知らせ](#) (last modified 2018/11/25) **NEW**
- [インストール | について](#) (last modified 2018/11/15) **NEW**
- インストール | R本体 | 最新版 | [Win用](#) (last modified 2018/11/12) 推奨 **NEW**
- インストール | R本体 | 最新版 | [Mac用](#) (last modified 2018/11/12) 推奨 **NEW**
- 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) **NEW**
- インストール | Rパッケージ | [ほぼ全て](#) (削除予定) (last modified 2015/05/25)
- 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 2018/11/20) 推奨 **NEW**
- インストール | Rパッケージ | [必要最小限](#) (削除予定) (last modified 2015/05/25)
- インストール | Rパッケージ | [個別\(2018年11月以降\)](#) (last modified 2018/11/12) **NEW**
- 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) **NEW**

[トップページへ](#)

コメント

特にやらなくてもいいコマンド

プログラム実行時に目的に応じて変更すべき箇所

応用

①例題1は、②1列目でキーワード検索するやり方でした。つまり、③の列が対象でした。

イントロ | 一般 | 任意のキーワードを含む行を抽出(基礎)

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

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ペ。

1. 目的のタブ区切りテキストファイル(annotation.txt)中の第1列目をキーとして、リストファイル(genelist.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 <- "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(出力ファイル)
param <- 1 #アノテーションファイルの列番号
```

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

#本番

```
obj <- is.element(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で抽出した結果をoutに格納
dim(out) #オブジェクトoutの行数と列数を表示
```

```
#ファイルに出力
write.table(out, out_f, sep="\t", as=F) #outの中身を指定したファイル名で出力
```

コメント

特にやらなくてもいいコマンド

プログラム実行時に目的に応じて変更すべき箇所

	A	B	C	D
1	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

応用

イントロ | 一般 | 任意のキーワードを含む行を抽出(基礎)

①例題1は、②1列目でキーワード検索するやり方でした。つまり、③の列が対象でした。応用として、別のリストファイルを読み込んで、④4列目で検索するやり方を示します。

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

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(genelist.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 <- "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(出力ファイル)
param <- 1 #アノテーションファイルの列番号
```

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

#本番

```
obj <- is.element(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,] #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
```

#ファイルに出力

```
write.table(out, out_f, sep="\t", quote="")
```

コメント

特にやらなくてもいいコマンド

プログラム実行時に目的に応じて変更すべき箇所

	A	B	C	D
1	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

	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

おさらい。例題1の①リストファイル(genelist1.txt)中の、キーワードリストは②のように書かれていた。

リストファイルの作成

イントロ | 一般 | 任意のキーワードを含む行を抽出(基礎)

例えばタブ区切りテキストファイルが手元があり、この中からリストファイル中の文字列を含む行を抽出するやり方を示します。Linux (UNIX)のgrepコマンドのようなものであり、perlのハッシュのようなものです。
「ファイル」 - 「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

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

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



gene1
gene7
gene9
|

① 名を指定してin_f1に格納(アノテーションファイル)
② 名を指定してin_f2に格納(リストファイル)
名を指定してout_fに格納
in_f1ファイル中の検索したい列番号を指定
#in_f1で指定したファイルの読み込み
読み込みしたファイルの読み込み
dataの行数と列数を表示

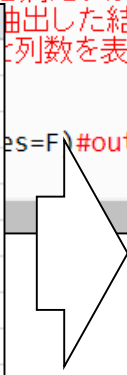
```
#入力ファイルの読み込み
data <- read.table(in_f1, header=TRUE, as.is=TRUE, sep=" ")
keywords <- readLines(in_f2)
dim(data)

#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,]
dim(out)

#ファイルに出力
write.table(out, out_f, sep="F", #out
```

	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

	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



リストファイルの作成

イントロ | 一般 | 任意のキーワードを含む行を抽出(基本)

今回は、①のようなnuclearとmembraneを含む行を抽出したいので、②のようなリストファイルを作成し、list.txtというファイル名で保存しておく。当然ながら、保存場所はannotation.txtと同じ場所です。

例えばタブ区切りテキストファイルが手元があり、この中からリストファイル中の文字列を含む行を抽出するやり方を示します。Linux (UNIX)のgrepコマンドのようなものであり、perlのハッシュのようなものです。
「ファイル」 - 「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

1. 目的のタブ区切りテキストファイル(annotation.txt)中の第1列目をキーとして、リストファイル(genelist.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 <- "genelist.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
```

	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

	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

②

```
nuclear
membrane
```

①

Rエディタを起動

①「ファイル - 新しいスクリプト」で、②Rエディタを起動。
もちろん、他のエディタをどうしても使いたければ、「二重クォーテーション問題」などに遭遇する可能性を踏まえ、自己責任でご利用ください。

イントロ | 一般 | 任意の

RGui (64-bit)

ファイル 編集 パッケージ ウィンドウ ヘルプ

- ① 新しいスクリプト Ctrl+N
- スクリプトを開く... Ctrl+O
- 保存 Ctrl+S
- 別名で保存...
- 印刷... Ctrl+P
- スクリプトを閉じる

R 無題 - Rエディタ

例えばタブ区切りテキストファイルが手
を示します。Linux (UNIX)のgrepコマ
「ファイル」 - 「ディレクトリの変更」

1. 目的のタブ区切りテキストファイル(4
(genelist1.txt)中のもが含まれる行

```
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, heade  
keywords <- readLines(in_f2)  
dim(data)
```

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

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

例題の赤枠内のコードを、
①Rエディタにコピペ。

テンプレートをコピペ

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ることを示します。Linux (UNIX)のgrepコマンドで「ファイル」-「ディレクトリの変更」で解

1. 目的のタブ区切りテキストファイル(ann
(genelist1.txt)中のものが含まれる行全体

```
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=T,  
keywords <- readLines(in_f2)  
dim(data)  
  
#本番  
obj <- is.element(as.character(data  
out <- data[obj,]  
dim(out)  
  
#ファイルに保存  
write.table(out, out_f, sep="\t",
```

The screenshot shows the RGui (64-bit) interface. The R Console window displays the R code from the previous block. A red arrow with the number 1 points to the R editor window titled '無題 - RIエディタ', which contains the same R code being copied into it. The R editor window also shows the same code with comments in Japanese explaining the steps: reading input files, loading keywords, performing the main operation, and saving the output to a file.

必要最小限の箇所を変更

必要最小限の変更箇所は、①リストファイルの名前と、②列番号情報。

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るとします。Linux (UNIX)のgrepコマンドで「ファイル」 - 「ディレクトリの変更」で検索

1. 目的のタブ区切りテキストファイル(annotation.txt) (genelist1.txt)中のものが含まれる行全体

```
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)
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",
```

The screenshot shows the RGui (64-bit) interface with an R Console window titled '無題 - RIデータ'. The console displays the following code with red annotations:

```
R は、自由な...
一定の条件に...
配布条件の詳...

R は多くの貢献...
詳しくは 'con...
また、R や R...

'citation()' と...
'demo()' と...
'help()' と...
'help.start'...
'q()' と入力...

[以前にセー...

> |

#入力ファイルの読み込み
data <- read.table(in_f1, header=TRUE, sep="\t", quote="") #in f
keywords <- readLines(in_f2) #in f2で指定したファイルの読み込
dim(data) #オブジェクト dataの行数と列数を表

#本番
obj <- is.element(as.character(data[,param]), keywords) #条件を満
out <- data[obj,] #objがTRUEとなる行のみ抽出した
dim(out) #オブジェクト outの行数と列数を表

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

Red arrows labeled ① and ② point to the changes in the code: ① points to the file name `"list.txt"` in the `in_f2` assignment, and ② points to the column index `4` in the `param` assignment.

必要最小限の箇所を変更

必要最小限の変更箇所は、①リストファイルの名前と、②列番号情報。③list.txt中のキーワードで、④4列目で探したいからです。

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ることを示します。Linux (UNIX)のgrepコマンドで「ファイル」-「ディレクトリの変更」で解決

1. 目的のタブ区切りテキストファイル(annotation.txt) (genelist1.txt)中のものが含まれる行全体

```
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)
```

```
#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",
```

RGui (64-bit) window showing the R Console with the following code and output:

```
R 無題 - RIデータ
in_f1 <- "annotation.txt"
in_f2 <- "list.txt"
out_f <- "hoge1.txt"
param <- 4

#入力ファイルの読み込み
data <- read.table(in_f1, header=TRUE, sep="\t", quote="")
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",
```

	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

Tips: CTRL + R

変更後のコードを全選択したのち、右クリックで「コピー (CTRL + C) & ペースト (CTRL + V)」とやらなくても、①「CTRL + R」でイけます。

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るとします。Linux (UNIX) の grep コマンドで「ファイル」 - 「ディレクトリの変更」で解決

1. 目的のタブ区切りテキストファイル (annotation.txt) (genelist1.txt) 中のものが含まれる行全体

```
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=T, 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",
```

The screenshot shows the RGui (64-bit) interface with the R Console window open. The console contains R code for reading files and filtering rows. A context menu is displayed over the code, with the 'Ctrl+R' option highlighted. A red circle with the number '1' is placed over the 'Ctrl+R' option in the menu.

RGui (64-bit) ファイル 編集 パッケージ ウィンドウ ヘルプ

R Console

R 無題 - RI データ

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

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

#本番
obj <- is.element(as
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=
```

カーソル行または選択中の R コードを実行 Ctrl+R

やり直し Ctrl+Z

カット Ctrl+X

コピー Ctrl+C

ペースト Ctrl+V

消去

全て選択 Ctrl+A

Tips: CTRL + R

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ることを示します。Linux (UNIX)のgrepコマンドで「ファイル」 - 「ディレクトリの変更」で解決

1. 目的のタブ区切りテキストファイル(annot1.txt) (genelist1.txt)中のものが含まれる行全体

```
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=T, sep="\t")
keywords <- readLines(in_f2)
dim(data)
```

#本番

```
obj <- is.element(as.character(keywords), data$V1)
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)
```

変更後のコードを全選択したのち、右クリックで「コピー(CTRL + C) & ペースト(CTRL + V)」とやらなくても、①「CTRL + R」でイケます。これは、②CTRL + ③Rでもよいことを意味します。

RGui (64-bit) window showing a script editor with R code. The context menu is open, showing options like 'Ctrl+R', 'Ctrl+Z', 'Ctrl+X', 'Ctrl+C', 'Ctrl+V', and 'Ctrl+A'. A keyboard diagram highlights the Ctrl and R keys. Red arrows point to the Ctrl key (labeled 2) and the R key (labeled 3) on the keyboard, and to the Ctrl+R option in the menu (labeled 1).

Tips: CTRL + R

イントロ | 一般 | 任意のキーワードを含む行を抽出

例えばタブ区切りテキストファイルが手元にあることを示します。Linux (UNIX)のgrepコマンドで「ファイル」 - 「ディレクトリの変更」で解

1. 目的のタブ区切りテキストファイル(annotation.txt) (genelist1.txt)中のものが含まれる行全体

```
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)
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",
```

RGui (64-bit)

ファイル 編集 パッケージ ウィンドウ ヘルプ

R Console

```
> #入力ファイル名を指定してin_f1に格納し
> data <- read.table(in_f1, header=TRUE)
> keywords <- readLines(in_f2)
> dim(data)
[1] 11 4
> #入力ファイルの読み込み
> #本番
> obj <- is.element(as.character(data[,param]), keywords)
> out <- data[obj,]
> dim(out)
[1] 7 4
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F, quote=F, row.names=
```

R 無題 - RIデータ

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

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

#本番
obj <- is.element(as.character(data[,param]), keywords)#条件を満たす
#objがTRUEとなる行のみ抽出したオブジェクトoutの行数と列数を表
```

実行後はこんな感じになります。①出力ファイル名は変更していないので、①hoge1.txtファイルに上書きされるはず。もしExcelなどでhoge1.txtを開いていると「Permission denied(書き込み権限がない)」みたいなエラーが出ます(後述する、ありがちなミス3)

list.files()で確認

①list.files()と打ち込んで確認。確かに出力ファイルに相当する名前は②hoge1.txtしかないですね。

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ることを示します。Linux (UNIX)のgrepコマンドで「ファイル」-「ディレクトリの変更」で解

1. 目的のタブ区切りテキストファイル(annotation.txt) (genelist1.txt)中のものが含まれる行全体

```
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=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")
```



```
RGui (64-bit)
ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ

R Console
> 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
[1] 11 4
>
> #本番
> obj <- is.element(as.character(data[,param]), keywords) #条件を満たす行のみ抽出したファイルの読み込み
> out <- data[obj,] #objがTRUEとなる行の行数と列数を表
> dim(out) #オブジェクトoutの行数と列数を表
[1] 7 4
>
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F, quote=F, row.names=)
>
> list.files()
[1] "annotation.txt" "genelist1.txt" "hoge1.txt"
[4] "list.txt"
> |
```



①outと打ち込んで確認。

outの中身を確認

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はありません。Linux (UNIX)のgrepコマンドで「ファイル」 - 「ディレクトリの変更」で解決

1. 目的のタブ区切りテキストファイル(annotation.txt)中のものが含まれる行全体(genelist1.txt)中のものが含まれる行全体

```

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=T)
keywords <- readLines(in_f2)
dim(data)

#本番
obj <- is.element(as.character(data[,1]), keywords)
out <- data[obj,]
dim(out)

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

```



RGui (64-bit) window showing R Console output:

```

>
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F, quote=F, row.names=
>
> list.files()
[1] "annotation.txt" "genelist1.txt" "hoge1.txt"
[4] "list.txt"
> out
  genename accession description subcellular_location
1    gene1   hoge01  plasma_mem             nuclear
2    gene2   hoge02    hohinu             membrane
5    gene5   hoge05      kamo             membrane
7    gene7   hoge07  tebasaki             nuclear
8    gene8   hoge08    biiru             nuclear
9    gene9   hoge09  nihonshu             nuclear
10   gene10   hoge10    agenel             membrane
>

```

outの中身を確認

①outと打ち込んで確認。確かに②list.txtファイル中に書き込まれたキーワードを含む行のみが抽出できていることが、③からわかります。

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ることを示します。Linux (UNIX)のgrepコマンドで「ファイル」 - 「ディレクトリの変更」で解

1. 目的のタブ区切りテキストファイル(annotation.txt)中のものが含まれる行全体 (genelist.txt)中のもが含まれる行全体

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

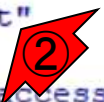
#入力ファイルの読み込み
data <- read.table(in_f1, header=T)
keywords <- readLines(in_f2)
dim(data)

#本番
obj <- is.element(as.character(data[,1]), keywords)
out <- data[obj,]
dim(out)

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

The screenshot shows the RGui (64-bit) interface with the R Console window open. The console displays the execution of R code and the resulting output table. The output table is highlighted with a red box, and three red arrows point to specific parts of the code and output.

```
>
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F, quote=F, row.names=)
>
> list.files()
[1] "annotation.txt" "genelist1.txt" "hoge1.txt"
[4] "list.txt"
> out
  genename accession description subcellular_location
1    gene1   hoge01  plasma_mem          nuclear
2    gene2   hoge02    hohinu             membrane
5    gene5   hoge05      kamo             membrane
7    gene7   hoge07  tebasaki          nuclear
8    gene8   hoge08    biiru             nuclear
9    gene9   hoge09  nihonshu          nuclear
10   gene10   hoge10    agenel            membrane
```



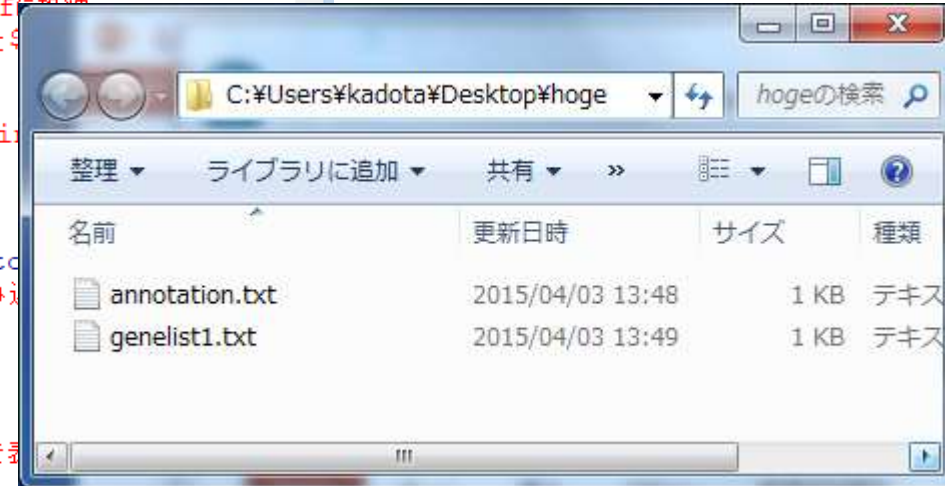
Contents

- 任意のキーワードを含む行を抽出(行列データの取り扱い)
- 色の説明と応用
- ありがちなミスや警告メッセージ
- コードの解説
 - 最初の4行分、read.table関数、quoteオプション(昔話)
 - Tips(上下左右の矢印キーを有効活用)、ヘッダ一行の有無
 - dim関数、nrow関数とncol関数、?関数名、class関数、dim関数マニュアルの続き
 - dim関数マニュアルの続き、dimnames、rownames、colnames、length
 - readLines、Tips(タブ補完)

ありがちなミス1

作業ディレクトリの変更を忘れていたため、①in_f1で指定した最初のファイルの読み込み段階で、②エラーが出る。つまり、③作業中のフォルダにはannotation.txtというファイルは存在しないということ。

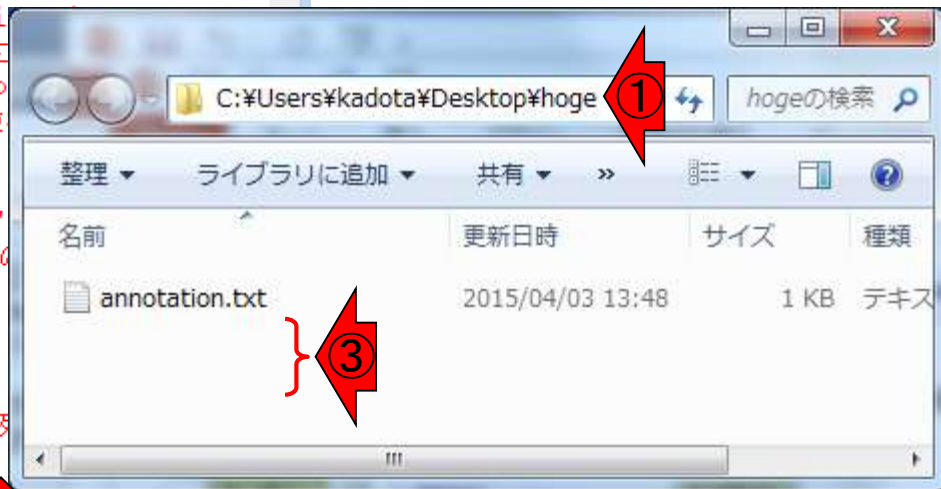
```
R Console
> in_f1 <- "annotation.txt" #入力ファイル名を指定してin_f1に格納
> in_f2 <- "list.txt" #入力ファイル名を指定してin_f2に格納
> out_f <- "hogel.txt" #出力ファイル名を指定してout_fに格納
> param <- 4 #アンテーションファイル中の検索した回数
>
> #入力ファイルの読み込み
> data <- read.table(in_f1, header=TRUE, sep="\t", quote="") #in_f1で指定したファイルの読み込み
file(file, "rt") 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) #in_f2で指定したファイルの読み込み
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 'list.txt' を開くことができません: No such file or directory
> dim(data) #オブジェクトdataの行数と列数を表示
NULL
>
> #本番
> obj <- is.element(as.character(data[,param]), keywords) #条件を満たすかどうか
data[, param] でエラー:
'closure' 型のオブジェクトは部分代入可能ではありません
> out <- data[obj,] #objがTRUEとなる行のみ抽出した結果
エラー: オブジェクト 'obj' がありません
> dim(out) #オブジェクトoutの行数と列数を表示
エラー: オブジェクト 'out' がありません
>
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F) #outの中
is.data.frame(x) でエラー: オブジェクト 'out' がありません
> getwd()
[1] "C:/Users/kadota/Documents"
> |
```



ありがちなミス2

必要な入力ファイルが①作業ディレクトリ中に存在しない。この場合、②in_f2で指定したgenelist1.txtが、③存在しないため、④その読み込み段階でエラーが出ている。それゆえ、⑤その情報を用いるコマンド部分でエラーが出ている。

```
R Console
> getwd()
[1] "C:/Users/kadota/Desktop/hoge"
> in_f1 <- "annotation.txt"
> in_f2 <- "list.txt"
> out_f <- "hoge1.txt"
> param <- 4
>
> #入力ファイルの読み込み
> data <- read.table(in_f1, header=TRUE, sep="\t", quote="")
> 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 'list.txt' を開くことができません: No such file or directo$
> dim(data)
[1] 11 4
>
> #本番
> obj <- is.element(as.character(data[,param]), keywords)
is.element(as.character(data[, param]), keywords) でエラー:
  オブジェクト 'keywords' がありません
> out <- data[obj,]
`[.data.frame`(data, obj, ) でエラー: オブジェクト 'obj' がありません
> dim(out)
エラー: オブジェクト 'out' がありません
>
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F)
is.data.frame(x) でエラー: オブジェクト 'out' がありません
>
> |
```

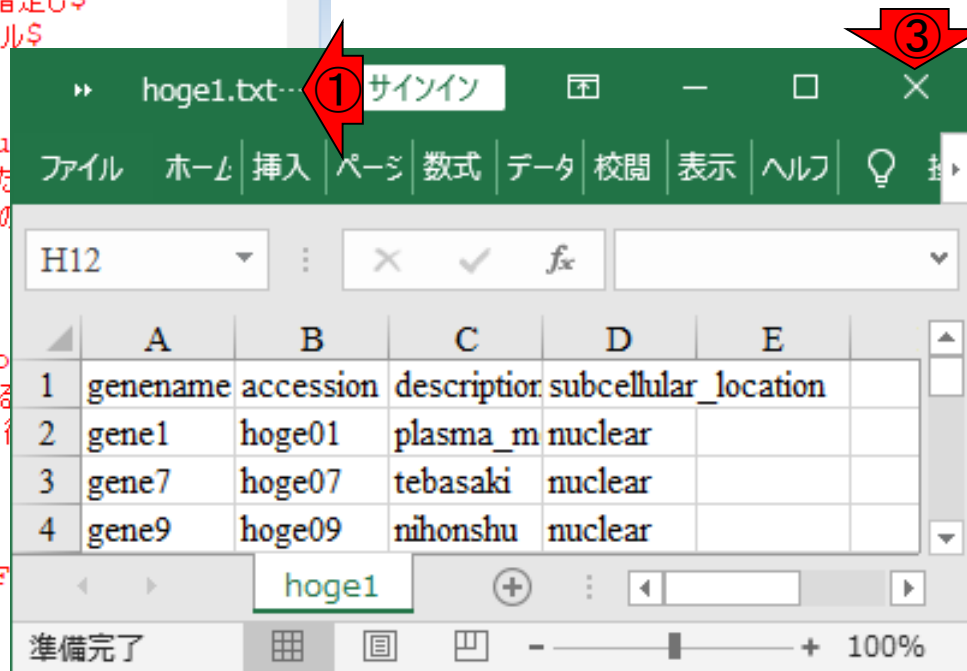


ありがちなミス3

①出力予定のファイル名と同じものをエクセルなど別のプログラムで開いているため、②最後のwrite.table関数のところでエラーが出る。対処法は、出力ファイル名を変更するか、③開いている別のプログラムを閉じる。

R Console

```
> in_f1 <- "annotation.txt" #入力ファイル名を指定し$
> in_f2 <- "list.txt" #入力ファイル名を指定してin_f$
> out_f <- "hoge1.txt" #出力ファイル名を指定し$
> param <- 4 #アノテーションファイル$
>
> #入力ファイルの読み込み
> data <- read.table(in_f1, header=TRUE, sep="\t", qu #in_f2で指定した
> keywords <- readLines(in_f2) #オブジェクトdataの
> dim(data)
[1] 11 4
>
> #本番
> obj <- is.element(as.character(data[,param]), keywo #objがTRUEとなる
> out <- data[obj,] #オブジェクトoutの行
> dim(out)
[1] 7 4
>
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F, quote=F #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
>
> |
```



ありがちなミス4

①実行スクリプトをコピーする際、最後の行のところで改行を含まずにR Console画面上でペーストしたため、②最後のコマンドが実行されない(出力ファイルが生成されない)。これも比較的ありがちなパターンです。CTRL + Rを利用するか、コピペ(CTRL + C、CTRL + V)実行する場合はコピペ後に無意識にリターンキーを押すことを心がけるだけでもよいでしょう。

```
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 <- "hogel.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

#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

> 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
```



警告メッセージ

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

```
R Console
> in_f1 <- "annotation.txt"
> in_f2 <- "list.txt"
> out_f <- "hoge2.txt"
> param <- 4
>
> #入力ファイルの読み込み
> data <- read.table(in_f1, header=TRUE, sep="$"
> keywords <- readLines(in_f2) #in_f1$
> dim(data) #オブ$
[1] 11 4
>
> #本番
> obj <- is.element(as.character(data[,param])$
> out <- data[obj,] #obj$
> dim(out) #オブ$
[1] 7 4
>
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F, $
> |
```

```
R Console
> in_f1 <- "annotation.txt"
> in_f2 <- "list.txt"
> out_f <- "hoge2.txt"
> param <- 4
>
> #入力ファイルの読み込み
> data <- read.table(in_f1, header=TRUE, sep="\t", quote="") #in_f1で指$
> keywords <- readLines(in_f2) #in_f2で指定したファイルの読$
警告メッセージ:
In readLines(in_f2) : 'list.txt' で不完全な最終行が見つかりました
> dim(data) #オブジェクト dataの行数と列数$
[1] 11 4
>
> #本番
> obj <- is.element(as.character(data[,param]), keywords) #条件を満たす$
> out <- data[obj,] #objがTRUEとなる行のみ抽出し$
> dim(out) #オブジェクト outの行数と列数$
[1] 7 4
>
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F) #ou$
> |
```

Contents

- 任意のキーワードを含む行を抽出(行列データの取り扱い)
- 色の説明と応用
- ありがちなミスや警告メッセージ
- コードの解説
 - 最初の4行分、read.table関数、quoteオプション(昔話)
 - Tips(上下左右の矢印キーを有効活用)、ヘッダ一行の有無
 - dim関数、nrow関数とncol関数、?関数名、class関数、dim関数マニュアルの続き
 - dim関数マニュアルの続き、dimnames、rownames、colnames、length
 - readLines、Tips(タブ補完)

コードの解説

これまでは、例題のコードをテンプレートとして利用し、①必要な部分のみを変更して目的を達成する手段を解説してきました。

イントロ | 一般 | 任意のキーワードを含む行を抽出(基礎)

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

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

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

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



#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)
#入力ファイル名を指定してin_f2に格納(リストファイル)
#出力ファイル名を指定してout_fに格納
#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
```

コードの解説

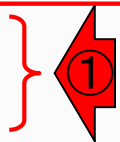
イントロ | 一般 | 任意のキーワードを含む行を抽出(基礎)

例えばタブ区切りテキストファイルが手元があり、この中からリストファイル中の文字列を抽出します。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。

③「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(genelist.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 <- "genelist.txt"
out_f <- "hoge1.txt"
param <- 1
```



#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)
#入力ファイル名を指定してin_f2に格納(リストファイル)
#出力ファイル名を指定してout_fに格納
#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
```

これまでは、例題のコードをテンプレートとして利用し、①必要な部分のみを変更して目的を達成する手段を解説してきました。ここからは、②コードの中身を解説していきます。まずは③例題1でやろうとしていることのおさらい。


全体像

目的: タブ区切りテキストファイル(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	gene1	hoge01	plasma_mem	nuclear
2	gene2	hoge02	hohinu	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

出力: hoge1.txt



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

入力: リストファイル(genelist1.txt)

	A
1	gene1
2	gene7
3	gene9

①作業ディレクトリ中に、②に相当する、③2つのファイルが存在することを確認しておいてください。

コードの解説の前に

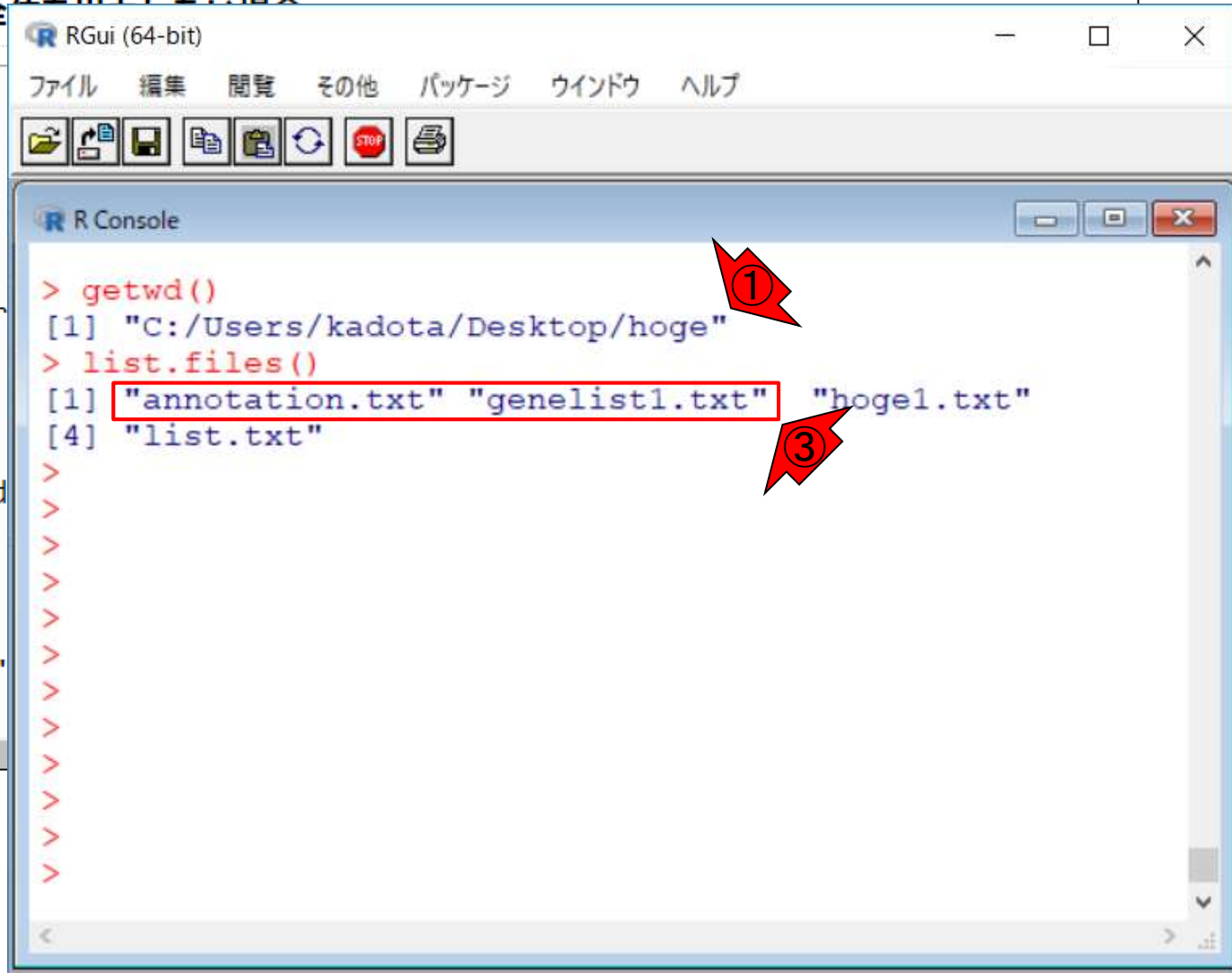
1. 目的のタブ区切りテキストファイル(annotation.txt)中の第1列目をキーとして、リストファイル(genelist1.txt)中のものが含まれる行全体を抽出する

```
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, as.is=TRUE)
keywords <- readLines(in_f2)
dim(data)

#本番
obj <- is.element(as.character(data[,1]), keywords)
out <- data[obj,]
dim(out)

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



```
RGui (64-bit)
ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ

R Console
> getwd()
[1] "C:/Users/kadota/Desktop/hoge"
> list.files()
[1] "annotation.txt" "genelist1.txt" "hoge1.txt"
[4] "list.txt"
>
>
>
>
>
>
>
>
>
>
```

コードの解説

①の赤枠内では、必要に応じて変更する入出力ファイル名や必要なパラメータ(ここでは1列目の情報をターゲットにするという数値情報に相当)の情報を格納する作業を行っています。

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

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



```
#入力ファイル名を指定してin_f1に格納(アノテーションファイル)
#入力ファイル名を指定してin_f2に格納(リストファイル)
#出力ファイル名を指定してout_fに格納
#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
```


コードの解説

1. 目的のタブ区切りテキストファイル(annota
(genelist1.txt)中のもが含まれる行全体を出

```
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)  
dim(data)
```

#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,]  
dim(out)
```

#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
```

①の赤枠内では、必要に応じて変更する入出力ファイル名や必要なパラメータ(ここでは1列目の情報をターゲットにするという数値情報に相当)の情報を格納する作業を行っています。文字列として認識させたいものは、②のように二重クォーテーションで囲います。③は数字として認識させたい列番号情報なので、二重クォーテーションではありません。

#入力ファイル名を指定してin_f2に格納(リストファイル)
#出力ファイル名を指定してout_fに格納
#アノテーションファイル中の検索したい列番号を指定

#in_f1で指定したファイルの読み込み
#in_f2で指定したファイルの読み込み
#オブジェクトdataの行数と列数を表示

#条件を満たすかどうかを判定した結果をobjに格納
#objがTRUEとなる行のみ抽出した結果をoutに格納
#オブジェクトoutの行数と列数を表示

①の赤枠内を再度コピー実行。多少見栄えが異なっても構いません。重要なのは②のようにコピー実行することのみ。

コードの解説

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.txt)中のものが含まれる行全体を抽出する

```
in_f1 <- "annotation.txt"
in_f2 <- "genelist.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, as.is=TRUE)
keywords <- readLines(in_f2)
dim(data)

#本番
obj <- is.element(as.character(data[,1]), keywords)
out <- data[obj,]
dim(out)

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



```
RGui (64-bit)
ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ

R Console
> getwd()
[1] "C:/Users/kadota/Desktop/hoge"
> list.files()
[1] "annotation.txt" "genelist.txt" "hoge1.txt"
[4] "list.txt"
> in_f1 <- "annotation.txt" #入力ファイル$
> in_f2 <- "genelist.txt" #入力ファイル$
> out_f <- "hoge1.txt" #出力ファイル$
> param <- 1 #アノテーション$
```

in_f2の中身

①の赤枠内を再度コピー実行。多少見栄えが異なっても構いません。重要なのは②のようにコピー実行することのみ。
③in_f2と打ってリターン。④でgenelist1.txtという文字列を、in_f2というオブジェクト名で取り扱えることがわかります。

1. 目的のタブ区切りテキストファイル(annota (genelist1.txt)中のものが含まれる行全体を

```
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, as.is=TRUE)  
keywords <- readLines(in_f2)  
dim(data)
```

#本番

```
obj <- is.element(as.character(data[,1]), keywords)  
out <- data[obj,]  
dim(out)
```

#ファイルに保存

```
write.table(out, out_f, sep="\t", as.is=TRUE)
```



```
> getwd()  
[1] "C:/Users/kadota/Desktop/hoge"  
> list.files()  
[1] "annotation.txt" "genelist1.txt" "hoge1.txt"  
[4] "list.txt"  
> in_f1 <- "annotation.txt" #入力ファイル$  
> in_f2 <- "genelist1.txt" #入力ファイル$  
> out_f <- "hoge1.txt" #出力ファイル$  
> param <- 1 #アノテーション$  
> in_f2  
[1] "genelist1.txt"  
> |
```

paramの中身

①の赤枠内を再度コピー実行。多少見栄えが異なっても構いません。重要なのは②のようにコピー実行することのみ。③in_f2と打ってリターン。④でgenelist1.txtという文字列を、in_f2というオブジェクト名で取り扱えることがわかります。⑤paramの中身も妥当ですね。

1. 目的のタブ区切りテキストファイル(annota (genelist1.txt)中のもが含まれる行全

```
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, as.is=TRUE)
keywords <- readLines(in_f2)
dim(data)
```

#本番

```
obj <- is.element(as.character(data[,1]), keywords)
out <- data[obj,]
dim(out)
```

#ファイルに保存

```
write.table(out, out_f, sep="\t", as.is=TRUE)
```



ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ

R Console

```
> getwd()
[1] "C:/Users/kadota/Desktop/hoge"
> list.files()
[1] "annotation.txt" "genelist1.txt" "hoge1.txt"
[4] "list.txt"
> in_f1 <- "annotation.txt"
> in_f2 <- "genelist1.txt"
> out_f <- "hoge1.txt"
> param <- 1
> in_f2
[1] "genelist1.txt"
> param
[1] 1
> |
```

①のあたりに見えている[1]という数字が気になって仕方がないヒトが一定数いると思われませんが、これはベクトルの1番目の要素という意味です。

[1]って何？

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.txt)中のものが含まれる行全体を抽出したい場合

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

#入力ファイルの読み込み

```
data <- read.table(in_f1, header=TRUE)
keywords <- readLines(in_f2)
dim(data)
```

#本番

```
obj <- is.element(as.character(data[,1]), keywords)
out <- data[obj,]
dim(out)
```

#ファイルに保存

```
write.table(out, out_f, sep="\t")
```

```
RGui (64-bit)
ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ

R Console
> getwd()
[1] "C:/Users/kadota/Desktop/hoge"
> list.files()
[1] "annotation.txt" "genelist.txt" "hoge1.txt"
[4] "list.txt"
> in_f1 <- "annotation.txt" #入力ファイル$
> in_f2 <- "genelist.txt" #入力ファイル$
> out_f <- "hoge1.txt" #出力ファイル$
> param <- 1 #アノテーション$
> in_f2
[1] "genelist.txt"
> param
[1] 1
|
```

①のあたりに見えている[1]という数字が気になって仕方がないヒトが一定数いると思われませんが、これはベクトルの1番目の要素という意味です。②ここに[4]という数字も見えていますが、これは③list.files()実行結果のうちの4番目の要素という意味です。

[4]って何？

1. 目的のタブ区切りテキストファイル(annota
(genelist1.txt)中のものが含まれる行全

```
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")
keywords <- readLines(in_f2)
dim(data)
```

#本番

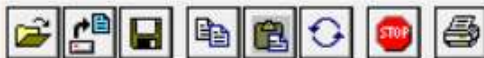
```
obj <- is.element(as.character(data[,1]), keywords)
out <- data[obj,]
dim(out)
```

#ファイルに保存

```
write.table(out, out_f, sep="\t")
```

R

ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ



R Console

```
> getwd()
[1] "C:/Users/kadota/Desktop/hoge"
> list.files()
[1] "annotation.txt" "genelist1.txt" "hoge1.txt"
[4] "list.txt"
> in_f1 <- "annotation.txt" #入力ファイル$
> in_f2 <- "genelist1.txt" #入力ファイル$
> out_f <- "hoge1.txt" #出力ファイル$
> param <- 1 #アノテーション$
> in_f1
[1] "genelist1.txt"
> param
[1] 1
> |
```

特定要素抽出

①のあたりに見えている[1]という数字が気になって仕方がないヒトが一定数いると思われませんが、これはベクトルの1番目の要素という意味です。②ここに[4]という数字も見えていますが、これは③list.files()実行結果のうちの4番目の要素という意味です。私はlist.files()と組み合わせて使うことはありませんが、実用上は、④のような感じで特定の要素を抽出したい場合に頻用するテクニックです。しかるべき箇所でまた説明します。

1. 目的のタブ区切りテキストファイル(annota
(genelist1.txt)中のものが含まれる行全

```
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, as.is=TRUE)
keywords <- readLines(in_f2)
dim(data)
```

#本番

```
obj <- is.element(as.character(data[,1]), keywords)
out <- data[obj,]
dim(out)
```

#ファイルに保存

```
write.table(out, out_f, sep="\t")
```

R RG

ファイル

ファイル

R Console

```
> getwd()
[1] "C:/Users/kadota/Desktop/hoge"
> list.files()
[1] "annotation.txt" "genelist1.txt" "hoge1.txt"
[4] "list.txt"
> in_f1 <- "annotation.txt" #入力ファイル$
> in_f2 <- "genelist1.txt" #入力ファイル$
> out_f <- "hoge1.txt" #出力ファイル$
> param <- 1 #アノテーション$
> in_f1
[1] "genelist1.txt"
> param
[1] 1
> list.files()[4]
[1] "list.txt"
> |
```

Contents

- 任意のキーワードを含む行を抽出(行列データの取り扱い)
- 色の説明と応用
- ありがちなミスや警告メッセージ
- コードの解説
 - 最初の4行分、read.table関数、quoteオプション(昔話)
 - Tips(上下左右の矢印キーを有効活用)、ヘッダ一行の有無
 - dim関数、nrow関数とncol関数、?関数名、class関数、dim関数マニュアルの続き
 - dim関数マニュアルの続き、dimnames、rownames、colnames、length
 - readLines、Tips(タブ補完)

read.table関数

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

```

read.table関数

1. 目的のタブ区切りテキストファイル(annota
(genelist1.txt)中のものが含まれる行全体を出

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

②

入力ファイルの読み込み部分の説明です。赤枠は、①in_f1で与えている②annotation.txtを、③read.tableという名前の関数を用いて読み込んだ結果を、④dataという名前のオブジェクトに格納せよという命令です。「オブジェクト」というのは、単純にR上で取り扱うための「もの」とか「物体」という風に解釈すればよいです。例えば、dataというものに格納している、と解釈すればよいです。

#出力ファイル名を指定してout_fに格納

#アノテーションファイル中の検索したい列番号を指定

④

①

```
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)
```

```
dim(data)
```

#in_f2で指定したファイルの読み込み

#オブ

#本番

```
obj <- is.element(as.character(data[,param]), keywords)
```

```
out <- data[obj,]
```

```
dim(out)
```

#objが

#オブ

#ファイルに保存

```
write.table(out, out_f, sep="\t", append=F, col.names=FALSE)
```

	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

read.table関数

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.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 <- "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               #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)
```

#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, col.names=FALSE)
```

	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

①の部分は、②read.table関数のオプションです。③in_f1で与えている、④annotation.txtの中身は、⑤のような感じです。

read.table関数

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.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 <- "genelist.txt"  
out_f <- "hoge1.txt"  
param <- 1
```

④

#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)
#入力ファイル名を指定してin_f2に格納(リストファイル)
#出力ファイル名を指定してout_fに格納
#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)
```

```
dim(data)
```

②

①

⑤

#in_f2で指定したファイルの読み込み

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

#objが
#オブ

#ファイルに保存

```
write.table(out, out_f, sep="\t", append=F, col.names=FALSE)
```

	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

header

1. 目的のタブ区切りテキストファイル(annota
(genelist1.txt)中のものが含まれる行全体を出

```
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)
```

#本番

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

#ファイルに保存

```
write.table(out, out_f, sep="\t", append=F, c
```

①このファイルの最初の1行目の情報は、各列にどのような情報が含まれているかを表す「列名」、つまり「ヘッダー行」に相当するもの。②headerオプションは、入力ファイル中にヘッダー行が含まれるか否かをTRUE or FALSEで指定する。このオプションのデフォルトはFALSEなので、明示的に②「header=TRUE」と与えている。尚、「header = TRUE」でも「header = T」でも「header=T」でもよい。

#出力ファイル名を指定してout_fに格納

#アノテーションファイル中の検索したい列番号を指定

①

	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

タブ区切りファイル

①このファイルは、タブ区切りテキストファイルです。タブ区切りの場合は、②sepオプション(←separator characterの略)のところで、`¥t`と指定する(「逆スラッシュ」ですが、文字コードの関係上表示できません)。

1. 目的のタブ区切りテキストファイル(`annotation.txt`)(`genelist1.txt`)中のものが含まれる行全体を出力したい場合:

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

#入力ファイル名を指定してin_f1に格納(アノテーションファイル)
#入力ファイル名を指定してin_f2に格納(リストファイル)
#出力ファイル名を指定してout_fに格納
#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) #オブジェクトの次元
```

②

#本番

```
obj <- is.element(as.character(data[,param]), keywords) #objがTRUEの行番号を抽出
out <- data[obj,] #オブジェクトを抽出
dim(out)
```

①

#ファイルに保存

```
write.table(out, out_f, sep="\t", append=F, as.is=T)
```

	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

Contents

- 任意のキーワードを含む行を抽出(行列データの取り扱い)
- 色の説明と応用
- ありがちなミスや警告メッセージ
- コードの解説
 - 最初の4行分、read.table関数、quoteオプション(昔話)
 - Tips(上下左右の矢印キーを有効活用)、ヘッダ一行の有無
 - dim関数、nrow関数とncol関数、?関数名、class関数、dim関数マニュアルの続き
 - dim関数マニュアルの続き、dimnames、rownames、colnames、length
 - readLines、Tips(タブ補完)

quoteオプション

①quoteオプションは、マニアックすぎるのでこれまで説明は省略してきました。しかし、毎年必ず数名から質問を受ける事柄なので参考資料として提示しておきます。まず、②のような中身の場合はquoteオプションはなくてもよいです。あっても悪さをすることがないのでつけています。

1. 目的のタブ区切りテキストファイル(annota (genelist1.txt)中のもが含まれる行全体を出

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

#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)
#入力ファイル名を指定してin_f2に格納(リストファイル)
#出力ファイル名を指定してout_fに格納
#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)
dim(data)
```

#in_f2で指定したファイルの読み込み
#オブ

#本番

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

#objが
#オブ

#ファイルに保存

```
write.table(out, out_f, sep="\t", append=F, c
```

	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

quoteオプション

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
```

quoteオプション

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抽出したい場合

```
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$)
keywords <- readLines(in_f2)
dim(data)
```

#本番

```
obj <- is.element(as.character(data[,1]), keywords)
out <- data[obj,]
dim(out)
```

#ファイルに保存

```
write.table(out, out_f, sep="\t")
```

```
> data <- read.table(in_f1, header=TRUE, sep="\t", quote$)
> data
  gene name accession description subcellular_location
1    gene1   hoge01 plasma_mem          nuclear
2    gene2   hoge02      hohinu          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   agen1          membrane
11   gene11   hoge11   iyaaaa    endoplasmic
> |
```

quoteオプション

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抽出したい場合

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

②

#入力ファイルの読み込み

```
data <- read.table(in_f1, header=TRUE, sep="\t")
dim(data)
```

#本番

```
obj <- is.element(as.character(data[,1]), in_f2)
out <- data[obj,]
dim(out)
```

#ファイルに保存

```
write.table(out, out_f, sep="\t")
```

```
> data <- read.table(in_f1, header=TRUE, sep="\t")
> data
  geneaccessiondescriptionsubcellular_location
1  gene1      hoge01  plasma_mem             nuclear
2  gene2      hoge02    hohinu             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    agen1             membrane
11 gene11     hoge11    iyaaaa             endoplasmic
> |
```

①

ものすごく昔の話で記憶が曖昧ですが、かつて「3' PDE」みたいな感じで「アポストロフィ(')」または「プライム(')」みたいなものがついた遺伝子名を含むファイルをうまく読み込めず苦悩しました。

quoteオプション

1. 目的のタブ区切りテキストファイル(annota
(genelist1.txt)中のものが含まれる行全体を出力したい場合:

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

#入力ファイル名を指定してin_f1に格納(アノテーションファイル)
#入力ファイル名を指定してin_f2に格納(リストファイル)
#出力ファイル名を指定してout_fに格納
#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) #オブジェクトの次元
```

#本番

```
obj <- is.element(as.character(data[,param]), keywords) #objがTRUEの行番号
out <- data[obj,] #オブジェクトの抽出
dim(out)
```

#ファイルに保存

```
write.table(out, out_f, sep="\t", append=F, col.names=TRUE)
```

	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

ものすごく昔の話で記憶が曖昧ですが、かつて「3' PDE」みたいな感じで「アポストロフィ(')」または「プライム(')」みたいなものがついた遺伝子名を含むファイルをうまく読み込めず苦悩しました。例えば①をエディタで開いて、②gene5の部分を3' PDEみたいな感じにして保存し、③quoteオプションの有無による挙動の違いを実感してみるとよいと思います。

quoteオプション

1. 目的のタブ区切りテキストファイル(annota (genelist1.txt)中のものが含まれる行全体を出

```

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)

#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, c

```



#入力ファイル名を指定してin_f2に格納(リストファイル)
#出力ファイル名を指定してout_fに格納
#アノテーションファイル中の検索したい列番号を指定

#in_f1で指定したファイルの読み込み
#in_f2で指定したファイルの読み込み
#オブ

	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	3' PDE	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

gene5の部分を3' PDEにして保存した状態で、①quoteオプションなしの実行結果。②警告メッセージが出ていることがわかります。

quoteなし

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

```
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")
keywords <- readLines(in_f2)
dim(data)

#本番
obj <- is.element(as.character(data[,1]), keywords)
out <- data[obj,]
dim(out)

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

RGui (64-bit)

R Console

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

> data <- read.table(in_f1, header=TRUE, sep="\t")

警告メッセージ:

1: scan(file = file, what = what, sep = sep, quote = q\$
クォートで囲まれた文字列中にEOFがあります

2: scan(file = file, what = what, sep = sep, quote = q\$
読み込まれた項目数が、列数の倍数ではありませんでした

gene5の部分を3' PDEにして保存した状態で、①quoteオプションなしの実行結果。②警告メッセージが出ていることがわかります。③dataオブジェクトの中身を表示。変になっていることがわかります。

quoteなし

1. 目的のタブ区切りテキストファイル(annota (genelist1.txt)中のものが含まれる行全体を抽出

```
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, head = FALSE, as.is = TRUE)
keywords <- readLines(in_f2)
dim(data)

#本番
obj <- is.element(as.character(data[,1]), keywords)
out <- data[obj,]
dim(out)

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



RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ

R Console

読み込まれた項目数が、列数の倍数ではありませんでした

```
> data
1
2
3
4
5 3PDE\thoge05\tkamo\tmembrane\ngene6\thoge06\tnetteba\
accession description subcellular_location
1 hoge01 plasma_mem nuclear
2 hoge02 hohinu membrane
3 hoge03 agribio endoplasmic
4 hoge04 genesis endoplasmic
5
> |
```

gene5の部分を3' PDEにして保存した状態で、①quoteオプションありの実行結果。②警告メッセージが出ていないことがわかります。

quoteあり

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.txt)中のものが含まれる行全体を抽出する

```
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", as.is=TRUE)
keywords <- readLines(in_f2)
dim(data)
```

#本番

```
obj <- is.element(as.character(data[,1]), keywords)
out <- data[obj,]
dim(out)
```

#ファイルに保存

```
write.table(out, out_f, sep="\t", as.is=TRUE)
```

```
RGui (64-bit)
ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ

R Console
> data
1
2
3
4
5 3PDE\thoge05\tkamo\tmembrane\ngene6\thoge06\tnetteba\
accession description subcellular_location
1 hoge01 plasma_mem nuclear
2 hoge02 hohinu membrane
3 hoge03 agribio endoplasmic
4 hoge04 genesis endoplasmic
5
> data <- read.table(in_f1, header=TRUE, sep="\t", quot$
> |
```


gene5の部分を3' PDEにして保存した状態で、①quoteオプションありの実行結果。②警告メッセージが出ていないことがわかります。③dataオブジェクトの中身を表示。

quoteあり

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.txt)中のものが含まれる行全体を抽出する

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

#入力ファイルの読み込み

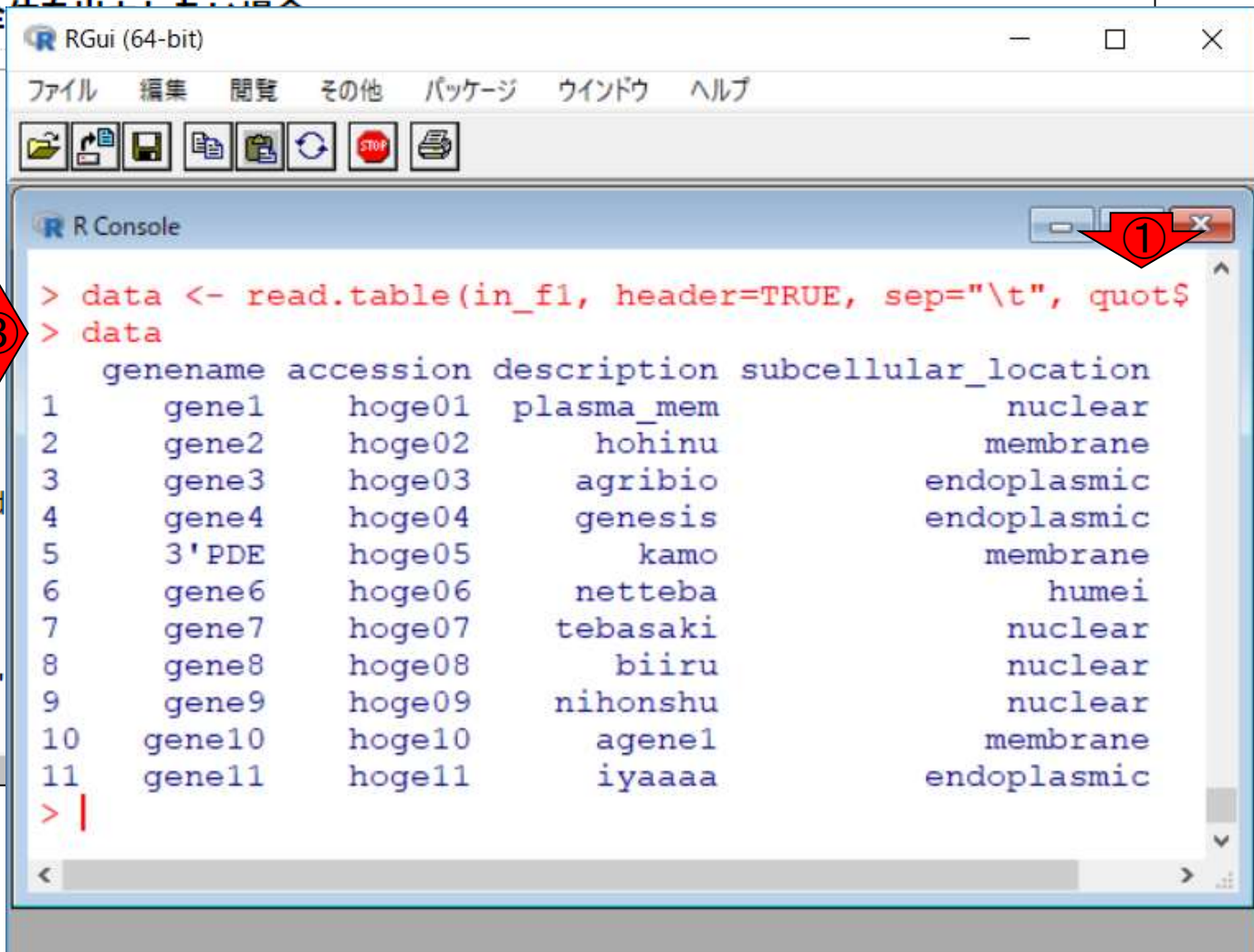
```
data <- read.table(in_f1, header=TRUE, sep="\t", quote="$")
keywords <- readLines(in_f2)
dim(data)
```

#本番

```
obj <- is.element(as.character(data[,1]), keywords)
out <- data[obj,]
dim(out)
```

#ファイルに保存

```
write.table(out, out_f, sep="\t")
```



```
> data <- read.table(in_f1, header=TRUE, sep="\t", quote="$")
> data
```

	gene name	accession	description	subcellular_location
1	gene1	hoge01	plasma_mem	nuclear
2	gene2	hoge02	hohinu	membrane
3	gene3	hoge03	agribio	endoplasmic
4	gene4	hoge04	genesis	endoplasmic
5	3' PDE	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	agen1	membrane
11	gene11	hoge11	iyaaaa	endoplasmic

gene5の部分を3' PDEにして保存した状態で、①quoteオプションありの実行結果。②警告メッセージが出ていないことがわかります。③dataオブジェクトの中身を表示。④正しく読み込めていることがわかります。このようにquoteオプションをつけておくほうが無難なので、read.table関数を利用する際はおまじない的につけています。

quoteあり

1. 目的のタブ区切りテキストファイル(annota (genelist1.txt)中のものが含まれる行全

```
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, head
keywords <- readLines(in_f2)
dim(data)
```

#本番

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

#ファイルに保存

```
write.table(out, out_f, sep="\t"
```

```
> data <- read.table(in_f1, header=TRUE, sep="\t", quot$
> data
```

	gene name	accession	description	subcellular_location
1	gene1	hoge01	plasma_mem	nuclear
2	gene2	hoge02	hohinu	membrane
3	gene3	hoge03	agribio	endoplasmic
4	gene4	hoge04	genesis	endoplasmic
5	3' PDE	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	agen1	membrane
11	gene11	hoge11	iyaaaa	endoplasmic

Contents

- 任意のキーワードを含む行を抽出(行列データの取り扱い)
- 色の説明と応用
- ありがちなミスや警告メッセージ
- コードの解説
 - 最初の4行分、read.table関数、quoteオプション(昔話)
 - Tips(上下左右の矢印キーを有効活用)、ヘッダ一行の有無
 - dim関数、nrow関数とncol関数、?関数名、class関数、dim関数マニュアルの続き
 - dim関数マニュアルの続き、dimnames、rownames、colnames、length
 - readLines、Tips(タブ補完)

①入力ファイルの中身が、②のような状態で、③をコピー実行。

read.table実行

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.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 <- "genelist.txt"
out_f <- "hoge1.txt"
param <- 1
```

①

#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)
#入力ファイル名を指定してin_f2に格納(リストファイル)
#出力ファイル名を指定してout_fに格納
#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="")
```

③

f1で指定したファイルの読み込み

```
keywords <- readLines(in_f2)
dim(data)
```

#in_f2で指定したファイルの読み込み
#オブジェクト

②

#本番

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

#objがTRUEの行
#オブジェクト

#ファイルに保存

```
write.table(out, out_f, sep="\t", append=F, col.names=FALSE)
```

	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

- ①入力ファイルの中身が、②のような状態で、③をコピー実行。
④dataの中身を表示。

read.table実行

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.txt)中のものが含まれる行全体を抽出する場合

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

#入力ファイルの読み込み

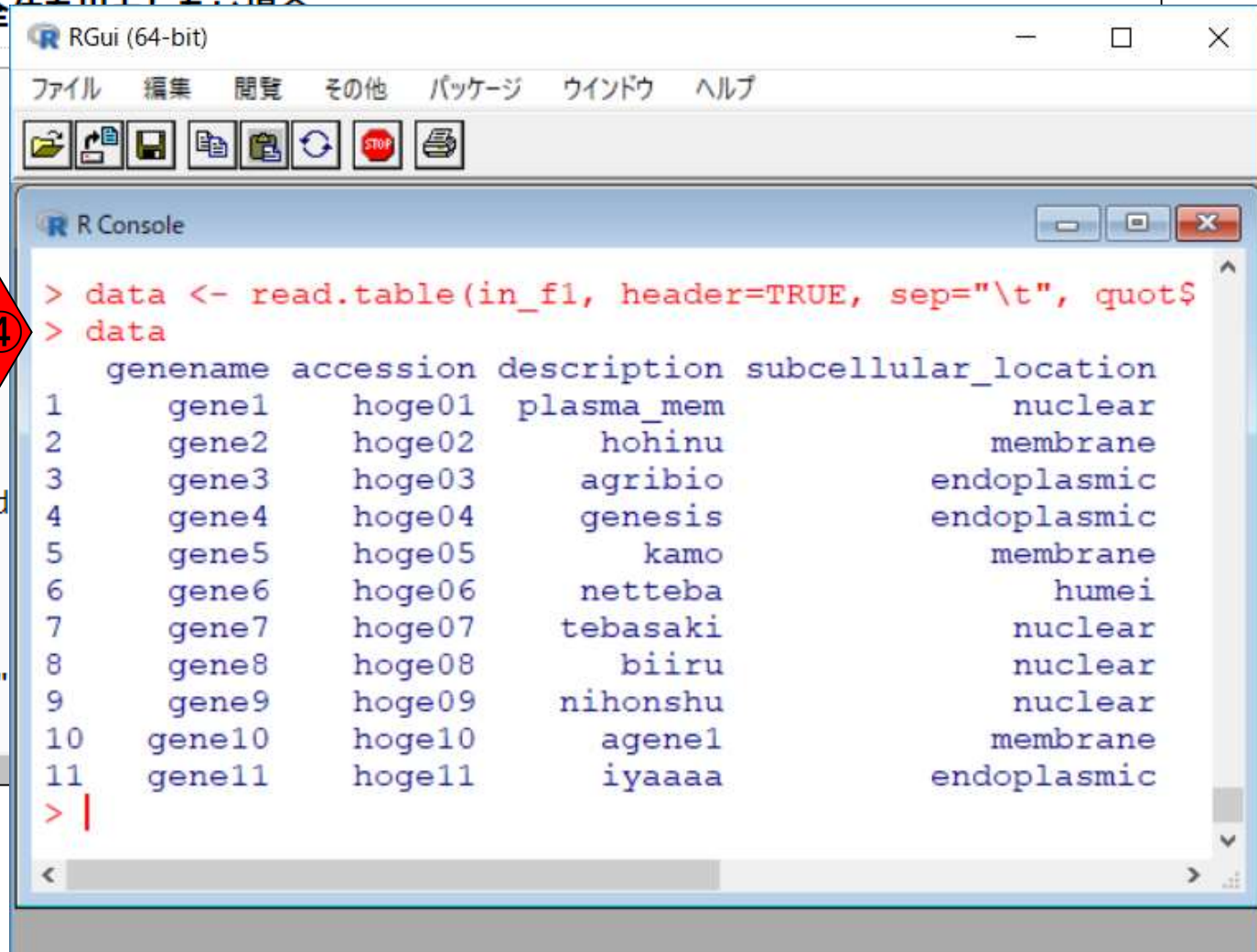
```
data <- read.table(in_f1, head  
keywords <- readLines(in_f2)  
dim(data)
```

#本番

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

#ファイルに保存

```
write.table(out, out_f, sep="\t"
```



```
> data <- read.table(in_f1, header=TRUE, sep="\t", quot$  
> data  
  gene  accession  description  subcellular_location  
1  gene1    hoge01    plasma_mem          nuclear  
2  gene2    hoge02      hohinu            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    agen1             membrane  
11 gene11    hoge11    iyaaaa            endoplasmic  
> |
```

read.table実行

①入力ファイルの中身が、②のような状態で、③をコピー実行。
④dataの中身を表示。⑤header=TRUEとしているので、⑥入力ファイルの1行目が列名のような位置づけになっていることがわかる。

1. 目的のタブ区切りテキストファイル(annota
(genelist1.txt)中のものが含まれる行全

```
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

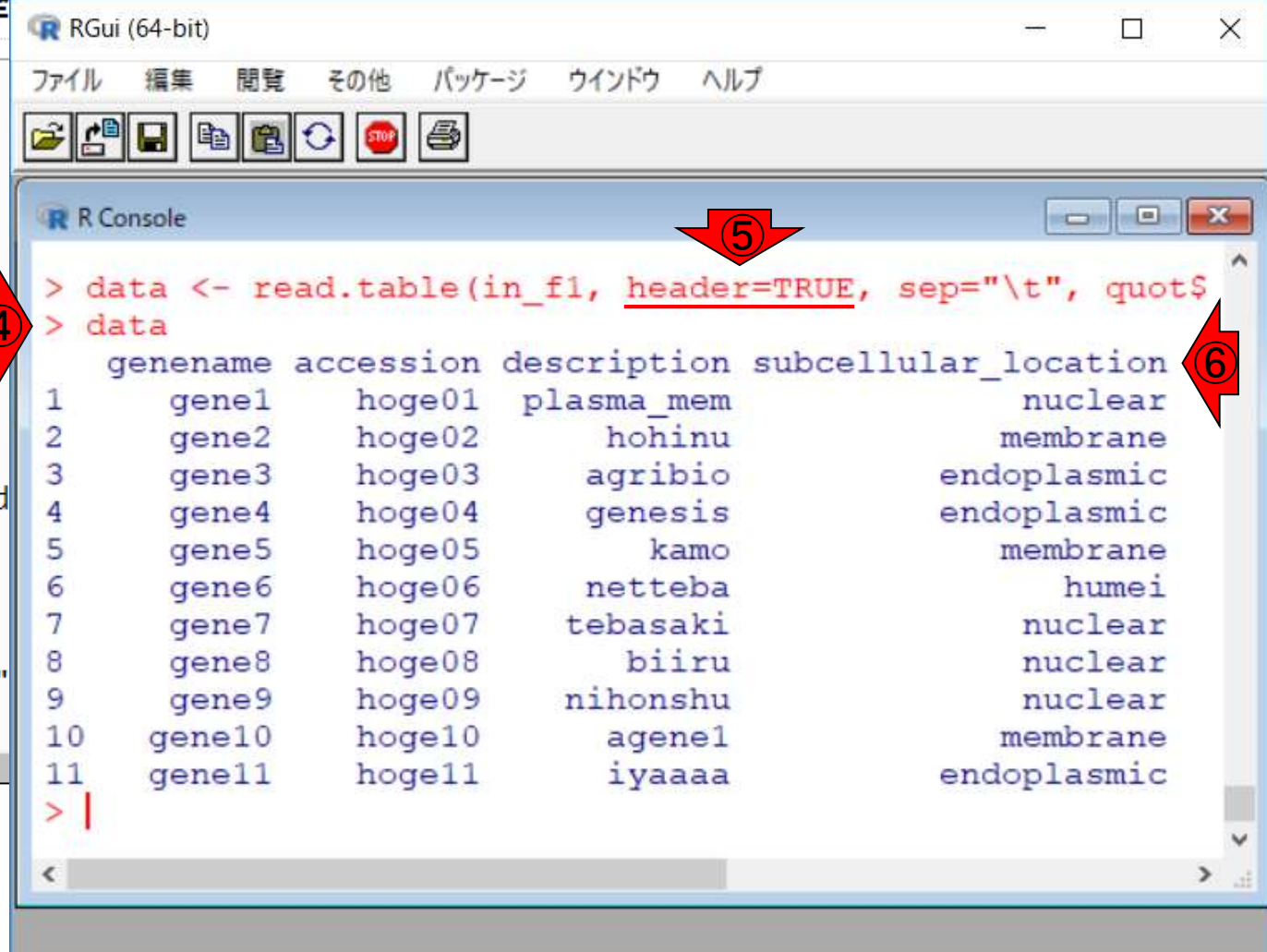
```
data <- read.table(in_f1, head  
keywords <- readLines(in_f2)  
dim(data)
```

#本番

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

#ファイルに保存

```
write.table(out, out_f, sep="\t"
```



```
> data <- read.table(in_f1, header=TRUE, sep="\t", quot$  
> data  
  gene  accession  description  subcellular_location  
1  gene1    hoge01    plasma_mem          nuclear  
2  gene2    hoge02      hohinu            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    agen1              membrane  
11 gene11    hoge11    iyaaaa             endoplasmic  
> |
```

read.table実行

①入力ファイルの中身が、②のような状態で、③をコピー実行。
④dataの中身を表示。⑤header=TRUEとしているので、⑥入力ファイルの1行目が列名のような位置づけになっていることがわかる。⑦入力ファイルは本来12行からなるが、ヘッダ行を除いているので11行になっていると解釈すればよい。

1. 目的のタブ区切りテキストファイル(annota (genelist1.txt)中のもが含まれる行全

```
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, head
keywords <- readLines(in_f2)
dim(data)
```

#本番

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

#ファイルに保存

```
write.table(out, out_f, sep="\t"
```

The screenshot shows the R Console window with the following code and output:

```
> data <- read.table(in_f1, header=TRUE, sep="\t", quot$
> data
```

	gene	accession	description	subcellular_location
1	gene1	hoge01	plasma_mem	nuclear
2	gene2	hoge02	hohinu	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	agen1	membrane
11	gene11	hoge11	iyaaaa	endoplasmic

Tips: 矢印キー

①header=TRUEの記述をなくす(あるいはheader=FALSEにする)とどのような結果になるかを調べる。と同時に②上矢印キーを押して以前打ち込んだコマンドを再利用するテクニックを思い出してもらう。②上矢印キーを2回押すと、③2つ前に打ち込んだコマンドが表示されるはずです。

1. 目的のタブ区切りテキストファイル(annota
(genelist1.txt)中のもが含まれる行全

```
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, head  
keywords <- readLines(in_f2)  
dim(data)
```

#本番

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

#ファイルに保存

```
write.table(out, out_f, sep="\t"
```

The screenshot shows the R Studio interface. The R Console window displays the following output:

```
> data <- read.table(in_f1, header=TRUE, sep="\t", quot$  
> data  
  gene name accession description subcellular_location  
1     gene1     hoge01  plasma_mem             nuclear  
2     gene2  
3     gene3  
4     gene4  
5     gene5  
6     gene6  
7     gene7  
8     gene8  
9     gene9  
10    gene10  
11    gene11  
> |
```

A red arrow labeled '1' points to the header=TRUE text in the console. A red arrow labeled '2' points to the Up Arrow key on a keyboard. A red arrow labeled '3' points to the Up Arrow key in the R Console window.

Tips: 矢印キー

①header=TRUEの記述をなくす(あるいはheader=FALSEにする)とどのような結果になるかを調べる。と同時に②上矢印キーを押して以前打ち込んだコマンドを再利用するテクニックを思い出してもらおう。②上矢印キーを2回押すと、③2つ前に打ち込んだコマンドが表示されるはずですが、こんな感じになると思うので、左矢印キーを押して行ってheader=TRUEのところをheader=FALSEにしてみましょう。

1. 目的のタブ区切りテキストファイル(annota
(genelist1.txt)中のものが含まれる行全

```
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,  
keywords <- readLines(in_f2)  
dim(data)
```

#本番

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

#ファイルに保存

```
write.table(out, out_f, sep="\t"
```



```
R Console  
$sep="\t", quote="") #in_f1で指定したファイルの読み込み  
  
$ular_location  
$ nuclear  
$ membrane  
$ endoplasmic  
$ endoplasmic  
$ membrane  
$ humei  
$ nuclear  
$ nuclear  
$ nuclear  
$ membrane  
$ endoplasmic  
$sep="\t", quote="") #in_f1で指定したファイルの読み込み
```

Tips: 矢印キー

1. 目的のタブ区切りテキストファイル(annota
(genelist1.txt)中のもが含まれる行全

```
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  
keywords <- readLines(in_f2)  
dim(data)
```

#本番

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

#ファイルに保存

```
write.table(out, out_f, sep="\t"
```

①header=TRUEの記述をなくす(あるいはheader=FALSEにする)とどのような結果になるかを調べる。と同時に②上矢印キーを押して以前打ち込んだコマンドを再利用するテクニックを思い出してもらおう。②上矢印キーを2回押すと、③2つ前に打ち込んだコマンドが表示されるはず。こんな感じになると思うので、左矢印キーを押して行ってheader=TRUEのところをheader=FALSEにしてみましょう。つまり④こういう状態にしましょう、ということです。リターンキーを押して、もう一度dataの中身を表示。

```
> data <- read.table(in_f1, header=TRUE, sep="\t", quo$  
> data  
  geneName accession description subcellular_location  
1     gene1     hoge01  plasma_mem             nuclear  
2     gene2     hoge02    hohinu             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   agen1              membrane  
11    gene11     hoge11   iyaaaa             endoplasmic  
> data <- read.table(in_f1, header=FALSE, sep="\t", quo$
```



header=FALSE

①dataの中身を表示した結果。②ヘッダ一行部分も普通のデータと同じ扱いになっているので、これが③1行目になっていることがわかる。それゆえ、④最終行が12行目となる。こんな感じで自分のデータのヘッダ行の有無に応じて、オプションをうまく変更して読み込みましょう。私はこのデータはheader=TRUEとするのが正解だと思います。

1. 目的のタブ区切りテキストファイル(annota
(genelist1.txt)中のものが含まれる行全

```
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

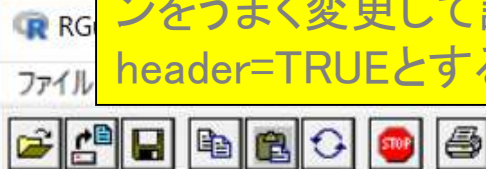
```
data <- read.table(in_f1, head  
keywords <- readLines(in_f2)  
dim(data)
```

#本番

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

#ファイルに保存

```
write.table(out, out_f, sep="\t"
```



R Console

```
> data
```

	V1	V2	V3	V4
1	genename	accession	description	subcellular_location
2	gene1	hoge01	plasma_mem	nuclear
3	gene2	hoge02	hohinu	membrane
4	gene3	hoge03	agribio	endoplasmic
5	gene4	hoge04	genesis	endoplasmic
6	gene5	hoge05	kamo	membrane
7	gene6	hoge06	netteba	humei
8	gene7	hoge07	tebasaki	nuclear
9	gene8	hoge08	biiru	nuclear
10	gene9	hoge09	nihonshu	nuclear
11	gene10	hoge10	agen1	membrane
12	gene11	hoge11	iyaaaa	endoplasmic



Contents

- 任意のキーワードを含む行を抽出(行列データの取り扱い)
- 色の説明と応用
- ありがちなミスや警告メッセージ
- コードの解説
 - 最初の4行分、read.table関数、quoteオプション(昔話)
 - Tips(上下左右の矢印キーを有効活用)、ヘッダ一行の有無
 - dim関数、nrow関数とncol関数、?関数名、class関数、dim関数マニュアルの続き
 - dim関数マニュアルの続き、dimnames、rownames、colnames、length
 - readLines、Tips(タブ補完)

dim関数

①このdataオブジェクトはheader=FALSEとして読み込んだものではあるが、とりあえず気にせずこのdataオブジェクトに対してdim関数を実行することを意味する②dim(data)を実行。

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.txt)中のものが含まれる行全体を抽出する場合

```
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=FALSE, as.is=TRUE)
keywords <- readLines(in_f2)
dim(data)
```

#本番

```
obj <- is.element(as.character(keywords), data[,1])
out <- data[obj,]
dim(out)
```

#ファイルに保存

```
write.table(out, out_f, sep="\t")
```

```
RGui (64-bit)
ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ

R Console
> data
      V1      V2      V3      V4
1  gene1  hoge01  plasma_mem  nuclear
2  gene2  hoge02  hohinu      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  agen1      membrane
11 gene11  hoge11  iyaaaa    endoplasmic
12 gene11  hoge11  iyaaaa    endoplasmic
> dim(data)
```

dim関数

①このdataオブジェクトはheader=FALSEとして読み込んだものではあるが、とりあえず気にせずこのdataオブジェクトに対してdim関数を実行することを意味する②dim(data)を実行。実行結果。③このdataオブジェクトは12行×4列からなるデータだということを意味します。

1. 目的のタブ区切りテキストファイル(annota (genelist1.txt)中のものが含まれる行全

```
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

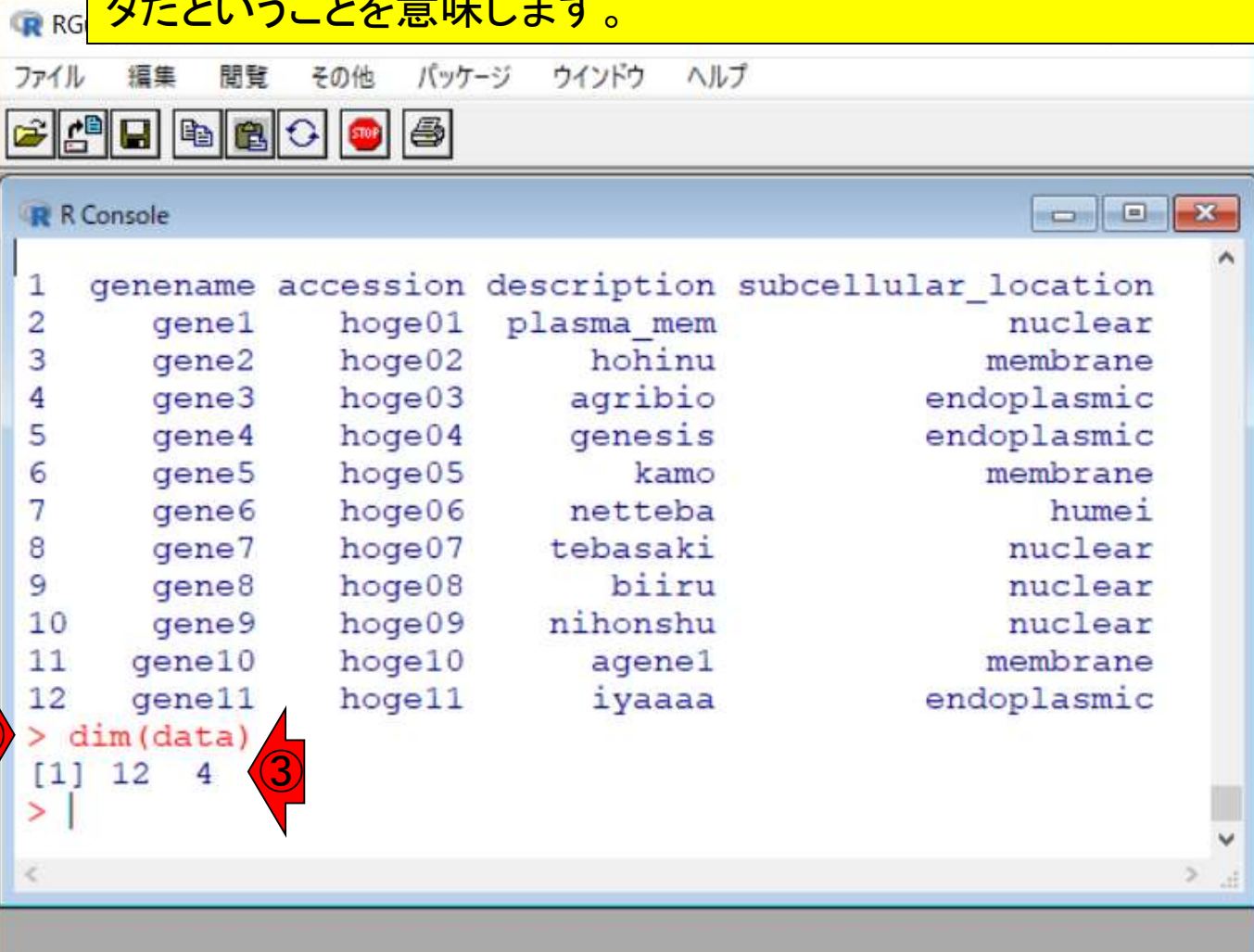
```
data <- read.table(in_f1, header=
keywords <- readLines(in_f2)
dim(data)
```

#本番

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

#ファイルに保存

```
write.table(out, out_f, sep="\t"
```



The screenshot shows the R Console window with the following content:

```
R Console
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
> dim(data)
[1] 12  4
> |
```

dim関数

①このdataオブジェクトはheader=FALSEとして読み込んだものではあるが、とりあえず気にせずこのdataオブジェクトに対してdim関数を実行することを意味する②dim(data)を実行。実行結果。③このdataオブジェクトは12行×4列からなるデータだということを意味します。④確かにそうですね。

1. 目的のタブ区切りテキストファイル(annota (genelist1.txt)中のものが含まれる行全

```
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=
keywords <- readLines(in_f2)
dim(data)
```

#本番

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

#ファイルに保存

```
write.table(out, out_f, sep="\t"
```

R Console

	gene	accession	description	subcellular_location
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	agen1	membrane
11	gene11	hoge11	iyaaaa	endoplasmic

```
> dim(data)
[1] 12 4
> |
```

dim関数

1. 目的のタブ区切りテキストファイル(annota (genelist1.txt)中のものが含まれる行全

```
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=
keywords <- readLines(in_f2)
dim(data)
```

#本番

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

#ファイルに保存

```
write.table(out, out_f, sep="\t"
```

①このdataオブジェクトはheader=FALSEとして読み込んだものではあるが、とりあえず気にせずこのdataオブジェクトに対してdim関数を実行することを意味する②dim(data)を実行。実行結果。③このdataオブジェクトは12行×4列からなるデータだということを意味します。④確かにそうですね。ちなみに③dim関数実行結果は、2つの要素からなるベクトルです。1つ目の要素が行数に相当し、2つ目の要素が列数に相当します。

```
R Console
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

> dim(data)
[1] 12  4
> |
```


nrowとncol

1. 目的のタブ区切りテキストファイル(annota
(genelist1.txt)中のものが含まれる行全

```
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=
keywords <- readLines(in_f2)
dim(data)
```

#本番

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

#ファイルに保存

```
write.table(out, out_f, sep="\t"
```

①このdataオブジェクトはheader=FALSEとして読み込んだものではあるが、とりあえず気にせずこのdataオブジェクトに対してdim関数を実行することを意味する②dim(data)を実行。実行結果。③このdataオブジェクトは12行×4列からなるデータだということを意味します。④確かにそうですね。ちなみに③dim関数実行結果は、2つの要素からなるベクトルです。1つ目の要素が行数に相当し、2つ目の要素が列数に相当します。これらの結果は、⑤nrow関数、⑥ncol関数でも代用できます。いずれの関数も、日常的に利用します。

```
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
> dim(data)
[1] 12 4
> nrow(data)
[1] 12
> ncol(data)
[1] 4
> |
```

Contents

- 任意のキーワードを含む行を抽出(行列データの取り扱い)
- 色の説明と応用
- ありがちなミスや警告メッセージ
- コードの解説
 - 最初の4行分、read.table関数、quoteオプション(昔話)
 - Tips(上下左右の矢印キーを有効活用)、ヘッダ一行の有無
 - dim関数、nrow関数とncol関数、?関数名、class関数、dim関数マニュアルの続き
 - dim関数マニュアルの続き、dimnames、rownames、colnames、length
 - readLines、Tips(タブ補完)

?関数名

「?関数名」または「help(関数名)」で、どのようなデータを入力として与えると、どのような結果を返すのか、どのようなオプションが利用可能なのか、オプションのデフォルトは何か?を知ることができます。

1. 目的のタブ区切りテキストファイル(annota
(genelist1.txt)中のもが含まれる行全

```
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

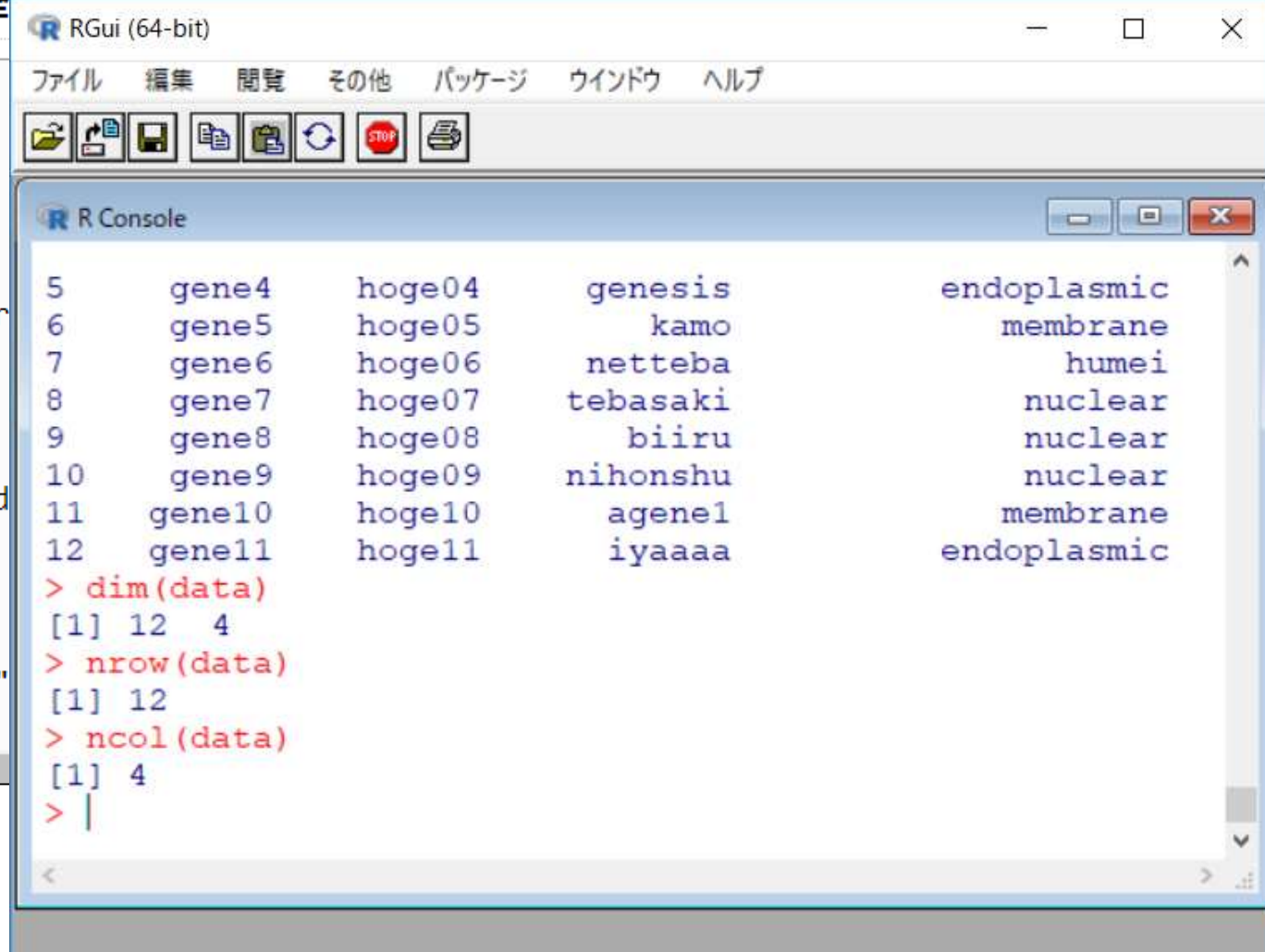
```
data <- read.table(in_f1, header=TRUE, as.is=TRUE)  
keywords <- readLines(in_f2)  
dim(data)
```

#本番

```
obj <- is.element(as.character(data[,1]), keywords)  
out <- data[obj,]  
dim(out)
```

#ファイルに保存

```
write.table(out, out_f, sep="\t")
```



```
RGui (64-bit)  
ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ  
R Console  
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  
> dim(data)  
[1] 12 4  
> nrow(data)  
[1] 12  
> ncol(data)  
[1] 4  
> |
```

?dim

「?dim」または「help(dim)」で、dim関数の詳細を知ることができます。①この状態でリターンキーを押すと…

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抽出する

```
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)
keywords <- readLines(in_f2)
dim(data)
```

#本番

```
obj <- is.element(as.character(data[,1]), keywords)
out <- data[obj,]
dim(out)
```

#ファイルに保存

```
write.table(out, out_f, sep="\t")
```

```
RGui (64-bit)
ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ
R Console
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
> dim(data)
[1] 12 4
> nrow(data)
[1] 12
> ncol(data)
[1] 4
> ?dim|
```



?dim

「?dim」または「help(dim)」で、dim関数の詳細を知ることができます。①この状態でリターンキーを押すと、②こんな感じになって…

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抽出したい場合

```
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

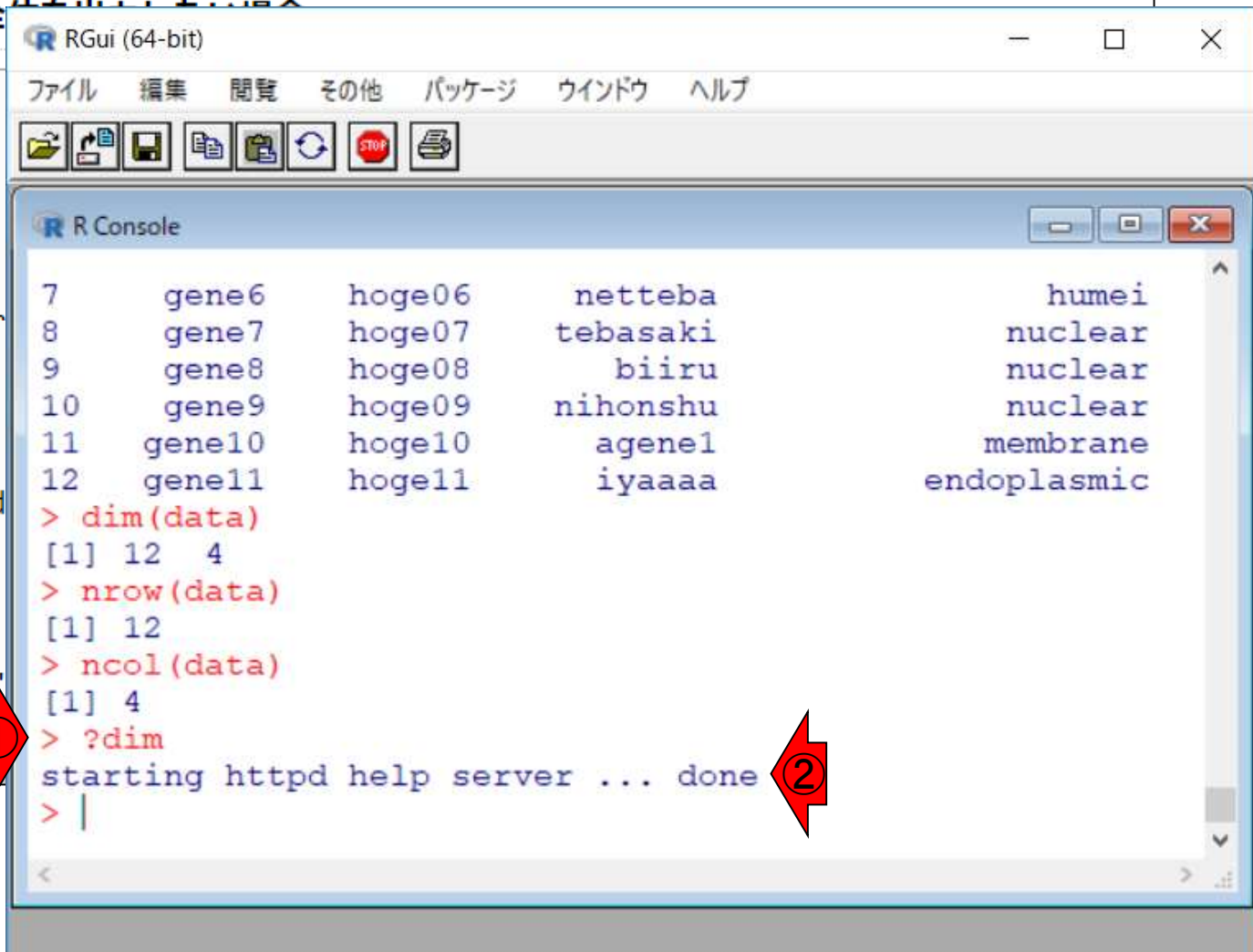
```
data <- read.table(in_f1, header=TRUE)
keywords <- readLines(in_f2)
dim(data)
```

#本番

```
obj <- is.element(as.character(keywords), data[,1])
out <- data[obj,]
dim(out)
```

#ファイルに保存

```
write.table(out, out_f, sep="\t", as.is=TRUE)
```



```
RGui (64-bit)
ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ

R Console
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

> dim(data)
[1] 12 4
> nrow(data)
[1] 12
> ncol(data)
[1] 4
> ?dim
starting httpd help server ... done
> |
```

?dim

「?dim」または「help(dim)」で、dim関数の詳細を知ることができます。①この状態でリターンキーを押すと、②こんな感じになって…ウェブブラウザが起動します。

R: Dimensions of an Object

127.0.0.1:11648/library/base/html/dim.html

dim {base}

R Documentation

Dimensions of an Object

Description

Retrieve or set the dimension of an object.

Usage

```
dim(x)
dim(x) <- value
```

Arguments

x
an R object, for example a matrix, array or data frame.

value

humei
nuclear
nuclear
nuclear
membrane
adoplasmic

?dim

①今マニュアルを調べている関数名。②関数のタイトル。この関数の挙動が既知の状態で見れば、「入力として与えられたオブジェクトの次元(dimension)」の意味が理解できるでしょう。

R: Dimensions of an Object

127.0.0.1:11648/library/base/html/dim.html

dim {base}

R Documentation

Dimensions of an Object

Description

Retrieve or set the dimension of an object.

Usage

```
dim(x)
dim(x) <- value
```

Arguments

x

an R object, for example a matrix, array or data frame.

value

```
humi
nuclear
nuclear
nuclear
membrane
adoplasmic
```

Description

①今マニュアルを調べている関数名。②関数のタイトル。この関数の挙動が既知の状態で見れば、「入力として与えられたオブジェクトの次元(dimension)」の意味が理解できるでしょう。③Descriptionのところ、④より詳細な関数の説明。

The screenshot shows the R documentation page for the 'dim' function. The browser address bar shows the URL '127.0.0.1:11648/library/base/html/dim.html'. The page content includes the function name 'dim {base}', the title 'Dimensions of an Object', the description 'Retrieve or set the dimension of an object.', the usage 'dim(x)' and 'dim(x) <- value', and the argument 'x' described as 'an R object, for example a matrix, array or data frame.'. Annotations with red arrows point to: ① the function name 'dim {base}', ② the title 'Dimensions of an Object', ③ the description 'Retrieve or set the dimension of an object.', and ④ the argument 'x'. A search bar on the right shows the results of a search for 'humi', 'nuclear', and 'membrane'.

R: Dimensions of an Object

127.0.0.1:11648/library/base/html/dim.html

dim {base}

R Documentation

Dimensions of an Object

Description

Retrieve or set the dimension of an object.

Usage

```
dim(x)
dim(x) <- value
```

Arguments

x

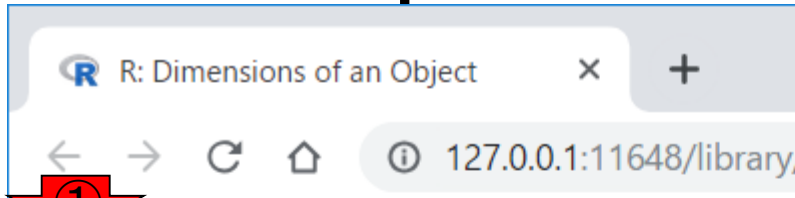
an R object, for example a matrix, array or data frame.

value

humi
nuclear
nuclear
nuclear
membrane
adoplasmic

Description

①今マニュアルを調べている関数名。②関数のタイトル。この関数の挙動が既知の状態で見れば、「入力として与えられたオブジェクトの次元(dimension)」の意味が理解できるでしょう。③Descriptionのところ、④より詳細な関数の説明。⑤retrieve(回収)という単語と結びつけることで、「dataオブジェクトの次元情報(行数と列数情報)を回収して表示しているのだ」みたいな感じで記述のノリに慣れていくとよい。



dim {base}

R Documentation

Dimensions of an Object

Description



Retrieve or set the dimension of an object.



Usage

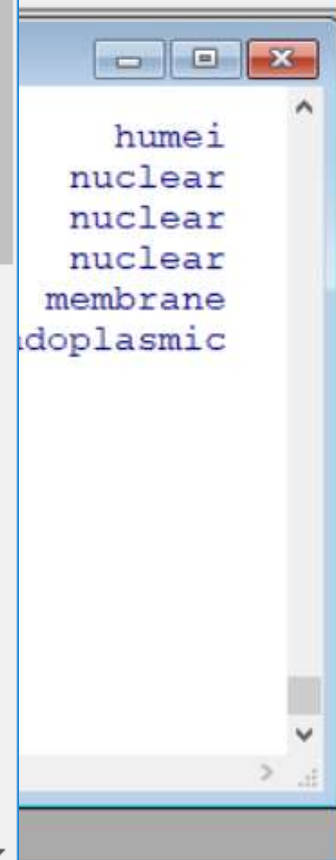


```
dim(x)  
dim(x) <- value
```

Arguments

x
an R object, for example a matrix, array or data frame.

value



Usage

①Usage(利用法)のところで、②のような感じで利用すればよいのだということを知る。

R: Dimensions of an Object

127.0.0.1:11648/library/base/html/dim.html

dim {base}

Dimensions of an Object

Description

Retrieve or set the dimension of an object.

Usage

```
dim(x)
dim(x) <- value
```

Arguments

x
an R object, for example a matrix, array or data frame.

value

```
humei
nuclear
nuclear
nuclear
membrane
adoplasmic
```

上手に手を抜く

① Usage (利用法) のところで、② のような感じで利用すればよいのだということを知る。③ のような使い方もあるようです。おそらくこれが、④ set に相当するのでしょうか。私は dim 関数をこのような形で利用したことは一度もありません。利用できる局面もあるのかもしれませんが、何でもかんでも完璧に理解しなければいけないというわけではないというよい例です。上手に手抜きして、無理なく学んでいけばよいと思います。

R: Dimensions of an Object x +
← → ↻ ⏠ ⓘ 127.0.0.1:11648/library

dim {base}

R Documentation

Dimensions of an Object

Description

Retrieve or set the dimension of an object.



Usage

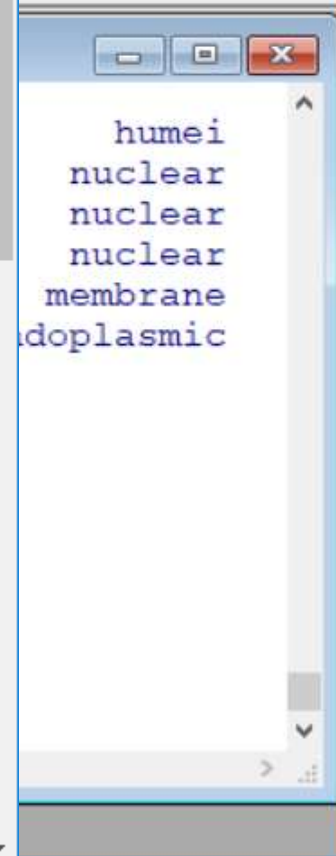
dim(x)
dim(x) <- value



Arguments

x
an R object, for example a matrix, array or data frame.

value



Arguments

①Arguments (引数; ひきすう) のところでは、入力としてどのようなものを与えるべきか、どのようなオプションが利用可能な
のか、オプションのデフォルトは何か、などの情報を知ることが
できます。②半ページほど下部に移動。

The screenshot shows a web browser displaying the R documentation for the `dim` function. The page title is "dim {base}" and the subtitle is "Dimensions of an Object". The main content includes a description: "Retrieve or set the dimension of an object." and a usage example: `dim(x)` and `dim(x) <- value`. The "Arguments" section is highlighted with a red arrow and a circled "1", showing the parameter `x` as "an R object, for example a matrix, array or data frame." On the right side, a scrollable list of related terms is visible, with a red arrow and a circled "2" pointing to the bottom of the list, indicating the instruction to scroll down.

dim {base}

Dimensions of an Object

Description

Retrieve or set the dimension of an object.

Usage

```
dim(x)
dim(x) <- value
```

Arguments

x

an R object, for example a matrix, array or data frame.

humei
nuclear
nuclear
nuclear
membrane
adoplasmic

Arguments

②これくらいまで移動。dim関数のArgumentsの中身は③のみです。④文字の大きさが同じDetailsが次の項目。この理由は、この関数自体の役割(行数と列数を表示するだけ)がシンプルで、オプションなども存在しないからです。

R: Dimensions of an Object

127.0.0.1:11648/library/base/html/dim.html

Usage

```
dim(x)  
dim(x) <- value
```

Arguments

①

x
an R object, for example a matrix, array or data frame.

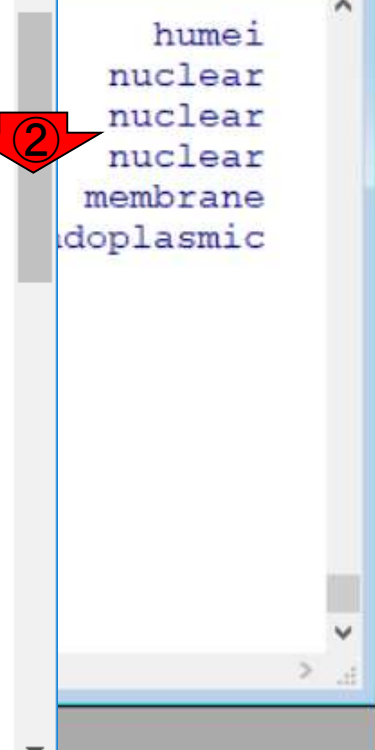
value
For the default method, either `NULL` or a numeric vector, which is coerced to integer (by truncation).

③

Details

④

The functions `dim` and `dim<-` are [internal generic primitive](#) functions.



②

Arguments

①でも見えていますが、入力として与えるのは②xというものののみです。③がxについての説明です。xは、「例えばmatrix, array, data frameという形式のRオブジェクト」だと解釈します。

The screenshot shows the R documentation for the `dim` function. The browser address bar shows the URL `127.0.0.1:11648/library/base/html/dim.html`. The page content includes a 'Usage' section with `dim(x)` and `dim(x) <- value`, an 'Arguments' section, and a 'Details' section. A red box highlights the 'Arguments' section, which contains the following text:

x ② an R object, for example a matrix, array or data frame. ③

value
For the default method, either `NULL` or a numeric vector, which is coerced to integer (by truncation).

Details
The functions `dim` and `dim<-` are [internal generic primitive](#) functions.

On the right side of the screenshot, a terminal window shows the output of the `dim` function applied to a character vector: `humi`, `nuclear`, `nuclear`, `nuclear`, `membrane`, and `adoplasmic`.

Arguments

①でも見えていますが、入力として与えるのは②xというもののみです。③がxについての説明です。xは、「例えばmatrix, array, data frameという形式のRオブジェクト」と解釈します。このことから、dim関数を実行できている④dataオブジェクトは、matrix, array, data frameのような形式だろうと予想できます。

R: Dimensions of an Object

Usage

①

```
dim(x)
dim(x) <- value
```

Arguments

②

x an R object, for example a matrix, array, or data frame

value For the default method, either a vector of integers or a character vector coerced to integer (by truncation)

Details

The functions dim and dim<- are generic functions. The default method for dim is implemented in base::dim.default. The default method for dim<- is implemented in base::dim.default<-.

R Console

```
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
```

```
> dim(data)
[1] 12 4
> nrow(data)
[1] 12
> ncol(data)
[1] 4
> ?dim
starting httpd help server ... done
> |
```

Contents

- 任意のキーワードを含む行を抽出(行列データの取り扱い)
- 色の説明と応用
- ありがちなミスや警告メッセージ
- コードの解説
 - 最初の4行分、read.table関数、quoteオプション(昔話)
 - Tips(上下左右の矢印キーを有効活用)、ヘッダ一行の有無
 - dim関数、nrow関数とncol関数、?関数名、class関数、dim関数マニュアルの続き
 - dim関数マニュアルの続き、dimnames、rownames、colnames、length
 - readLines、Tips(タブ補完)

①class関数は、オブジェクトの形式を返すものです。②dataオブジェクトは、③data.frameという形式であることが分かります。

class関数

R: Dimensions of an Object

Usage

```
dim(x)  
dim(x) <- value
```

Arguments

x
an R object, for example a matrix

value
For the default method, either a vector of length 2 or a vector of length 3, coerced to integer (by truncation)

Details

The functions dim and dim<- are

RGui (64-bit)

R Console

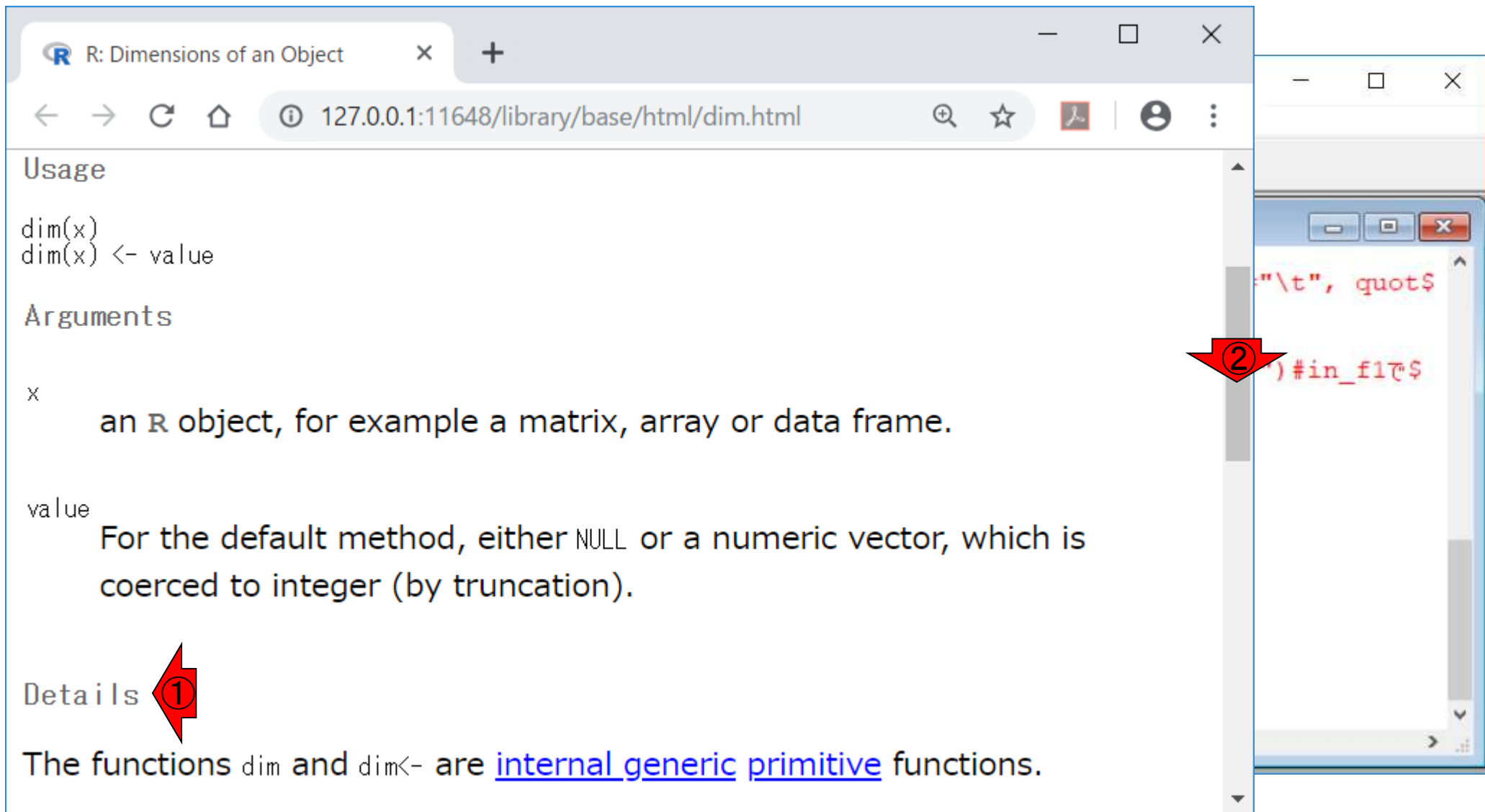
```
> data <- read.table(in_f1, header=TRUE, sep="\t", quote="$"  
> dim(data)  
[1] 11 4  
> data <- read.table(in_f1, sep="\t", quote="")#in_f1で$  
> dim(data)  
[1] 12 4  
> nrow(data)  
[1] 12  
> ncol(data)  
[1] 4  
> ?dim  
starting up help server ... done  
> class(data)  
[1] "data.frame"  
> |
```

① points to the 'x' argument in the Usage section of the left panel.

② points to the output of the `?dim` command in the R Console.

③ points to the output of the `class(data)` command in the R Console.

Details



The image shows a browser window displaying the R help page for the 'dim' function. The page is titled 'R: Dimensions of an Object' and the URL is '127.0.0.1:11648/library/base/html/dim.html'. The page content includes sections for 'Usage', 'Arguments', and 'Details'. A red arrow labeled '1' points to the 'Details' section header. A second red arrow labeled '2' points to the bottom of the page, indicating the scroll action.

Usage

```
dim(x)  
dim(x) <- value
```

Arguments

x
an R object, for example a matrix, array or data frame.

value
For the default method, either `NULL` or a numeric vector, which is coerced to integer (by truncation).

Details ①

The functions `dim` and `dim<-` are [internal generic primitive](#) functions.

②

①Detailsが一番上になるように、②ページ下部に移動。こんな感じ。

Details

R: Dimensions of an Object x +

127.0.0.1:11648/library/base/html/dim.html

Details ①

The functions `dim` and `dim<-` are [internal generic primitive](#) functions.

`dim` has a method for [data.frames](#), which returns the lengths of the `row.names` attribute of `x` and of `x` (as the numbers of rows and columns respectively).

Value

For an array (and hence in particular, for a matrix) `dim` retrieves the `dim` attribute of the object. It is `NULL` or a vector of mode [integer](#).

The replacement method changes the "dim" attribute (provided the new value is compatible) and removes any "dimnames" *and* "names" attributes.

References

②

Details

①Detailsが一番上になるように、②ページ下部に移動。こんな感じ。③dim関数についてのより詳細な記述。何を言いたいのか理解しがたいですが、dim関数を使う上では不都合ないので読み飛ばします。

R: Dimensions of an Object

127.0.0.1:11648/library/base/html/dim.html

Details

The functions `dim` and `dim<-` are [internal generic primitive](#) functions.

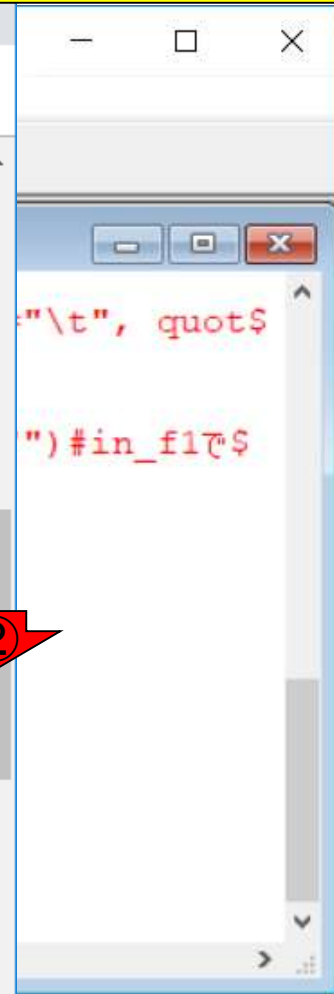
`dim` has a method for [data.frames](#), which returns the lengths of the `row.names` attribute of `x` and of `x` (as the numbers of rows and columns respectively).

Value

For an array (and hence in particular, for a matrix) `dim` retrieves the `dim` attribute of the object. It is `NULL` or a vector of mode [integer](#).

The replacement method changes the "dim" attribute (provided the new value is compatible) and removes any "dimnames" *and* "names" attributes.

References



Value

①Valueは、②dim関数を実行するとどういう結果が返されるかについての説明部分。「行数と列数という2つの要素からなる数値ベクトルを返す」この関数自体がシンプルすぎるため、かえって難解な印象を受けますが…

R: Dimensions of an Object

127.0.0.1:11648/library/base/html/dim.html

Details

The functions `dim` and `dim<-` are [internal generic primitive](#) functions.

`dim` has a method for [data.frames](#), which returns the lengths of the `row.names` attribute of `x` and of `x` (as the numbers of rows and columns respectively).

Value

For an array (and hence in particular, for a matrix) `dim` retrieves the `dim` attribute of the object. It is `NULL` or a vector of mode [integer](#).

The replacement method changes the "dim" attribute (provided the new value is compatible) and removes any "dimnames" *and* "names" attributes.

References

```
"\t", quot$  
")#in_f1で$
```

Value

①Valueは、②dim関数を実行するとどうい結果が返されるかについての説明部分。「行数と列数という2つの要素からなる数値ベクトルを返す」この関数自体がシンプルすぎるため、かえって難解な印象を受けますが、実際の挙動と一致する③「整数(integer)のベクトル」という記述を見ると安心できます。

R: Dimensions of an Object

127.0.0.1:11648/library/base/html/dim.html

Details

The functions `dim` and `dim<-` are [internal generic primitive](#) functions.

`dim` has a method for [data.frames](#), which returns the lengths of the `row.names` attribute of `x` and of `x` (as the numbers of rows and columns respectively).

Value

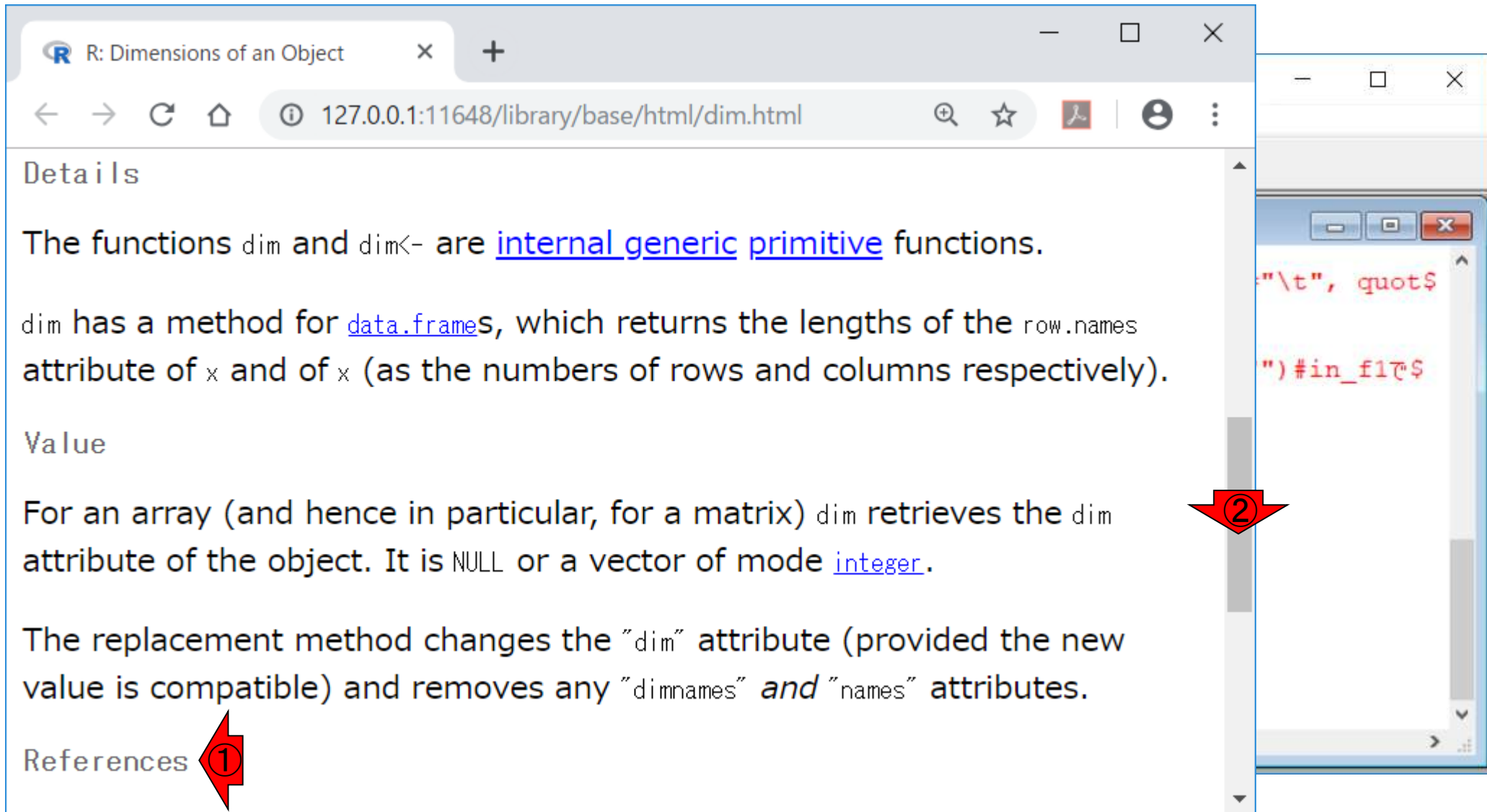
For an array (and hence in particular, for a matrix) `dim` retrieves the `dim` attribute of the object. It is `NULL` or a vector of mode [integer](#).

The replacement method changes the "dim" attribute (provided the new value is compatible) and removes any "dimnames" *and* "names" attributes.

References

```
"\t", quot$  
")#in_f1で$
```

References



The screenshot shows a web browser window displaying the R documentation for the `dim` function. The page is titled "R: Dimensions of an Object" and the URL is `127.0.0.1:11648/library/base/html/dim.html`. The content includes sections for "Details", "Value", and "References". A red arrow labeled "1" points to the "References" section at the bottom of the page. A second red arrow labeled "2" points to the scrollbar on the right side of the page, indicating the page is scrolled down to the bottom.

Details


The functions `dim` and `dim<-` are [internal generic primitive](#) functions.

`dim` has a method for [data.frames](#), which returns the lengths of the `row.names` attribute of `x` and of `x` (as the numbers of rows and columns respectively).


Value

For an array (and hence in particular, for a matrix) `dim` retrieves the `dim` attribute of the object. It is `NULL` or a vector of mode [integer](#).

The replacement method changes the "dim" attribute (provided the new value is compatible) and removes any "dimnames" *and* "names" attributes.

References 

```
"\t", quot$  
")#in_f1で$
```



References

①Referencesが一番上になるように、②ページ下部に移動。こんな感じ。③dim関数のReference(文献情報)です。現実問題として、利用した際にどこまで丁寧にReferenceを調べて記載するかはヒトそれぞれだと思いますが、感謝の気持ちをもって使いましょう。

References

1

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

3

See Also

[ncol](#), [nrow](#) and [dimnames](#).

Examples

```
x <- 1:12 ; dim(x) <- c(3,4)
x

# simple versions of nrow and ncol could be defined as follows
nrow0 <- function(x) dim(x)[1]
ncol0 <- function(x) dim(x)[2]
```

[Package *base* version 3.5.1 [Index](#)]

2

Contents

- 任意のキーワードを含む行を抽出(行列データの取り扱い)
- 色の説明と応用
- ありがちなミスや警告メッセージ
- コードの解説
 - 最初の4行分、read.table関数、quoteオプション(昔話)
 - Tips(上下左右の矢印キーを有効活用)、ヘッダ一行の有無
 - dim関数、nrow関数とncol関数、?関数名、class関数、dim関数マニュアルの続き
 - dim関数マニュアルの続き、dimnames、rownames、colnames、length
 - readLines、Tips(タブ補完)

See Also

The screenshot shows a web browser window displaying the R help page for 'Dimensions of an Object'. The page content includes a 'References' section with a citation for Becker et al. (1988), a 'See Also' section with a red box around the text '[ncol](#), [nrow](#) and [dimnames](#).', and an 'Examples' section with R code. A red box highlights the 'See Also' text, and two red arrows point to it from the left and right. A yellow box at the top right contains Japanese text explaining that the 'See Also' section provides useful information, specifically mentioning 'ncol' and 'nrow'. A second browser window is partially visible on the right side of the main window.

R: Dimensions of an Object

127.0.0.1:11648/library/base/html/dim.html

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

See Also ①

② [ncol](#), [nrow](#) and [dimnames](#).

Examples

```
x <- 1:12 ; dim(x) <- c(3,4)
x

# simple versions of nrow and ncol could be defined as follows
nrow0 <- function(x) dim(x)[1]
ncol0 <- function(x) dim(x)[2]
```

[Package *base* version 3.5.1 [Index](#)]

①See Alsoのところは、文字通り②で示す関数なども参考になるよ的な情報です。ncolやnrowは、③でも示していますので挙動既知ですね。

See Also

R: Dimensions of an Object

127.0.0.1:11648/

References

Becker, R. A., Chambers, J. M.
Language. Wadsworth & Brook

See Also

①

[ncol](#), [nrow](#) and [dimnames](#).

②

Examples

```
x <- 1:12 ; dim(x) <- c(3,4)
```

```
x
```

```
# simple versions of nrow and ncol co
nrow0 <- function(x) dim(x)[1]
ncol0 <- function(x) dim(x)[2]
```

[Package base version 3.5.1 [index](#)]

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ

R Console

```
> data <- read.table(in_f1, header=TRUE, sep="\t", quot$
> dim(data)
[1] 11 4
> data <- read.table(in_f1, sep="\t", quote="")#in_f1で$
> dim(data)
[1] 12 4
> nrow(data)
[1] 12
> ncol(data)
[1] 4
> ?dim
starting httpd help server ... done
> class(data)
[1] "data.frame"
> |
```

③

①See Alsoのところは、文字通り②で示す関数なども参考になるよ的な情報です。ncolやnrowは、③でも示していますので挙動既知ですね。④については、⑤dimnames(data)の⑥結果と...

See Also

R: Dimensions of an Object

127.0.0.1:11648/

References

Becker, R. A., Chambers, J. M. *Language*. Wadsworth & Brook

See Also

[ncol](#), [nrow](#) and [dimnames](#).



Examples

```
x <- 1:12 ; dim(x) <- c(3,4)
x

# simple versions of nrow and ncol co
nrow0 <- function(x) dim(x)[1]
ncol0 <- function(x) dim(x)[2]
```

[Package base version 3.5.1 [index](#)]

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウインドウ ヘルプ

R Console

```
> ncol(data)
[1] 4
> ?dim
starting httpd help server ... done
> class(data)
[1] "data.frame"
> dimnames(data)
[[1]]
 [1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10"
[11] "11" "12"

[[2]]
 [1] "v1" "v2" "v3" "v4"

> |
```

⑤ (points to the dimnames command)

⑥ (points to the output of dimnames)

See Also

①See Alsoのところは、文字通り②で示す関数なども参考になるよ的な情報です。ncolやnrowは、③でも示していますので挙動既知ですね。④については、⑤dimnames(data)の⑥結果と、⑦dataの結果を見比べると理解できるのではないのでしょうか?つまり⑥は、⑧と⑨の部分を表示しているということです。

R: Dimensions of an Object

127.0.0.1:11648/

References

Becker, R. A., Chambers, J. M.
Language. Wadsworth & Bro

See Also

[ncol](#), [nrow](#) and [dimnames](#).

Examples

```
x <- 1:12 ; dim(x) <- c(3,4)
x
# simple versions of nrow and ncol co
nrow0 <- function(x) dim(x)[1]
ncol0 <- function(x) dim(x)[2]
```

R Console

```
> data
```

	v1	v2	v3	v4
1	gene1	hoge01	plasma_mem	nuclear
2	gene2	hoge02	hohinu	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	agen1	membrane
11	gene11	hoge11	iyaaaa	endoplasmic
12				

[Package base version 3.5.1 [index](#)]

rownames関数

R: Dimensions of an Object

127.0.0.1:11648/

References

Becker, R. A., Chambers, J. M.
Language. Wadsworth & Brook

See Also

[ncol](#), [nrow](#) and [dimnames](#).

Examples

```
x <- 1:12 ; dim(x) <- c(3,4)
```

```
x
```

```
# simple versions of nrow and ncol co
```

```
nrow0 <- function(x) dim(x)[1]
```

```
ncol0 <- function(x) dim(x)[2]
```

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ

R Console

```
data
```

	V1	V2	V3	V4
1	gene1	hoge01	plasma_mem	nuclear
2	gene2	hoge02	hohinu	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	agen1	membrane
11	gene11	hoge11	iyaaaa	endoplasmic

```
> rownames(data)|
```

[Package base version 3.5.1 [index](#)]

同様に、私は①列名情報を抽出したい場合は、②colnames関数を使います。同じ目的を達成する上でも、様々な手段があります。見たことがない関数でも「?関数名」で調べるなりして、徐々に慣れていくとよいでしょう。

colnames関数

R: Dimensions of an Object

← → ↻ 🏠 ⓘ 127.0.0.1:11648/

References

Becker, R. A., Chambers, J. M.
Language. Wadsworth & Brook

See Also

[ncol](#), [nrow](#) and [dimnames](#).

Examples

```
x <- 1:12 ; dim(x) <- c(3,4)
```

```
x
```

```
# simple versions of nrow and ncol col
nrow0 <- function(x) dim(x)[1]
ncol0 <- function(x) dim(x)[2]
```

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ



R Console

```
> data
```

	V1	V2	V3	V4
1	gene1	hoge01	plasma_mem	nuclear
2	gene2	hoge02	hohinu	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	agen1	membrane
11	gene11	hoge11	iyaaaa	endoplasmic

```
> colnames(data)|
```

[Package base version 3.5.1 [index](#)]

Examples

①Examplesのところに、②関数の利用例があります。このコードを眺めたり実際にコピー実行したりして理解を深めていくこともできます。

R: Dimensions of an Object

127.0.0.1:11648/library/base/html/dim.html

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

See Also

[ncol](#), [nrow](#) and [dimnames](#).

Examples

```
x <- 1:12 ; dim(x) <- c(3,4)
```

```
x
```

```
# simple versions of nrow and ncol could be defined as follows
```

```
nrow0 <- function(x) dim(x)[1]
```

```
ncol0 <- function(x) dim(x)[2]
```

[Package *base* version 3.5.1 [Index](#)]

Examples

①Examplesのところに、②関数の利用例があります。このコードを眺めたり実際にコピー実行したりして理解を深めていくこともできます。例えば③の行は、④と⑤の2つのコマンドから構成されます。逆に言えば、⑥セミコロン(;)を間に挟むことで、独立した2つのコマンドを同一行に書くこともできます。こういう書き方をする人も一定数いると思います。

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

See Also

[ncol](#), [nrow](#) and [dimnames](#).

Examples

```
x <- 1:12 ; dim(x) <- c(3,4)
```

x
④ ⑥ ⑤



```
# simple versions of nrow and ncol could be defined as follows  
nrow0 <- function(x) dim(x)[1]  
ncol0 <- function(x) dim(x)[2]
```

[Package *base* version 3.5.1 [Index](#)]

Examples

①の部分のコピペ実行。これは、「1から12の整数からなる数値ベクトルを作成した結果を、xというオブジェクト名で取り扱えるようにしている」ことに相当します。

The screenshot shows the RGui (64-bit) interface. The R Console window displays the following output:

```
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
> rownames(data)
 [1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10"
 [11] "11" "12"
> colnames(data)
 [1] "V1" "V2" "V3" "V4"
> x <- 1:12
> |
```

The web browser window shows the R documentation page for `ncol`, `nrow`, and `dimnames`. The code example in the documentation is:

```
x <- 1:12 ; dim(x) <- c(3,4)
x
# simple versions of nrow and ncol
nrow0 <- function(x) dim(x)[1]
ncol0 <- function(x) dim(x)[2]
```

Red arrows with the number 1 point to the `x` variable in the code example and the `> x <- 1:12` command in the R Console output.

[Package base version 3.5.1 [index](#)]

Examples

①の部分のコピー実行。これは、「1から12の整数からなる数値ベクトルを作成した結果を、xというオブジェクト名で取り扱えるようにしている」ことに相当します。②xの中身を表示。確かに12個の要素からなる整数のベクトルになっていますね。

R: Dimensions of an Object x +
127.0.0.1:11648/

References

Becker, R. A., Chambers, J. M.
Language. Wadsworth & Brook

See Also

[ncol](#), [nrow](#) and [dimnames](#).

Examples

```
x <- 1:12 ; dim(x) <- c(3,4)
```

```
x  
# simple versions of nrow and ncol  
nrow0 <- function(x) dim(x)[1]  
ncol0 <- function(x) dim(x)[2]
```

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ

R Console

```
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  
> rownames(data)  
[1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10"  
[11] "11" "12"  
> colnames(data)  
[1] "V1" "V2" "V3" "V4"  
> x <- 1:12  
> x  
[1] 1 2 3 4 5 6 7 8 9 10 11 12  
> |
```

[Package base version 3.5.1 [index](#)]

Contents

- 任意のキーワードを含む行を抽出(行列データの取り扱い)
- 色の説明と応用
- ありがちなミスや警告メッセージ
- コードの解説
 - 最初の4行分、read.table関数、quoteオプション(昔話)
 - Tips(上下左右の矢印キーを有効活用)、ヘッダ一行の有無
 - dim関数、nrow関数とncol関数、?関数名、class関数、dim関数マニュアルの続き
 - dim関数マニュアルの続き、dimnames、rownames、colnames、length
 - readLines、Tips(タブ補完)

length関数

①の部分のコピー実行。これは、「1から12の整数からなる数値ベクトルを作成した結果を、xというオブジェクト名で取り扱えるようにしている」ことに相当します。②xの中身を表示。確かに12個の要素からなる整数のベクトルになっていますね。④(ベクトルの)要素数を調べたい場合はlength関数を使うのが一般的だと思います。

R: Dimensions of an Object

127.0.0.1:11648/

References

Becker, R. A., Chambers, J. M.
Language. Wadsworth & Brook

See Also

[ncol](#), [nrow](#) and [dimnames](#).

Examples

```
x <- 1:12 ; dim(x) <- c(3,4)
```

x

```
# simple versions of nrow and ncol  
nrow0 <- function(x) dim(x)[1]  
ncol0 <- function(x) dim(x)[2]
```

R RG

ファイル



R Console

```
9      gene8      hoge08      biiru      nuclear  
10     gene9      hoge09      nihonshu   nuclear  
11     gene10     hoge10     agene1     membrane  
12     gene11     hoge11     iyaaaa    endoplasmic  
> rownames(data)  
[1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10"  
[11] "11" "12"  
> colnames(data)  
[1] "v1" "v2" "v3" "v4"  
> x <- 1:12  
> x  
[1] 1 2 3 4 5 6 7 8 9 10 11 12  
> length(x)  
[1] 12  
> |
```

[Package base version 3.5.1 [index](#)]

dim(x)

ちなみに、⑤このオブジェクトxに対してdim関数を実行しても、⑥NULLとなります。理由は、xが行列ではなくベクトルだからです。

The image shows a screenshot of an R environment. On the left is the R documentation page for `dim(x)`. On the right is the R Console window showing a series of commands and their outputs. Red arrows with circled numbers 1 through 6 point to specific parts of the code and output.

References
Becker, R. A., Chambers, J. M. *Language*. Wadsworth & Brook
See Also
[ncol](#), [nrow](#) and [dimnames](#).

Examples

```
x <- 1:12 ; dim(x) <- c(3,4)
```

simple versions of nrow and ncol
nrow0 <- function(x) dim(x)[1]
ncol0 <- function(x) dim(x)[2]

R Console Output:

```
11  gene10  hoge10  agene1  membrane  
12  gene11  hoge11  iyaaaa  endoplasmic  
> rownames(data)  
[1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10"  
[11] "11" "12"  
> colnames(data)  
[1] "v1" "v2" "v3" "v4"  
> x <- 1:12  
> x  
[1] 1 2 3 4 5 6 7 8 9 10 11 12  
> length(x)  
[1] 12  
> dim(x)  
NULL  
> |
```

[Package base version 3.5.1 [index](#)]

class(x)

ちなみに、⑤このオブジェクトxに対してdim関数を実行しても、⑥NULLとなります。理由は、xが行列ではなくベクトルだからです。⑦class(x)で、オブジェクトxがintegerという形式であることを知る。確かに整数(integer)からなるベクトルなので妥当。

R: Dimensions of an Object
127.0.0.1:11648/

References
Becker, R. A., Chambers, J. M.
Language. Wadsworth & Brook

See Also
[ncol](#), [nrow](#) and [dimnames](#).

Examples
`x <- 1:12` ; `dim(x) <- c(3,4)`
x
simple versions of nrow and ncol
`nrow0 <- function(x) dim(x)[1]`
`ncol0 <- function(x) dim(x)[2]`

```
RGui (64-bit)
ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ

R Console
> rownames(data)
[1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10"
[11] "11" "12"
> colnames(data)
[1] "v1" "v2" "v3" "v4"
> x <- 1:12
> x
[1] 1 2 3 4 5 6 7 8 9 10 11 12
> length(x)
[1] 12
> dim(x)
NULL
> class(x)
[1] "integer"
> |
```

[Package base version 3.5.1 [index](#)]

まだExamplesの続きです。①の部分のコピペ実行。これは、xの次元を3行×4列に変更せよ、という命令です。

Examples

R: Dimensions of an Object

127.0.0.1:11648/

References

Becker, R. A., Chambers, J. M.
Language. Wadsworth & Brook

See Also

[ncol](#), [nrow](#) and [dimnames](#).

Examples

```
x <- 1:12 ; dim(x) <- c(3,4)
```

```
x  
# simple versions of nrow and ncol  
nrow0 <- function(x) dim(x)[1]  
ncol0 <- function(x) dim(x)[2]
```



[Package base version 3.5.1 [index](#)]

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ

R Console

```
[1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10"  
[11] "11" "12"  
> colnames(data)  
[1] "v1" "v2" "v3" "v4"  
> x <- 1:12  
> x  
[1] 1 2 3 4 5 6 7 8 9 10 11 12  
> length(x)  
[1] 12  
> dim(x)  
NULL  
> class(x)  
[1] "integer"  
> dim(x) <- c(3,4)  
> |
```


Examples

まだExamplesの続きです。①の部分のコピペ実行。これは、xの次元を3行×4列に変更せよ、という命令です。②xの中身を表示。確かに3行×4列のデータに変更されていますね。

R: Dimensions of an Object

127.0.0.1:11648/

References

Becker, R. A., Chambers, J. M.
Language. Wadsworth & Brook

See Also

[ncol](#), [nrow](#) and [dimnames](#).

Examples

```
x <- 1:12 ; dim(x) <- c(3,4)
```

```
x  
# simple versions of nrow and ncol co  
nrow0 <- function(x) dim(x)[1]  
ncol0 <- function(x) dim(x)[2]
```

[Package base version 3.5.1 [index](#)]

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ

R Console

```
> x  
[1] 1 2 3 4 5 6 7 8 9 10 11 12  
> length(x)  
[1] 12  
> dim(x)  
NULL  
> class(x)  
[1] "integer"  
> dim(x) <- c(3,4)  
> x  
      [,1] [,2] [,3] [,4]  
[1,]  1   4   7  10  
[2,]  2   5   8  11  
[3,]  3   6   9  12  
> |
```

class(x)

まだExamplesの続きです。①の部分のコピペ実行。これは、xの次元を3行×4列に変更せよ、という命令です。②xの中身を表示。確かに3行×4列のデータに変更されていますね。もう一度③class(x)を実行。この際、打ち込まずに「上矢印キーを3回押して、④以前実行したコマンドを有効利用」するなどして効率的に進めよう。

R: Dimensions of an Object

127.0.0.1:11648/

References

Becker, R. A., Chambers, J. M.
Language. Wadsworth & Brook

See Also

[ncol](#), [nrow](#) and [dimnames](#).

Examples

```
x <- 1:12 ; dim(x) <- c(3,4)
```

```
x  
# simple versions of nrow and ncol  
nrow0 <- function(x) dim(x)[1]  
ncol0 <- function(x) dim(x)[2]
```

RG

ファイル



R Console

```
> length(x)  
[1] 12  
> dim(x)  
NULL  
> class(x)  
[1] "integer"  
> dim(x) <- c(3,4)  
> x  
      [,1] [,2] [,3] [,4]  
[1,]    1    4    7   10  
[2,]    2    5    8   11  
[3,]    3    6    9   12  
> class(x)  
[1] "matrix"  
> |
```

[Package base version 3.5.1 [index](#)]

class(x)

まだExamplesの続きです。①の部分のコピペ実行。これは、xの次元を3行×4列に変更せよ、という命令です。②xの中身を表示。確かに3行×4列のデータに変更されていますね。もう一度③class(x)を実行。この際、打ち込まずに「上矢印キーを3回押して、④以前実行したコマンドを有効利用」するなどして効率的に進めよう。⑤xがintegerからmatrixに変わっていることが分かります。こんな感じで**基本的な関数やテクニックを駆使してstep-by-step**で着実に自分のペースで理解していこう。

R: Dimensions of an Object
127.0.0.1:11648/

References
Becker, R. A., Chambers, J. M.
Language. Wadsworth & Brook

See Also
[ncol](#), [nrow](#) and [dimnames](#).

```
Examples  
x <- 1:12 ; dim(x) <- c(3,4)  
x  
# simple versions of nrow and ncol  
nrow0 <- function(x) dim(x)[1]  
ncol0 <- function(x) dim(x)[2]
```

```
R Console  
> length(x)  
[1] 12  
> dim(x)  
NULL  
> class(x)  
[1] "integer"  
> dim(x) <- c(3,4)  
> x  
      [,1] [,2] [,3] [,4]  
[1,]    1    4    7   10  
[2,]    2    5    8   11  
[3,]    3    6    9   12  
> class(x)  
[1] "matrix"  
> |
```

[Package base version 3.5.1 [index](#)]

Contents

- 任意のキーワードを含む行を抽出(行列データの取り扱い)
- 色の説明と応用
- ありがちなミスや警告メッセージ
- コードの解説
 - 最初の4行分、read.table関数、quoteオプション(昔話)
 - Tips(上下左右の矢印キーを有効活用)、ヘッダ一行の有無
 - dim関数、nrow関数とncol関数、?関数名、class関数、dim関数マニュアルの続き
 - dim関数マニュアルの続き、dimnames、rownames、colnames、length
 - readLines、Tips(タブ補完)

これまでは①の行を除く、②赤枠内のコードの解説を行いました。②赤枠内を再度コピー実行。

コードの解説

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
```



コードの解説

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抽出したい場合

```
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", as.is=TRUE)
keywords <- readLines(in_f2)
dim(data)
```

```
#本番
obj <- is.element(as.character(data[,1]), keywords)
out <- data[obj,]
dim(out)
```

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

RGui (64-bit) window showing the R Console output and code. The console output shows a matrix of data:

```
[2,] 2 5 8 11
[3,] 3 6 9 12
```

The R Console shows the execution of the code with comments:

```
> class(x)
[1] "matrix"
> in_f1 <- "annotation.txt" #入力ファイル$
> in_f2 <- "genelist1.txt" #入力ファイル$
> out_f <- "hoge1.txt" #出力ファイル$
> param <- 1 #アノテーション$
>
> #入力ファイルの読み込み
> data <- read.table(in_f1, header=TRUE, sep="\t", quot$
> keywords <- readLines(in_f2) #in_f2で指定し$
> dim(data) #オブジェクトd$
[1] 11 4
> |
```

コードの解説

こんな感じになります。①dataオブジェクトが11行×4列になっている理由は、②header=TRUEとした状態で読み込んでいるので、最初の行がヘッダ行として認識されているからです。

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

```
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)
keywords <- readLines(in_f2)
dim(data)
```

#本番

```
obj <- is.element(as.character(keywords), data[,1])
out <- data[obj,]
dim(out)
```

#ファイルに保存

```
write.table(out, out_f, sep="\t")
```

```
RGui (64-bit)
ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ

R Console
[2,] 2 5 8 11
[3,] 3 6 9 12
> class(x)
[1] "matrix"
> in_f1 <- "annotation.txt" #入力ファイル$
> in_f2 <- "genelist1.txt" #入力ファイル$
> out_f <- "hoge1.txt" #出力ファイル$
> param <- 1 #アノテーション$
>
> #入力ファイルの読み込み
> data <- read.table(in_f1, header=TRUE, sep="\t", quot$
> keywords <- readLines(in_f2) #in_f2で指定し$
> dim(data) #オブジェクトd$
[1] 11 4
> |
<
```

readLines

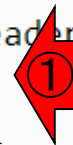
1. 目的のタブ区切りテキストファイル(annotation.txt)中の第1列目をキーとして、リストファイル(genelist.txt)中のものが含まれる行全体を抽出したい場合

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



#入力ファイルの読み込み

```
data <- read.table(in_f1, header=TRUE, sep="\t", as.is=TRUE)
keywords <- readLines(in_f2)
dim(data)
```



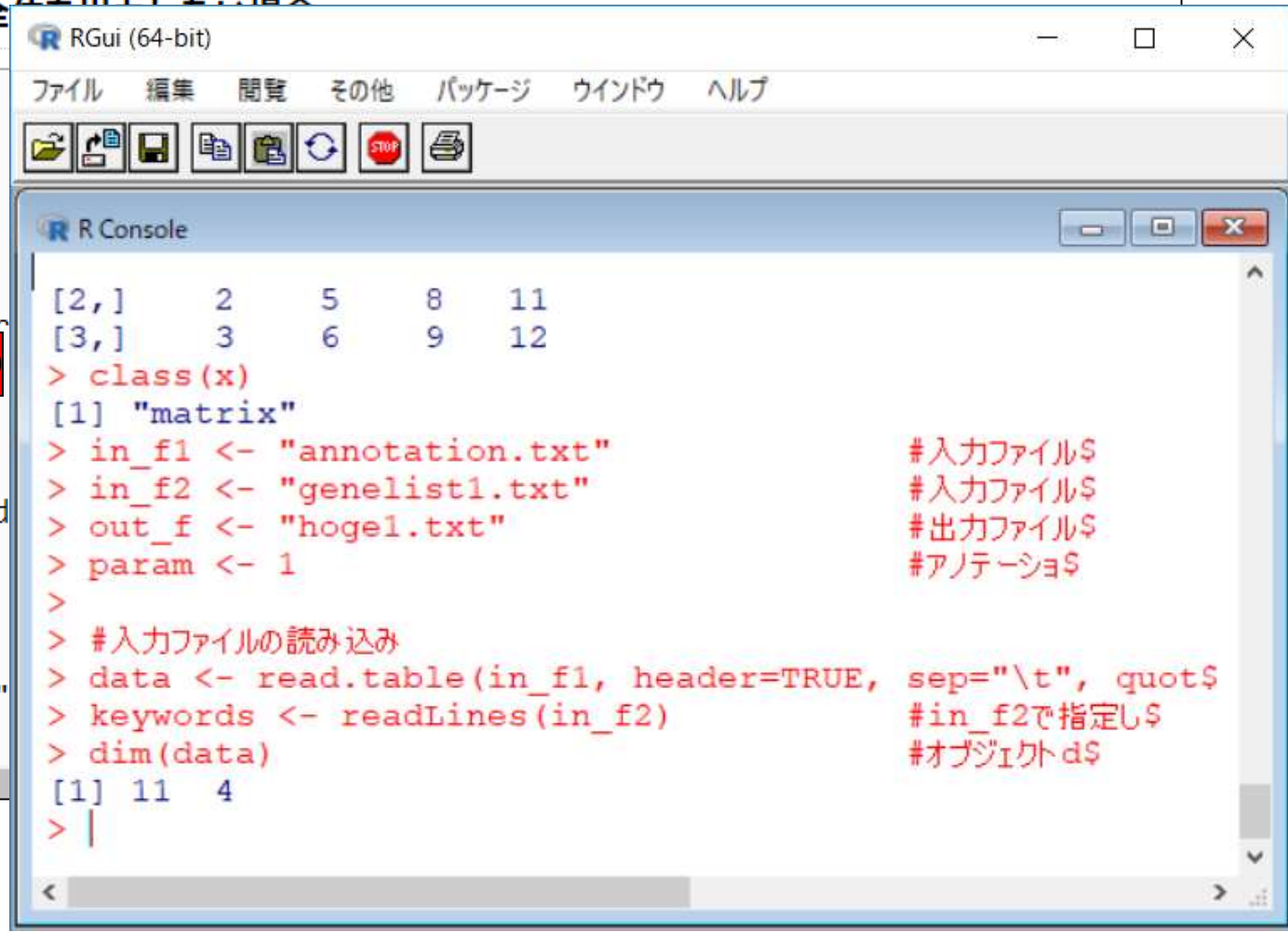
#本番

```
obj <- is.element(as.character(data[,1]), keywords)
out <- data[obj,]
dim(out)
```

#ファイルに保存

```
write.table(out, out_f, sep="\t", as.is=TRUE)
```

```
gene1 ↓
gene7 ↓
gene9 ↓
|
```



readLines

①では、中身が②の、③in_f2を、④readLines関数を用いて読み込んだ結果を、⑤keywordsというオブジェクト名で保存しています。

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

```
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", as.is=TRUE)  
keywords <- readLines(in_f2)
```

#本番

```
obj <- is.element(as.character(data[,1]), keywords)  
out <- data[obj,]  
dim(out)
```

#ファイルに保存

```
write.table(out, out_f, sep="\t", as.is=TRUE)
```

```
gene1 ↓  
gene7 ↓  
gene9 ↓  
|
```

```
RGui (64-bit)  
ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ  
R Console  
[2,] 2 5 8 11  
[3,] 3 6 9 12  
> class(x)  
[1] "matrix"  
> in_f1 <- "annotation.txt" #入力ファイル$  
> in_f2 <- "genelist1.txt" #入力ファイル$  
> out_f <- "hoge1.txt" #出力ファイル$  
> param <- 1 #アノテーション$  
>  
> #入力ファイルの読み込み  
> data <- read.table(in_f1, header=TRUE, sep="\t", quot$  
> keywords <- readLines(in_f2) #in_f2で指定し$  
> dim(data) #オブジェクトd$  
[1] 11 4  
> |
```

タブ補完

①では、中身が②の、③in_f2を、④readLines関数を用いて読み込んだ結果を、⑤keywordsというオブジェクト名で保存しています。⑤keywordsの中身を表示させて確認しますが、コピー実行しないで既存のオブジェクトを打ち込む際は、スペルミス防止のため、タブ補完を有効利用する癖をつけてください。

1. 目的のタブ区切りテキストファイル(annota
(genelist1.txt)中のもが含まれる行全

```
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", as.is=TRUE)  
keywords <- readLines(in_f2)
```

#本番

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

#ファイルに保存

```
write.table(out, out_f, sep="\t", as.is=TRUE)
```

```
gene1 ↓  
gene7 ↓  
gene9 ↓  
|
```

The screenshot shows the R Studio interface. The menu bar includes 'ファイル', '編集', '閲覧', 'その他', 'パッケージ', 'ウインドウ', and 'ヘルプ'. The R Console window displays the following output:

```
[2,] 2 5 8 11  
[3,] 3 6 9 12  
> class(x)  
[1] "matrix"  
> in_f1 <- "annotation.txt" #入力ファイル$  
> in_f2 <- "genelist1.txt" #入力ファイル$  
> out_f <- "hoge1.txt" #出力ファイル$  
> param <- 1 #アノテーション$  
>  
> #入力ファイルの読み込み  
> data <- read.table(in_f1, header=TRUE, sep="\t", quot$  
> keywords <- readLines(in_f2) #in_f2で指定し$  
> dim(data) #オブジェクトd$  
[1] 11 4  
> |
```

タブ補完

①では、中身が②の、③in_f2を、④readLines関数を用いて読み込んだ結果を、⑤keywordsというオブジェクト名で保存しています。⑤keywordsの中身を表示させて確認しますが、コピペ実行しないで既存のオブジェクトを打ち込む際は、スペルミス防止のため、タブ補完を有効利用する癖をつけてください。具体的には、⑥keyまで(keまででもよい)打ち込んでから…

1. 目的のタブ区切りテキストファイル(annota (genelist1.txt)中のもが含まれる行全

```
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", as.is=TRUE)
keywords <- readLines(in_f2)
```

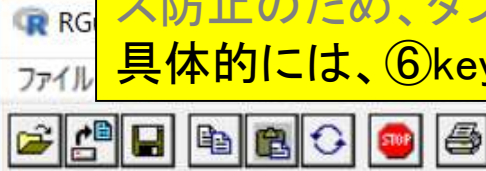
#本番

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

#ファイルに保存

```
write.table(out, out_f, sep="\t", as.is=TRUE)
```

```
gene1↓
gene7↓
gene9↓
|←
```



```
R Console
[2,] 2 5 8 11
[3,] 3 6 9 12
> class(x)
[1] "matrix"
> in_f1 <- "annotation.txt" #入力ファイル$
> in_f2 <- "genelist1.txt" #入力ファイル$
> out_f <- "hoge1.txt" #出力ファイル$
> param <- 1 #アノテーション$
>
> #入力ファイルの読み込み
> data <- read.table(in_f1, header=TRUE, sep="\t", quot$
> keywords <- readLines(in_f2) #in_f2で指定し$
> dim(data) #オブジェクトd$
[1] 11 4
> key|
```

タブ補完

①では、中身が②の、③in_f2を、④readLines関数を用いて読み込んだ結果を、⑤keywordsというオブジェクト名で保存しています。⑤keywordsの中身を表示させて確認しますが、コピペ実行しないで既存のオブジェクトを打ち込む際は、スペルミス防止のため、タブ補完を有効利用する癖をつけてください。具体的には、⑥keyまで(keまででもよい)打ち込んでから、⑦Tabキーを押すと…

1. 目的のタブ区切りテキストファイル(annota
(genelist1.txt)中のもが含まれる行全

```
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)  
keywords <- readLines(in_f2)
```

#本番

```
obj <- is.element(as.character(data[,1]), keywords)  
out <- data[obj,]  
dim(out)
```

#ファイルに保存

```
write.table(out, out_f, sep="\t")
```

```
gene1↓  
gene7↓  
gene9↓  
|
```



```
R Console  
[2,] 2 5 8 11  
[3,] 3 6 9 12  
> class(x)  
[1] "matrix"  
> 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)  
> keywords <- readLines(in_f2)  
> dim(data)  
[1] 11 4  
> key|
```



タブ補完

①では、中身が②の、③in_f2を、④readLines関数を用いて読み込んだ結果を、⑤keywordsというオブジェクト名で保存しています。⑤keywordsの中身を表示させて確認しますが、コピペ実行しないで既存のオブジェクトを打ち込む際は、スペルミス防止のため、タブ補完を有効利用する癖をつけてください。具体的には、⑥keyまで(keまででもよい)打ち込んでから、⑦Tabキーを押すと、⑧こんな感じになります。

1. 目的のタブ区切りテキストファイル(annota
(genelist1.txt)中のもが含まれる行全

```
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)  
keywords <- readLines(in_f2)
```

#本番

```
obj <- is.element(as.character(data[,1]), keywords)  
out <- data[obj,]  
dim(out)
```

#ファイルに保存

```
write.table(out, out_f, sep="\t")
```

```
gene1↓  
gene7↓  
gene9↓  
|
```



```
R Console  
[2,] 2 5 8 11  
[3,] 3 6 9 12  
> class(x)  
[1] "matrix"  
> 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)  
> keywords <- readLines(in_f2)  
> dim(data)  
[1] 11 4  
> keywords|
```



①リターンキーを押した結果。②中身が一致しており妥当ですね。

タブ補完

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.txt)中のものが含まれる行全体を抽出したい場合

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

#入力ファイルの読み込み

```
data <- read.table(in_f1, header=TRUE, sep="\t", as.is=TRUE)  
keywords <- readLines(in_f2)  
dim(data)
```

#本番

```
obj <- is.element(as.character(data[,1]), keywords)  
out <- data[obj,]  
dim(out)
```

#ファイルに保存

```
write.table(out, out_f, sep="\t", as.is=TRUE)
```

```
gene1 ↓  
gene7 ↓  
gene9 ↓  
|
```

```
> class(x)  
[1] "matrix"  
> in_f1 <- "annotation.txt" #入力ファイル$  
> in_f2 <- "genelist.txt" #入力ファイル$  
> out_f <- "hoge1.txt" #出力ファイル$  
> param <- 1 #アノテーション$  
>  
> #入力ファイルの読み込み  
> data <- read.table(in_f1, header=TRUE, sep="\t", quot$  
> keywords <- readLines(in_f2) #in_f2で指定し$  
> dim(data) #オブジェクトd$  
[1] 11 4  
> keywords  
[1] "gene1" "gene7" "gene9"  
> |
```

classで確認

①リターンキーを押した結果。②中身が一致しており妥当ですね。③keywordsに対してclass関数を実行すると、④character(文字列)という結果。数値ではないので妥当ですね。

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.txt)中のものが含まれる行全体を抽出したい場合

```
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", as.is=TRUE)
keywords <- readLines(in_f2)
dim(data)
```

#本番

```
obj <- is.element(as.character(data[,1]), keywords)
out <- data[obj,]
dim(out)
```

#ファイルに保存

```
write.table(out, out_f, sep="\t", as.is=TRUE)
```

```
gene1↓
gene7↓
gene9↓
|
```

```
> in_f1 <- "annotation.txt" #入力ファイル$
> in_f2 <- "genelist1.txt" #入力ファイル$
> out_f <- "hoge1.txt" #出力ファイル$
> param <- 1 #アノテーション$
>
> #入力ファイルの読み込み
> data <- read.table(in_f1, header=TRUE, sep="\t", quot$
> keywords <- readLines(in_f2) #in_f2で指定し$
> dim(data) #オブジェクトd$
[1] 11 4
> keywords
[1] "gene1" "gene7" "gene9"
> class(keywords)
[1] "character"
> |
```