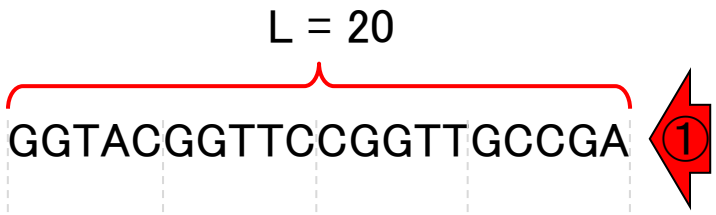
Contents

- Introduction、出現頻度解析（k=2）、出現頻度解析（k=1）
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図（例題10）、作図（例題11）、作図（例題12）
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ（例題32）、被覆率（coverage）、基本的な考え方(例題7)
  - 例題8（k=2)、例題9（k=3）、1,000塩基の仮想ゲノム（サンプルデータの例題33）
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

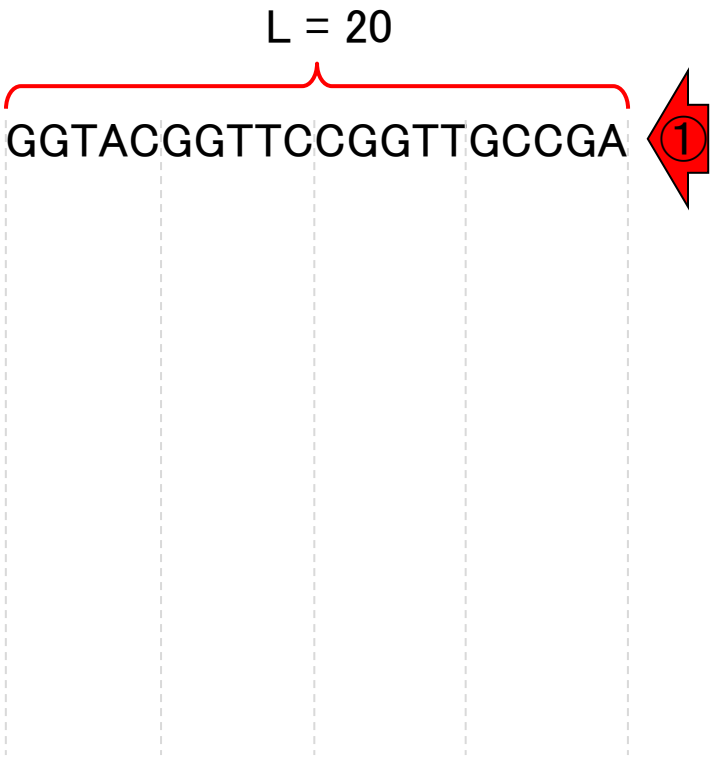
Introduction1

NGS解析で、①のようなL=20塩基からなるリードが得られたとする。



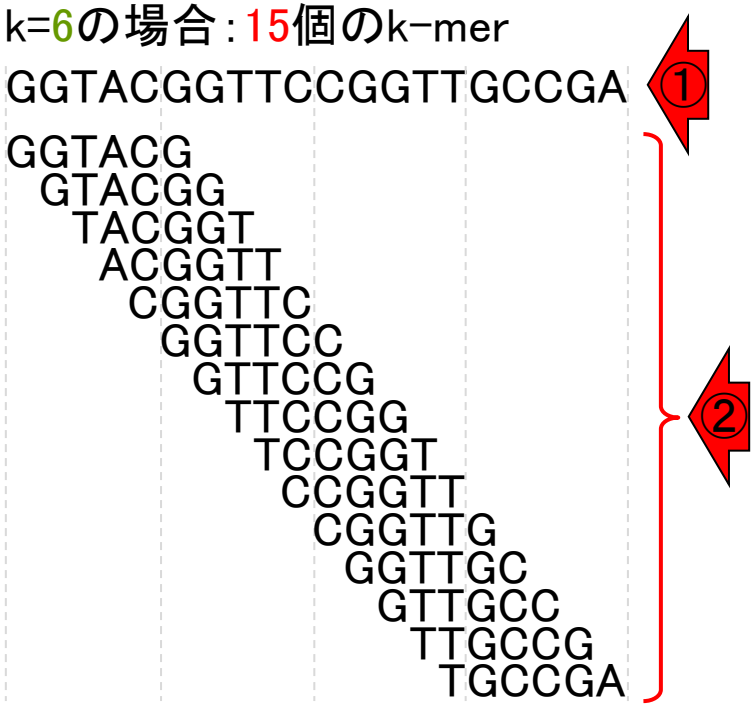
Introduction2

NGS解析で、①のようなL=20塩基からなるリードが得られたとする。k-mer解析は、k塩基の長さからなる部分塩基配列を生成して、出現頻度を調べたりする作業のこと。



Introduction3

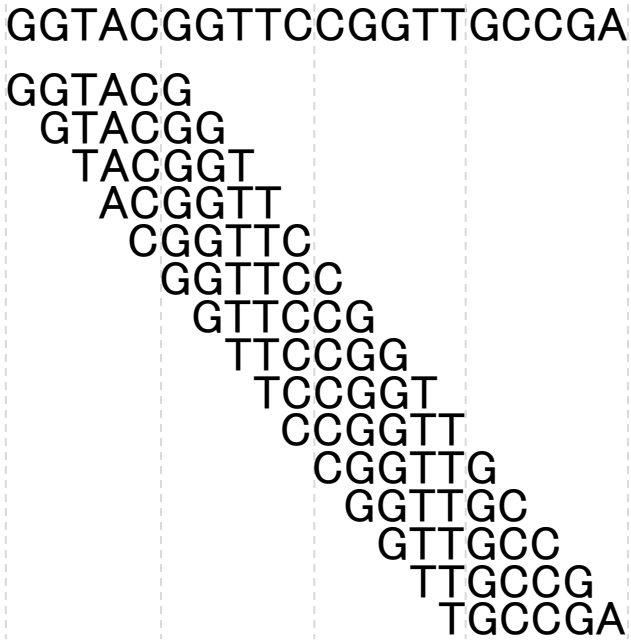
NGS解析で、①のようなL=20塩基からなるリードが得られたとする。k-mer解析は、k塩基の長さからなる部分塩基配列を生成して、出現頻度を調べたりする作業のこと。kの値は、Lよりも短い値にします（つまり、k＜L）。例えばk=6の場合は、②15個のk-merを生成可能。



Introduction4

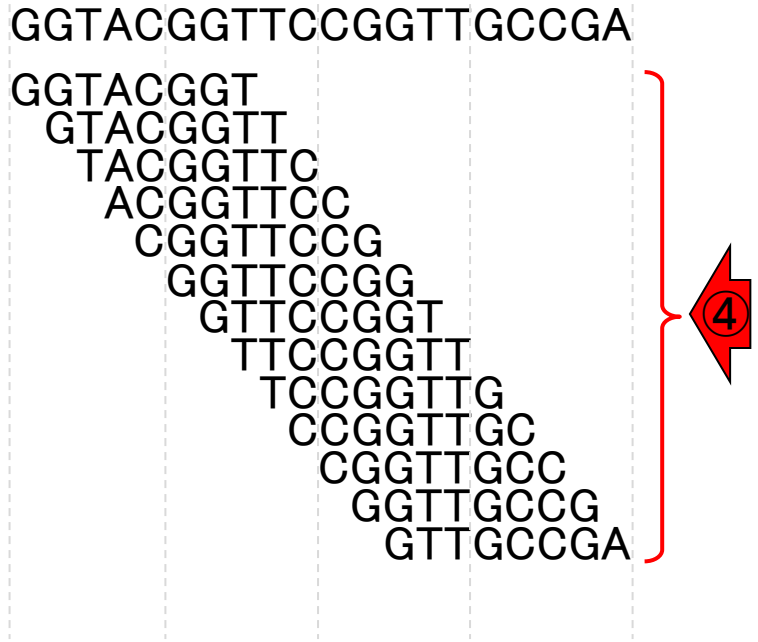
同様にして、③k=8の場合は、④13個のk-merを生成可能。

k=6の場合：15個のk-mer



③

k=8の場合：13個のk-mer



④

Introduction5

同様にして、③k=8の場合は、④13個のk-merを生成可能。また、⑤k=10の場合は11個のk-merを生成可能。

k=6の場合：15個のk-mer

```
GGTACGGTTCCGGTTGCCGA
GGTACG
  GTACGG
    TACGGT
     ACGGTT
      CGGTTC
       GGTTCC
        GTTCCG
         TTCCGG
          TCCGGT
           CCGGTT
            CGGTTG
             GGTTGC
              GTTGCC
               TTGCCG
                TGCCGA
```

k=8の場合：13個のk-mer

```
GGTACGGTTCCGGTTGCCGA
GGTACGGT
  GTACGGTT
    TACGGTTC
     ACGGTTCC
      CGGTTCCG
       GGTTCCGG
        GTTCCGGT
         TTCCGGTT
          TCCGGTTG
           CCGGTTGC
            CGGTTGCC
             GGTTGCCG
              GTTGCCGA
```

k=10の場合：11個のk-mer

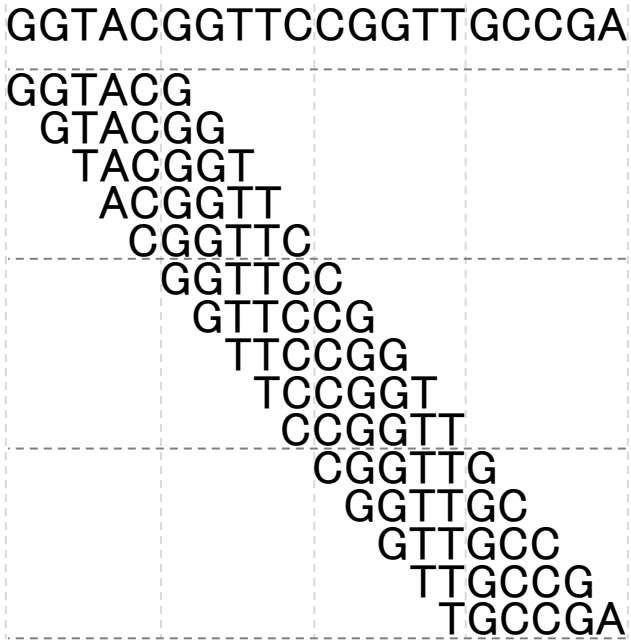
```
GGTACGGTTCCGGTTGCCGA
GGTACGGTTC
  GTACGGTTCC
    TACGGTTCCG
     ACGGTTCCGG
      CGGTTCCGGT
       GGTTCCGGTT
        GTTCCGGTTG
         TTCCGGTTGC
          TCCGGTTGCC
           CCGGTTGCCG
            CGGTTGCCGA
```

Introduction6

一般化すると、「L塩基長のリードをk-merで分割すると、(L − k + 1) 個のk-merを生成可能」です。

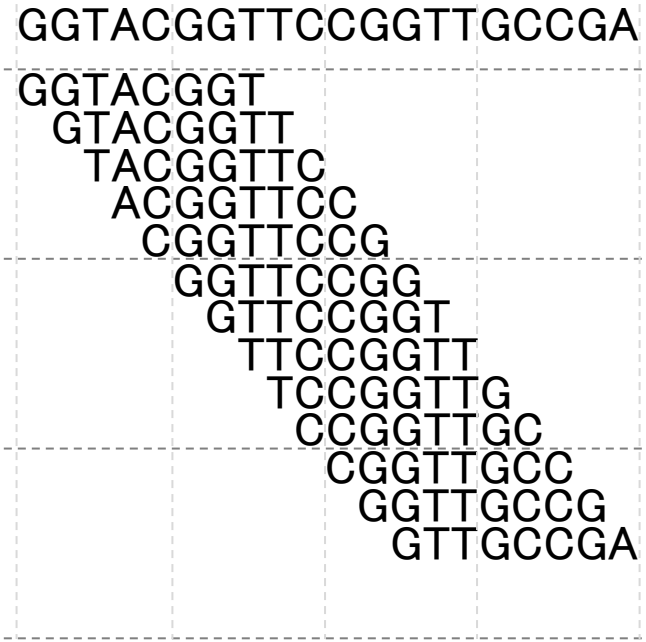
(L − k + 1) = 20 − 6 + 1 = 15

k=6の場合：15個のk-mer



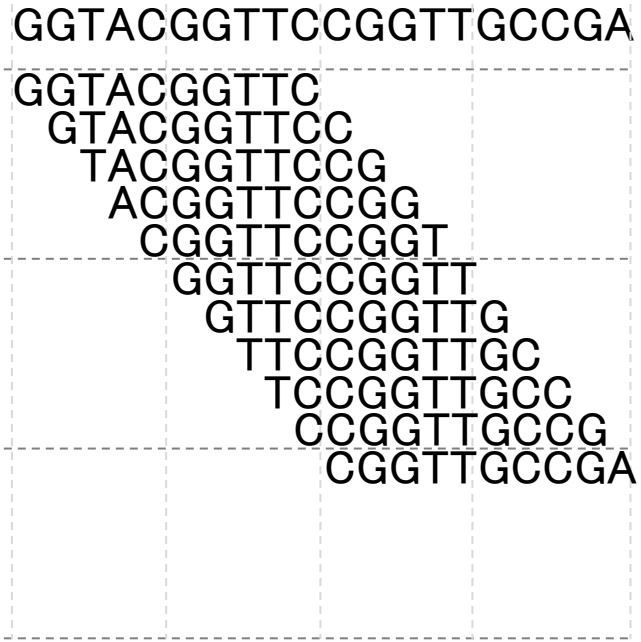
(L − k + 1) = 20 − 8 + 1 = 13

k=8の場合：13個のk-mer



(L − k + 1) = 20 − 10 + 1 = 11

k=10の場合：11個のk-mer



Introduction7

分野によって表現方法が異なりますが、k-spectrum　kernel（k-スペクトラムカーネル）とか、k-mer出現頻度解析みたいな表現もなされており、様々な局面で利用されています。

k=6の場合：15個のk-mer

GGTACGGTTCCGGTTGCCGA

GGTACG
GTACGG
TACGGT
ACGGTT
CGGTTC
GGTTCC
GTTCCG
TTCCGG
TCCGGT
CCGGTT
CGGTTG
GGTTGC
GTTGCC
TTGCCG
TGCCGA

k=8の場合：13個のk-mer

GGTACGGTTCCGGTTGCCGA

GGTACGGT
GTACGGTT
TACGGTTC
ACGGTTCC
CGGTTCCG
GGTTCCGG
GTTCCGGT
TTCCGGTT
TCCGGTTG
CCGGTTGC
CGGTTGCC
GGTTGCCG
GTTGCCGA

k=10の場合：11個のk-mer

GGTACGGTTCCGGTTGCCGA

GGTACGGTTC
GTACGGTTCC
TACGGTTCCG
ACGGTTCCGG
CGGTTCCGGT
GGTTCCGGTT
GTTCCGGTTG
TTCCGGTTGC
TCCGGTTGCC
CCGGTTGCCG
CGGTTGCCGA

Contents

- Introduction、出現頻度解析(k=2)、出現頻度解析(k=1)
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図(例題10)、作図(例題11)、作図(例題12)
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ(例題32)、被覆率(coverage)、基本的な考え方(例題7)
  - 例題8(k=2)、例題9(k=3)、1,000塩基の仮想ゲノム(サンプルデータの例題33)
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

出現頻度解析（k=2）1

例として、k-mer出現頻度解析（k=2）で、①の配列の出現頻度を調べる。ACGTの4種類のみで考えると、②k=2の場合に可能なk-merの種類数は「AA, AC, AG, …, TG, TT」の4^2 = 16。

①

GGTACGGTTCCGGTTGCCGA

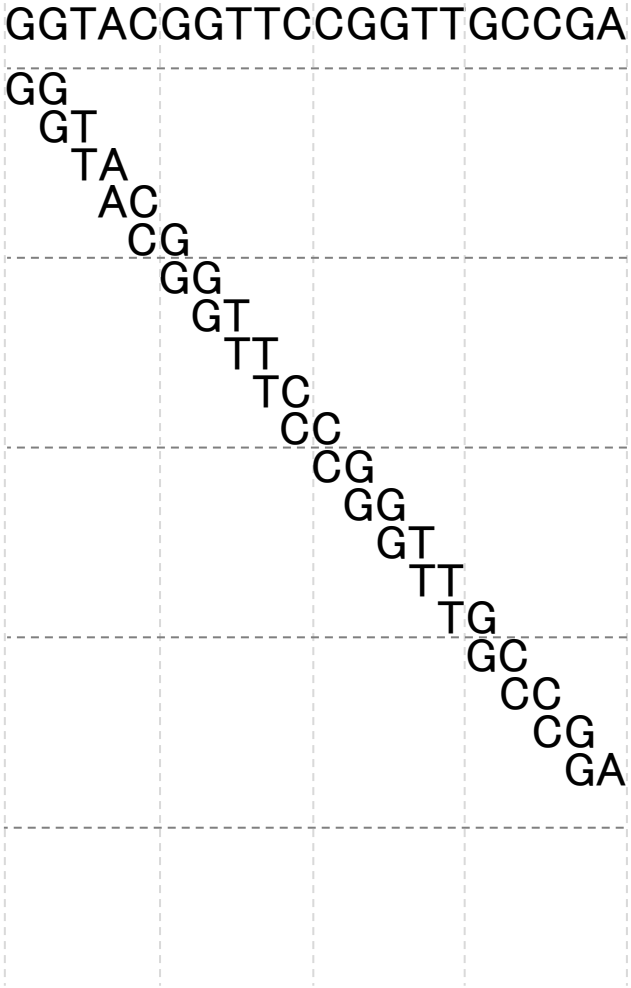
②

AA
AC
AG
AT
CA
CC
CG
CT
GA
GC
GG
GT
TA
TC
TG
TT

出現頻度解析(k=2)2

例として、k-mer出現頻度解析(k=2)で、①の配列の出現頻度を調べる。ACGTの4種類のみで考えると、②k=2の場合に可能なk-merの種類数は「AA, AC, AG, …, TG, TT」の4^2 = 16。③カウント結果。

①

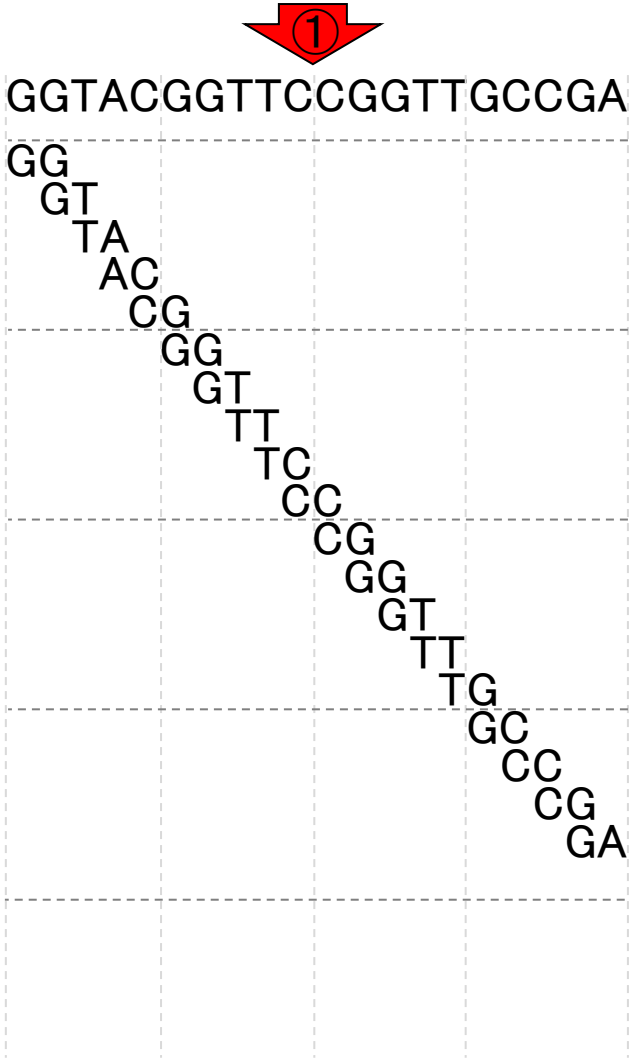


③

AA	0
AC	1
AG	0
AT	0
CA	0
CC	2
CG	3
CT	0
GA	1
GC	1
GG	3
GT	3
TA	1
TC	1
TG	1
TT	2

出現頻度解析(k=2)3

例として、k-mer出現頻度解析(k=2)で、①の配列の出現頻度を調べる。ACGTの4種類のみで考えると、②k=2の場合に可能なk-merの種類数は「AA, AC, AG, …, TG, TT」の4^2 = 16。③カウント結果。例えば④は、「CGという2連続塩基が3回出現した」という風に解釈する。



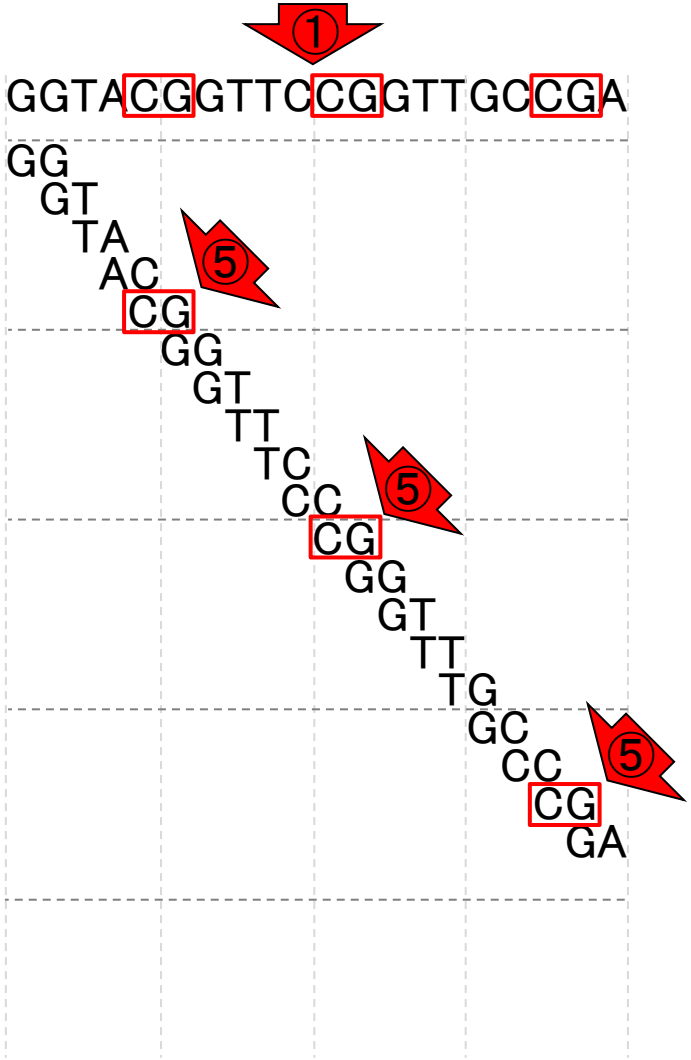
③

AA	0
AC	1
AG	0
AT	0
CA	0
CC	2
CG	3
CT	0
GA	1
GC	1
GG	3
GT	3
TA	1
TC	1
TG	1
TT	2

④

出現頻度解析（k=2）4

例として、k-mer出現頻度解析（k=2）で、①の配列の出現頻度を調べる。ACGTの4種類のみで考えると、②k=2の場合に可能なk-merの種類数は「AA, AC, AG, …, TG, TT」の4^2 = 16。③カウント結果。例えば④は、「CGという2連続塩基が3回出現した」という風に解釈する。確かに、⑤CGが3個ありますね。

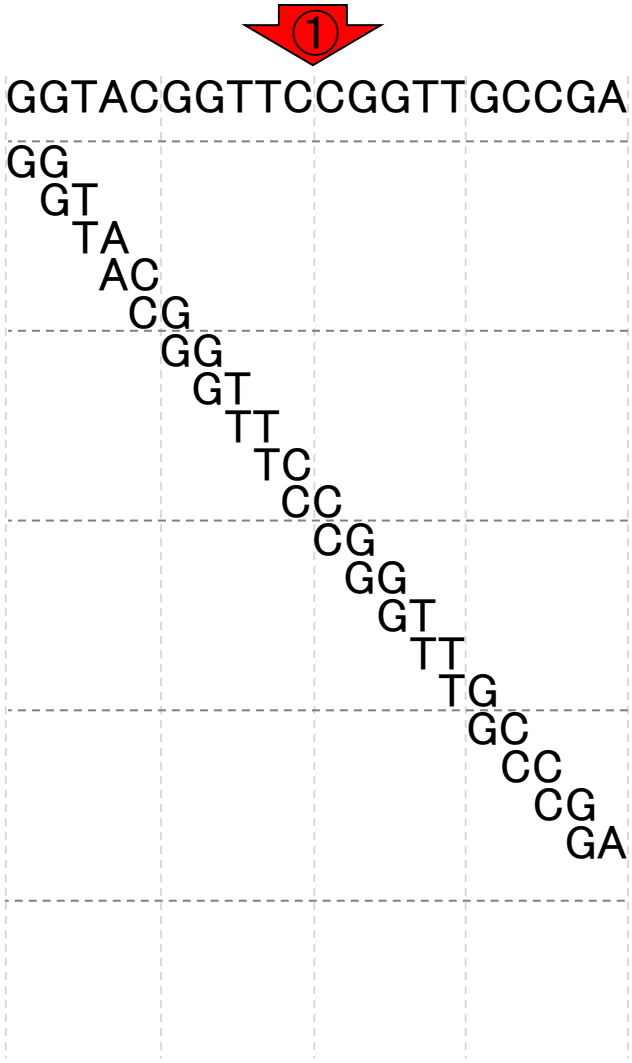


③

AA	0
AC	1
AG	0
AT	0
CA	0
CC	2
CG	3
CT	0
GA	1
GC	1
GG	3
GT	3
TA	1
TC	1
TG	1
TT	2

④

出現頻度解析(k=2)5



③

AA	0
AC	1
AG	0
AT	0
CA	0
CC	2
CG	3
CT	0
GA	1
GC	1
GG	3
GT	3
TA	1
TC	1
TG	1
TT	2

②

例として、k-mer出現頻度解析(k=2)で、①の配列の出現頻度を調べる。ACGTの4種類のみで考えると、②k=2の場合に可能なk-merの種類数は「AA, AC, AG, …, TG, TT」の4^2 = 16。③カウント結果。例えば④は、「CGという2連続塩基が3回出現した」という風に解釈する。確かに、⑤CGが3個ありますね。k-mer出現頻度解析のメリットは、①入力塩基配列の長さに関係なく、常に②一定の要素数からなる、③数値ベクトルが得られること。

出現頻度解析(k=2)6

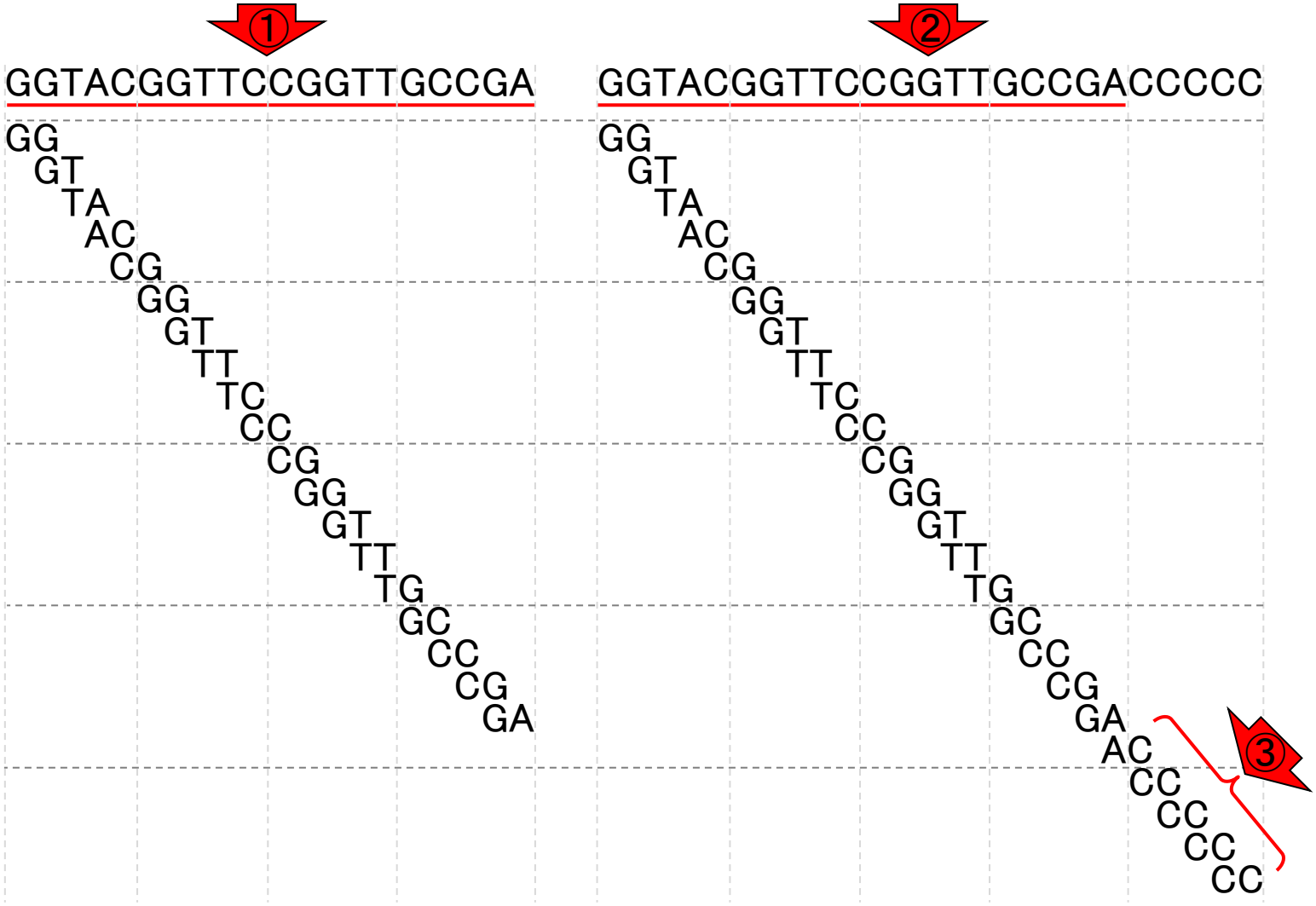
例えば、①と②の2つの配列比較(類似度の評価)が可能。赤下線部分は同一であり、②のほうが5つだけCが多い。

GGTACGGTTCCGGTTGCCGA

GGTACGGTTCCGGTTGCCGACCCCC

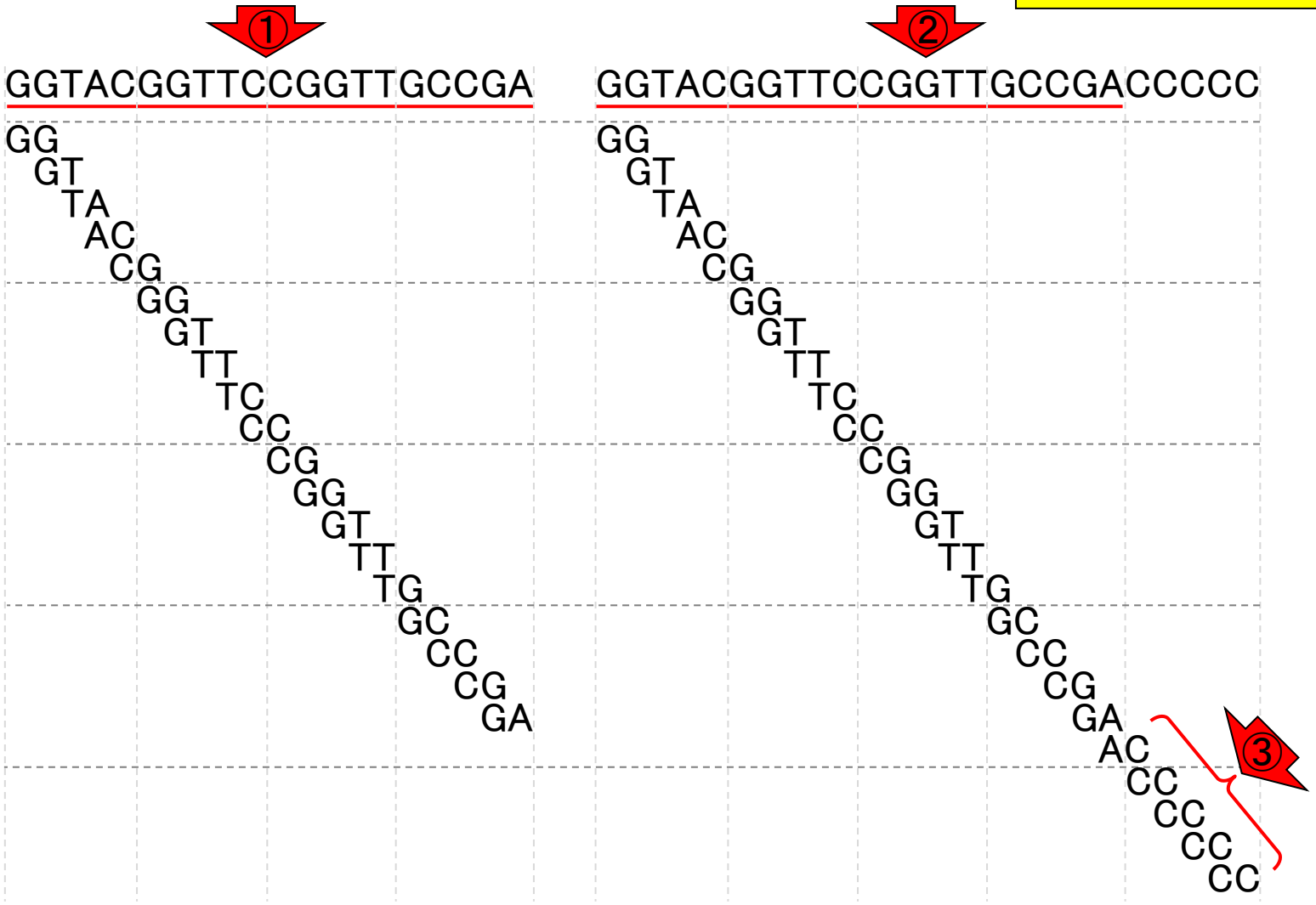
出現頻度解析(k=2)7

例えば、①と②の2つの配列比較(類似度の評価)が可能。赤下線部分は同一であり、②のほうが5つだけCが多い。それゆえ、③の分だけ出現頻度が上乗せされて…



出現頻度解析(k=2)8

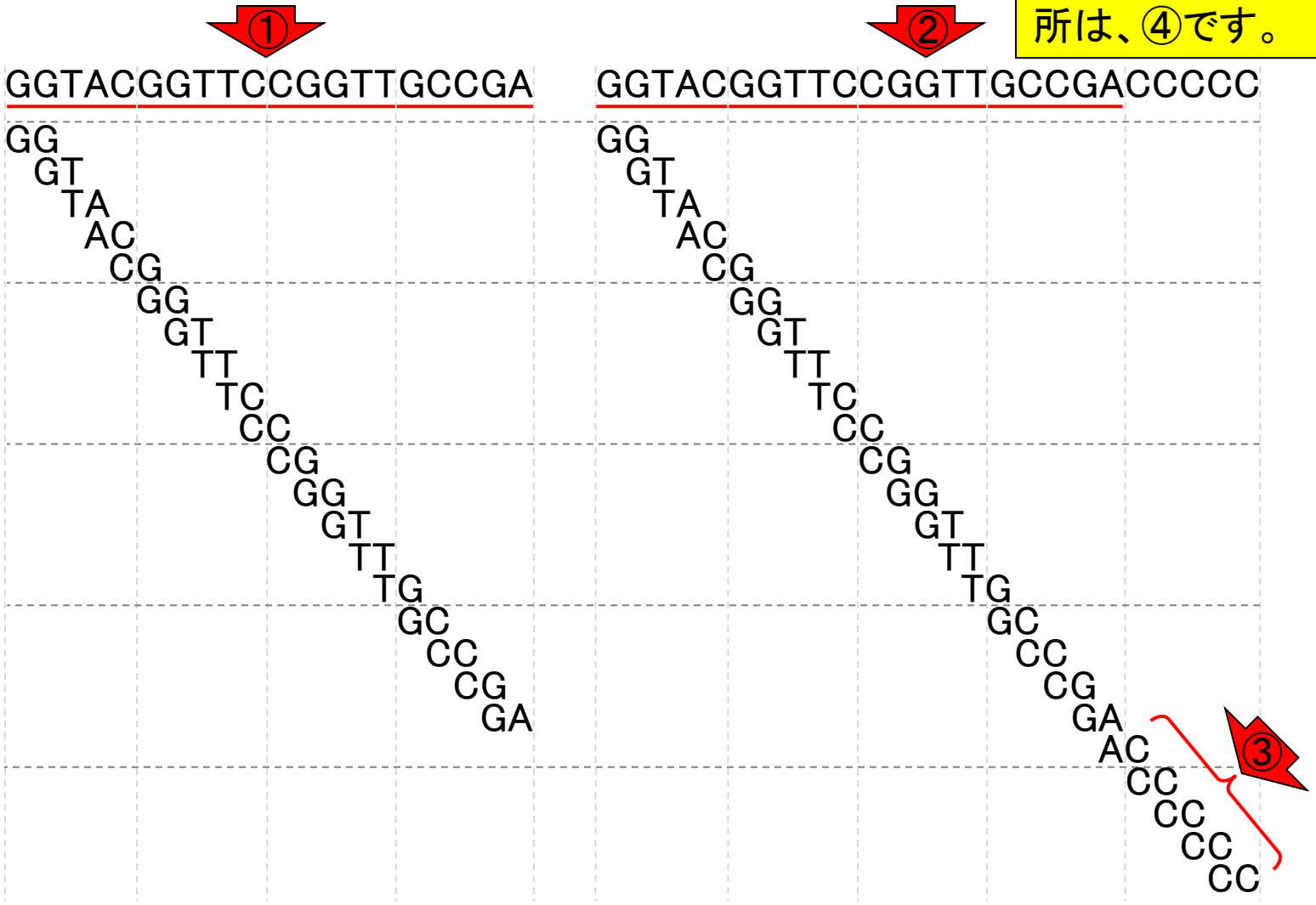
例えば、①と②の2つの配列比較(類似度の評価)が可能。赤下線部分は同一であり、②のほうが5つだけCが多い。それゆえ、③の分だけ出現頻度が上乗せされて、①と②の出現頻度は、それぞれこんな感じになります。



	①	②
AA	0	0
AC	1	2
AG	0	0
AT	0	0
CA	0	0
CC	2	6
CG	3	3
CT	0	0
GA	1	1
GC	1	1
GG	3	3
GT	3	3
TA	1	1
TC	1	1
TG	1	1
TT	2	2

出現頻度解析(k=2)9

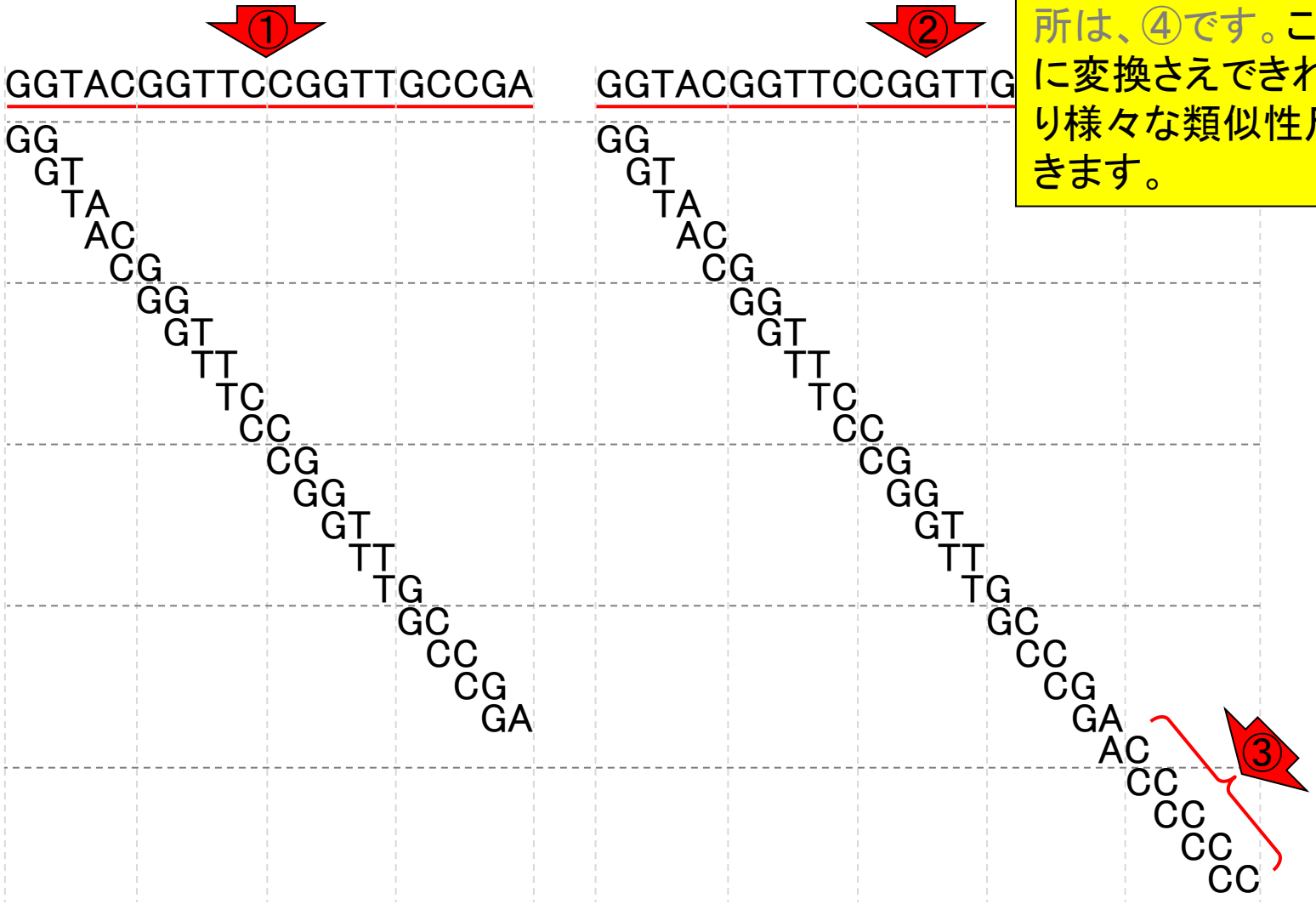
例えば、①と②の2つの配列比較(類似度の評価)が可能。赤下線部分は同一であり、②のほうが5つだけCが多い。それゆえ、③の分だけ出現頻度が上乗せされて、①と②の出現頻度は、それぞれこんな感じになります。③に関連する箇所は、④です。



	①	②	
AA	0	0	
AC	1	2	④
AG	0	0	
AT	0	0	
CA	0	0	
CC	2	6	④
CG	3	3	
CT	0	0	
GA	1	1	
GC	1	1	
GG	3	3	
GT	3	3	
TA	1	1	
TC	1	1	
TG	1	1	
TT	2	2	

出現頻度解析(k=2)10

例えば、①と②の2つの配列比較(類似度の評価)が可能。赤下線部分は同一であり、②のほうが5つだけCが多い。それゆえ、③の分だけ出現頻度が上乗せされて、①と②の出現頻度は、それぞれこんな感じになります。③に関連する箇所は、④です。こんな感じで数値ベクトルに変換さえできれば、あとは相関係数なり様々な類似性尺度を適用することができます。



AG	0	0
AT	0	0
CA	0	0
CC	2	6
CG	3	3
CT	0	0
GA	1	1
GC	1	1
GG	3	3
GT	3	3
TA	1	1
TC	1	1
TG	1	1
TT	2	2

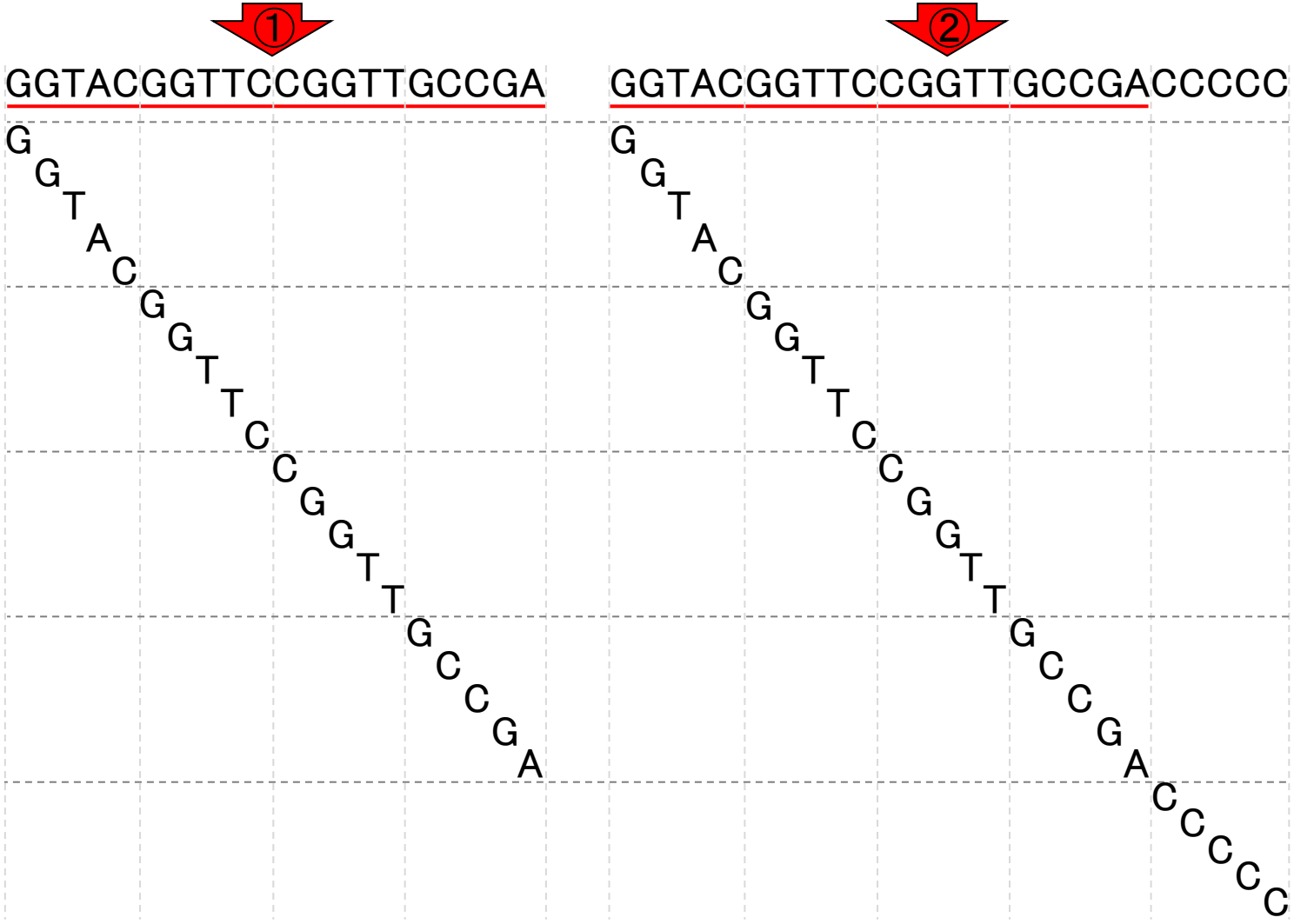
④

Contents

- Introduction、出現頻度解析（k=2）、出現頻度解析（k=1）
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図（例題10）、作図（例題11）、作図（例題12）
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ（例題32）、被覆率（coverage）、基本的な考え方(例題7)
  - 例題8（k=2)、例題9（k=3）、1,000塩基の仮想ゲノム（サンプルデータの例題33）
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

出現頻度解析(k=1)1

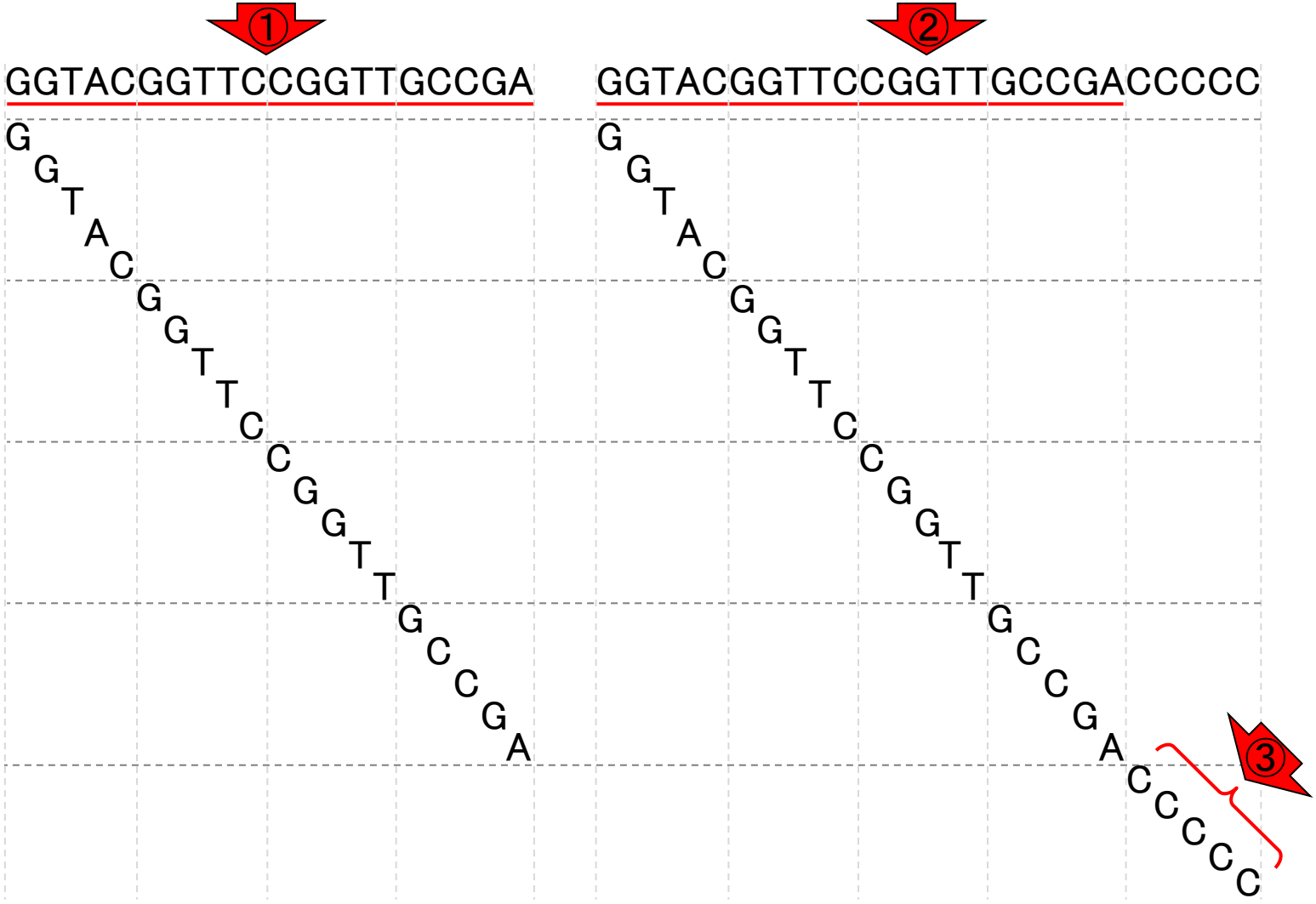
k=1でのk-mer出現頻度解析は、実質的に塩基ごとの出現回数をカウントする作業と同じです。



	①	②
A	2	2
C	5	10
G	8	8
T	5	5

出現頻度解析(k=1)2

k=1でのk-mer出現頻度解析は、実質的に塩基ごとの出現回数をカウントする作業と同じです。③に関連する箇所は、④です。

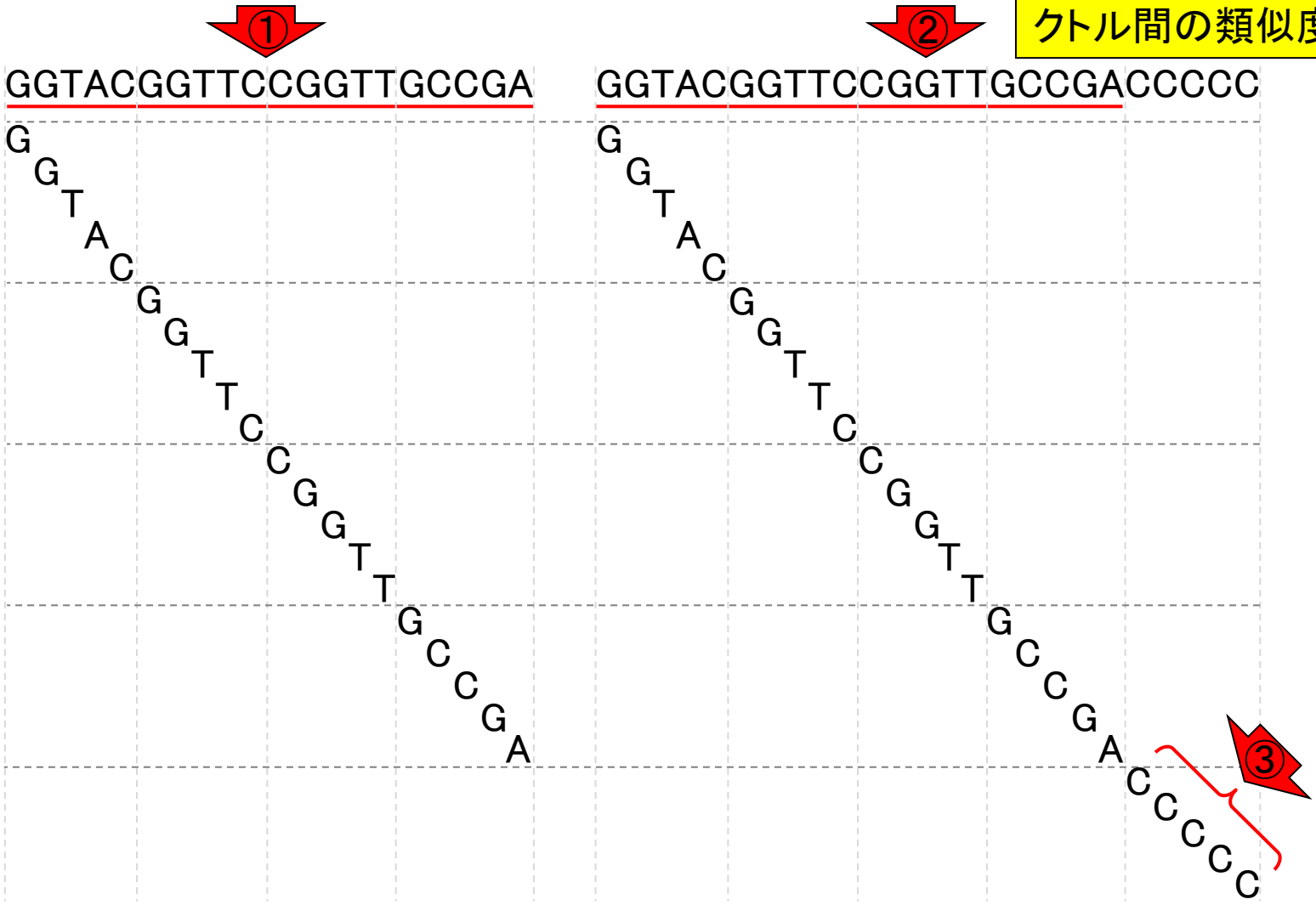


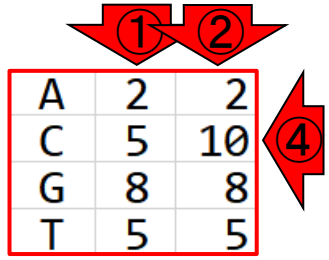
	①	②
A	2	2
C	5	10
G	8	8
T	5	5

④

出現頻度解析(k=1)3

k=1でのk-mer出現頻度解析は、実質的に塩基ごとの出現回数をカウントする作業と同じです。③に関連する箇所は、④です。この場合も、①と②それぞれ要素数が4で固定の数値ベクトルが生成されるので、配列長が異なっていても数値ベクトル間の類似度を評価可能。





Contents

- Introduction、出現頻度解析（k=2）、出現頻度解析（k=1）
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図（例題10）、作図（例題11）、作図（例題12）
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ（例題32）、被覆率（coverage）、基本的な考え方(例題7)
  - 例題8（k=2)、例題9（k=3）、1,000塩基の仮想ゲノム（サンプルデータの例題33）
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

k=1で実践1

①の配列を入力として、②の出現頻度解析結果を得る作業を行ってみましょう。
①の配列のFASTA形式ファイルは、seq_20.fastaです。

①

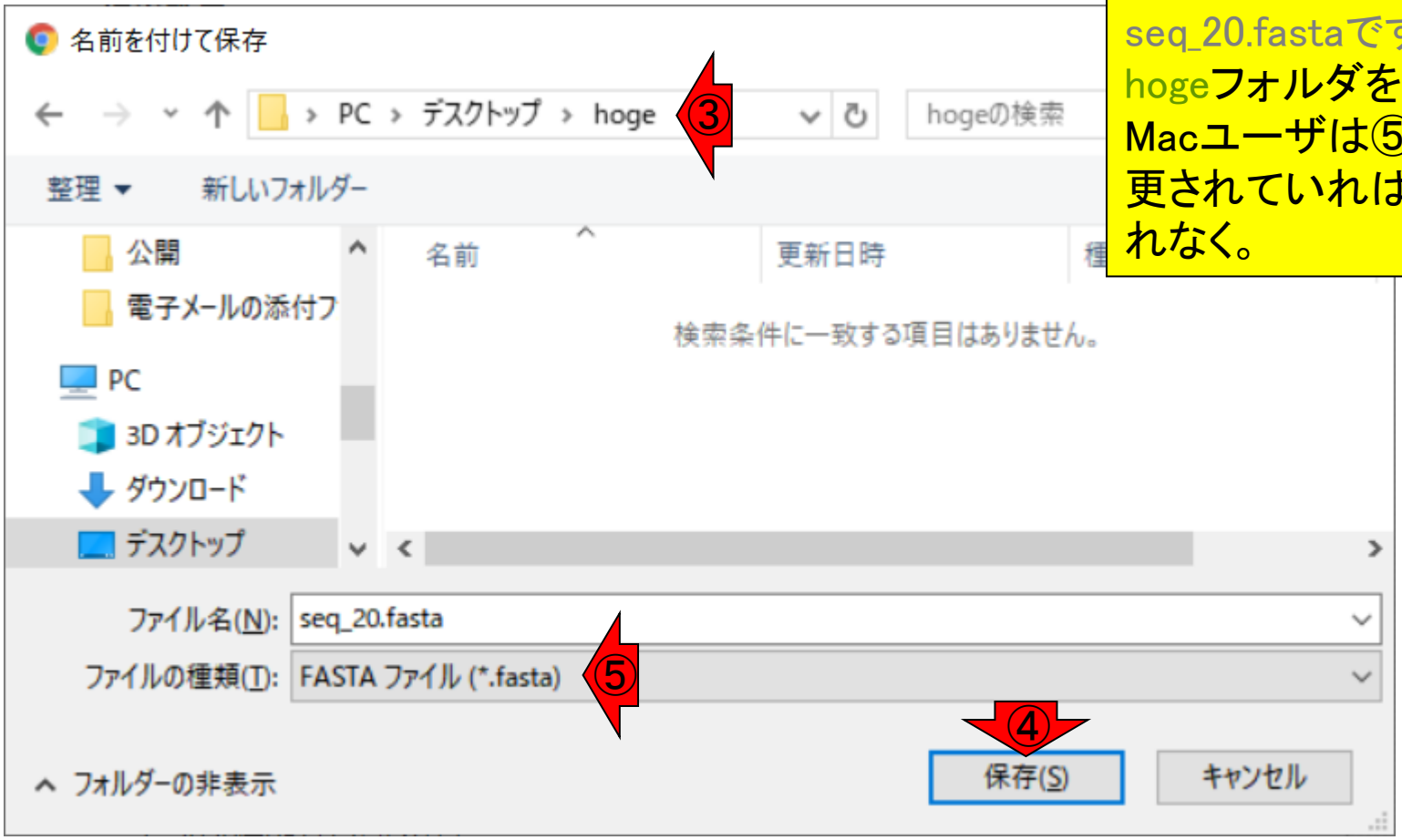
GGTACGGTTCCGGTTGCCGA

②

A	2
C	5
G	8
T	5

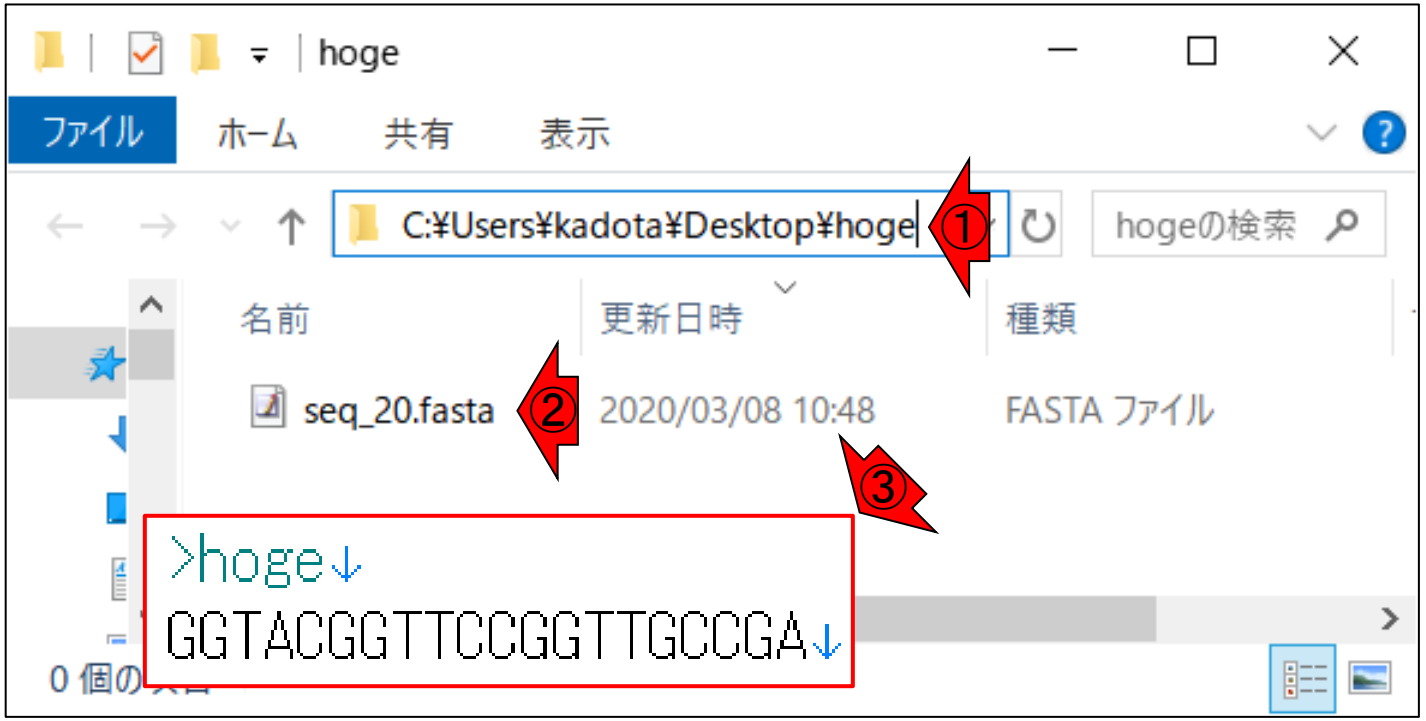
k=1で実践2

①の配列を入力として、②の出現頻度解析結果を得る作業を行ってみましょう。①の配列のFASTA形式ファイルは、seq_20.fastaです。③デスクトップ上にhogeフォルダを作成して④保存。特にMacユーザは⑤拡張子が.txtに勝手に変更されていれば.fastaに戻すことをお忘れなく。



k=1で実践3

①「デスクトップのhogeフォルダ」にある、
②seq_20.fastaの、③中身。

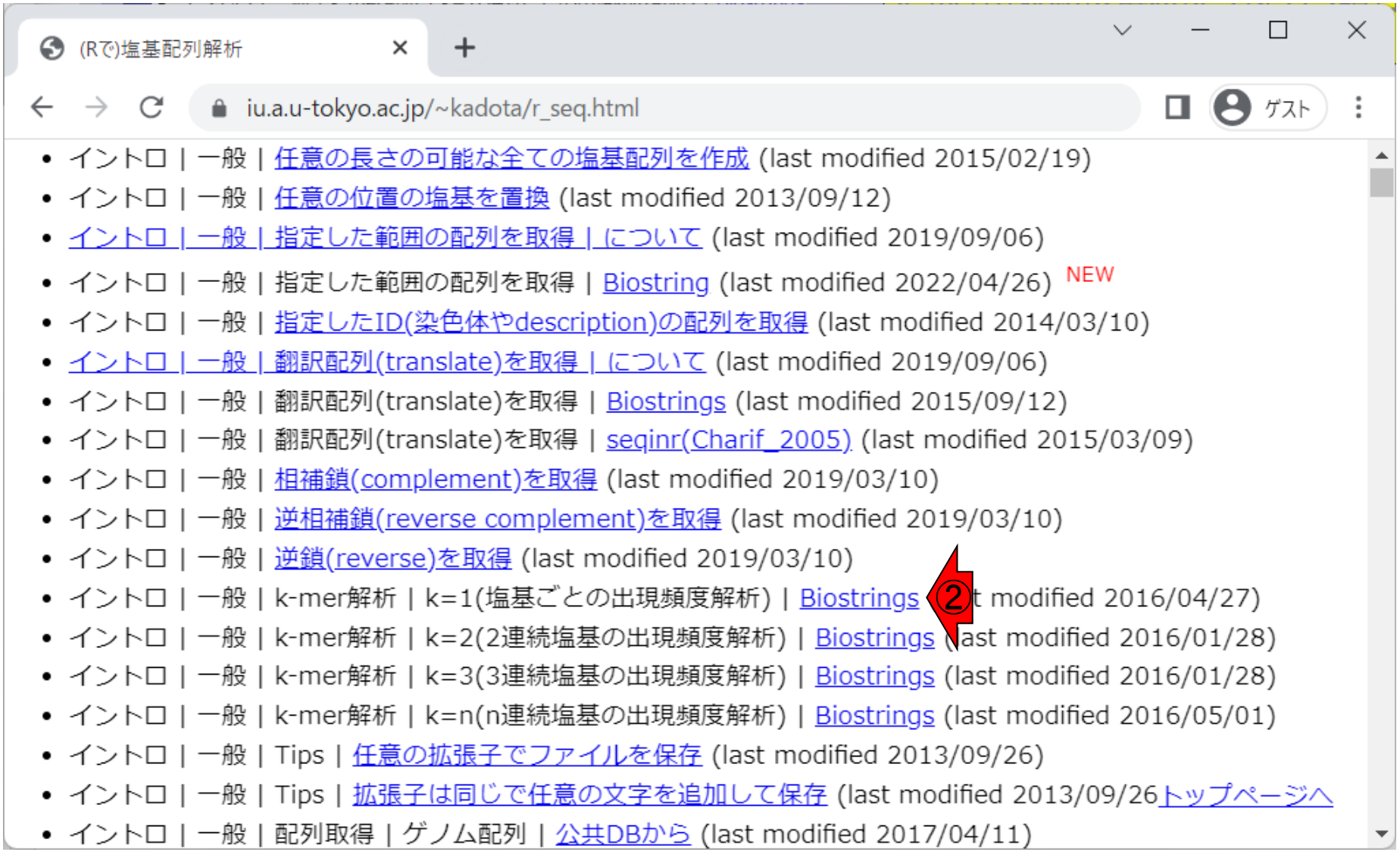


k=1で実践4

k-mer出現頻度解析(k=1)を行う項目は、①の中の…

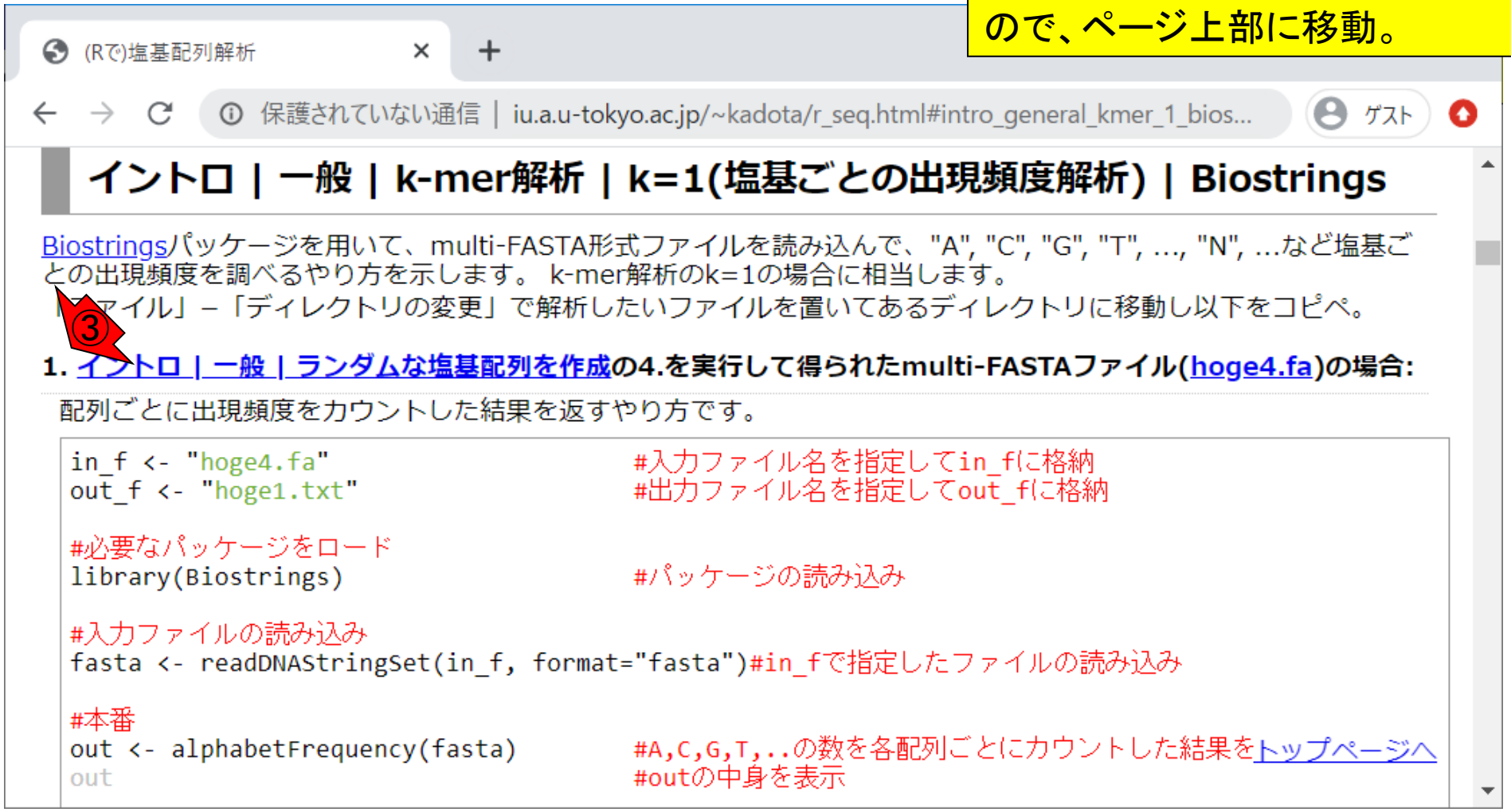
k=1で実践5

k-mer出現頻度解析(k=1)を行う項目は、①の中の、②です。



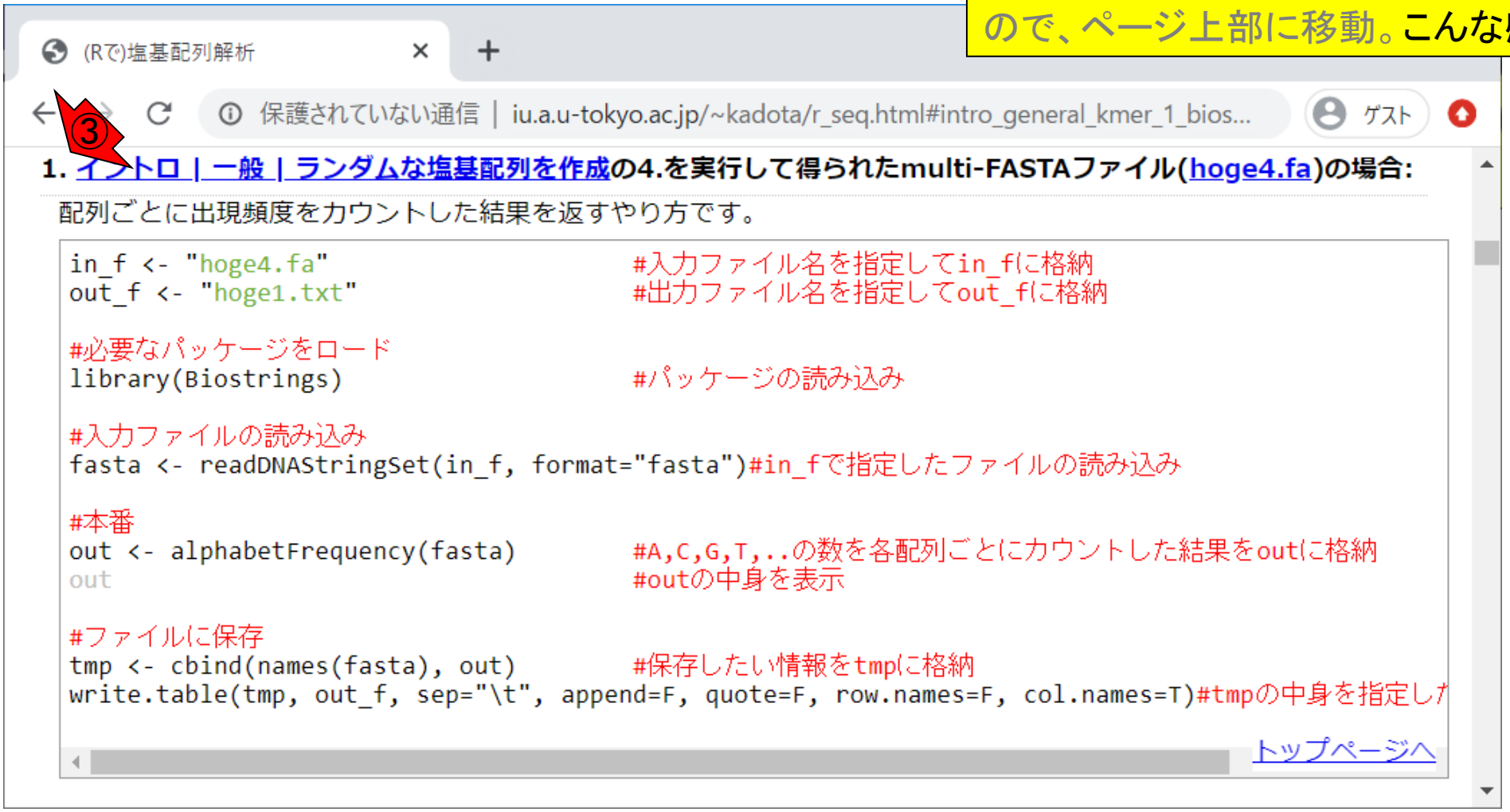
k=1で実践6

k-mer出現頻度解析(k=1)を行う項目は、①の中の、②です。こんな感じになります。③例題1をテンプレートとして用いるので、ページ上部に移動。



k=1で実践7

k-mer出現頻度解析(k=1)を行う項目は、①の中の、②です。こんな感じになります。③例題1をテンプレートとして用いるので、ページ上部に移動。こんな感じ。



1. イントロ | 一般 | ランダムな塩基配列を作成の4.を実行して得られたmulti-FASTAファイル(hoge4.fa)の場合:

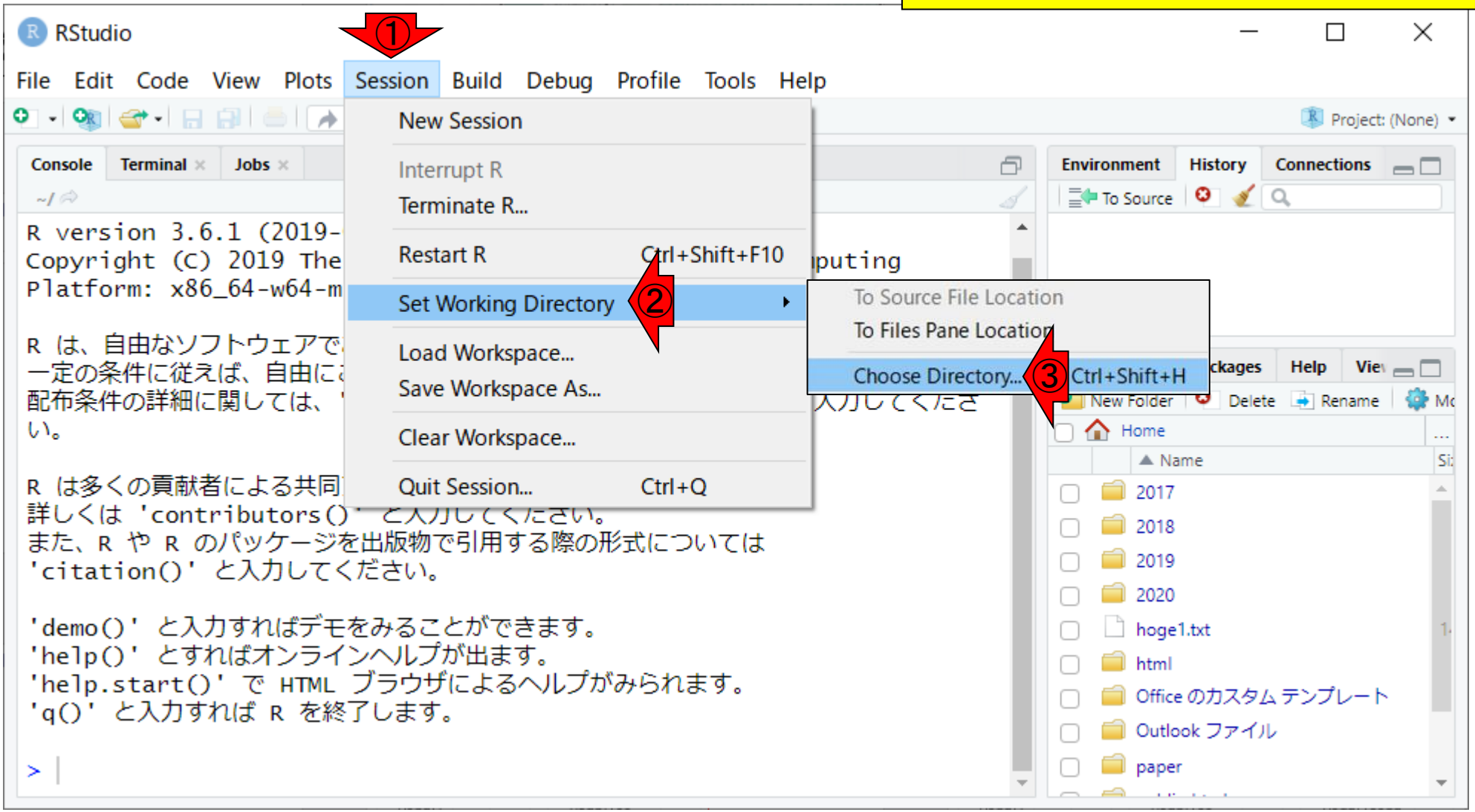
配列ごとに出現頻度をカウントした結果を返すやり方です。

```
in_f <- "hoge4.fa"                          #入力ファイル名を指定してin_fに格納
out_f <- "hoge1.txt"                        #出力ファイル名を指定してout_fに格納

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

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

#本番
out <- alphabetFrequency(fasta)             #A,C,G,T,..の数を各配列ごとにカウントした結果をoutに格納
out                                         #outの中身を表示

#ファイルに保存
tmp <- cbind(names(fasta), out)             #保存したい情報をtmpに格納
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F, col.names=T)#tmpの中身を指定した
```

トップページへ

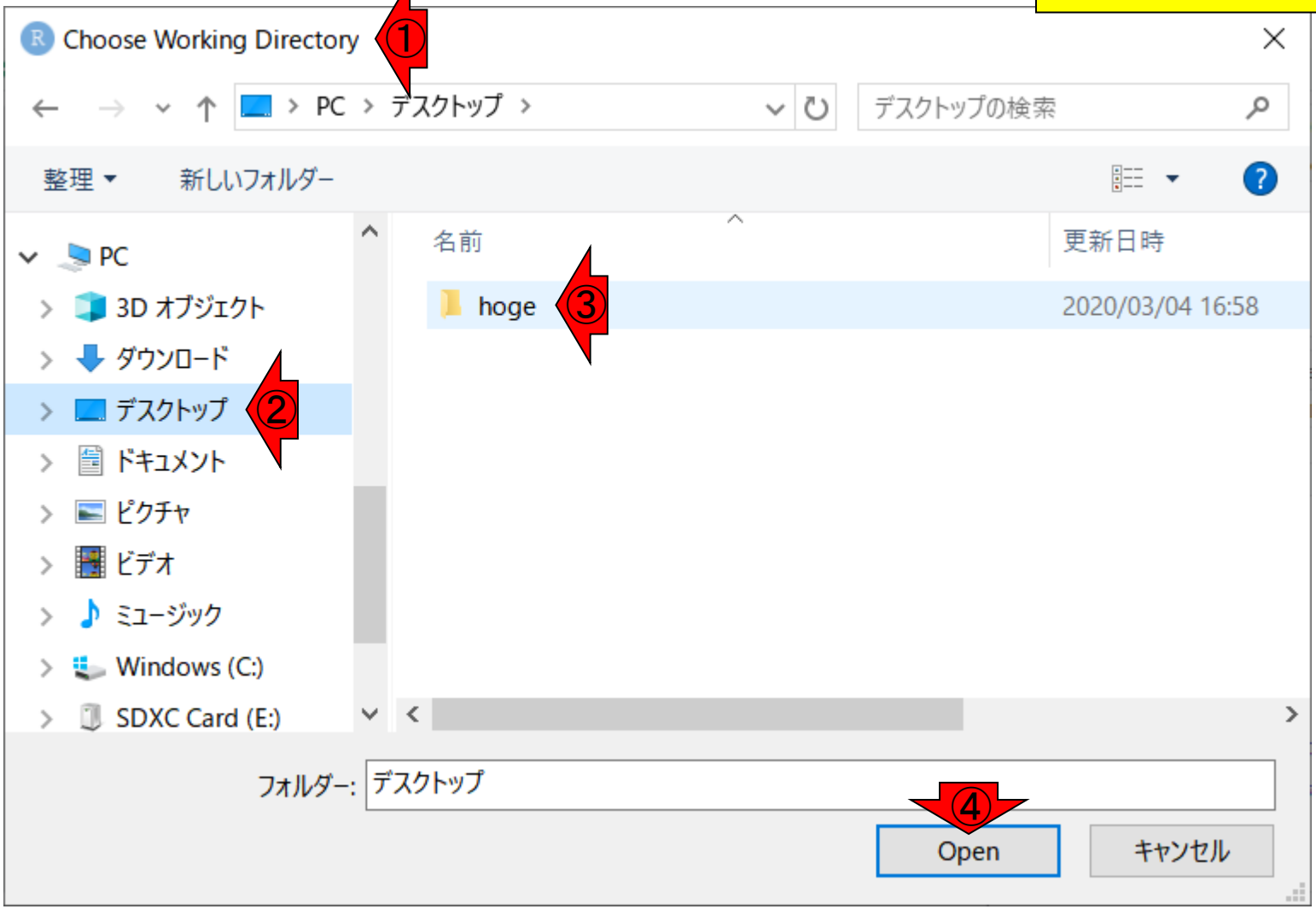
k=1で実践8

作業ディレクトリを「デスクトップのhoge」にすべく、①Session、②Set Working Directory、③Choose Directory…。



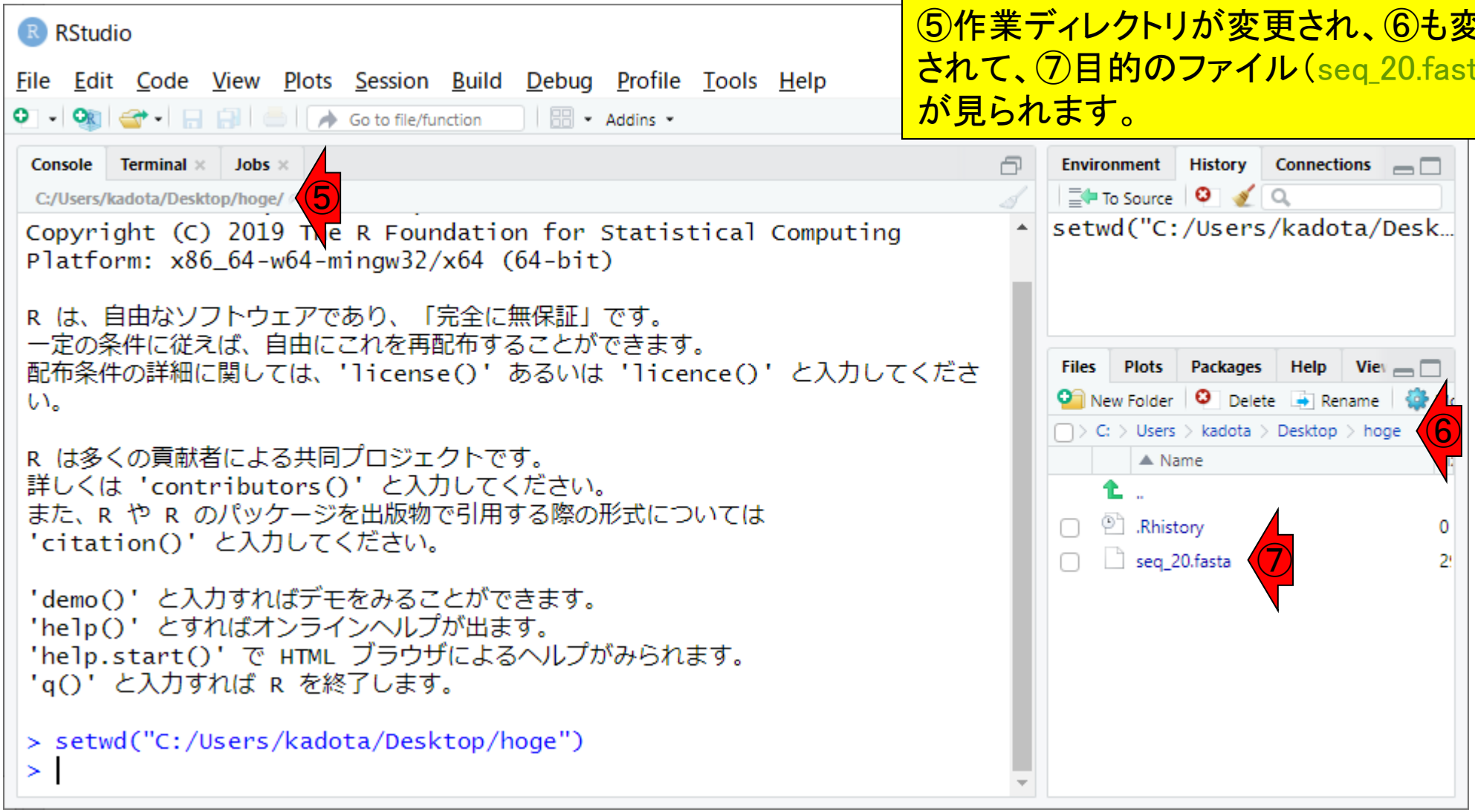
k=1で実践9

①こんな感じのフォルダ選択画面が出ますので、②③④のような感じで変更を完了してください。



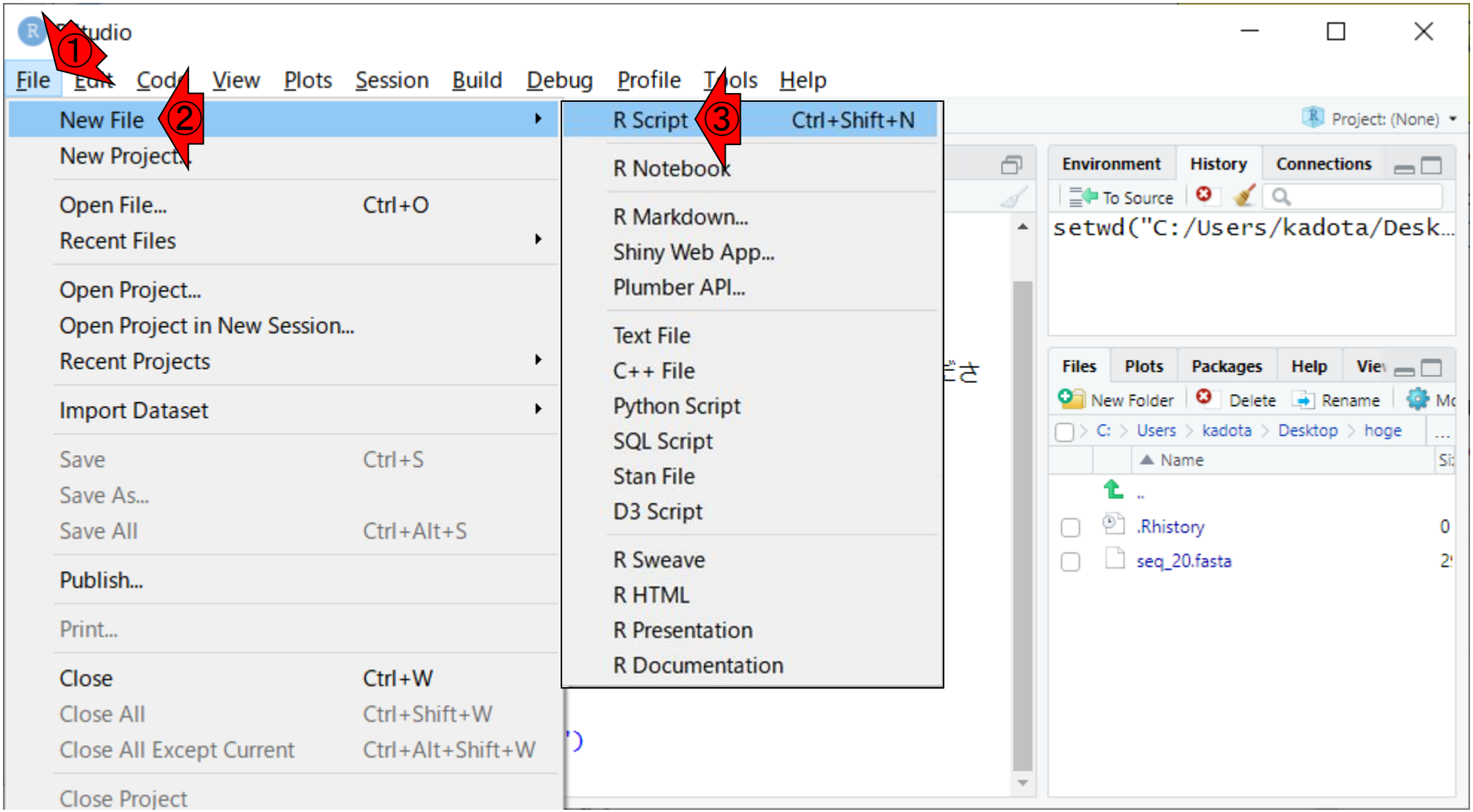
k=1で実践10

①こんな感じのフォルダ選択画面が出ますので、②③④のような感じで変更を完了してください。こんな感じになります。
⑤作業ディレクトリが変更され、⑥も変更されて、⑦目的のファイル（seq_20.fasta）が見られます。



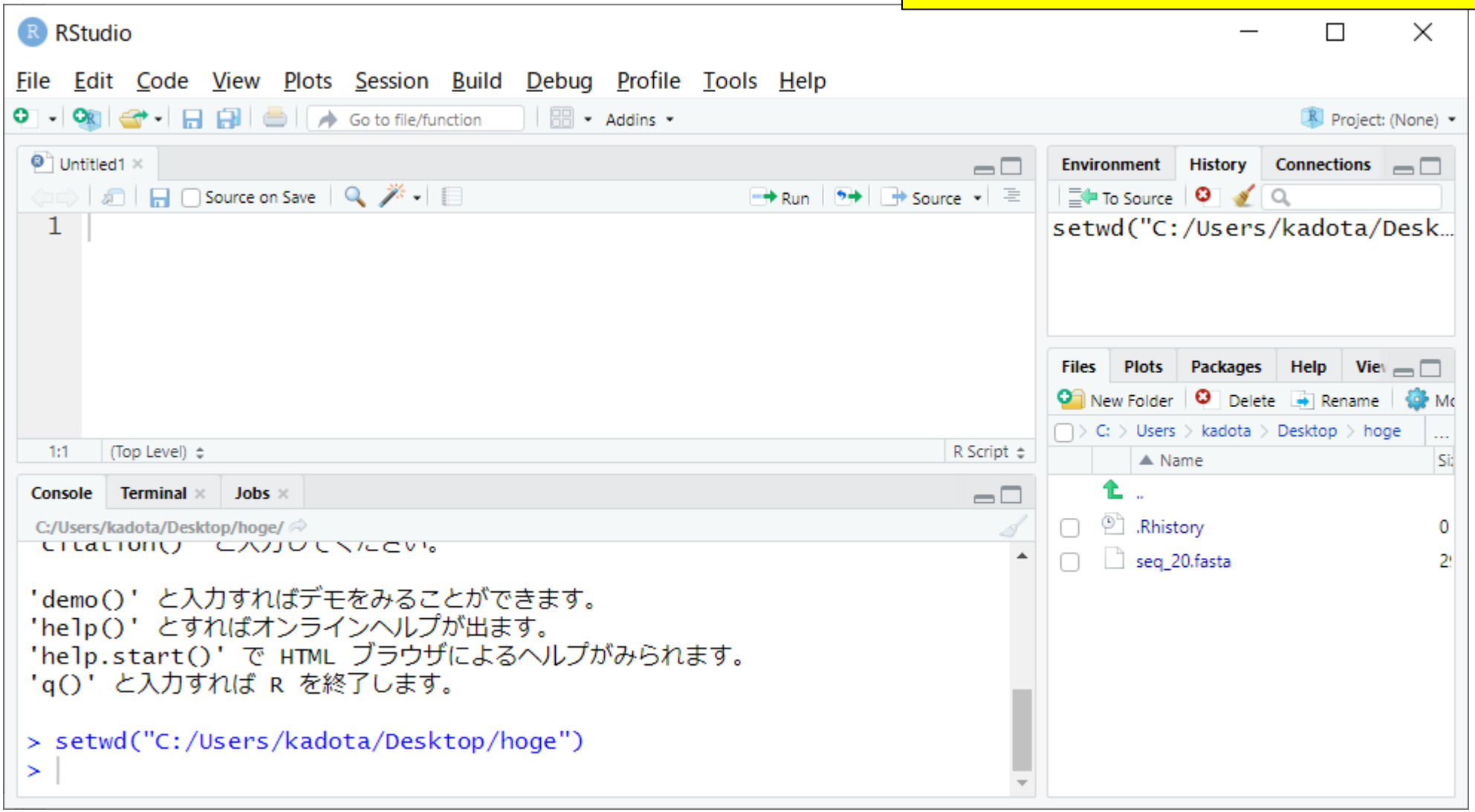
k=1で実践11

①File、②New File、③R Scriptとして、
「新規のRスクリプト」を開く。



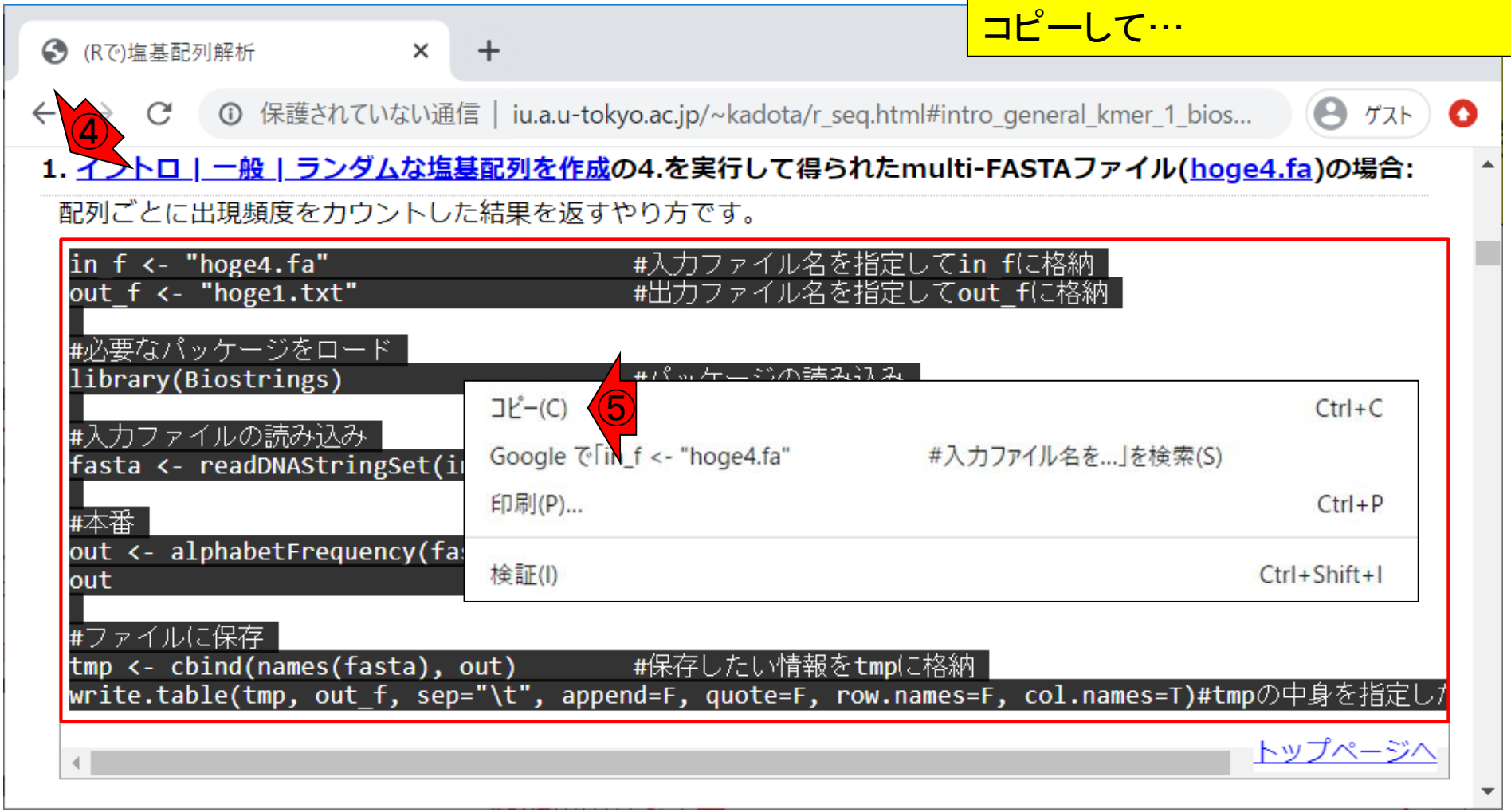
k=1で実践12

①File、②New File、③R Scriptとして、「新規のRスクリプト」を開く。こんな感じになります。



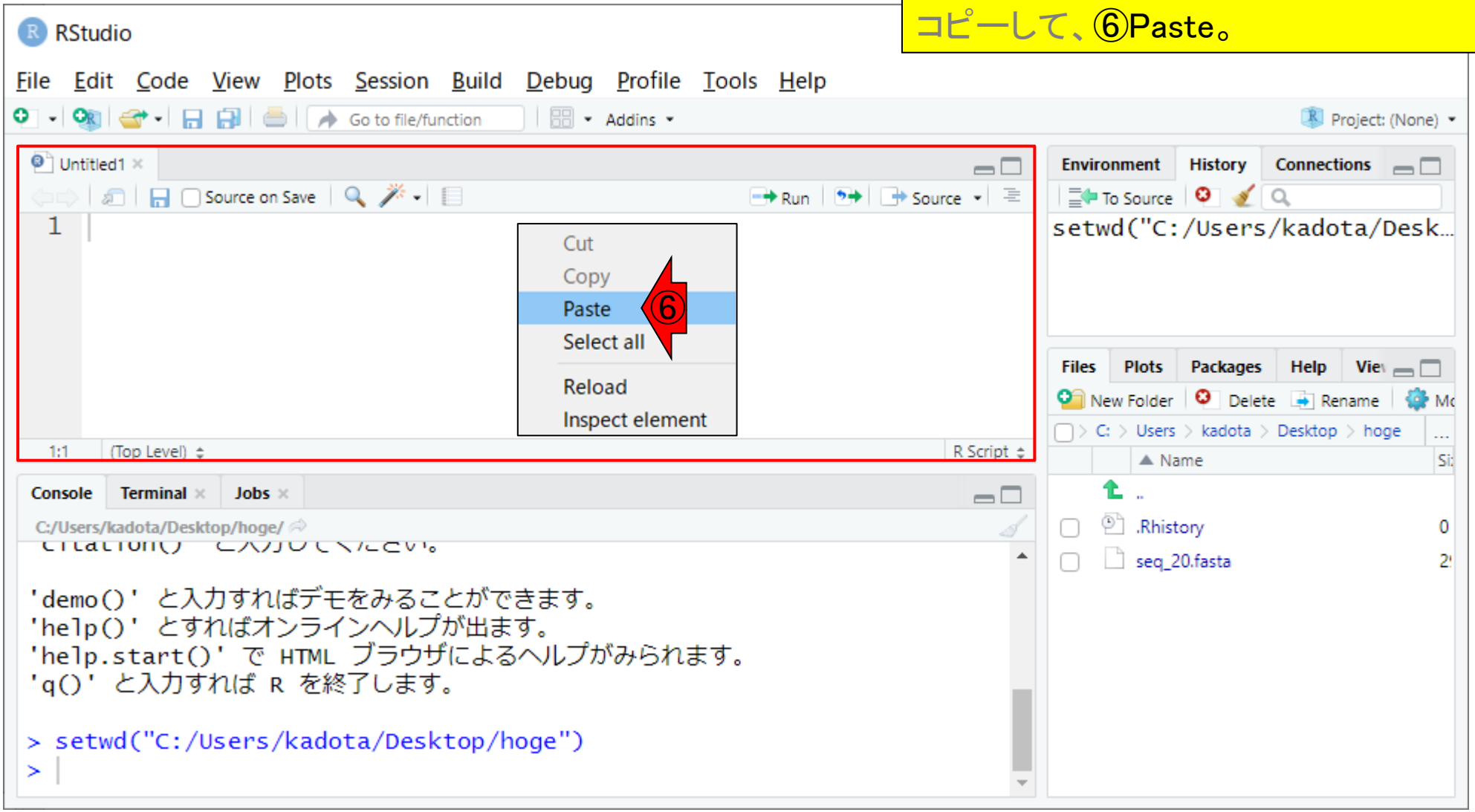
k=1で実践13

①File、②New File、③R Scriptとして、「新規のRスクリプト」を開く。こんな感じになります。④例題1のコード全体を、⑤コピーして…



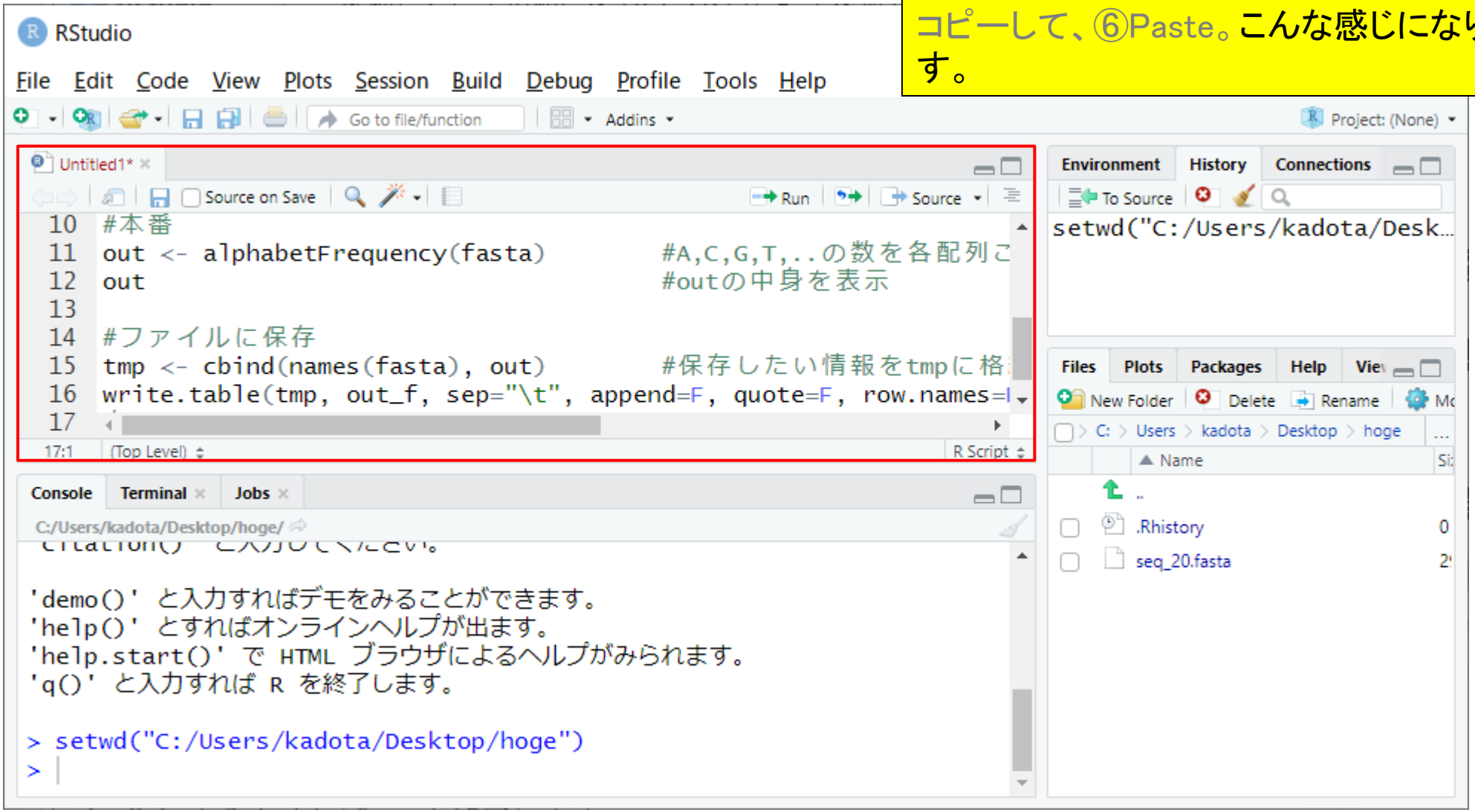
k=1で実践14

①File、②New File、③R Scriptとして、「新規のRスクリプト」を開く。こんな感じになります。④例題1のコード全体を、⑤コピーして、⑥Paste。



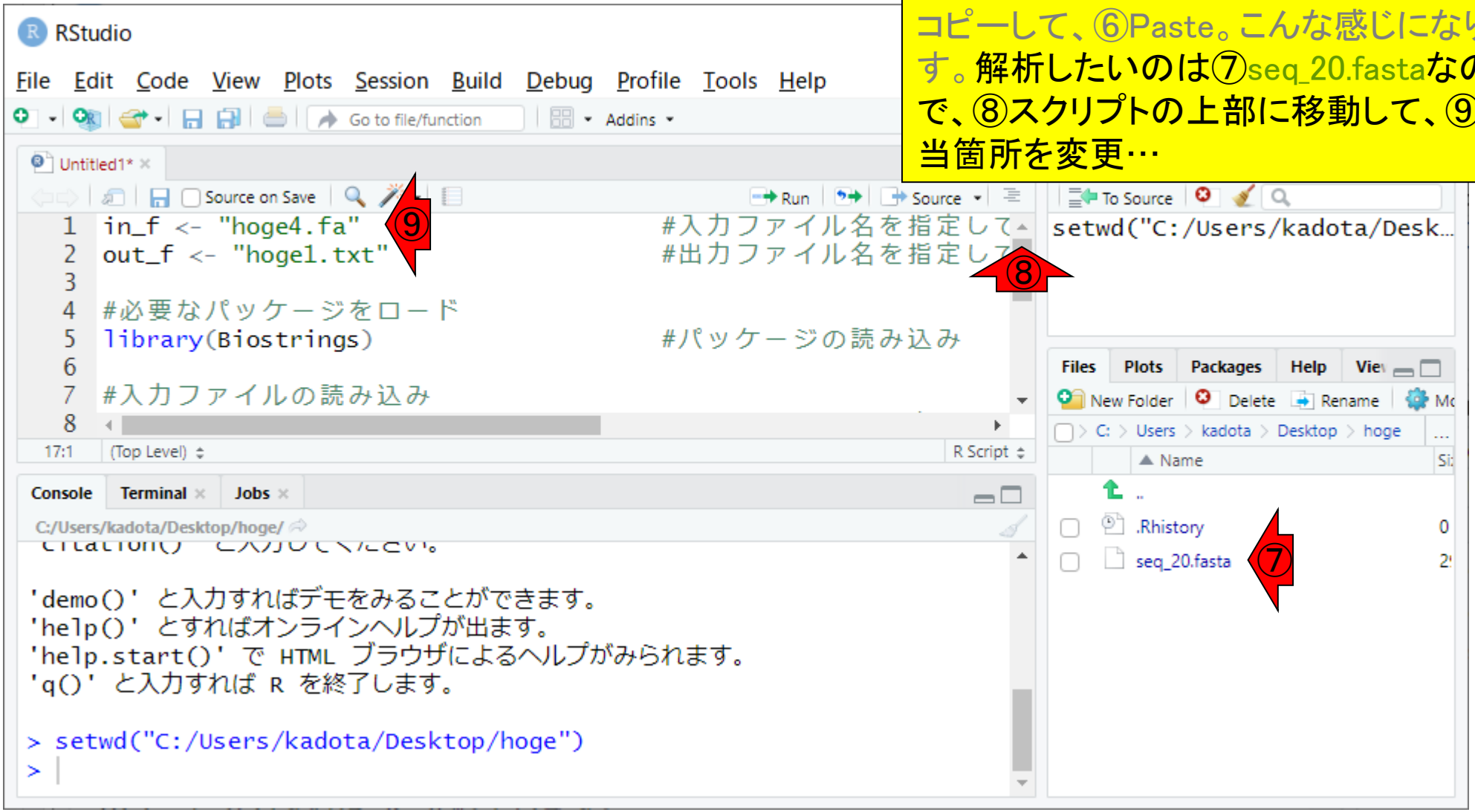
k=1で実践15

①File、②New File、③R Scriptとして、「新規のRスクリプト」を開く。こんな感じになります。④例題1のコード全体を、⑤コピーして、⑥Paste。こんな感じになります。



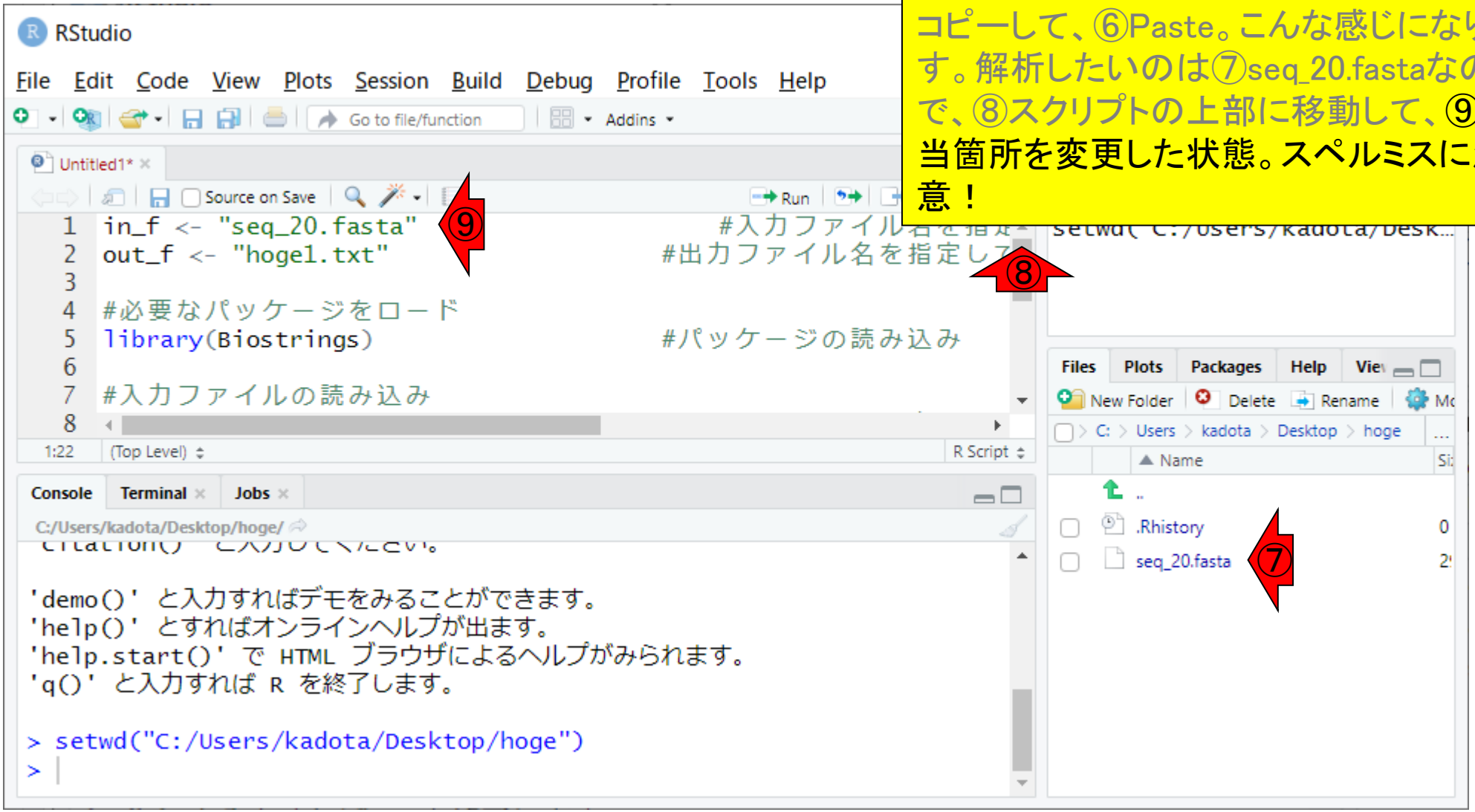
k=1で実践16

①File、②New File、③R Scriptとして、「新規のRスクリプト」を開く。こんな感じになります。④例題1のコード全体を、⑤コピーして、⑥Paste。こんな感じになります。解析したいのは⑦seq_20.fastaなので、⑧スクリプトの上部に移動して、⑨該当箇所を変更…



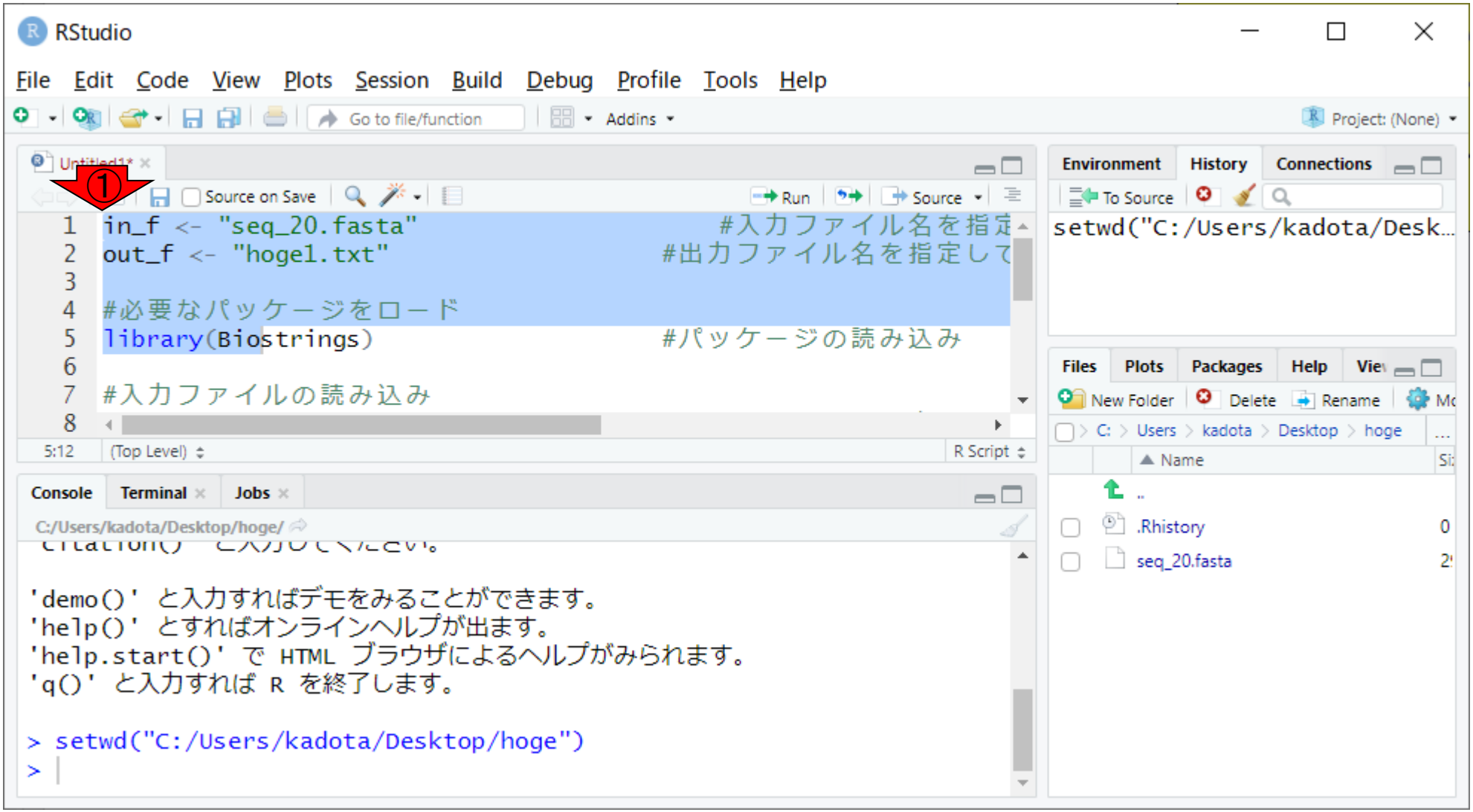
k=1で実践17

①File、②New File、③R Scriptとして、「新規のRスクリプト」を開く。こんな感じになります。④例題1のコード全体を、⑤コピーして、⑥Paste。こんな感じになります。解析したいのは⑦seq_20.fastaなので、⑧スクリプトの上部に移動して、⑨該当箇所を変更した状態。スペルミスに注意！



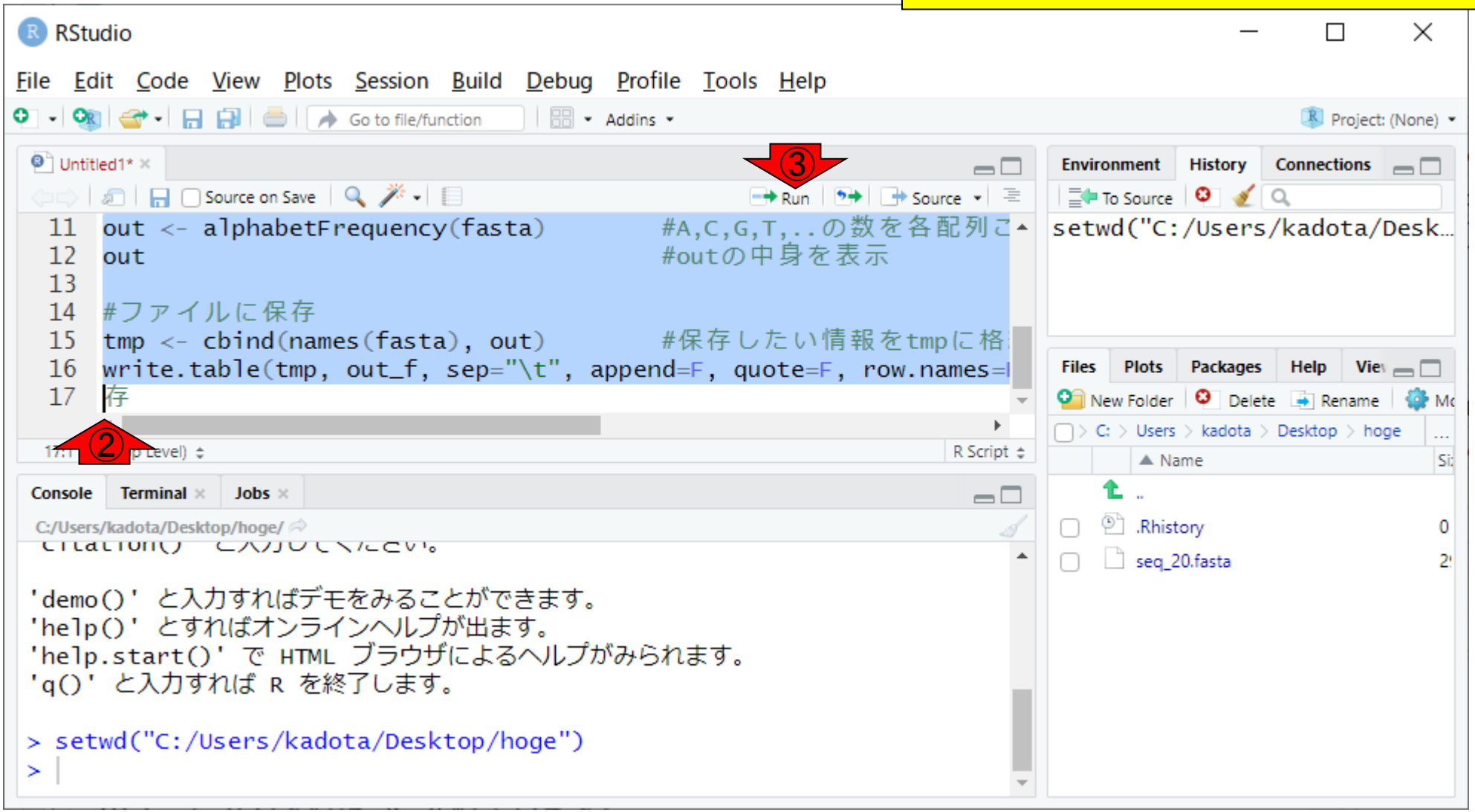
k=1で実践18

変更後のコード全体をコピペ実行すべく、
①左上から…



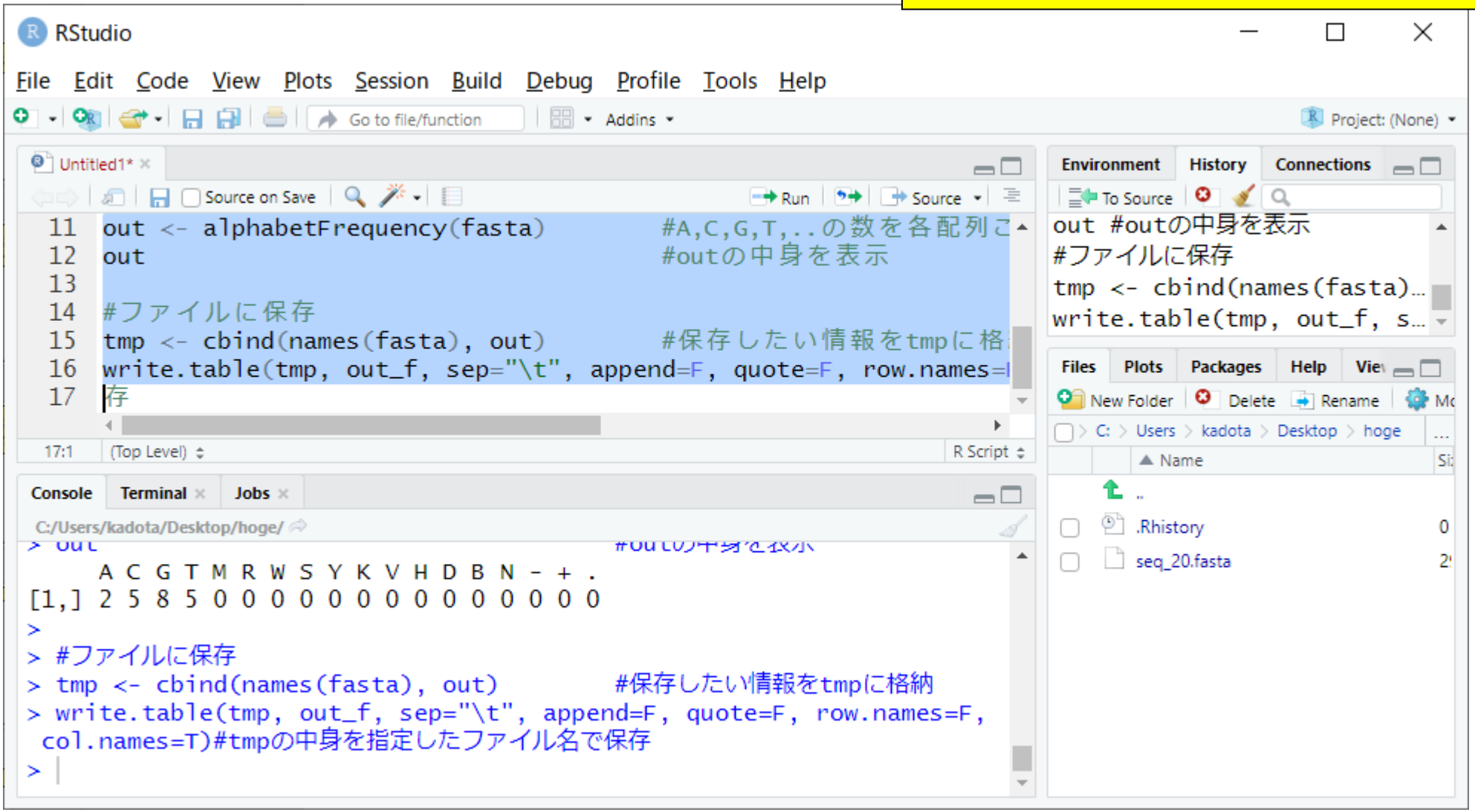
k=1で実践19

変更後のコード全体をコピペ実行すべく、
①左上から、②一番下まで反転させて、
③Run。



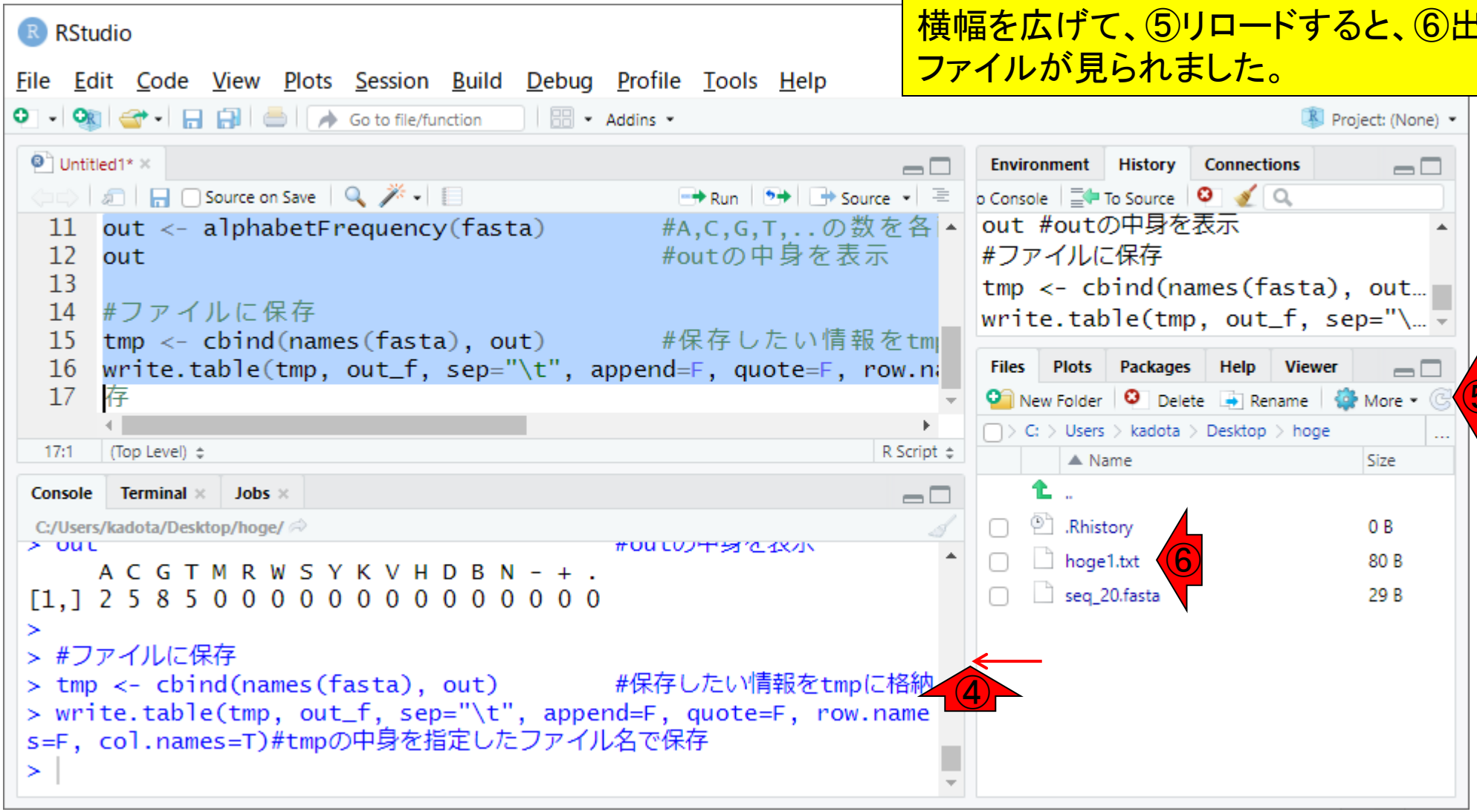
k=1で実践20

変更後のコード全体をコピペ実行すべく、
①左上から、②一番下まで反転させて、
③Run。実行完了。



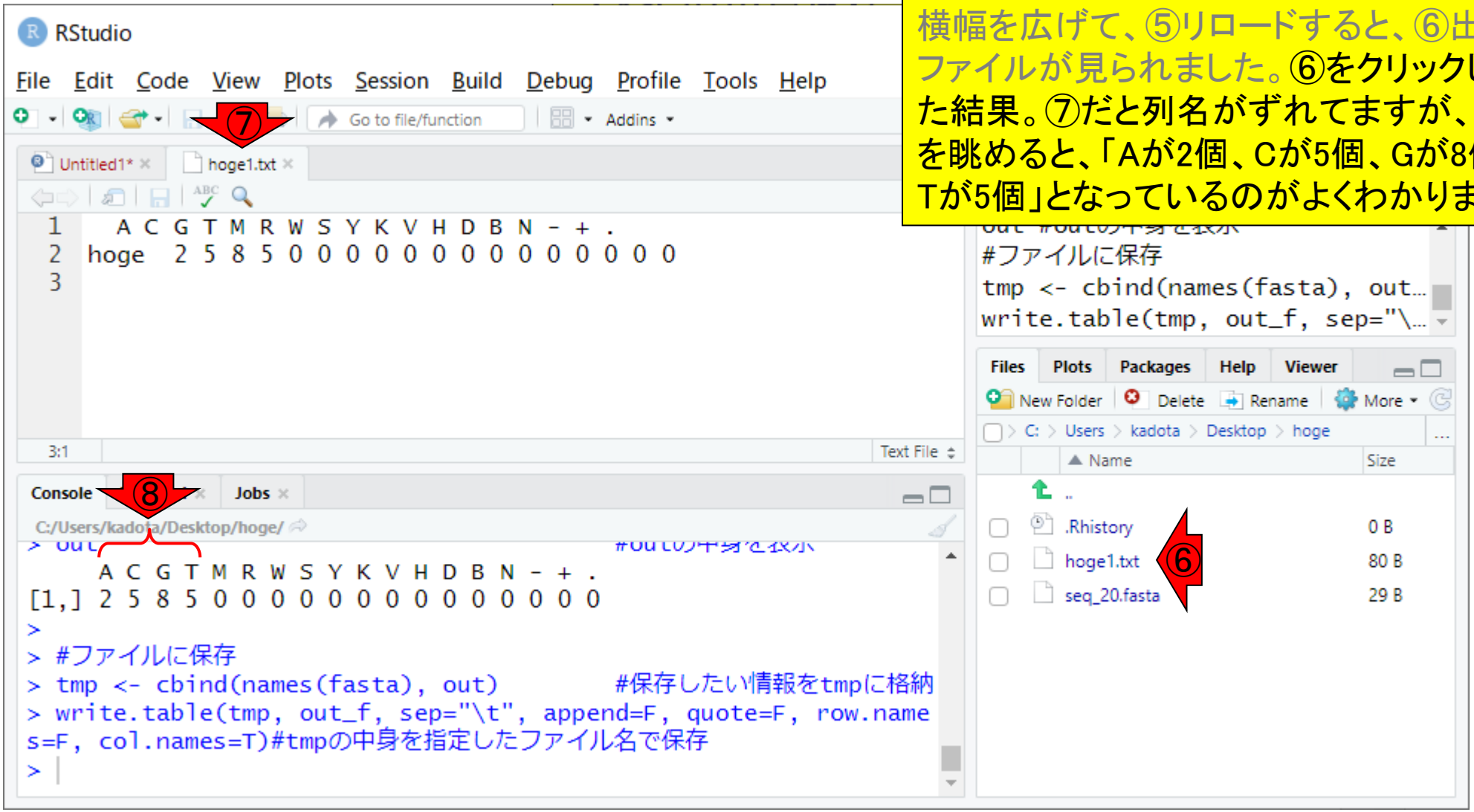
k=1で実践21

変更後のコード全体をコピペ実行すべく、①左上から、②一番下まで反転させて、③Run。実行完了。私は、④少し右側の横幅を広げて、⑤リロードすると、⑥出力ファイルが見られました。



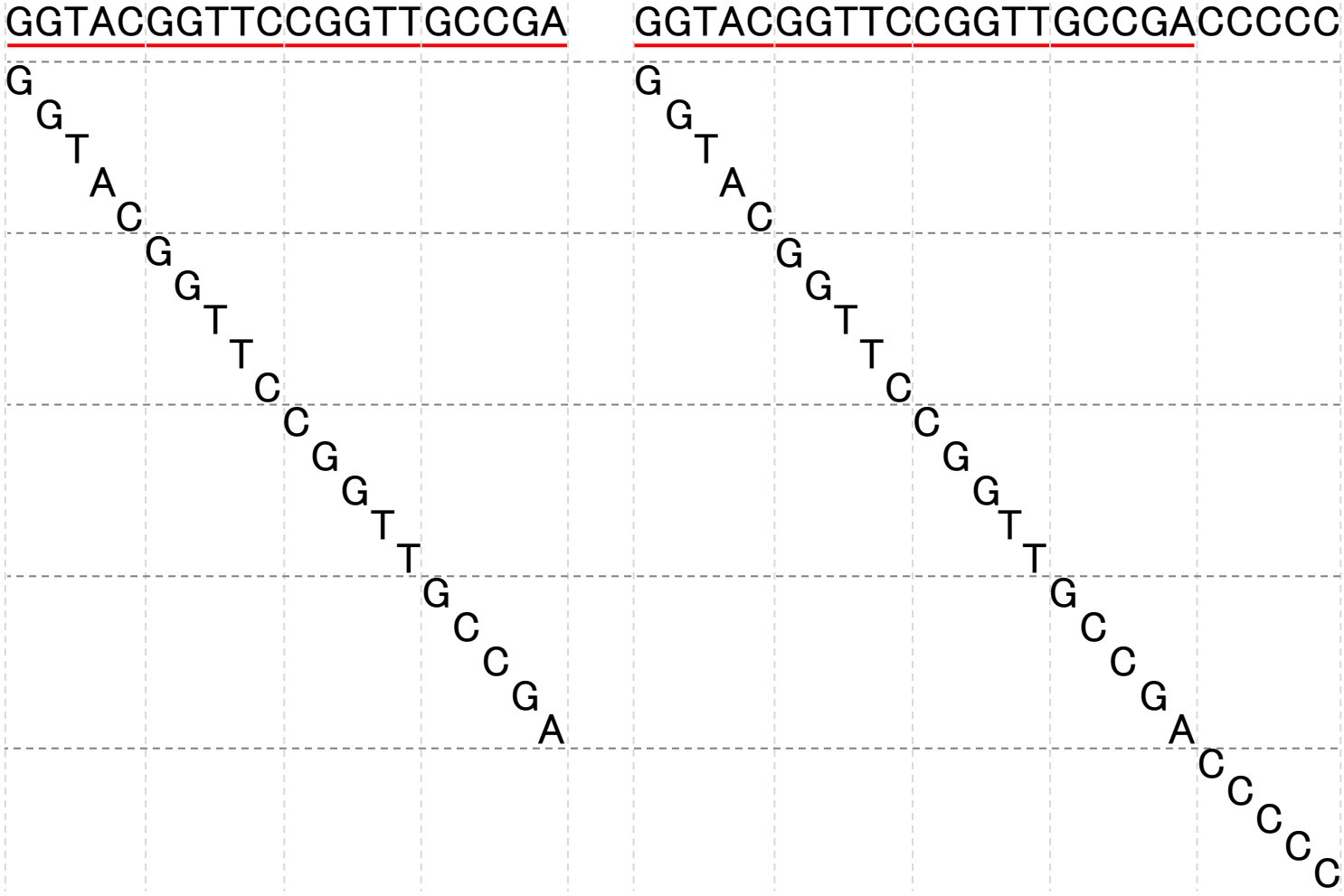
k=1で実践22

変更後のコード全体をコピペ実行すべく、①左上から、②一番下まで反転させて、③Run。実行完了。私は、④少し右側の横幅を広げて、⑤リロードすると、⑥出力ファイルが見られました。⑥をクリックした結果。⑦だと列名がずれてますが、⑧を眺めると、「Aが2個、Cが5個、Gが8個、Tが5個」となっているのがよくわかります。



確かに最初に示した⑨と同じ結果です。

k=1で実践23



⑨

A	2	2
C	5	10
G	8	8
T	5	5

Contents

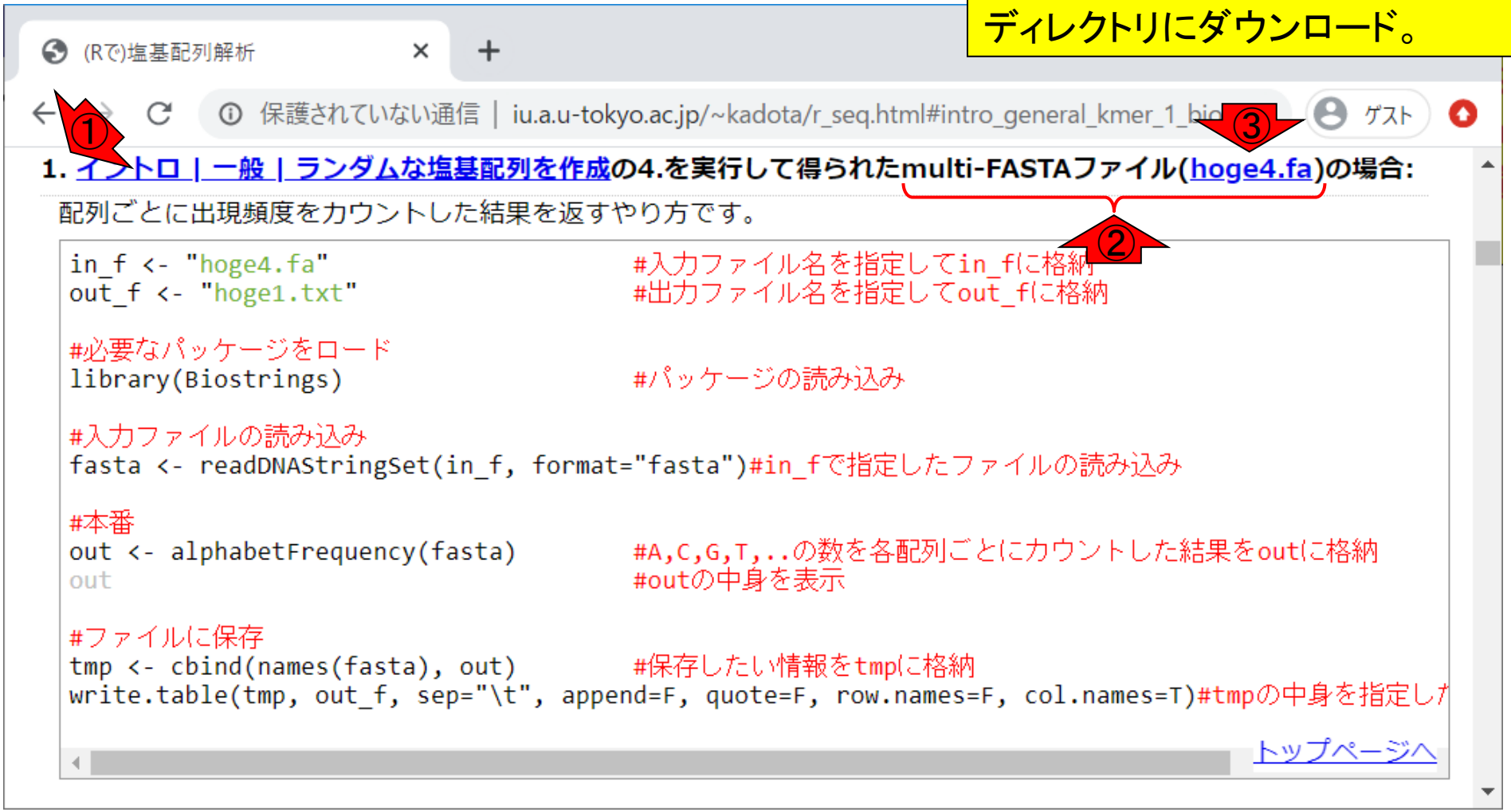
- Introduction、出現頻度解析（k=2）、出現頻度解析（k=1）
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図（例題10）、作図（例題11）、作図（例題12）
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ（例題32）、被覆率（coverage）、基本的な考え方(例題7)
  - 例題8（k=2)、例題9（k=3）、1,000塩基の仮想ゲノム（サンプルデータの例題33）
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

multi-FASTAファイル1

①を入力として得た、②の出力ファイルの形式が、③配列あたり1行で表現されている理由は、複数の配列を含むmulti-FASTA形式ファイルへの対応を意識しているからです。

multi-FASTAファイル2

さきほどテンプレートとして用いた①例題1は、②multi-FASTAファイルを入力として実行するものです。③hoge4.faを作業ディレクトリにダウンロード。



1. イントロ | 一般 | ランダムな塩基配列を作成の4.を実行して得られたmulti-FASTAファイル(hoge4.fa)の場合:

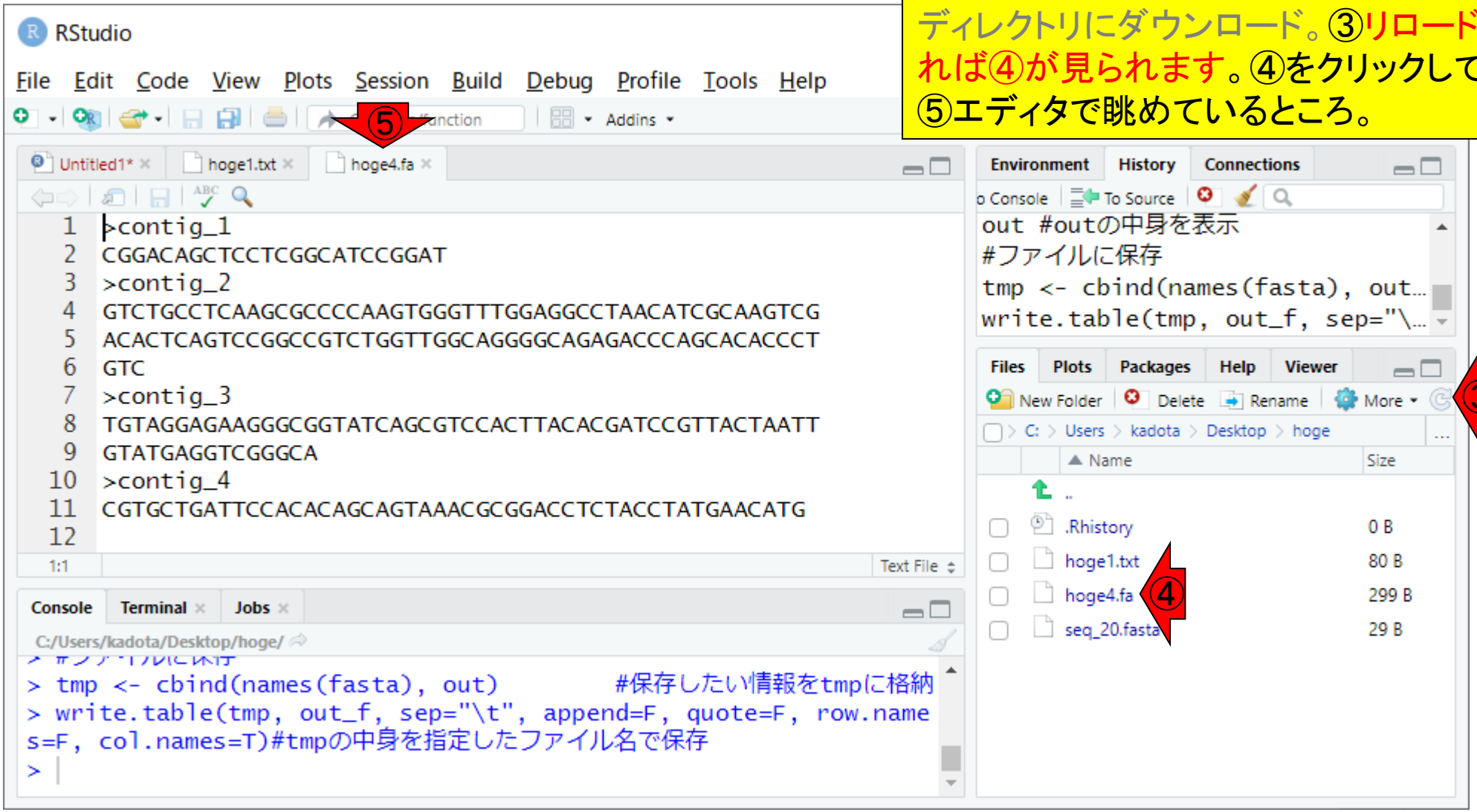
配列ごとに出現頻度をカウントした結果を返すやり方です。

```
in_f <- "hoge4.fa"                          #入力ファイル名を指定してin_fに格納
out_f <- "hoge1.txt"                        #出力ファイル名を指定してout_fに格納

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

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

#本番
out <- alphabetFrequency(fasta)             #A,C,G,T,..の数を各配列ごとにカウントした結果をoutに格納
out                                         #outの中身を表示

#ファイルに保存
tmp <- cbind(names(fasta), out)             #保存したい情報をtmpに格納
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F, col.names=T)#tmpの中身を指定した
```

トップページへ

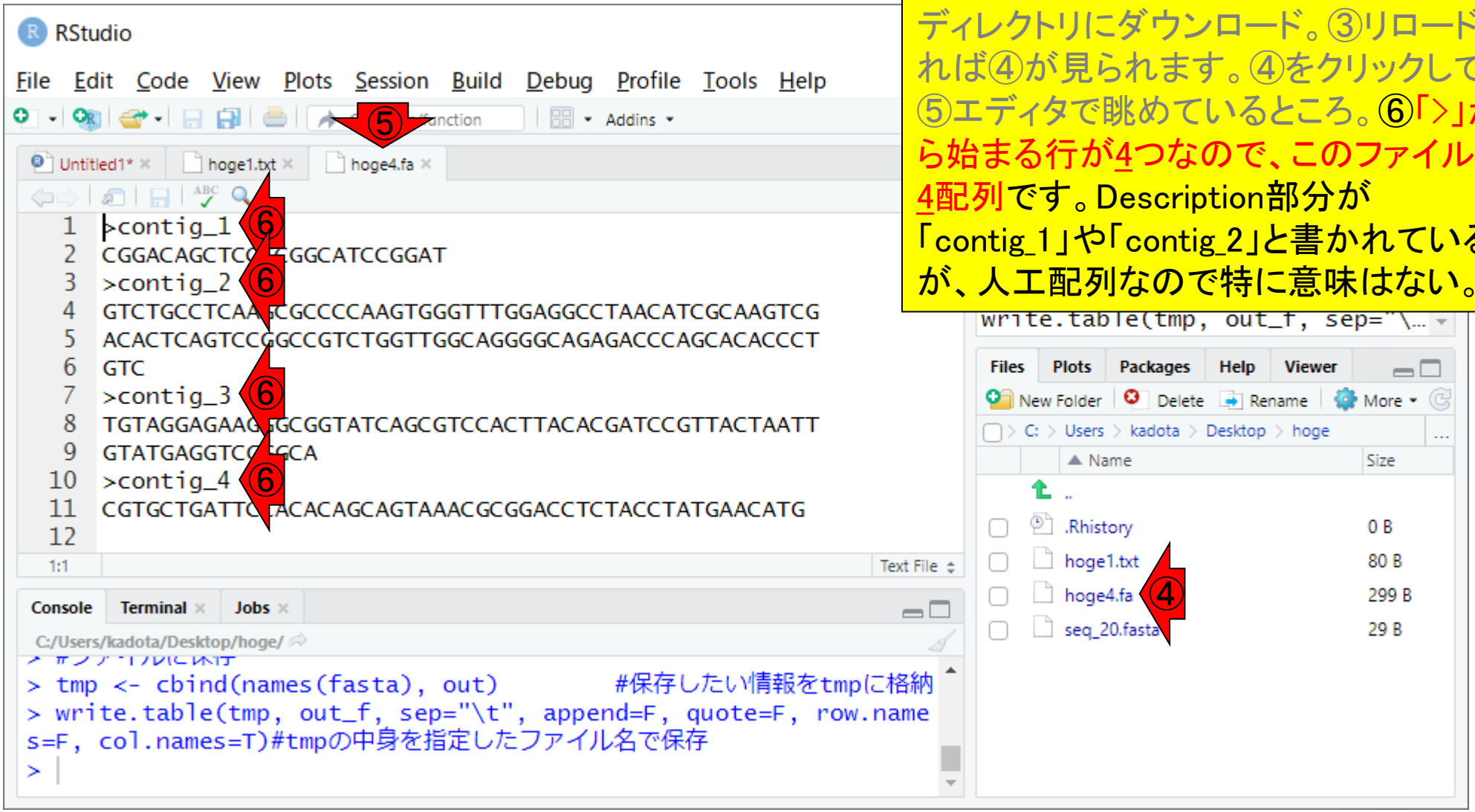
multi-FASTAファイル3

さきほどテンプレートとして用いた①例題1は、②multi-FASTAファイルを入力として実行するものです。③hoge4.faを作業ディレクトリにダウンロード。③リロードすれば④が見られます。④をクリックして、⑤エディタで眺めているところ。

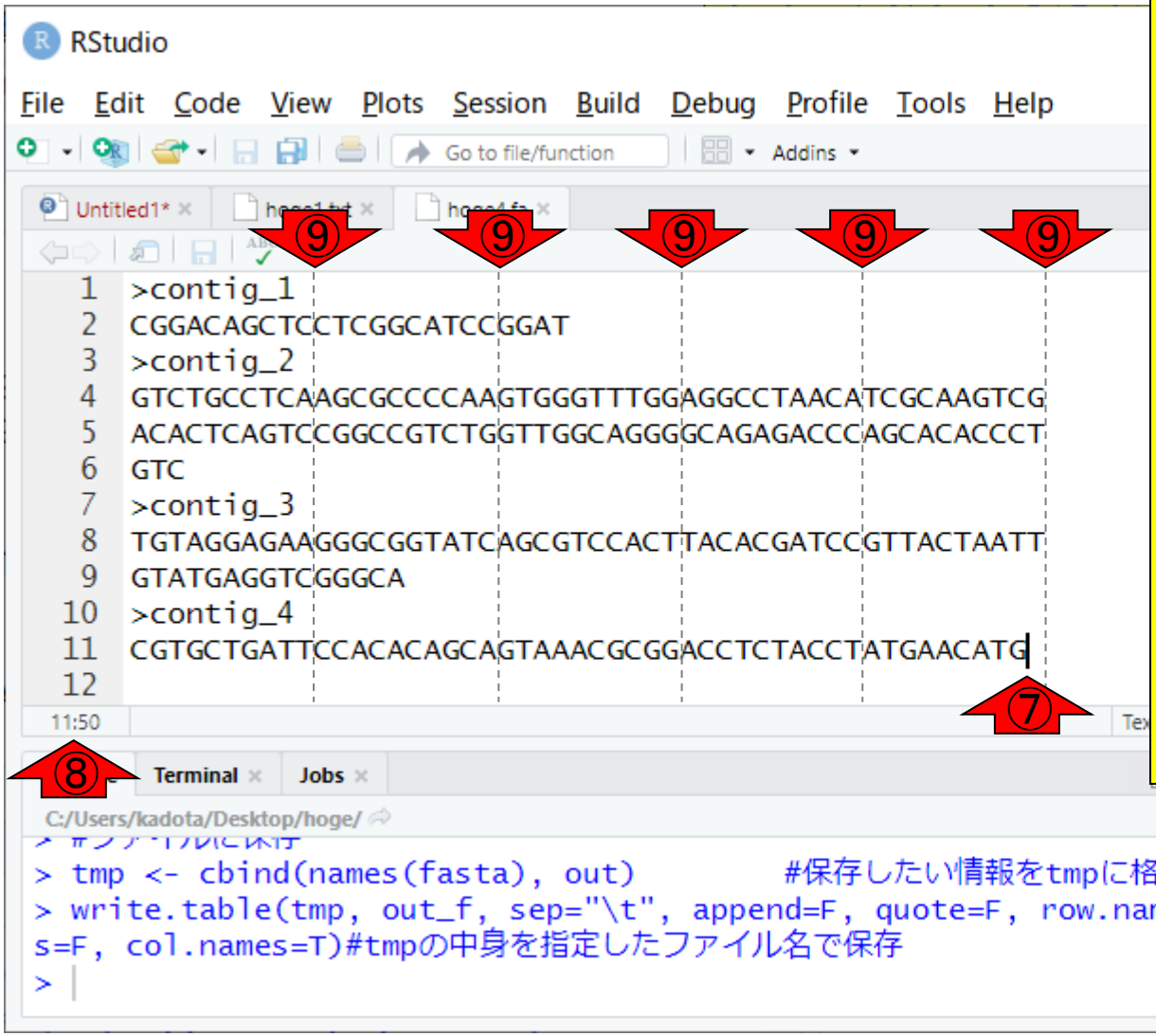


multi-FASTAファイル4

さきほどテンプレートとして用いた①例題1は、②multi-FASTAファイルを入力として実行するものです。③hoge4.faを作業ディレクトリにダウンロード。③リロードすれば④が見られます。④をクリックして、⑤エディタで眺めているところ。⑥「>」から始まる行が4つなので、このファイルは4配列です。Description部分が「contig_1」や「contig_2」と書かれているが、人工配列なので特に意味はない。



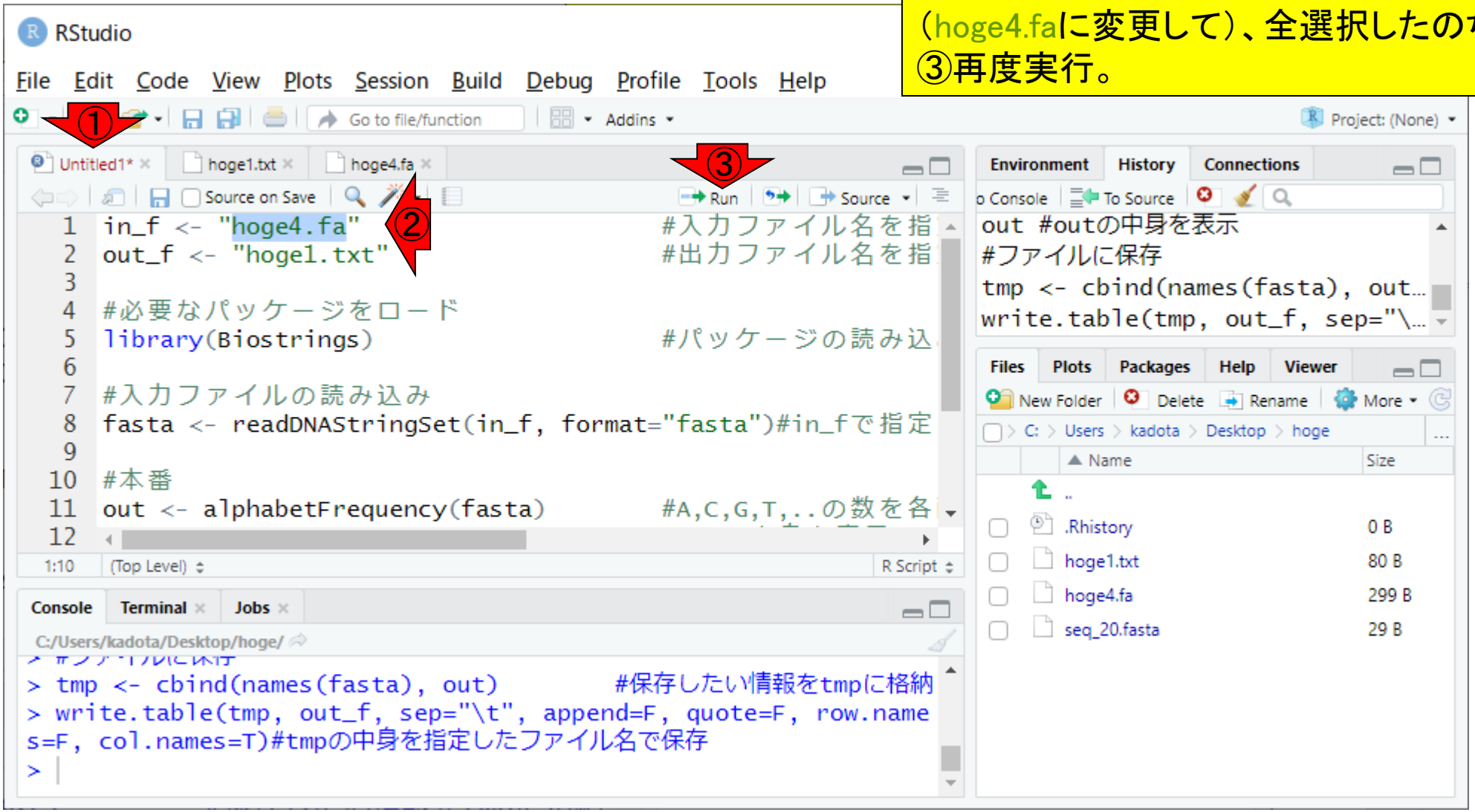
multi-FASTAファイル5



さきほどテンプレートとして用いた①例題1は、②multi-FASTAファイルを入力として実行するものです。③hoge4.faを作業ディレクトリにダウンロード。③リロードすれば④が見られます。④をクリックして、⑤エディタで眺めているところ。⑥「>」から始まる行が4つなので、このファイルは4配列です。Description部分が「contig_1」や「contig_2」と書かれているが、人工配列なので特に意味はない。⑦のカーソル位置が、⑧11行50列目を意味する「11:50」となっていることがわかる。また、⑨の破線は10塩基ごとに示している。このことから、「contig_1は24塩基、contig_2は103塩基、contig_3は65塩基、contig_4は49塩基」だということが目視でわかります。

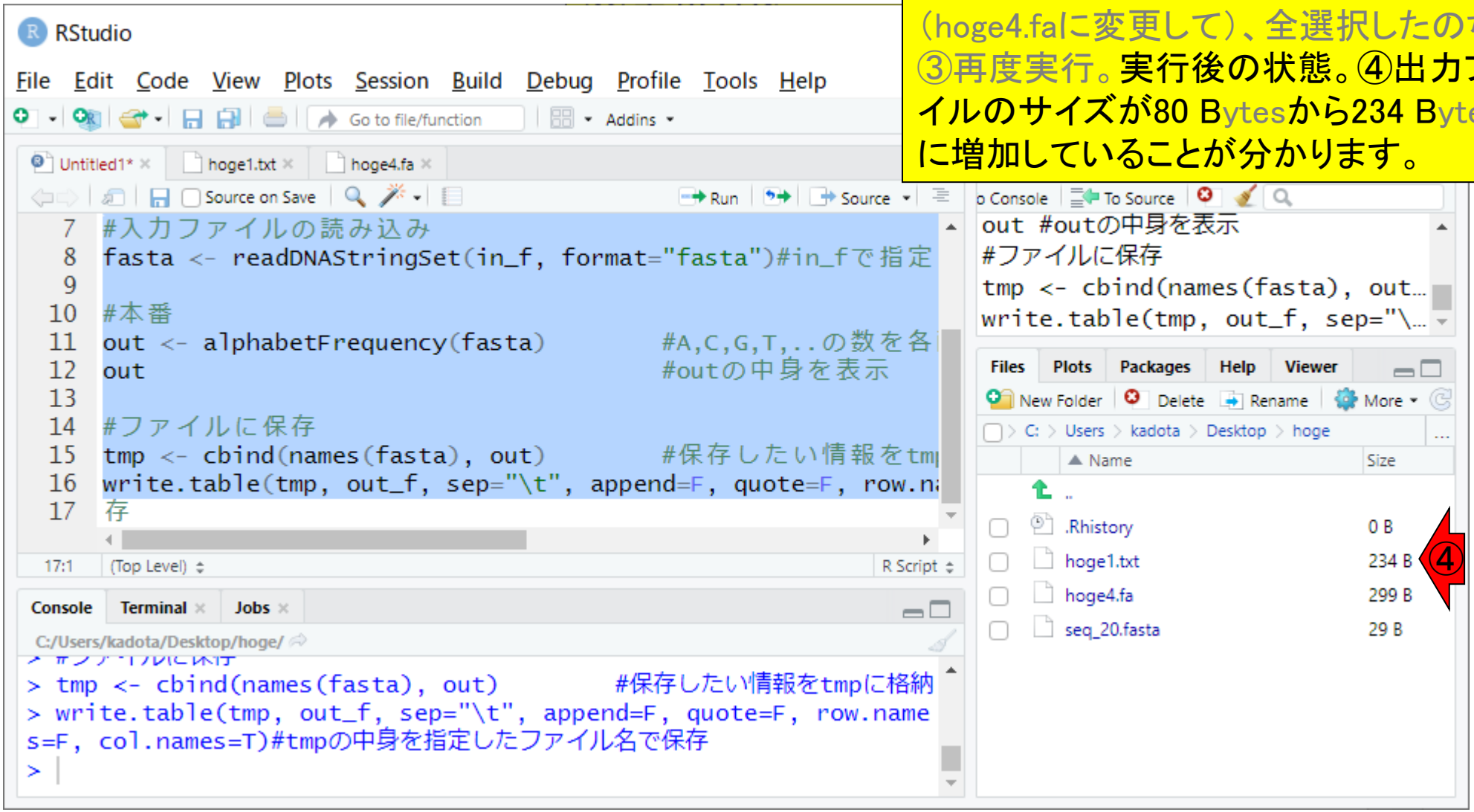
multi-FASTAファイル6

例題1のテンプレートをコピペ実行でもよいが、ここでは①のタブをアクティブにして、②入力ファイル名を元に戻して（hoge4.faに変更して）、全選択したのち、③再度実行。



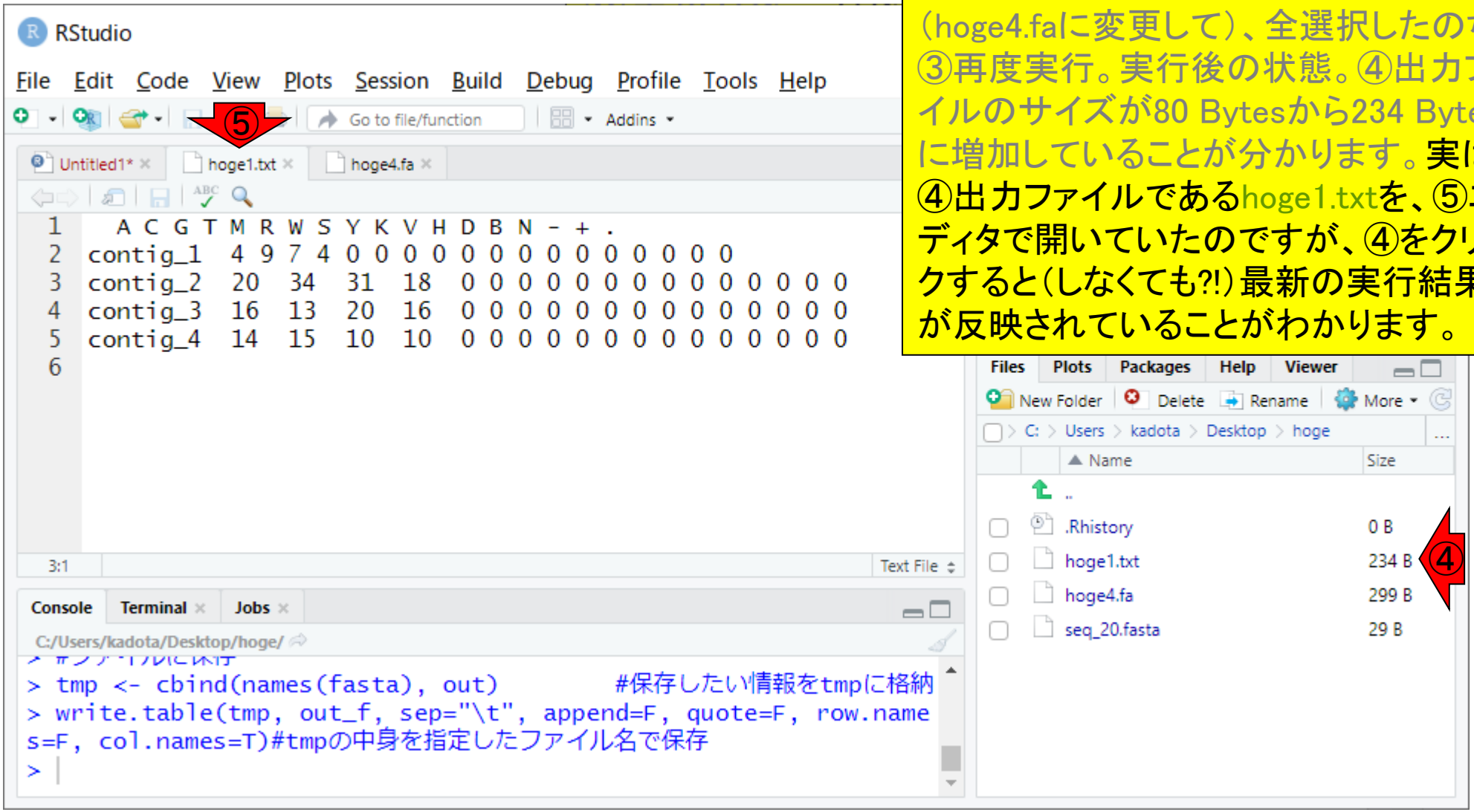
multi-FASTAファイル7

例題1のテンプレートをコピペ実行でもよいが、ここでは①のタブをアクティブにして、②入力ファイル名を元に戻して（hoge4.faに変更して）、全選択したのち、③再度実行。実行後の状態。④出力ファイルのサイズが80 Bytesから234 Bytesに増加していることが分かります。

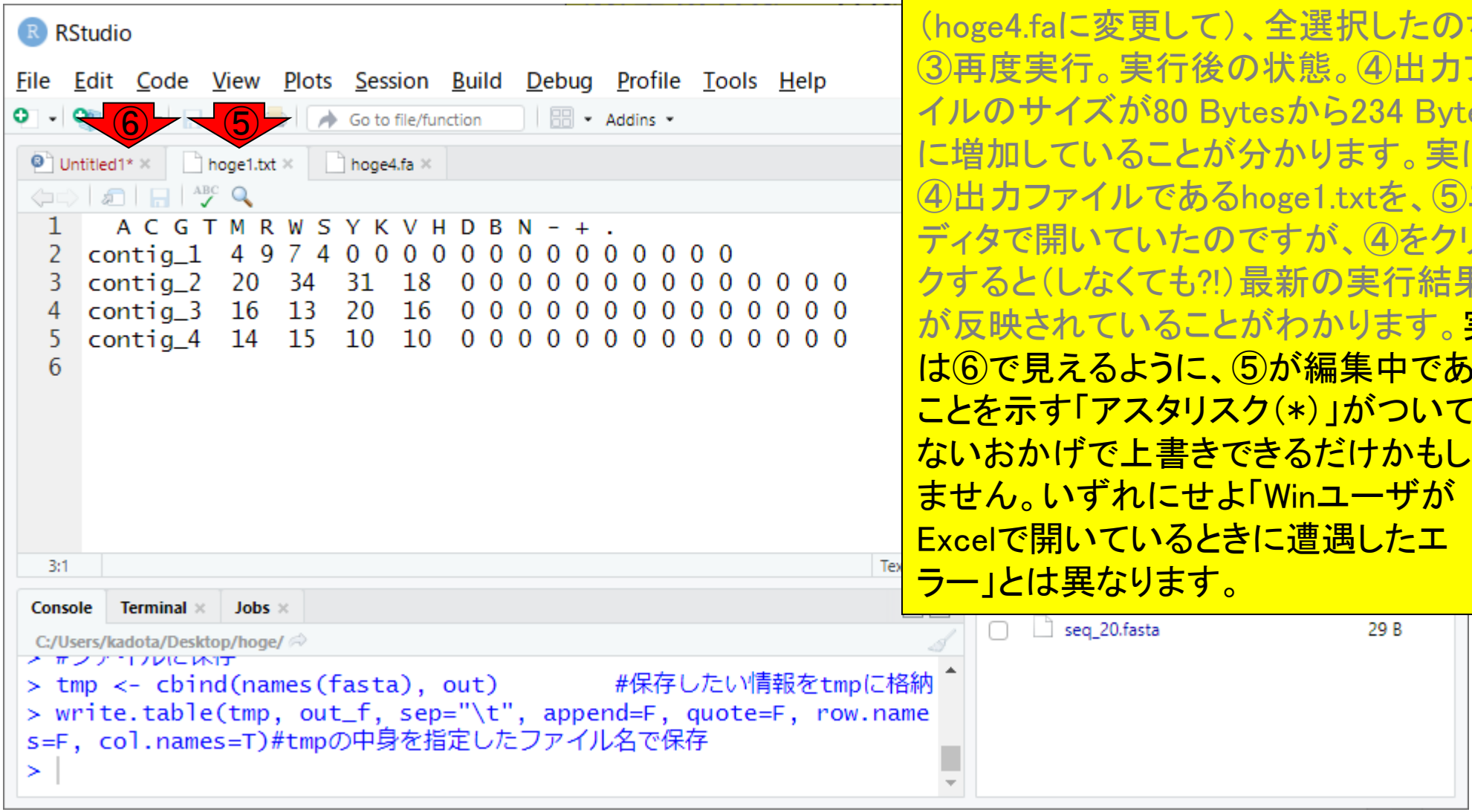


multi-FASTAファイル8

例題1のテンプレートをコピペ実行でもよいが、ここでは①のタブをアクティブにして、②入力ファイル名を元に戻して（hoge4.faに変更して）、全選択したのち、③再度実行。実行後の状態。④出力ファイルのサイズが80 Bytesから234 Bytesに増加していることが分かります。実は、④出力ファイルであるhoge1.txtを、⑤エディタで開いていたのですが、④をクリックすると（しなくても?!）最新の実行結果が反映されていることがわかります。



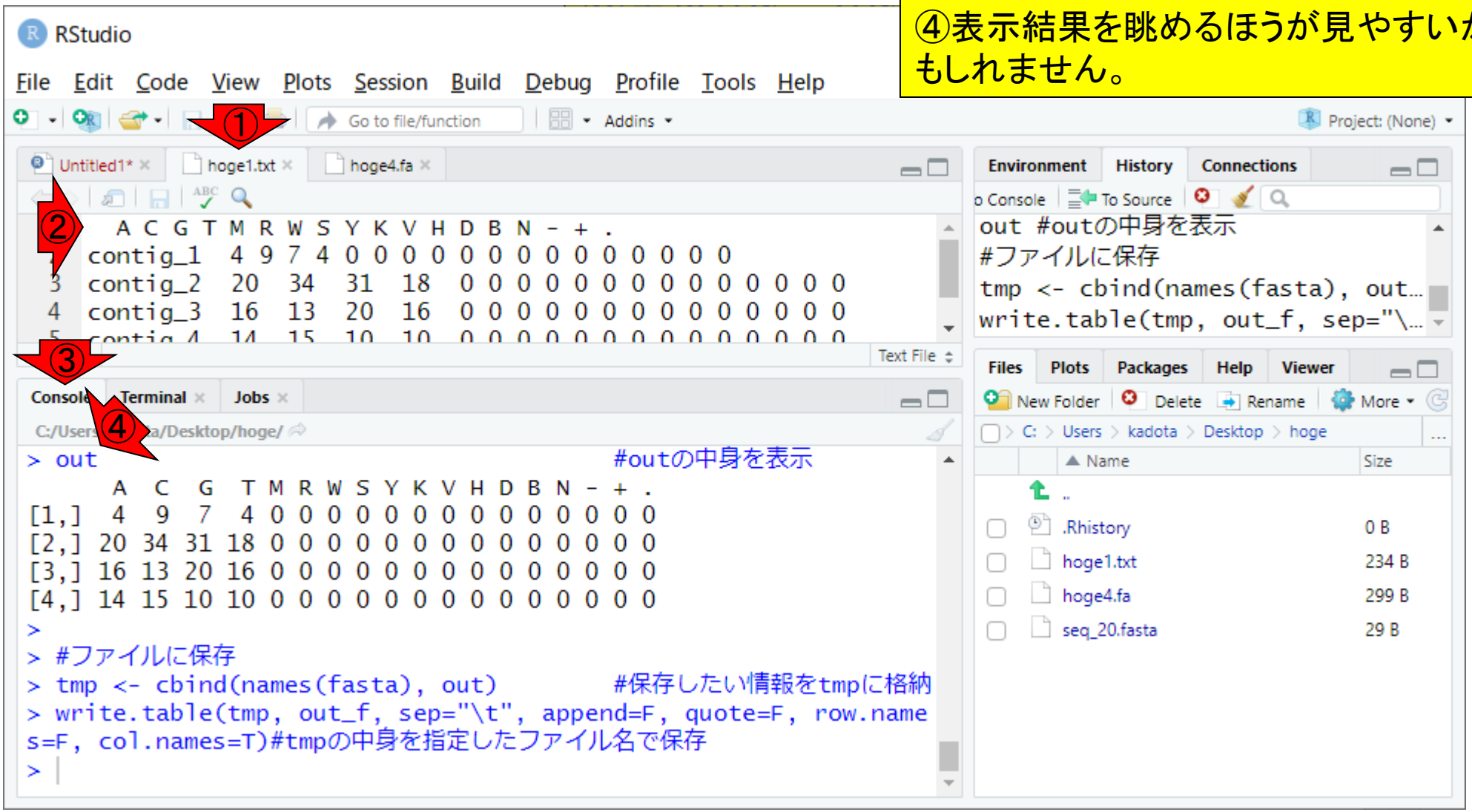
multi-FASTAファイル9



例題1のテンプレートをコピペ実行でもよいが、ここでは①のタブをアクティブにして、②入力ファイル名を元に戻して（hoge4.faに変更して）、全選択したのち、③再度実行。実行後の状態。④出力ファイルのサイズが80 Bytesから234 Bytesに増加していることが分かります。実は、④出力ファイルであるhoge1.txtを、⑤エディタで開いていたのですが、④をクリックすると（しなくても?!）最新の実行結果が反映されていることがわかります。実は⑥で見えるように、⑤が編集中であることを示す「アスタリスク（*）」がついていないおかげで上書きできるだけかもしれません。いずれにせよ「Winユーザが Excelで開いているときに遭遇したエラー」とは異なります。

multi-FASTAファイル10

①エディタで眺めると、②列名部分が左にずれて見辛い。慣れれば③Console画面上の、(この場合はoutオブジェクトの)④表示結果を眺めるほうが見やすいかもしれません。

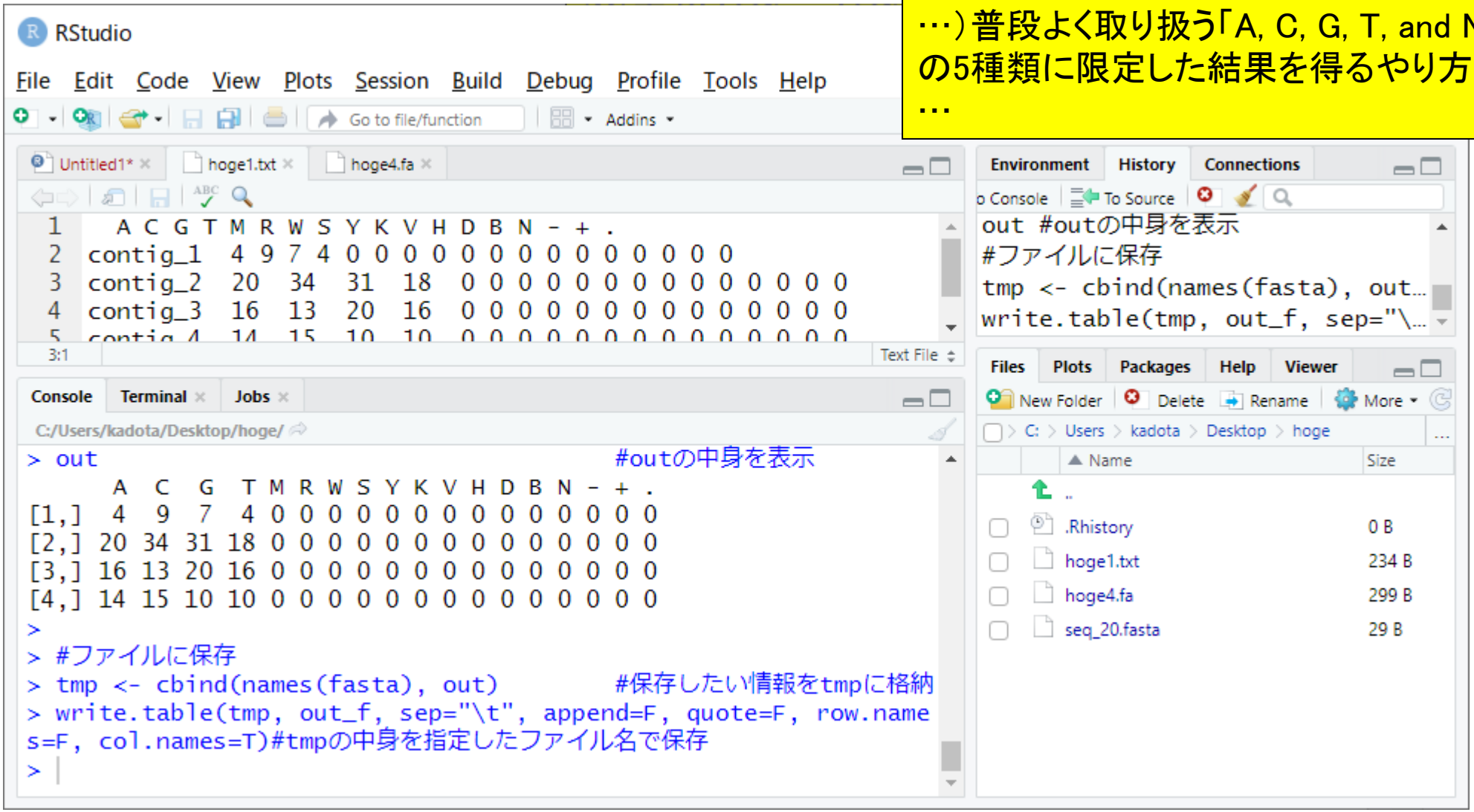


Contents

- Introduction、出現頻度解析（k=2）、出現頻度解析（k=1）
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図（例題10）、作図（例題11）、作図（例題12）
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ（例題32）、被覆率（coverage）、基本的な考え方(例題7)
  - 例題8（k=2)、例題9（k=3）、1,000塩基の仮想ゲノム（サンプルデータの例題33）
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

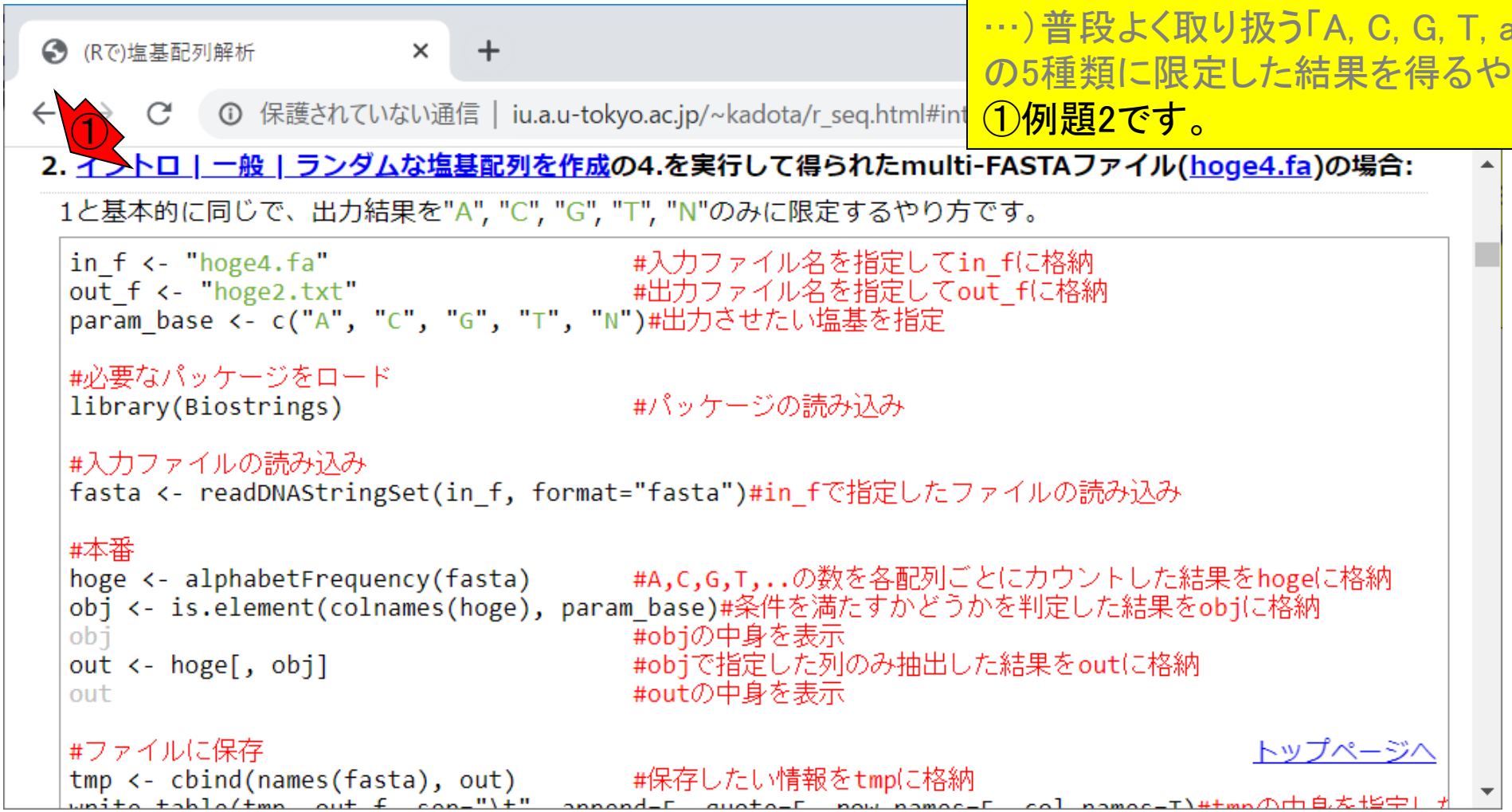
他の例題を実行1

（ごくまれに塩基の表記で「A or Cを意味するM」や「A or Gを意味するR」などを見かけますが実質的に無視でもよいので…）普段よく取り扱う「A, C, G, T, and N」の5種類に限定した結果を得るやり方が…



他の例題を実行2

（ごくまれに塩基の表記で「A or Cを意味するM」や「A or Gを意味するR」などを見かけますが実質的に無視でもよいので…）普段よく取り扱う「A, C, G, T, and N」の5種類に限定した結果を得るやり方が、①例題2です。



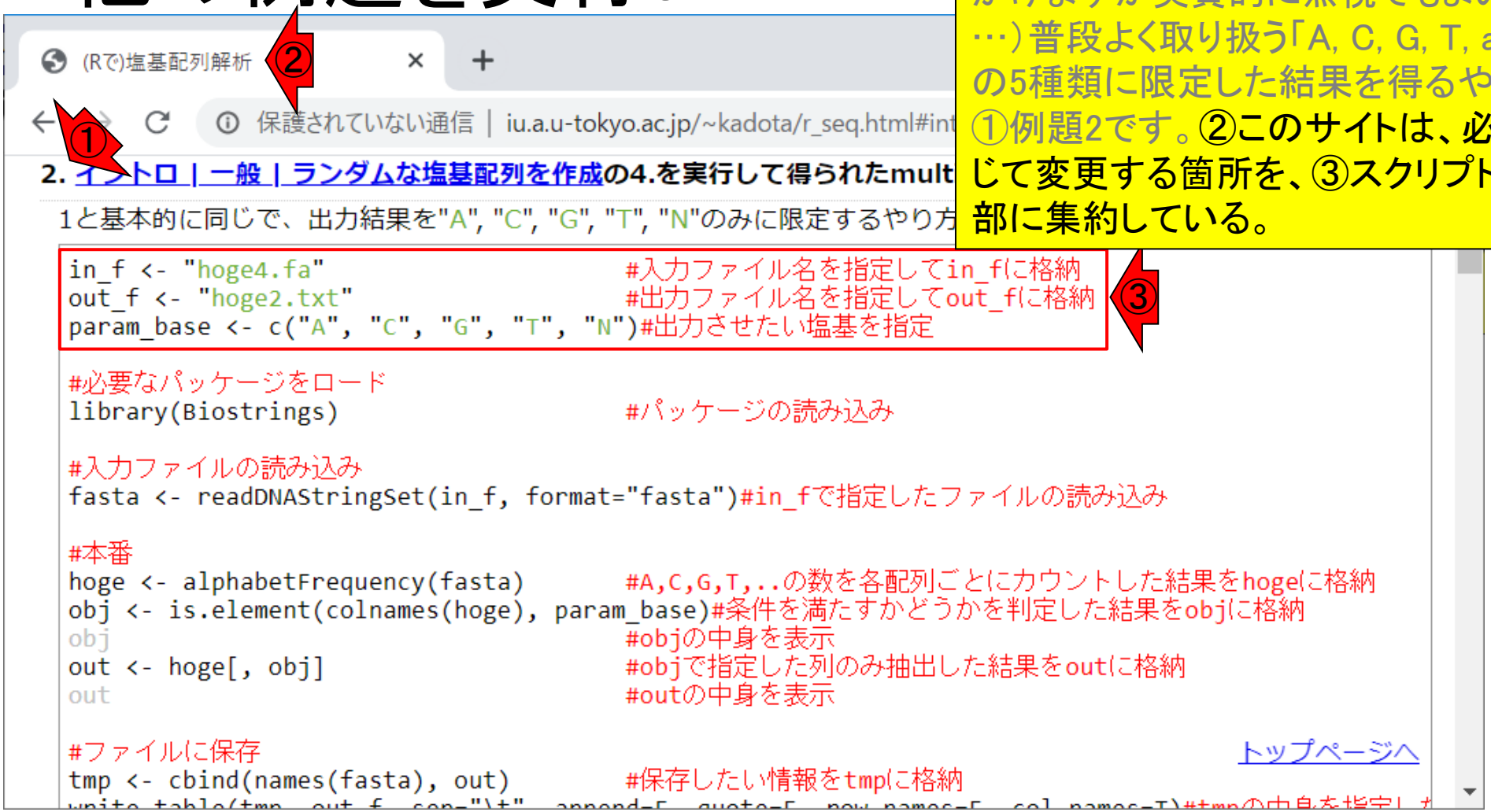
2. イントロ | 一般 | ランダムな塩基配列を作成の4.を実行して得られたmulti-FASTAファイル(hoge4.fa)の場合:

1と基本的に同じで、出力結果を"A", "C", "G", "T", "N"のみに限定するやり方です。

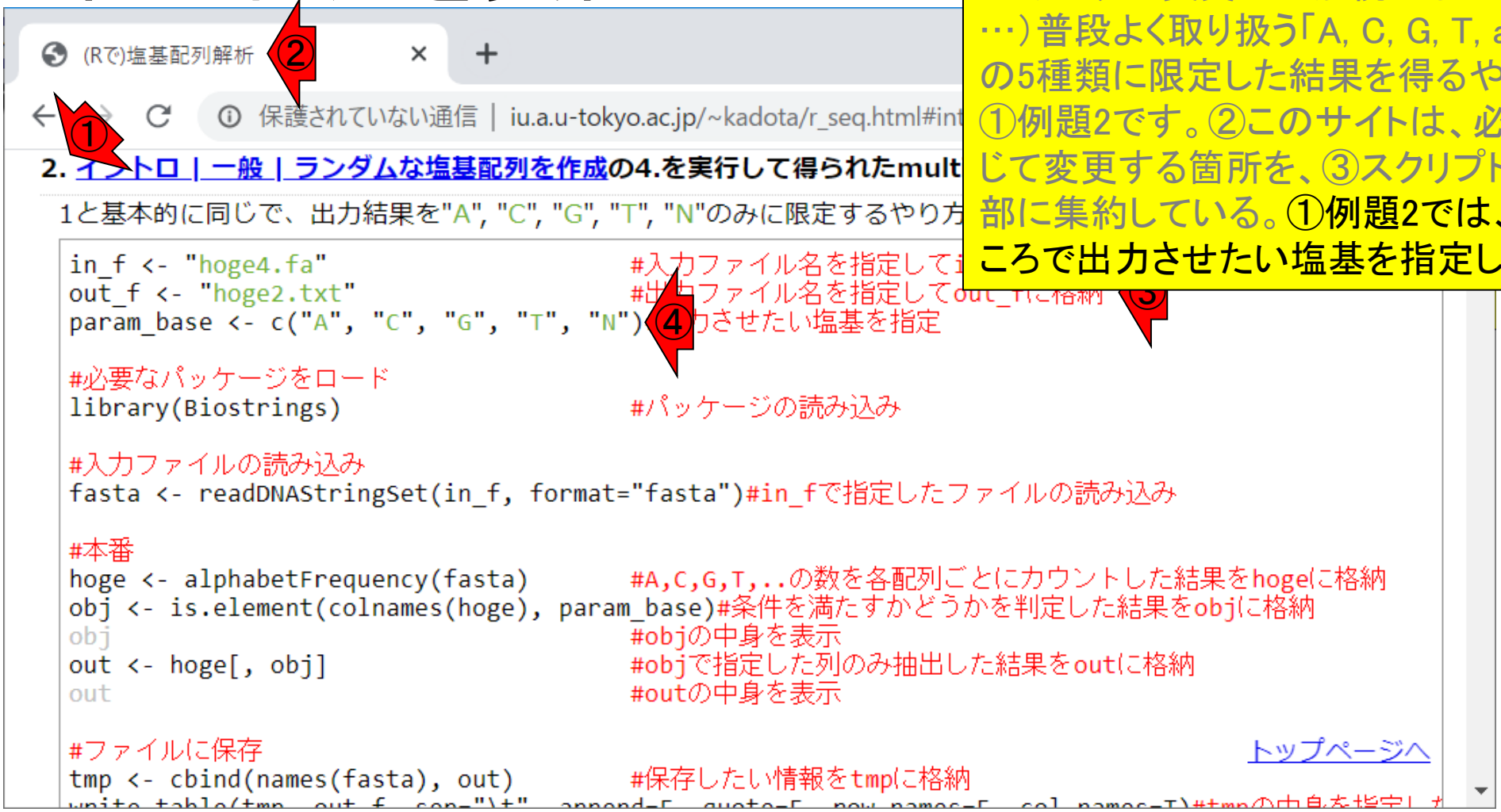
```
in_f <- "hoge4.fa"                          #入力ファイル名を指定してin_fに格納
out_f <- "hoge2.txt"                        #出力ファイル名を指定してout_fに格納
param_base <- c("A", "C", "G", "T", "N")#出力させたい塩基を指定

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

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

#本番
hoge <- alphabetFrequency(fasta)            #A,C,G,T,..の数を各配列ごとにカウントした結果をhogeに格納
obj <- is.element(colnames(hoge), param_base)#条件を満たすかどうかを判定した結果をobjに格納
obj                                         #objの中身を表示
out <- hoge[, obj]                          #objで指定した列のみ抽出した結果をoutに格納
out                                         #outの中身を表示

#ファイルに保存                                                    トップページへ
tmp <- cbind(names(fasta), out)             #保存したい情報をtmpに格納
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F, col.names=T)#tmpの中身を指定した
```

他の例題を実行3

(ごくまれに塩基の表記で「A or Cを意味するM」や「A or Gを意味するR」などを見かけますが実質的に無視でもよいので…)普段よく取り扱う「A, C, G, T, and N」の5種類に限定した結果を得るやり方が、①例題2です。②このサイトは、必要に応じて変更する箇所を、③スクリプトの上部に集約している。

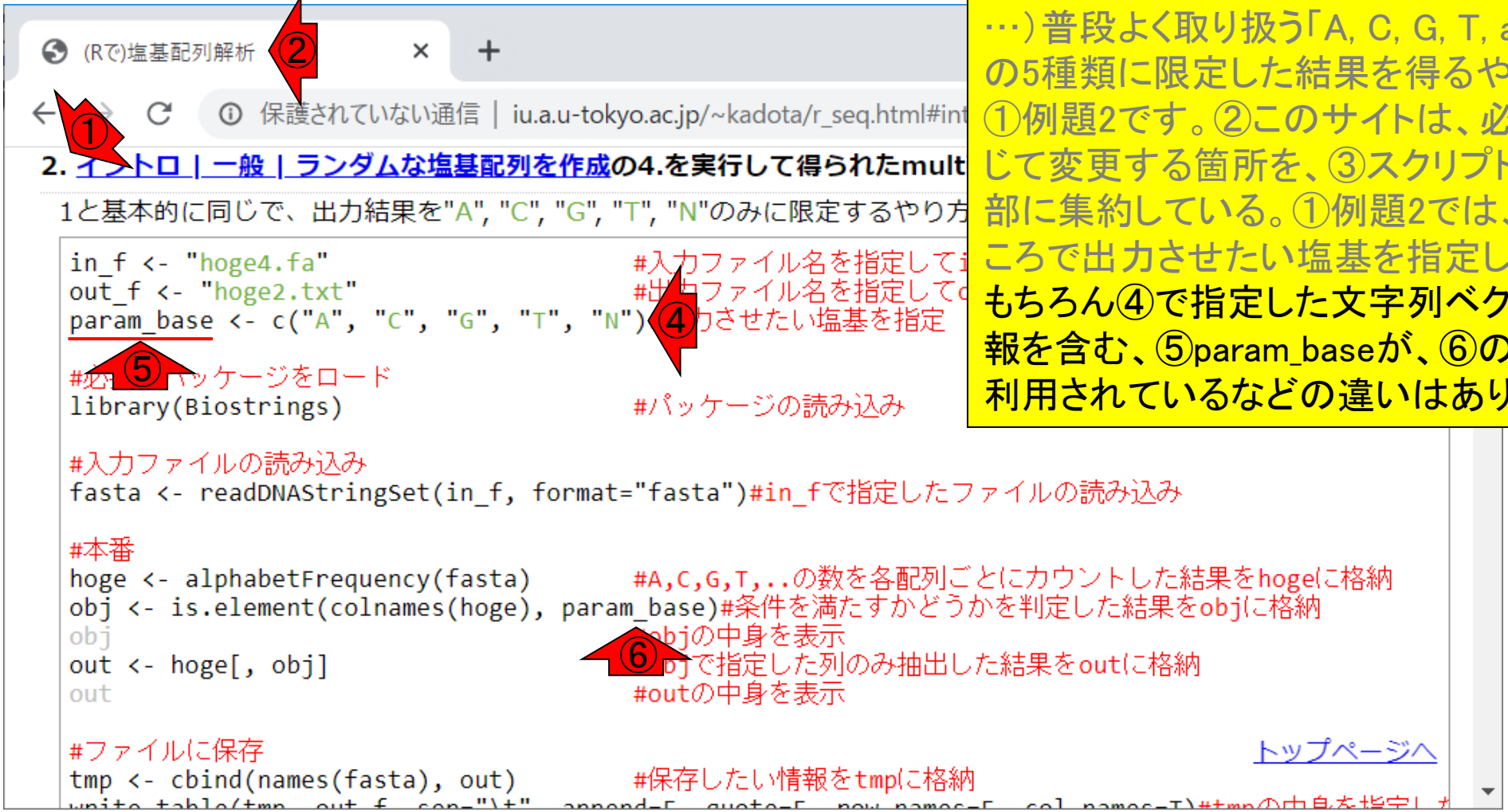


他の例題を実行4



（ごくまれに塩基の表記で「A or Cを意味するM」や「A or Gを意味するR」などを見かけますが実質的に無視でもよいので…）普段よく取り扱う「A, C, G, T, and N」の5種類に限定した結果を得るやり方が、①例題2です。②このサイトは、必要に応じて変更する箇所を、③スクリプトの上部に集約している。①例題2では、④のところで出力させたい塩基を指定している。

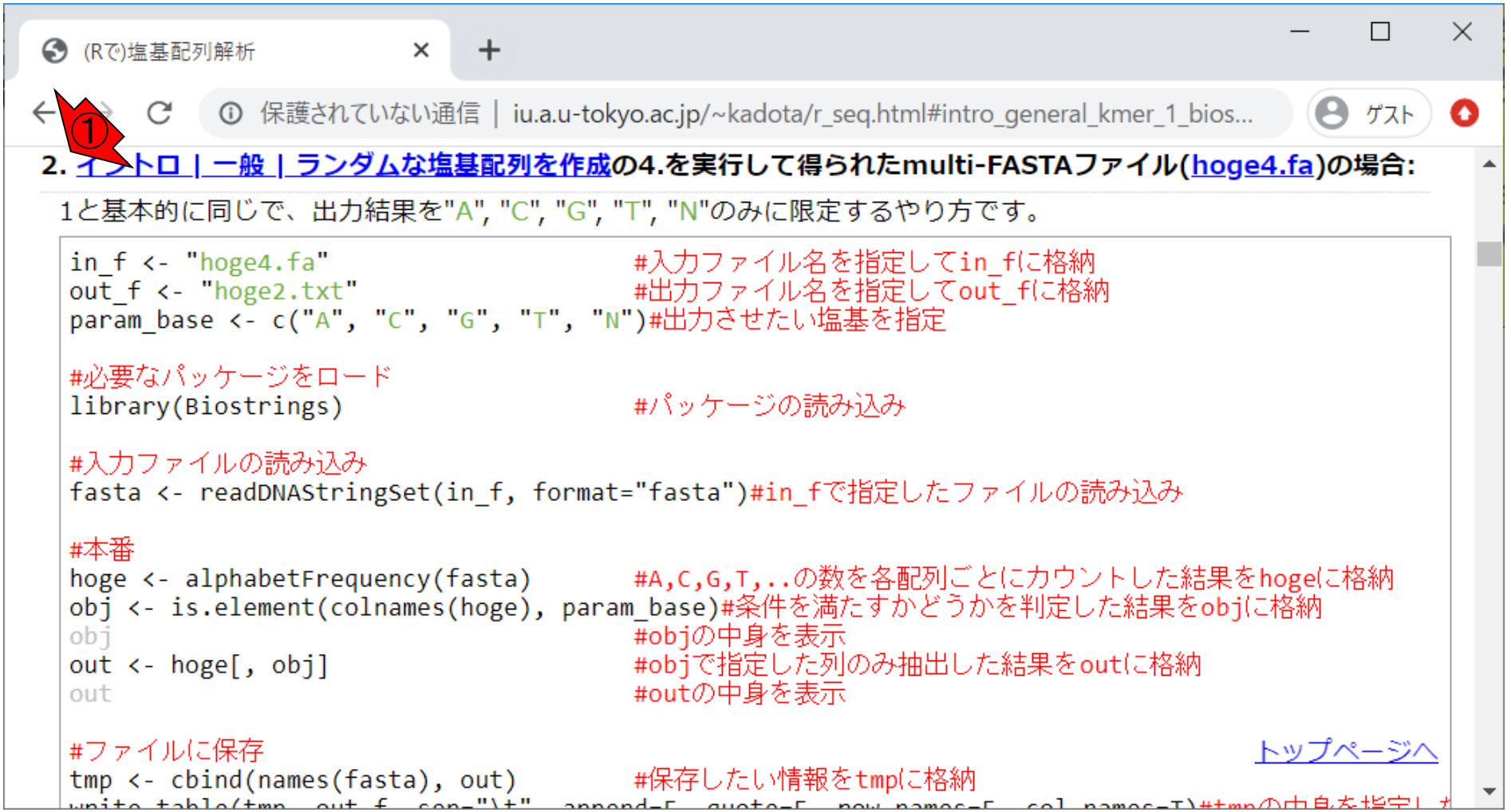
他の例題を実行5



（ごくまれに塩基の表記で「A or Cを意味するM」や「A or Gを意味するR」などを見かけますが実質的に無視でもよいので…）普段よく取り扱う「A, C, G, T, and N」の5種類に限定した結果を得るやり方が、①例題2です。②このサイトは、必要に応じて変更する箇所を、③スクリプトの上部に集約している。①例題2では、④のところで出力させたい塩基を指定している。もちろん④で指定した文字列ベクトル情報を含む、⑤param_baseが、⑥の部分で利用されているなどの違いはあります。

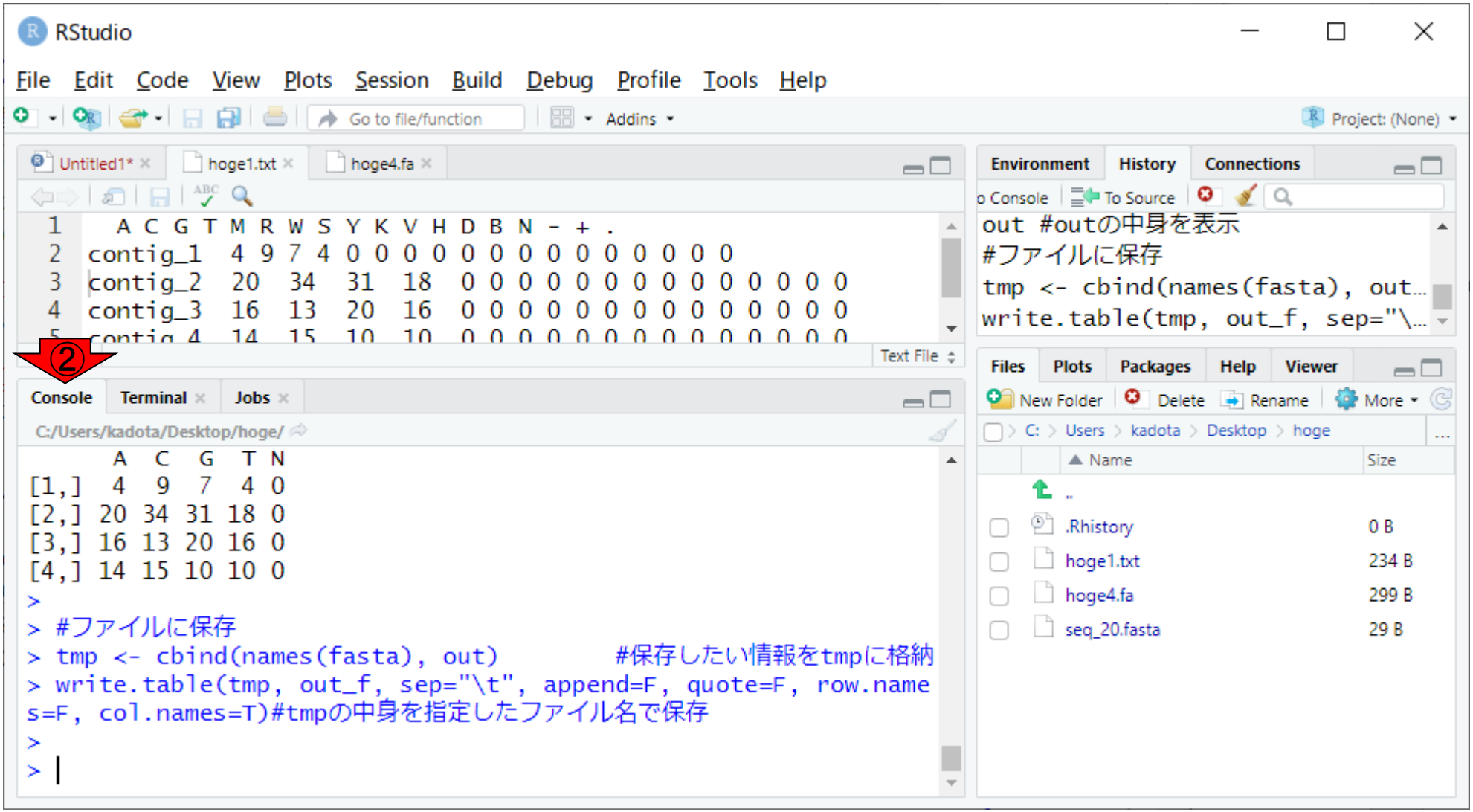
①例題2をコピペ実行。

他の例題を実行6



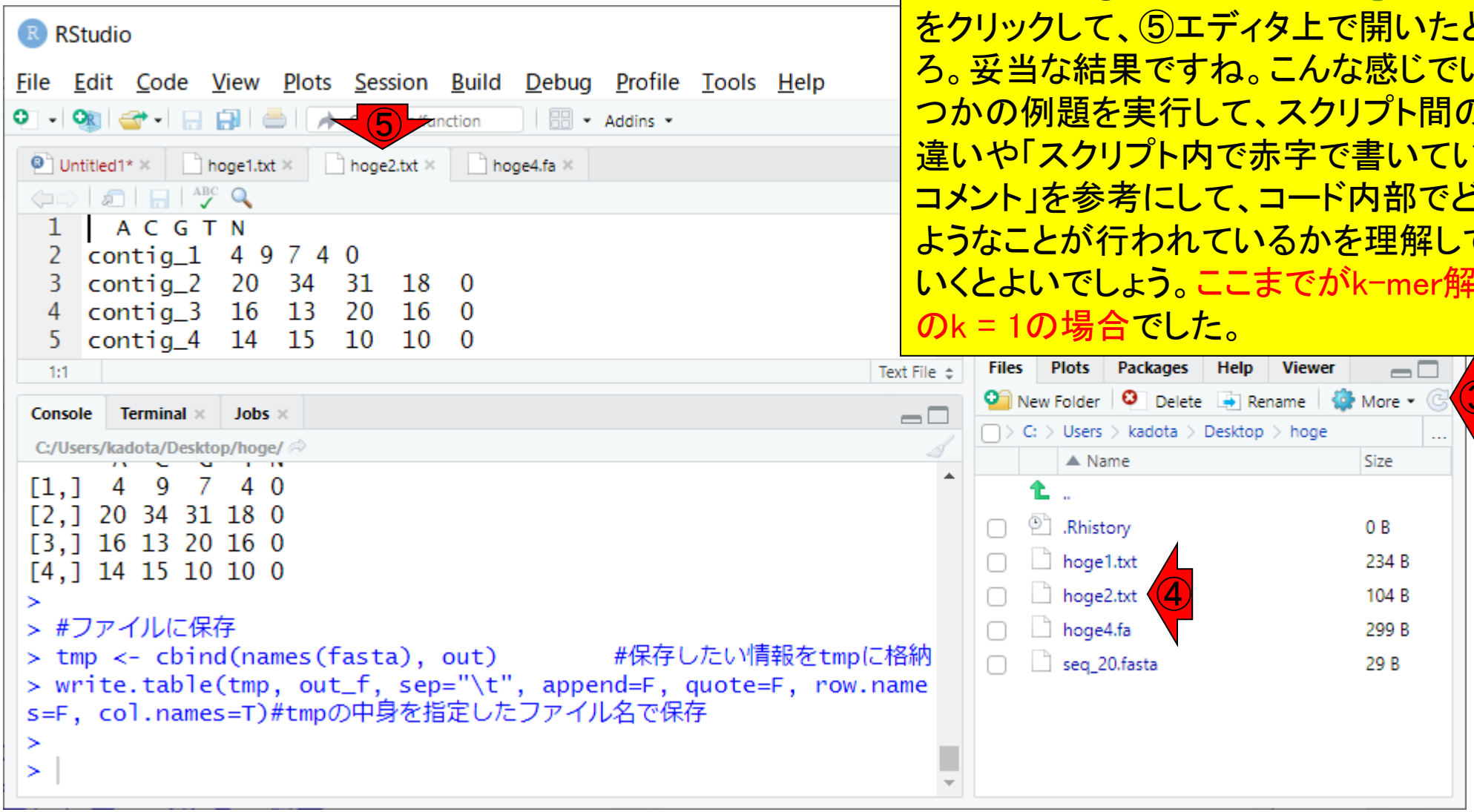
他の例題を実行7

①例題2をコピペ実行。②Console画面上にコピペ実行した直後の状態。



他の例題を実行8

①例題2をコピペ実行。②Console画面上にコピペ実行した直後の状態。③リロードして、④出力ファイル（hoge2.txt）をクリックして、⑤エディタ上で開いたところ。妥当な結果ですね。こんな感じでいくつかの例題を実行して、スクリプト間の違いや「スクリプト内で赤字で書いているコメント」を参考にして、コード内部でどのようなことが行われているかを理解していくとよいでしょう。ここまでがk-mer解析のk = 1の場合でした。

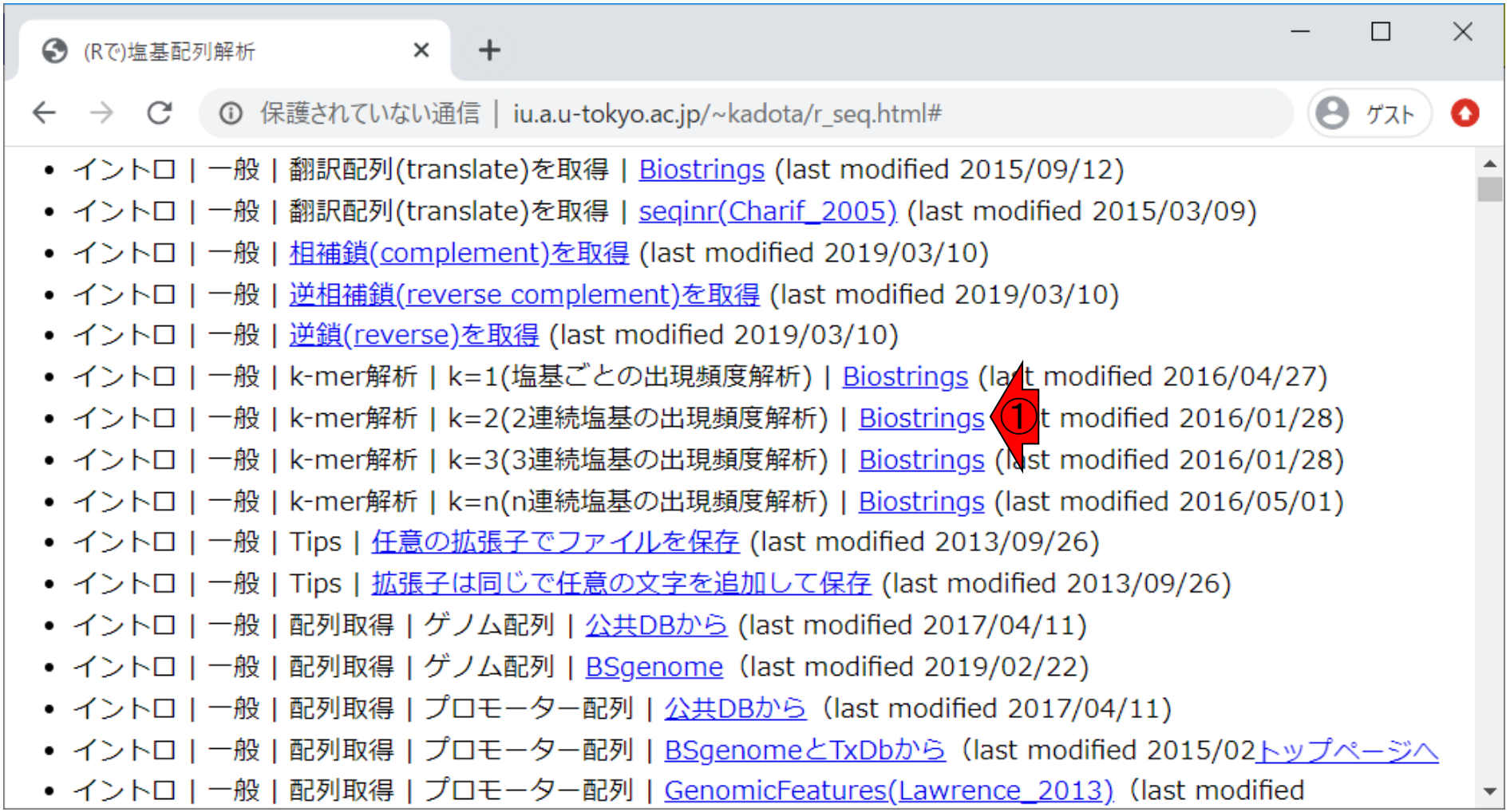


Contents

- Introduction、出現頻度解析（k=2）、出現頻度解析（k=1）
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図（例題10）、作図（例題11）、作図（例題12）
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ（例題32）、被覆率（coverage）、基本的な考え方(例題7)
  - 例題8（k=2)、例題9（k=3）、1,000塩基の仮想ゲノム（サンプルデータの例題33）
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

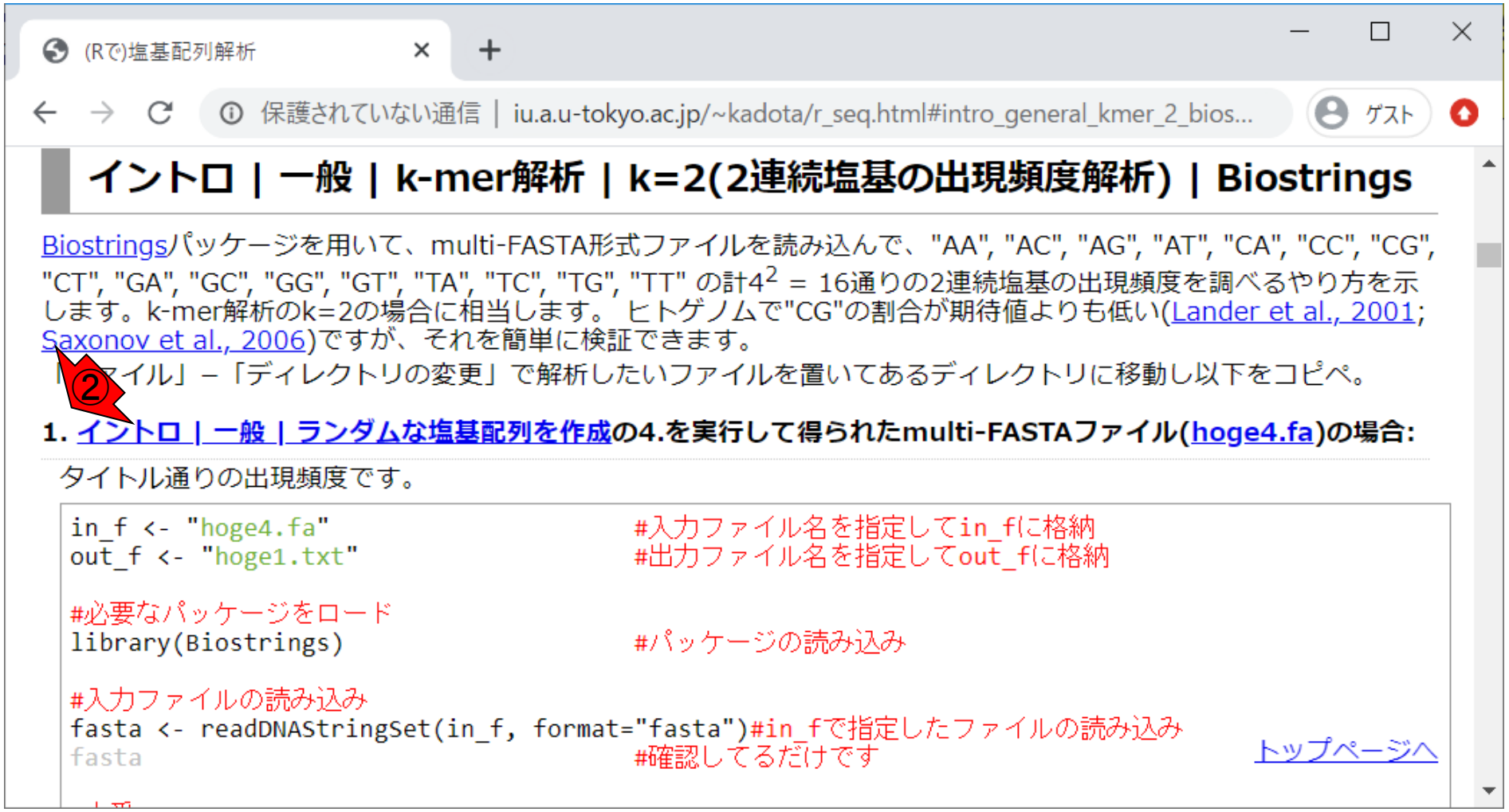
k=2で実践1

①次はk=2でk-mer出現頻度解析を行ってみます。



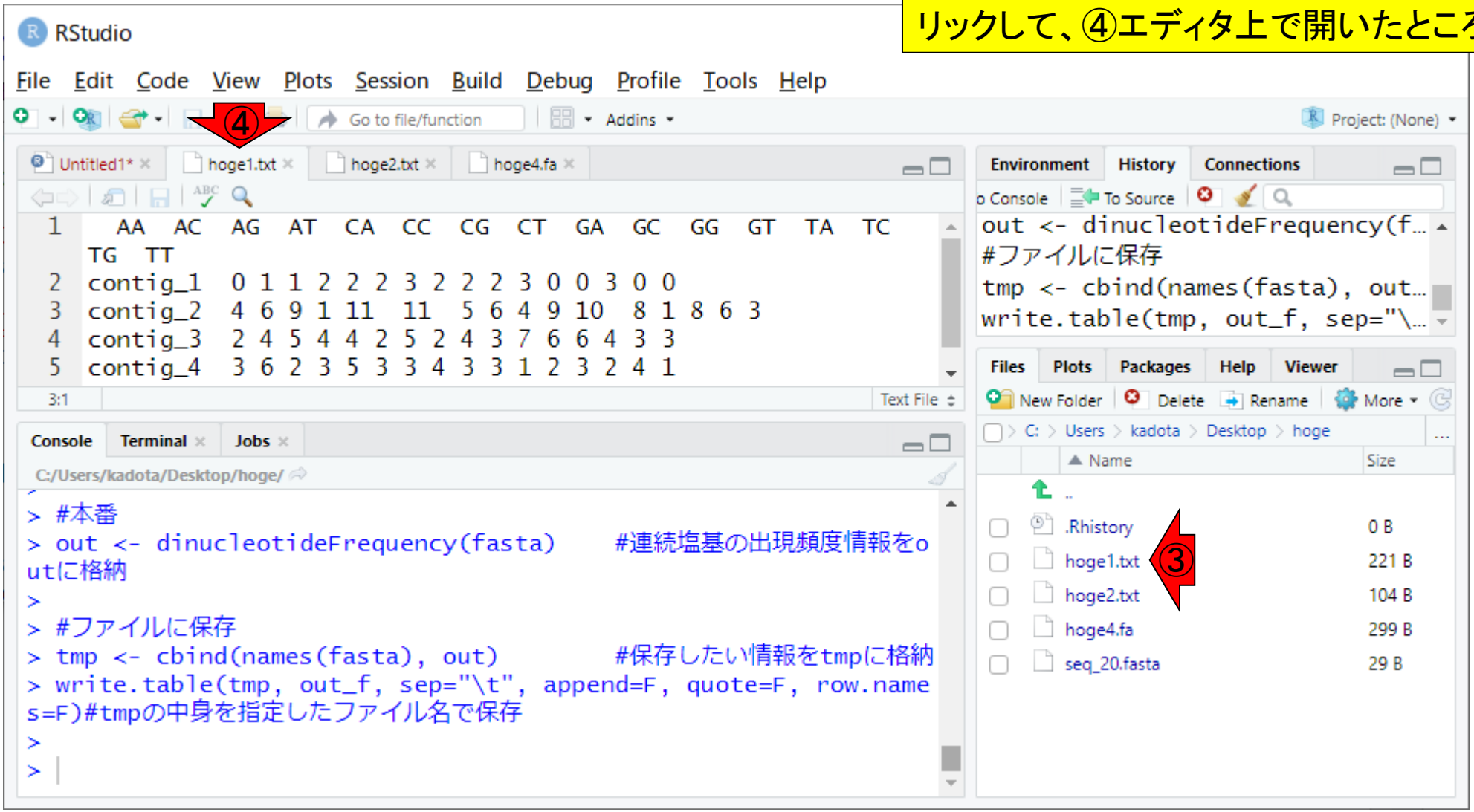
k=2で実践2

①次はk=2でk-mer出現頻度解析を行ってみます。②例題1をコピペ実行。



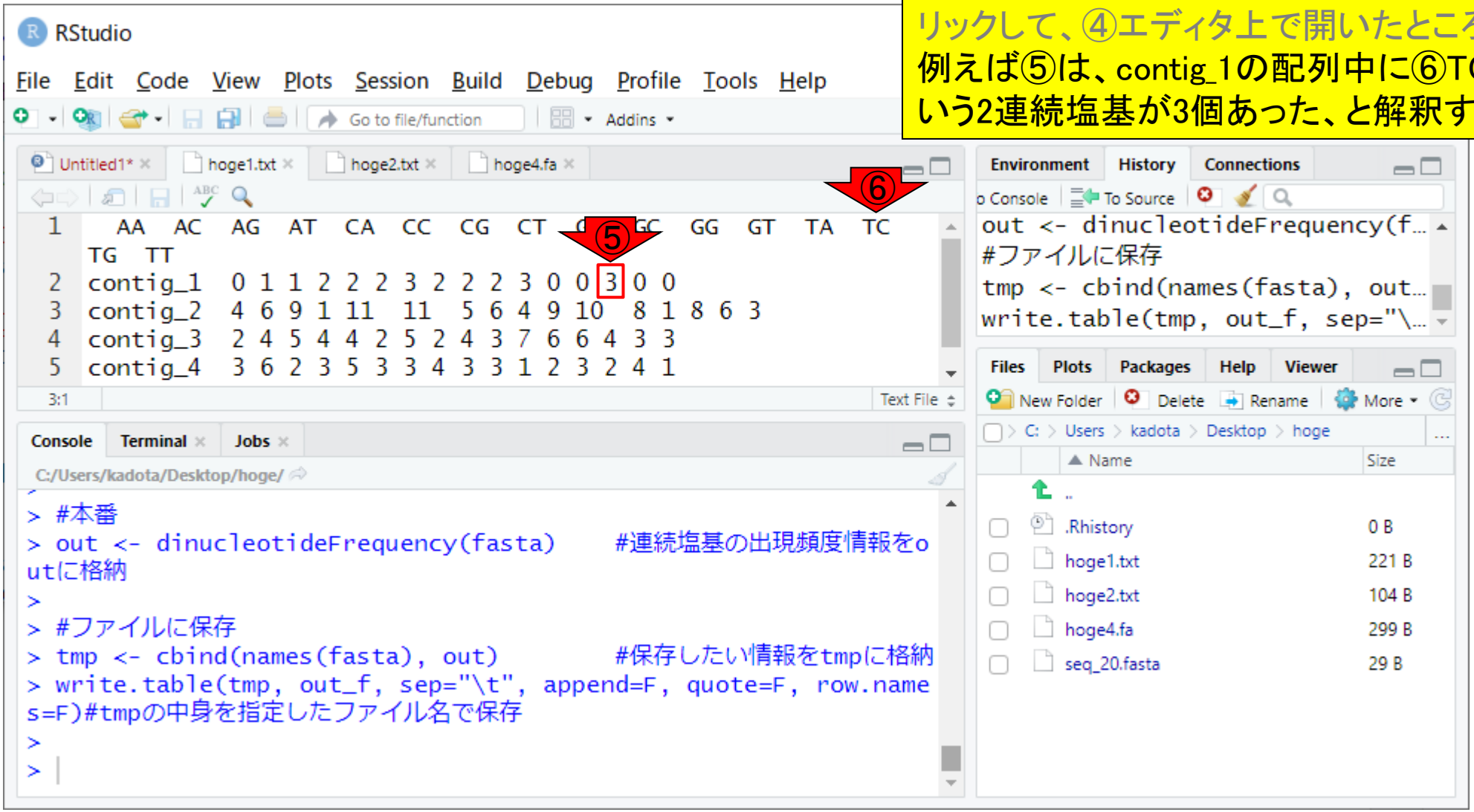
k=2で実践3

①次はk=2でk-mer出現頻度解析を行ってみます。②例題1をコピペ実行。実行後に、③出力ファイル(hoge1.txt)をクリックして、④エディタ上で開いたところ。



k=2で実践4

①次はk=2でk-mer出現頻度解析を行ってみます。②例題1をコピペ実行。実行後に、③出力ファイル(hoge1.txt)をクリックして、④エディタ上で開いたところ。
例えば⑤は、contig_1の配列中に⑥TCという2連続塩基が3個あった、と解釈する。



k=2で実践5

①次はk=2でk-mer出現頻度解析を行ってみます。②例題1をコピペ実行。実行後に、③出力ファイル(hoge1.txt)をクリックして、④エディタ上で開いたところ。例えば⑤は、contig_1の配列中に⑥TCという2連続塩基が3個あった、と解釈する。⑦入力ファイル(hoge4.fa)中の、⑧contig_1を眺めると、確かにTCが3つ存在しますね。

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/functio ⑦ Addins

Untitled1* | hoge1.txt | hoge2.txt | hoge4.fa

⑧

```
1 >contig_1
2 CGGACAGCTCCTCGGCATCCGGAT
3 >contig_2
4 GTCTGCCTCAAGCGCCCCAAGTGGGTTTGGAGGCCTAACATCGCAAGTCG
5 ACACTCAGTCCGGCCGTCTGGTTGGCAGGGGCAGAGACCCAGCACACCCT
6 GTC
```

11:50 Text File

Console Terminal Jobs

C:/Users/kadota/Desktop/hoge/

```
> #本番
> out <- dinucleotideFrequency(fasta)     #連続塩基の出現頻度情報をoutに格納
>
> #ファイルに保存
> tmp <- cbind(names(fasta), out)          #保存したい情報をtmpに格納
> write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定したファイル名で保存
>
>
```

```
tmp <- cbind(names(fasta), out...
write.table(tmp, out_f, sep="\...
```

Files Plots Packages Help Viewer

New Folder Delete Rename More

C: > Users > kadota > Desktop > hoge

Name	Size
..	
.Rhistory	0 B
hoge1.txt	221 B
hoge2.txt	104 B
hoge4.fa	299 B
seq_20.fasta	29 B

Contents

- Introduction、出現頻度解析（k=2）、出現頻度解析（k=1）
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図（例題10）、作図（例題11）、作図（例題12）
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ（例題32）、被覆率（coverage）、基本的な考え方(例題7)
  - 例題8（k=2)、例題9（k=3）、1,000塩基の仮想ゲノム（サンプルデータの例題33）
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

関数マニュアル1

「①さきほどk=2で実行した例題1（dinucleotideFrequency関数を使用）」と「その前にk=1で実行した例題1（alphabetFrequency関数を使用）」とは、②で利用している関数のみが異なります。

(Rで)塩基配列解析

保護されていない通信 | iu.a.u-tokyo.ac.jp/~kadota/r_seq.html#intro_general_kmer_2_bios...

ゲスト

①

1. イントロ | 一般 | ランダムな塩基配列を作成の4.を実行して得られたmulti-FASTAファイル(hoge4.fa)の場合:

タイトル通りの出現頻度です。

```
in_f <- "hoge4.fa"                          #入力ファイル名を指定してin_fに格納
out_f <- "hoge1.txt"                        #出力ファイル名を指定してout_fに格納

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

#入力ファイルの読み込み
fasta <- readDNAStringSet(in_f, format="fasta")#in_fで指定したファイルの読み込み
fasta                                       #確認してるだけです

#本番
out <- dinucleotideFrequency(fasta)         #連続塩基の出現頻度情報をoutに格納

#ファイルに保存
tmp <- cbind(names(fasta), out)             #保存したい情報をtmpに格納
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定したファイル名で保
```

②

トップページへ

関数マニュアル2

「①さきほどk=2で実行した例題1（dinucleotideFrequency関数を使用）」と「その前にk=1で実行した例題1（alphabetFrequency関数を使用）」とは、②で利用している関数のみが異なります。関数が異なると当然役割も異なります。RStudioでは関数名の最初の数文字分を手入力すれば該当する候補が表示されるので、目的の関数のマニュアルを眺めることができます。ここでは②の関数マニュアルを眺めるべく、③「dinu」まで入力してみます。

(Rで)塩基配列解析

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

①

1. イントロ | 一般 | ランダムな塩基配列を作成の4.を実行して得られたmult

タイトル通りの出現頻度です。

```
in_f <- "hoge4.fa"                          #入力ファイル名を指定して
out_f <- "hoge1.txt"                        #出力ファイル名を指定して

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

#入力ファイルの読み込み
fasta <- readDNAStringSet(in_f, format="fasta")#in_fで指定したファイルの読み込み
fasta                                       #確認してるだけです

#本番
out <- dinucleotideFrequency(fasta)         #連続塩基の出現頻度情報をoutに格納

#ファイルに保存
tmp <- cbind(names(fasta), out)             #保存したい情報をtmpに格納
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定したファイル名で保
```

③

②

トップページへ

関数マニュアル3

「①さきほどk=2で実行した例題1（dinucleotideFrequency関数を使用）」と「その前にk=1で実行した例題1（alphabetFrequency関数を使用）」とは、②で利用している関数のみが異なります。関数が異なると当然役割も異なります。RStudioでは関数名の最初の数文字分を手入力すれば該当する候補が表示されるので、目的の関数のマニュアルを眺めることができます。ここでは②の関数マニュアルを眺めるべく、③「dinu」まで入力してみます。④Console画面上で、⑤dinuまで打ち込むと…

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Untitled1* hoge1.txt hoge2.txt hoge4.fa

```
1 >contig_1
2 CGGACAGCTCCTCGGCATCCGGAT
3 >contig_2
4 GTCTGCCTCAAGCGCCCCAAGTGGGTTTGGAGGCCTAACATCGCAAGTCG
5 ACACTCAGTCCGGCCGTCTGGTTGGCAGGGGCAGAGACCCAGCACACCCT
6 GTC
```

④

Console Terminal Jobs

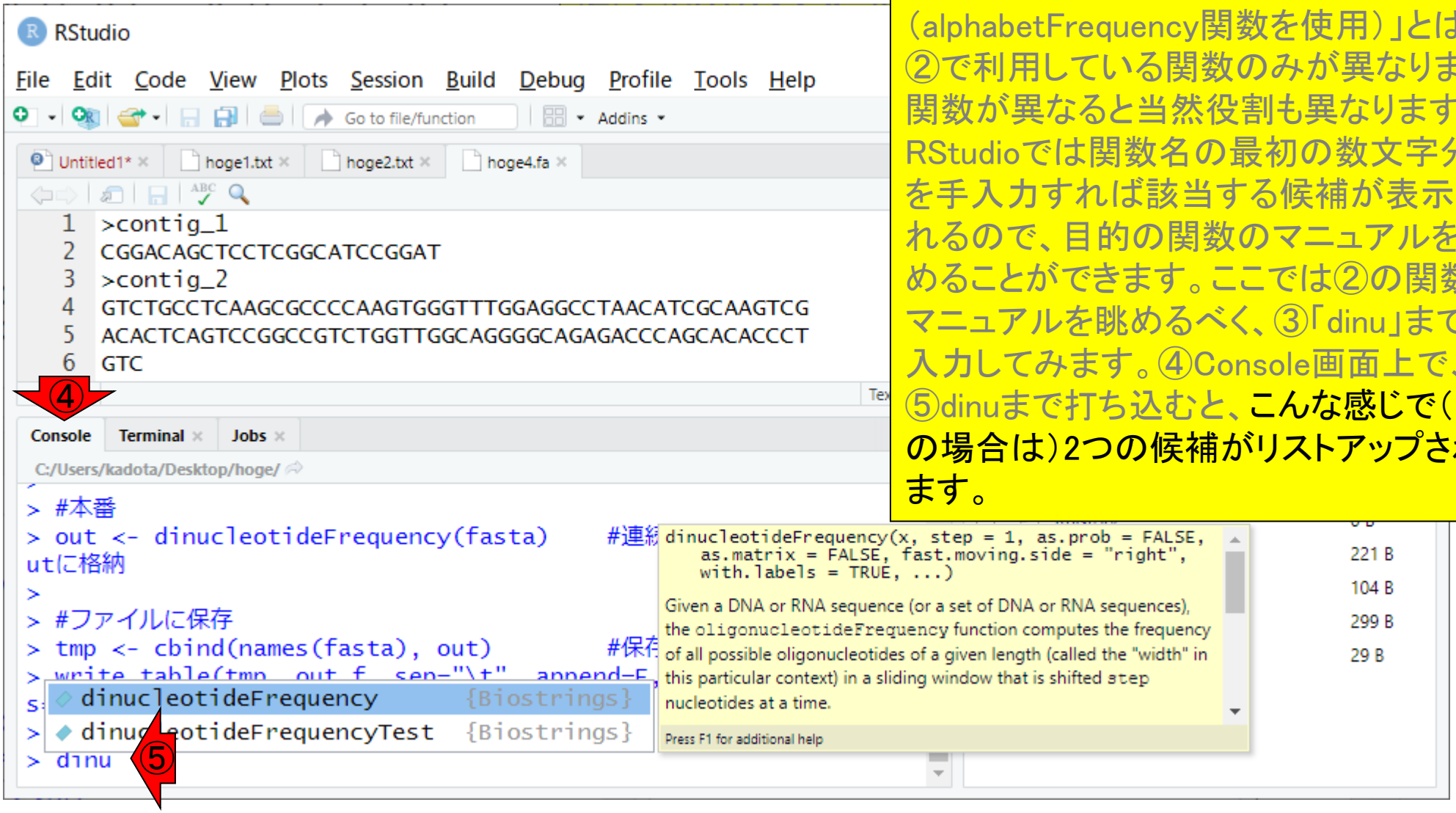
C:/Users/kadota/Desktop/hoge/

```
> #本番
> out <- dinucleotideFrequency(fasta)     #連続塩基の出現頻度情報をo
utに格納
>
> #ファイルに保存
> tmp <- cbind(names(fasta), out)          #保存したい情報をtmpに格納
> write.table(tmp, out_f, sep="\t", append=F, quote=F, row.name
s=F)#tmpの中身を指定したファイル名で保存
>
> dinu
```

⑤

Name	Size
..	
.Rhistory	0 B
hoge1.txt	221 B
hoge2.txt	104 B
hoge4.fa	299 B
seq_20.fasta	29 B

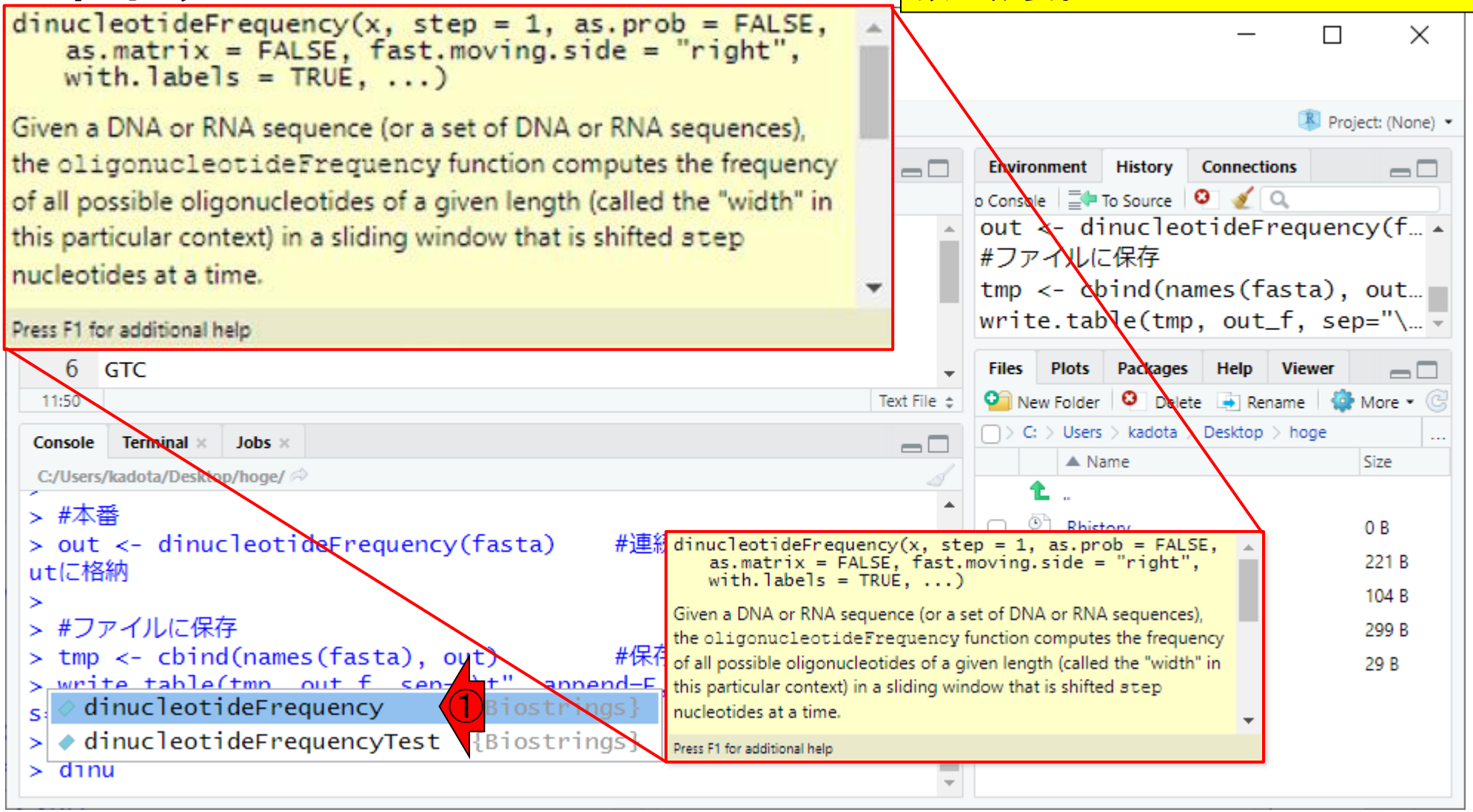
関数マニュアル4



「①さきほどk=2で実行した例題1（dinucleotideFrequency関数を使用）」と「その前にk=1で実行した例題1（alphabetFrequency関数を使用）」とは、②で利用している関数のみが異なります。関数が異なると当然役割も異なります。RStudioでは関数名の最初の数文字分を手入力すれば該当する候補が表示されるので、目的の関数のマニュアルを眺めることができます。ここでは②の関数マニュアルを眺めるべく、③「dinu」まで入力してみます。④Console画面上で、⑤dinuまで打ち込むと、こんな感じで（この場合は）2つの候補がリストアップされます。

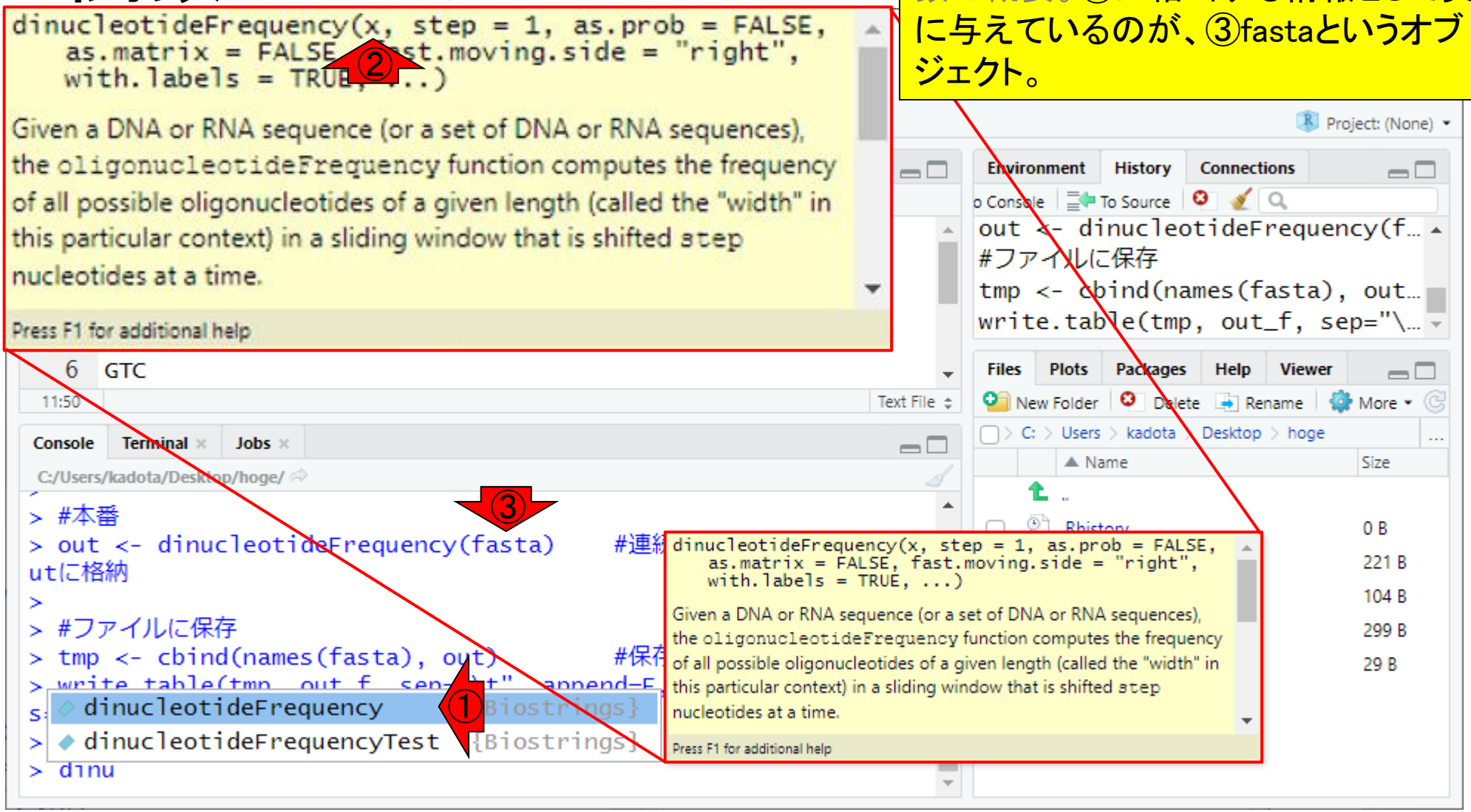
関数マニュアル5

今赤枠内で見えているのは、①でハイライトされているdinucleotideFrequency関数の概要。



関数マニュアル6

今赤枠内で見えているのは、①でハイライトされているdinucleotideFrequency関数の概要。②に相当する情報として実際に与えているのが、③fastaというオブジェクト。



関数マニュアル7

今赤枠内で見えているのは、①でハイライトされているdinucleotideFrequency関数の概要。②に相当する情報として実際に与えているのが、③fastaというオブジェクト。③は、④入力ファイル名情報の文字列である、⑤in_fを与えて読み込んだ情報。つまり実質的に入力ファイルの中身だということです。

(Rで)塩基配列解析

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

1. イントロ | 一般 | ランダムな塩基配列を作成の4.を実行して得られたmult

ル通りの出現頻度です。

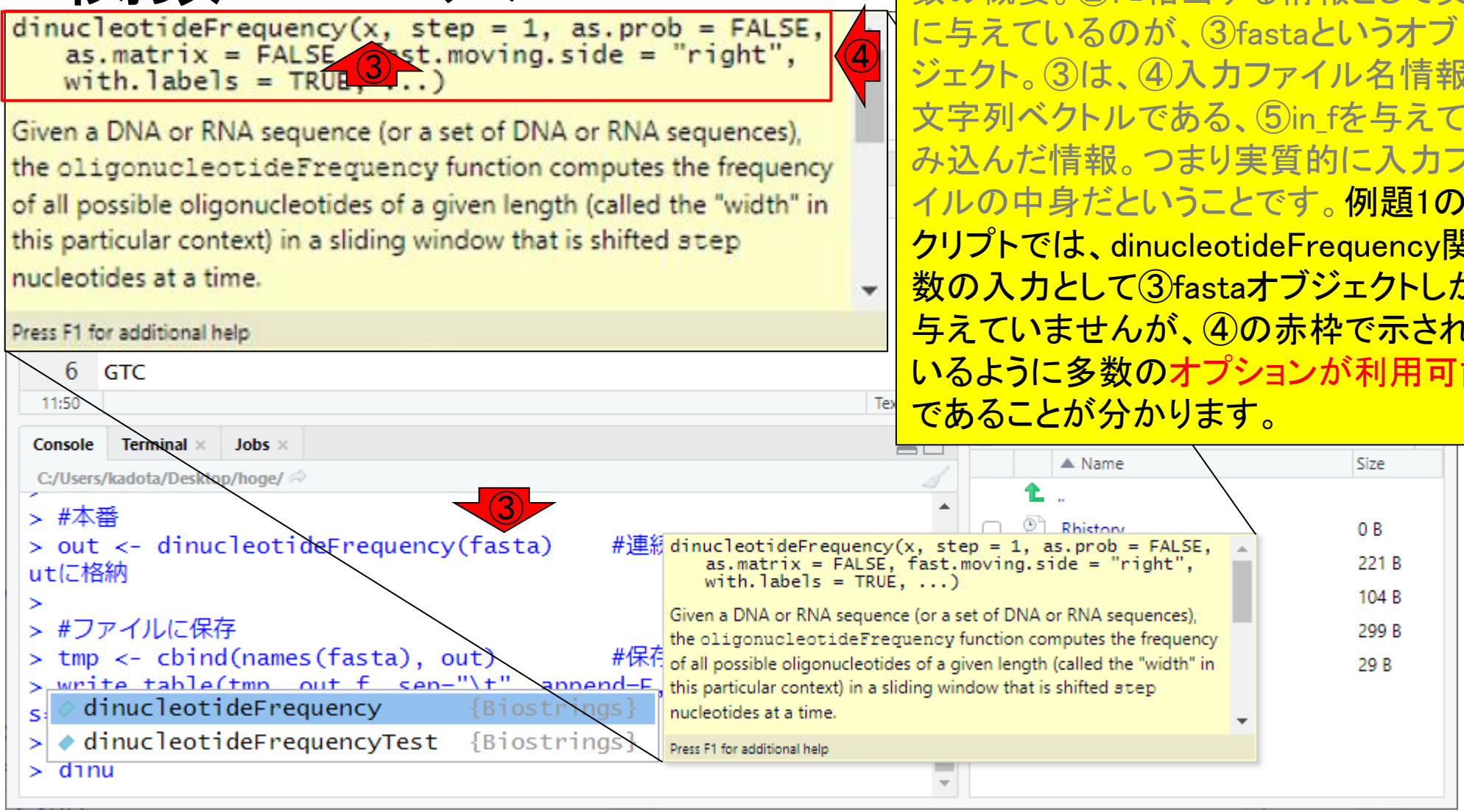
⑤ ④ ⑤ ③

```
in_f <- "hoge4.fa"                    #入力ファイル名を指定してin_fに格納
out_f <- "hoge1.txt"                  #出力ファイル名を指定してout_fに格納

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

#入力ファイルの読み込み
fasta <- readDNAStringSet(in_f, format="fasta")#in_fで指定したファイルの読み込み
fasta                                 #確認してるだけです

#本番
out <- dinucleotideFrequency(fasta)   #連続塩基の出現頻度情報をoutに格納

#ファイルに保存
tmp <- cbind(names(fasta), out)       #保存したい情報をtmpに格納
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定したファイル名で保
```

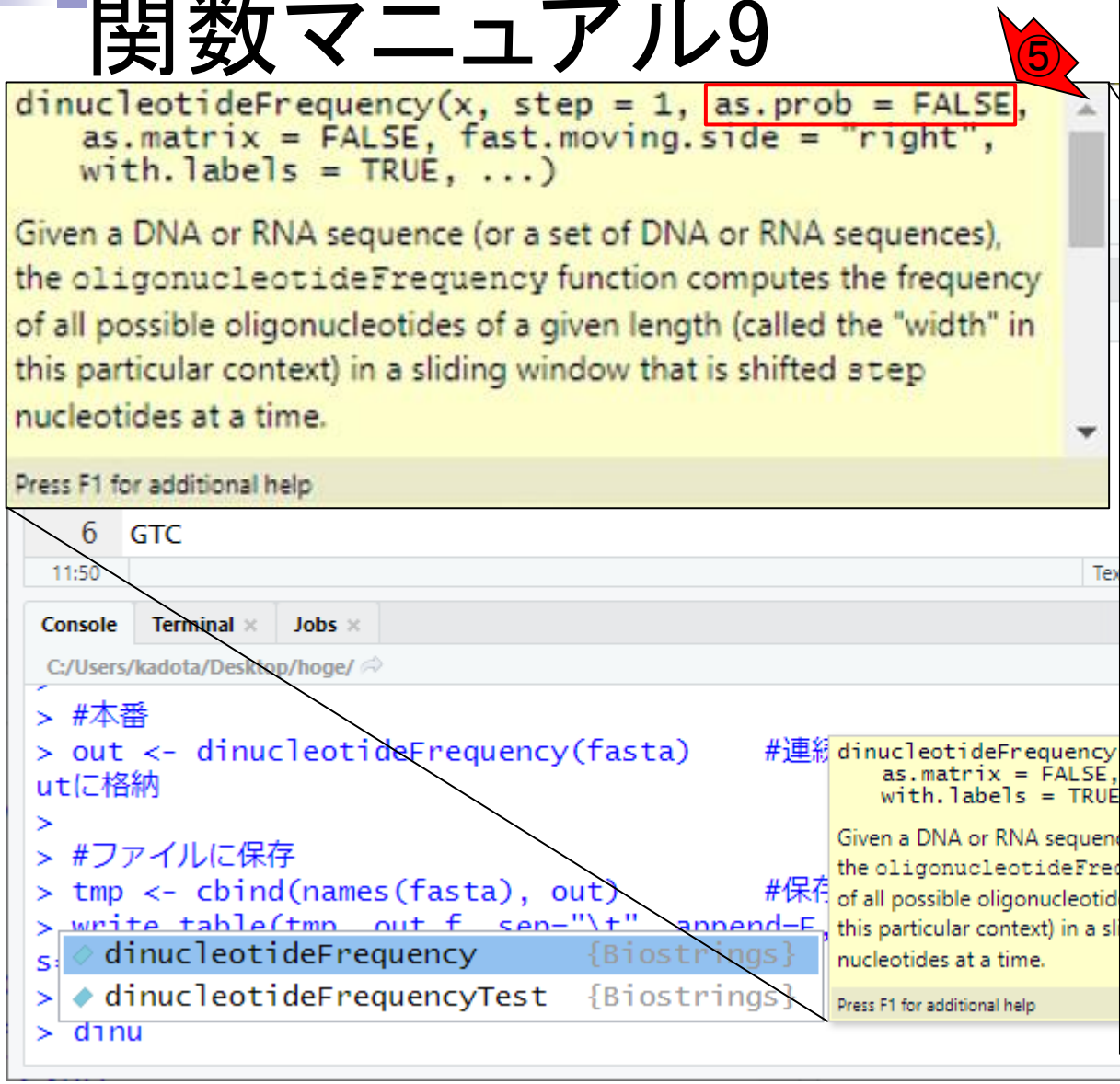
トップページへ

関数マニュアル8

今赤枠内で見えているのは、①でハイライトされているdinucleotideFrequency関数の概要。②に相当する情報として実際に与えているのが、③fastaというオブジェクト。③は、④入力ファイル名情報の文字列ベクトルである、⑤in_fを与えて読み込んだ情報。つまり実質的に入力ファイルの中身だということです。例題1のスクリプトでは、dinucleotideFrequency関数の入力として③fastaオブジェクトしか与えていませんが、④の赤枠で示されているように多数のオプションが利用可能であることが分かります。



関数マニュアル9



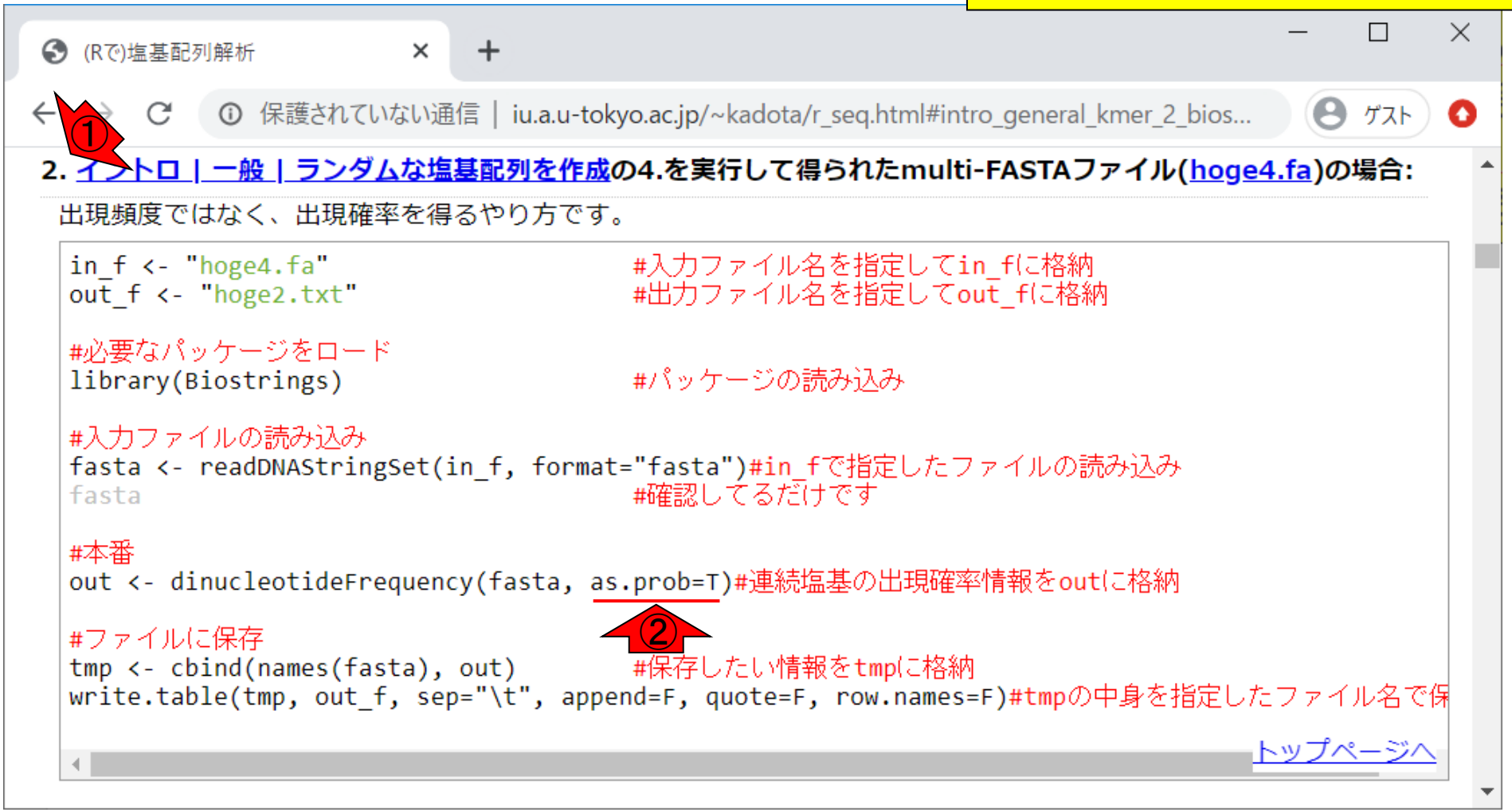
今赤枠内で見えているのは、①でハイライトされているdinucleotideFrequency関数の概要。②に相当する情報として実際に与えているのが、③fastaというオブジェクト。③は、④入力ファイル名情報の文字列ベクトルである、⑤in_fを与えて読み込んだ情報。つまり実質的に入力ファイルの中身だということです。例題1のスクリプトでは、dinucleotideFrequency関数の入力として③fastaオブジェクトしか与えていませんが、④の赤枠で示されているように多数のオプションが利用可能であることが分かります。例えば⑤as.probオプションは「(出現頻度ではなく)出現確率として(as probability)結果を返すかどうか」を指定するものです。デフォルトはFALSEなので「出現確率として返さない(つまり出現頻度として返す)」と宣言していることに相当します。これを「as.prob = TRUE(出現確率として返す)」としたものが、この項目の例題2です。

Contents

- Introduction、出現頻度解析（k=2）、出現頻度解析（k=1）
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図（例題10）、作図（例題11）、作図（例題12）
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ（例題32）、被覆率（coverage）、基本的な考え方(例題7)
  - 例題8（k=2)、例題9（k=3）、1,000塩基の仮想ゲノム（サンプルデータの例題33）
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

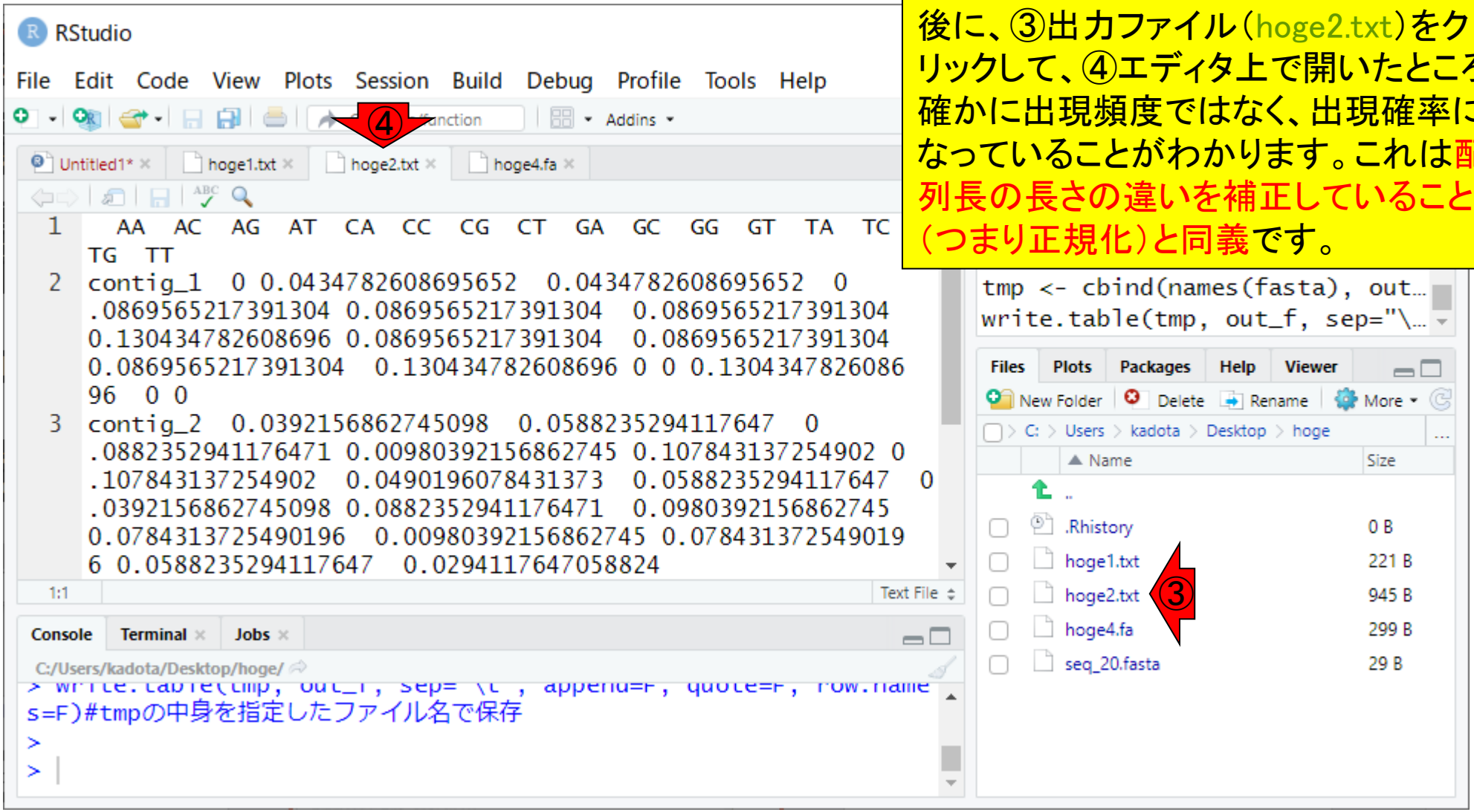
例題2を実行1

k=2の①例題2。例題1との違いは、結果を出現確率として返すべく、②as.prob=Tを追加しただけです。コピペ実行。



例題2を実行2

k=2の①例題2。例題1との違いは、結果を出現確率として返すべく、②as.prob=Tを追加しただけです。コピペ実行。実行後に、③出力ファイル(hoge2.txt)をクリックして、④エディタ上で開いたところ。確かに出現頻度ではなく、出現確率になっていることがわかります。これは配列長の長さの違いを補正していること(つまり正規化)と同義です。



例題2を実行3

k=2の①例題2のコードは、②入力ファイルを変更するだけで、例えばヒトゲノムの染色体ごとの2連続塩基の出現頻度解析を行うことも可能です。

(Rで)塩基配列解析

保護されていない通信 | iu.a.u-tokyo.ac.jp/~kadota/r_seq.html#intro_general_kmer_2_bios...

ゲスト

①

2. イントロ | 一般 | ランダムな塩基配列を作成の4.を実行して得られたmulti-FASTAファイル(hoge4.fa)の場合:

出現頻度ではなく、出現確率を得るやり方です。

②

```
in_f <- "hoge4.fa"                    #入力ファイル名を指定してin_fに格納
out_f <- "hoge2.txt"                  #出力ファイル名を指定してout_fに格納

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

#入力ファイルの読み込み
fasta <- readDNAStringSet(in_f, format="fasta")#in_fで指定したファイルの読み込み
fasta                                 #確認してるだけです

#本番
out <- dinucleotideFrequency(fasta, as.prob=T)#連続塩基の出現確率情報をoutに格納

#ファイルに保存
tmp <- cbind(names(fasta), out)       #保存したい情報をtmpに格納
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定したファイル名で保
```

トップページへ

Contents

- Introduction、出現頻度解析（k=2）、出現頻度解析（k=1）
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図（例題10）、作図（例題11）、作図（例題12）
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ（例題32）、被覆率（coverage）、基本的な考え方(例題7)
  - 例題8（k=2)、例題9（k=3）、1,000塩基の仮想ゲノム（サンプルデータの例題33）
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合
- 課題

例題7を実行1

k=2の①例題7。例題2では入力としてFASTA形式ファイルを与えたが、②ここではインストール済みのヒトゲノム情報を含むRパッケージを与えている。手順通りにR本体およびパッケージ群のインストール作業を行っていれば、特に意識せずともインストールされています。

(Rで)塩基配列解析

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

①

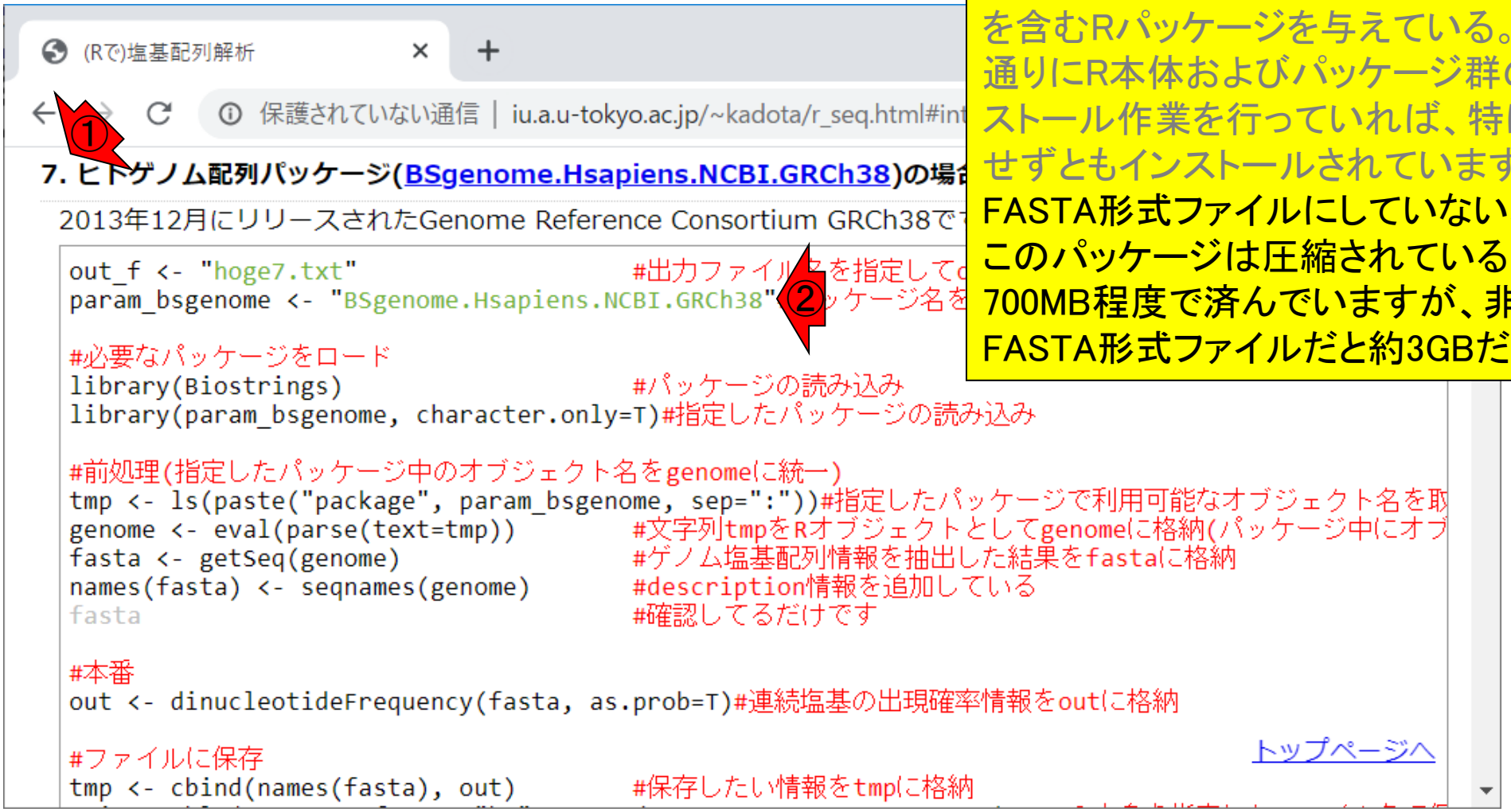
7. ヒトゲノム配列パッケージ(BSgenome.Hsapiens.NCBI.GRCh38)の場合

2013年12月にリリースされたGenome Reference Consortium GRCh38です。出力は出現確率です。

②

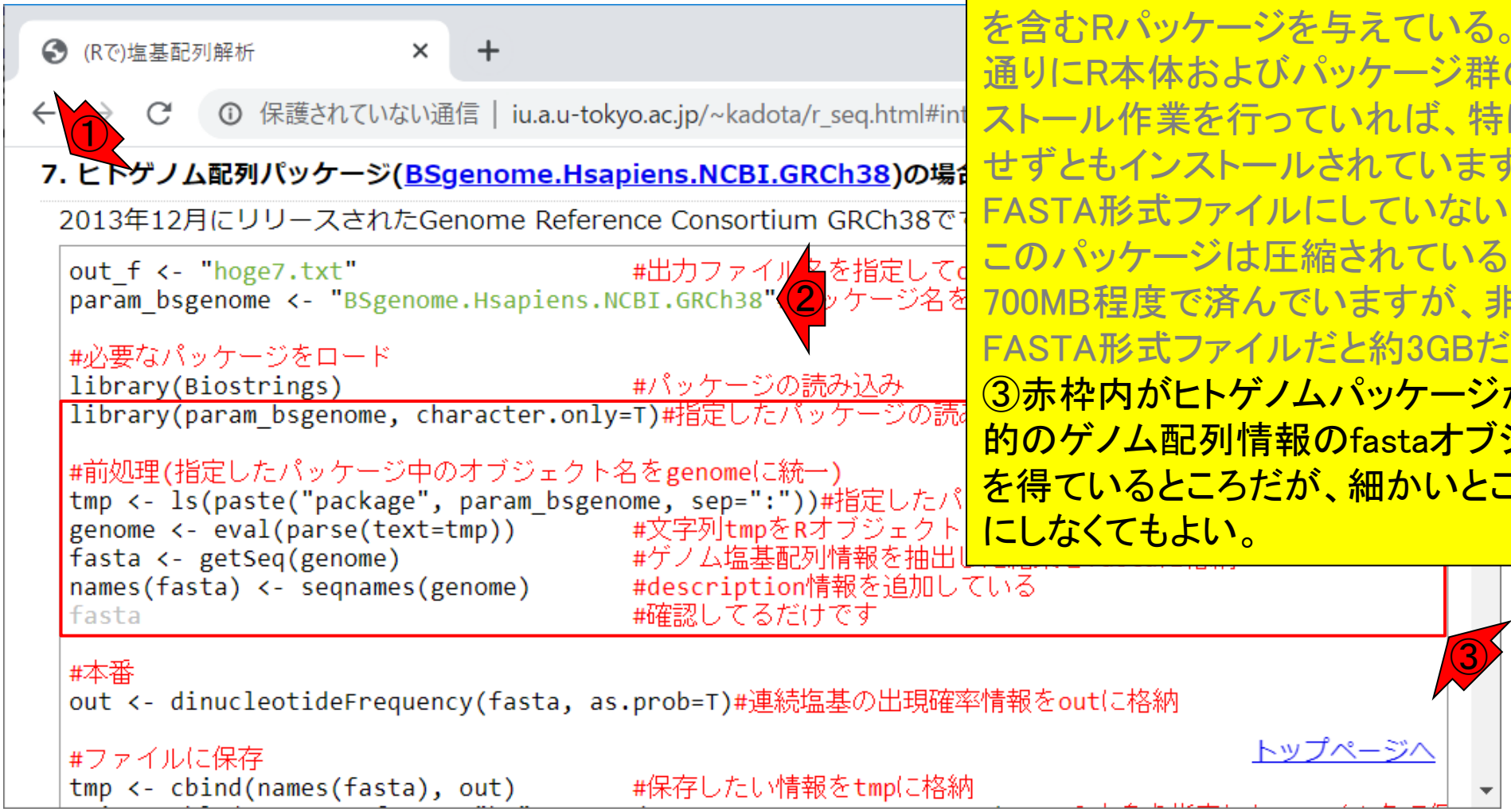
```
out_f <- "hoge7.txt"                    #出力ファイル名を指定してout_fに格納
param_bsgenome <- "BSgenome.Hsapiens.NCBI.GRCh38"ッケージ名を指定(BSgenome系のゲノムパッケージ)

#必要なパッケージをロード
library(Biostrings)                     #パッケージの読み込み
library(param_bsgenome, character.only=T)#指定したパッケージの読み込み

#前処理(指定したパッケージ中のオブジェクト名をgenomeに統一)
tmp <- ls(paste("package", param_bsgenome, sep=":"))#指定したパッケージで利用可能なオブジェクト名を取
genome <- eval(parse(text=tmp))         #文字列tmpをRオブジェクトとしてgenomeに格納(パッケージ中にオブ
fasta <- getSeq(genome)                 #ゲノム塩基配列情報を抽出した結果をfastaに格納
names(fasta) <- seqnames(genome)        #description情報を追加している
fasta                                   #確認してるだけです

#本番
out <- dinucleotideFrequency(fasta, as.prob=T)#連続塩基の出現確率情報をoutに格納

                                                                    トップページへ
#ファイルに保存
tmp <- cbind(names(fasta), out)         #保存したい情報をtmpに格納
```

例題7を実行2



k=2の①例題7。例題2では入力としてFASTA形式ファイルを与えたが、②ここではインストール済みのヒトゲノム情報を含むRパッケージを与えている。手順通りにR本体およびパッケージ群のインストール作業を行っていれば、特に意識せずともインストールされています。
FASTA形式ファイルにしていない理由は、このパッケージは圧縮されているので700MB程度で済んでいますが、非圧縮FASTA形式ファイルだと約3GBだから。

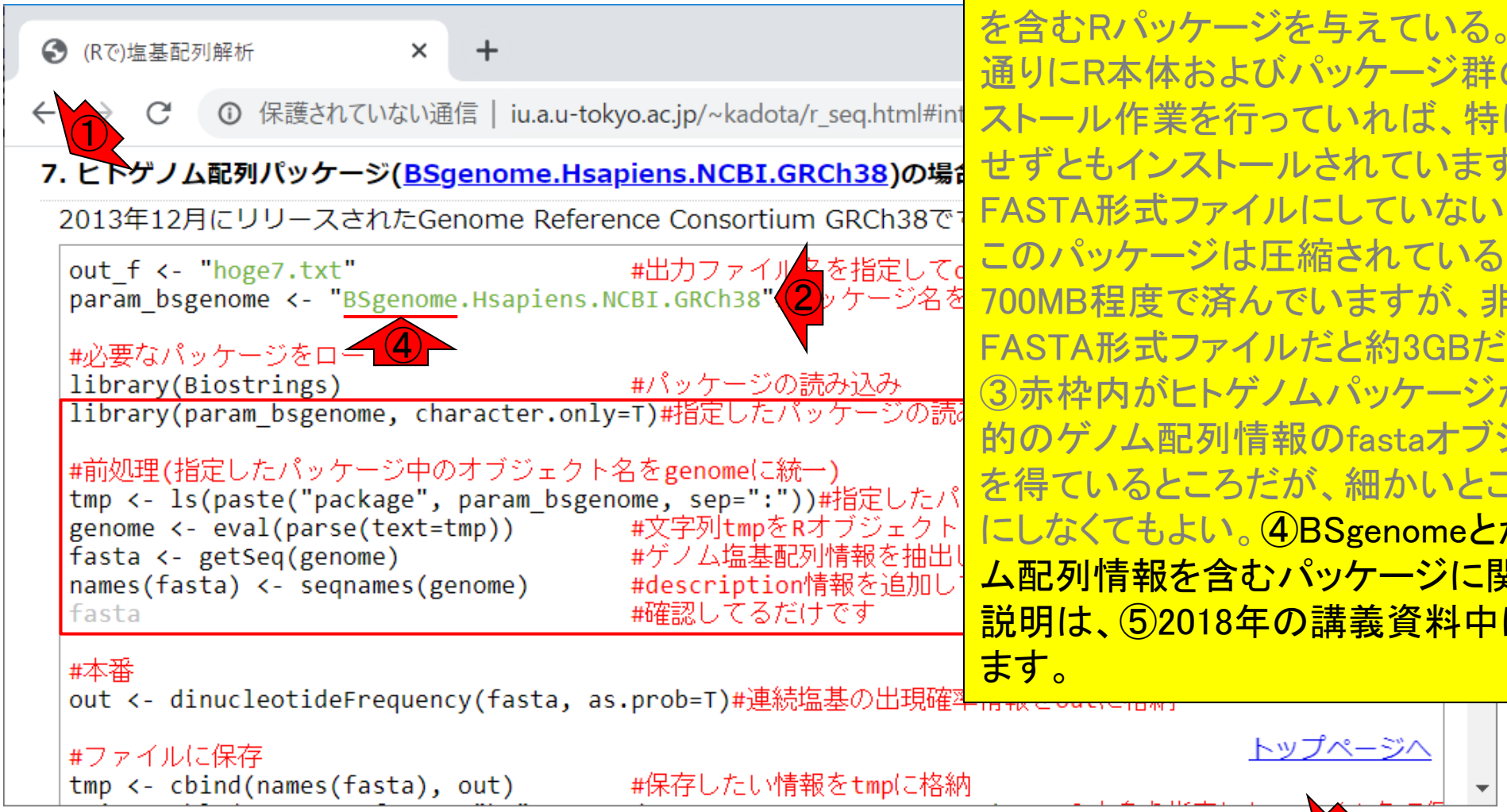
例題7を実行3



k=2の①例題7。例題2では入力としてFASTA形式ファイルを与えたが、②ここではインストール済みのヒトゲノム情報を含むRパッケージを与えている。手順通りにR本体およびパッケージ群のインストール作業を行っていれば、特に意識せずともインストールされています。FASTA形式ファイルにしていない理由は、このパッケージは圧縮されているので700MB程度で済んでいますが、非圧縮FASTA形式ファイルだと約3GBだから。
③赤枠内がヒトゲノムパッケージから目的のゲノム配列情報のfastaオブジェクトを得ているところだが、細かいところは気にしなくてもよい。

参考

例題7を実行4



k=2の①例題7。例題2では入力としてFASTA形式ファイルを与えたが、②ここではインストール済みのヒトゲノム情報を含むRパッケージを与えている。手順通りにR本体およびパッケージ群のインストール作業を行っていれば、特に意識せずともインストールされています。FASTA形式ファイルにしていない理由は、このパッケージは圧縮されているので700MB程度で済んでいますが、非圧縮FASTA形式ファイルだと約3GBだから。③赤枠内がヒトゲノムパッケージから目的のゲノム配列情報のfastaオブジェクトを得ているところだが、細かいところは気にしなくてもよい。④BSgenomeとかゲノム配列情報を含むパッケージに関する説明は、⑤2018年の講義資料中にあります。

http://www.iu.a.u-tokyo.ac.jp/~kadota/20180507_kadota.pdf

例題7を実行5

k=2の①例題7を実行すると、ヒトゲノムの染色体ごとの2連続塩基の出現確率が、②hoge7.txtに保存されます。コピペ実行。

(Rで)塩基配列解析

保護されていない通信 | iu.a.u-tokyo.ac.jp/~kadota/r_seq.html#intro_general_kmer_2_bios...

ゲスト

①

②

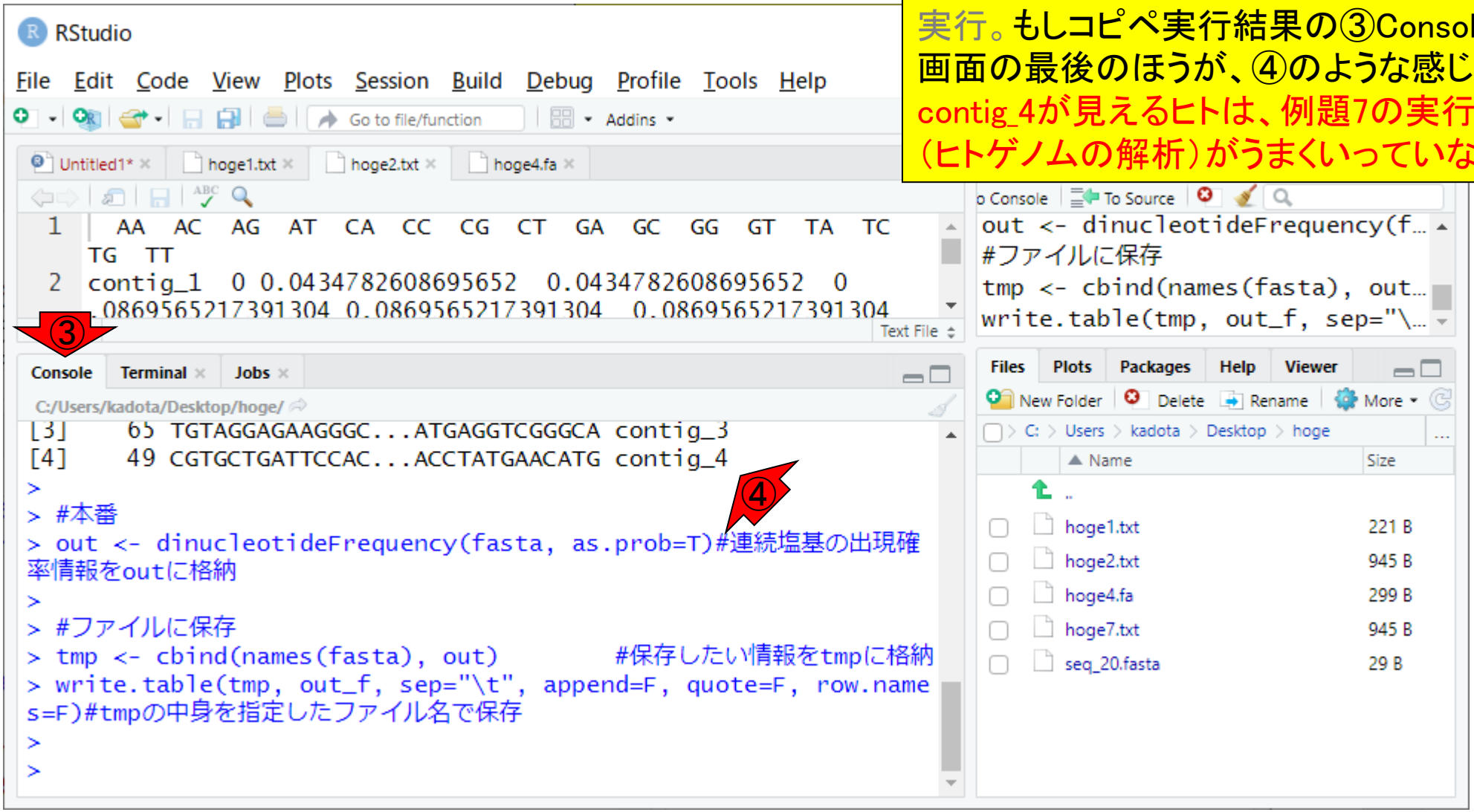
7. ヒトゲノム配列パッケージ(BSgenome.Hsapiens.NCBI.GRCh38)の場合：

2013年12月にリリースされたGenome Reference Consortium GRCh38です。出力は出現確率です。

```
out_f <- "hoge7.txt"                    #出力ファイル名を指定してout_fに格納
param_bsgenome <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定(BSgenome系のゲノムパッケージ)

#必要なパッケージをロード
library(Biostrings)                     #パッケージの読み込み
library(param_bsgenome, character.only=T)#指定したパッケージの読み込み

#前処理(指定したパッケージ中のオブジェクト名をgenomeに統一)
tmp <- ls(paste("package", param_bsgenome, sep=":"))#指定したパッケージで利用可能なオブジェクト名を取
genome <- eval(parse(text=tmp))         #文字列tmpをRオブジェクトとしてgenomeに格納(パッケージ中にオブ
fasta <- getSeq(genome)                 #ゲノム塩基配列情報を抽出した結果をfastaに格納
names(fasta) <- seqnames(genome)        #description情報を追加している
fasta                                   #確認してるだけです

#本番
out <- dinucleotideFrequency(fasta, as.prob=T)#連続塩基の出現確率情報をoutに格納

#ファイルに保存                                                    トップページへ
tmp <- cbind(names(fasta), out)         #保存したい情報をtmpに格納
```

例題7を実行6

k=2の①例題7を実行すると、ヒトゲノムの染色体ごとの2連続塩基の出現確率が、②hoge7.txtに保存されます。コピペ実行。もしコピペ実行結果の③Console画面の最後のほうが、④のような感じでcontig_4が見えるヒトは、例題7の実行（ヒトゲノムの解析）がうまくいっていない。



例題7を実行7

k=2の①例題7を実行すると、ヒトゲノムの染色体ごとの2連続塩基の出現確率が、②hoge7.txtに保存されます。コピペ実行。もしコピペ実行結果の③Console画面の最後のほうが、④のような感じでcontig_4が見えるヒトは、例題7の実行（ヒトゲノムの解析）がうまくいっていない。⑤上のほうにスクロールすると、おそらく⑥のような感じで当該パッケージが存在しない、というエラーメッセージが見られると思います。これは手順通りにインストール作業ができていないことを意味します。

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

Untitled1* | hoge1.txt | hoge2.txt | hoge4.fa

```
  AA   AC   AG   AT   CA   CC   CG   CT   GA   GC   GG   GT   TA   TC
TG   TT
contig_1   0 0.0434782608695652   0.0434782608695652   0
.0869565217391304 0.0869565217391304   0.0869565217391304
0.13043478260869 0.0869565217391304   0.0869565217391304
```

Console | Terminal | Jobs

C:/Users/kadota/Desktop/hoge/

```
> param_bsgenome <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定(BSgenome系のゲノムパッケージ)
>
> #必要なパッケージをロード
> library(Biostrings)                    #パッケージの読み込み
> library(param_bsgenome, character.only=T)#指定したパッケージの読み込み
 library(param_bsgenome, character.only = T) でエラー:
   'BSgenome.Hsapiens.NCBI.GRCh38' という名前のパッケージはありません

>
> #前処理(指定したパッケージ中のオブジェクト名をgenomeに統一)
> tmp <- ls(paste("package", param_bsgenome, sep=":"))#指定したパ
```

⑤

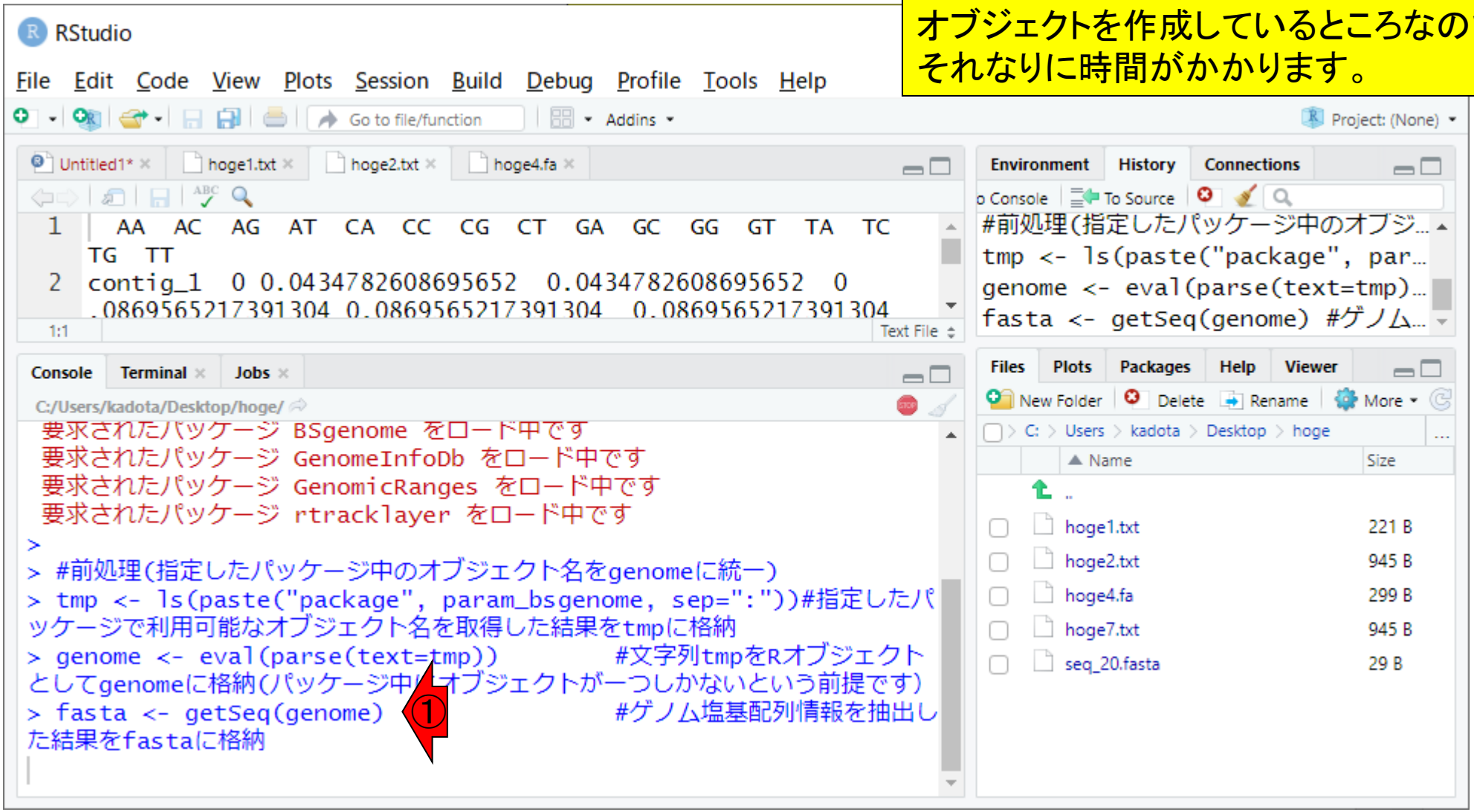
Name	Size
..	
hoge1.txt	221 B
hoge2.txt	945 B
hoge4.fa	299 B
seq_20.fasta	29 B

⑥

参考

うまく実行できているヒトは、①のあたりで数分程度かかると思います。ここがヒトゲノムパッケージからヒトゲノムのfastaオブジェクトを作成しているところなので、それなりに時間がかかります。

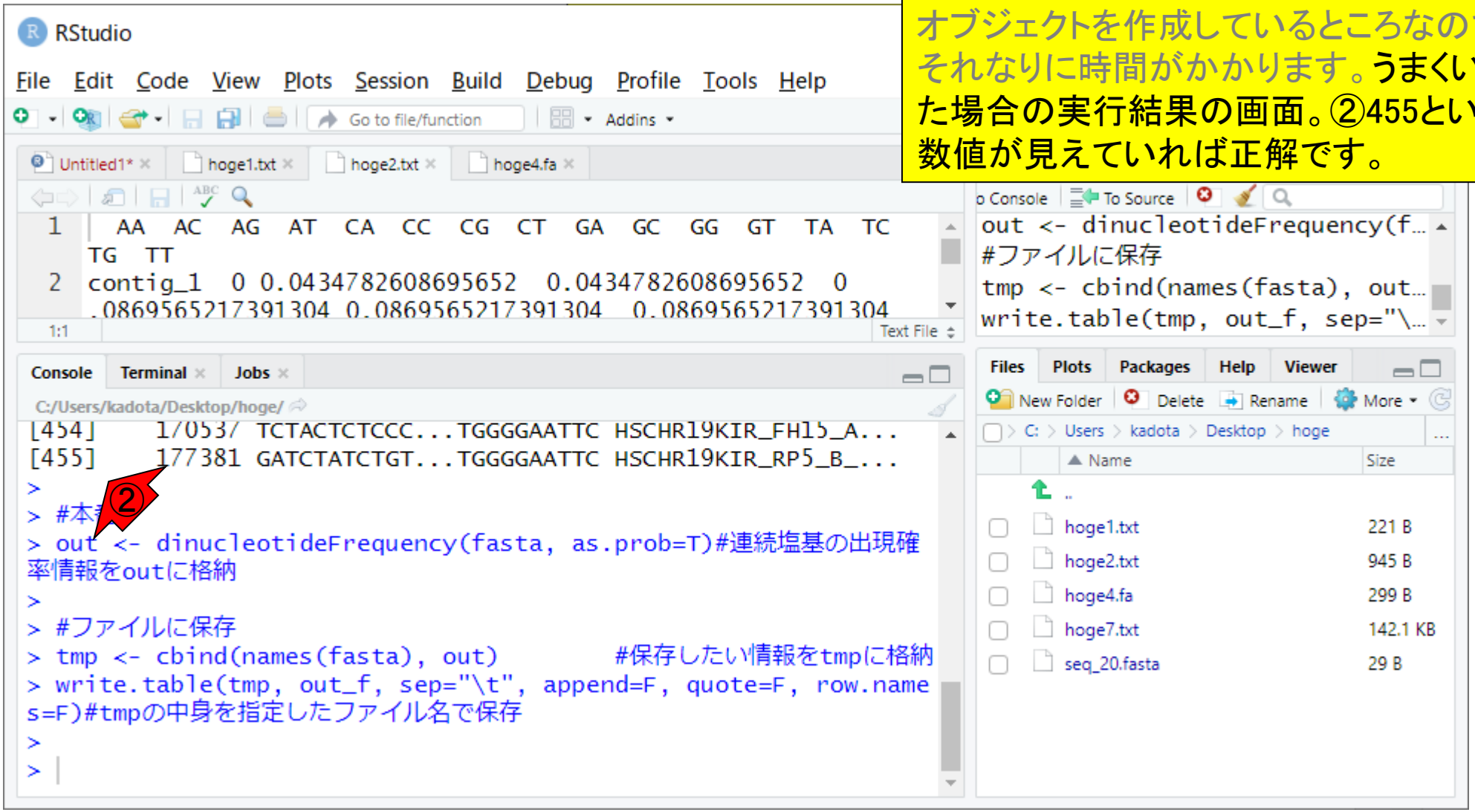
例題7を実行8



参考

例題7を実行9

うまく実行できているヒトは、①のあたりで数分程度かかると思います。ここがヒトゲノムパッケージからヒトゲノムのfastaオブジェクトを作成しているところなので、それなりに時間がかかります。うまくいった場合の実行結果の画面。②455という数値が見えていれば正解です。



例題7を実行10

例題7の出力ファイル（hoge7.txt）をExcelで開いて、数値以外の見栄えを加工したもの。

	AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
1	9.5%	5.0%	7.1%	7.4%	7.3%	5.4%	1.0%	7.1%	6.0%	4.4%	5.4%	5.0%	6.3%	6.0%	7.3%	9.6%
2	10.0%	5.0%	7.0%	7.9%	7.2%	5.0%	0.9%	7.0%	5.9%	4.1%	5.0%	5.0%	6.7%	5.9%	7.2%	10.0%
3	10.1%	5.0%	6.9%	8.0%	7.2%	4.9%	0.8%	6.9%	5.9%	4.0%	4.9%	5.0%	6.9%	5.9%	7.2%	10.2%
4	10.6%	5.0%	6.7%	8.5%	7.1%	4.5%	0.8%	6.7%	5.9%	3.8%	4.5%	5.0%	7.3%	5.8%	7.1%	10.6%
5	10.2%	5.0%	6.9%	8.1%	7.2%	4.8%	0.9%	6.9%	5.9%	4.0%	4.8%	5.1%	6.9%	5.9%	7.2%	10.3%
6	10.2%	5.0%	6.9%	8.1%	7.2%	4.8%	0.9%	6.9%	5.9%	4.0%	4.9%	5.0%	6.9%	5.9%	7.2%	10.2%
7	9.8%	5.0%	7.0%	7.7%	7.3%	5.1%	1.0%	7.0%	6.0%	4.2%	5.1%	5.1%	6.5%	5.9%	7.3%	10.0%
8	10.0%	5.1%	6.9%	7.9%	7.2%	5.0%	0.9%	6.9%	6.0%	4.1%	5.0%	5.0%	6.7%	5.9%	7.2%	10.0%
9	9.7%	5.1%	7.0%	7.6%	7.3%	5.3%	1.0%	7.0%	6.0%	4.3%	5.3%	5.0%	6.4%	6.0%	7.3%	9.7%
10	9.6%	5.0%	7.1%	7.5%	7.3%	5.3%	1.0%	7.1%	6.0%	4.4%	5.3%	5.1%	6.3%	6.0%	7.4%	9.7%
11	9.5%	5.1%	7.1%	7.5%	7.3%	5.3%	1.0%	7.1%	6.1%	4.3%	5.4%	5.0%	6.3%	6.0%	7.3%	9.6%
12	9.8%	5.0%	7.0%	7.7%	7.2%	5.1%	1.0%	7.0%	6.0%	4.2%	5.2%	5.1%	6.6%	6.0%	7.2%	9.9%
13	10.5%	5.0%	6.8%	8.4%	7.1%	4.5%	0.9%	6.7%	5.9%	3.8%	4.6%	5.0%	7.2%	5.8%	7.1%	10.6%
14	9.7%	5.0%	7.0%	7.7%	7.2%	5.1%	1.0%	7.0%	6.0%	4.2%	5.2%	5.1%	6.6%	5.9%	7.3%	9.9%
15	9.4%	5.1%	7.1%	7.3%	7.3%	5.4%	1.1%	7.1%	6.0%	4.5%	5.5%	5.1%	6.1%	6.0%	7.4%	9.5%
16	8.6%	5.1%	7.3%	6.7%	7.5%	6.1%	1.4%	7.2%	6.1%	5.0%	6.1%	5.1%	5.4%	6.1%	7.6%	8.8%
17	8.5%	5.1%	7.3%	6.4%	7.4%	6.3%	1.5%	7.4%	6.2%	5.1%	6.4%	5.0%	5.2%	6.1%	7.5%	8.6%
18	10.1%	5.1%	7.0%	7.9%	7.2%	4.7%	0.9%	6.9%	6.1%	4.0%	4.9%	5.1%	6.7%	5.9%	7.3%	10.3%
19	7.7%	5.1%	7.5%	5.7%	7.5%	7.0%	1.9%	7.4%	6.2%	5.6%	7.1%	5.2%	4.5%	6.2%	7.6%	7.9%
20	8.8%	5.0%	7.3%	6.8%	7.5%	5.8%	1.2%	7.3%	6.2%	4.8%	6.0%	5.1%	5.5%	6.1%	7.6%	9.1%
21	9.8%	5.1%	6.9%	7.7%	7.3%	5.1%	1.2%	6.9%	6.0%	4.3%	5.1%	5.1%	6.4%	6.0%	7.3%	9.9%

参考

例題7を実行11

例題7の出力ファイル(hoge7.txt)をExcelで開いて、数値以外の見栄えを加工したもの。①ヒトゲノムのドラフト配列決定論文でも言及されているが、確かに②ヒトゲノム中のCpG出現確率は低いですね。

②

	AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
1	9.5%	5.0%	7.1%	7.4%	7.3%	5.4%	1.0%	7.1%	6.0%	4.4%	5.4%	5.0%	6.3%	6.0%	7.3%	9.6%
2	10.0%	5.0%	7.0%	7.9%	7.2%	5.0%	0.9%	7.0%	5.9%	4.1%	5.0%	5.0%	6.7%	5.9%	7.2%	10.0%
3	10.1%	5.0%	6.9%	8.0%	7.2%	4.9%	0.8%	6.9%	5.9%	4.0%	4.9%	5.0%	6.9%	5.9%	7.2%	10.2%
4	10.6%	5.0%	6.7%	8.5%	7.1%	4.5%	0.8%	6.7%	5.9%	3.8%	4.5%	5.0%	7.3%	5.8%	7.1%	10.6%
5	10.2%	5.0%	6.9%	8.1%	7.2%	4.8%	0.9%	6.9%	5.9%	4.0%	4.8%	5.1%	6.9%	5.9%	7.2%	10.3%
6	10.2%	5.0%	6.9%	8.1%	7.2%	4.8%	0.9%	6.9%	5.9%	4.0%	4.9%	5.0%	6.9%	5.9%	7.2%	10.2%
7	9.8%	5.0%	7.0%	7.7%	7.3%	5.1%	1.0%	7.0%	6.0%	4.2%	5.1%	5.1%	6.5%	5.9%	7.3%	10.0%
8	10.0%	5.1%	6.9%	7.9%	7.2%	5.0%	0.9%	6.9%	6.0%	4.1%	5.0%	5.0%	6.7%	5.9%	7.2%	10.0%
9	9.7%	5.1%	7.0%	7.6%	7.3%	5.3%	1.0%	7.0%	6.0%	4.3%	5.3%	5.0%	6.4%	6.0%	7.3%	9.7%
10	9.6%	5.0%	7.1%	7.5%	7.3%	5.3%	1.0%	7.1%	6.0%	4.4%	5.3%	5.1%	6.3%	6.0%	7.4%	9.7%
11	9.5%	5.1%	7.1%	7.5%	7.3%	5.3%	1.0%	7.1%	6.1%	4.3%	5.4%	5.0%	6.3%	6.0%	7.3%	9.6%
12	9.8%	5.0%	7.0%	7.7%	7.2%	5.1%	1.0%	7.0%	6.0%	4.2%	5.2%	5.1%	6.6%	6.0%	7.2%	9.9%
13	10.5%	5.0%	6.8%	8.4%	7.1%	4.5%	0.9%	6.7%	5.9%	3.8%	4.6%	5.0%	7.2%	5.8%	7.1%	10.6%
14	9.7%	5.0%	7.0%	7.7%	7.2%	5.1%	1.0%	7.0%	6.0%	4.2%	5.2%	5.1%	6.6%	5.9%	7.3%	9.9%
15	9.4%	5.1%	7.1%	7.3%	7.3%	5.4%	1.1%	7.1%	6.0%	4.5%	5.5%	5.1%	6.1%	6.0%	7.4%	9.5%
16	8.6%	5.1%	7.3%	6.7%	7.5%	6.1%	1.4%	7.2%	6.1%	5.0%	6.1%	5.1%	5.4%	6.1%	7.6%	8.8%
17	8.5%	5.1%	7.3%	6.4%	7.4%	6.3%	1.5%	7.4%	6.2%	5.1%	6.4%	5.0%	5.2%	6.1%	7.5%	8.6%
18	10.1%	5.1%	7.0%	7.9%	7.2%	4.7%	0.9%	6.9%	6.1%	4.0%	4.9%	5.1%	6.7%	5.9%	7.3%	10.3%
19	7.7%	5.1%	7.5%	5.7%	7.5%	7.0%	1.9%	7.4%	6.2%	5.6%	7.1%	5.2%	4.5%	6.2%	7.6%	7.9%
20	8.8%	5.0%	7.3%	6.8%	7.5%	5.8%	1.2%	7.3%	6.2%	4.8%	6.0%	5.1%	5.5%	6.1%	7.6%	9.1%
21	9.8%	5.1%	6.9%	7.7%	7.3%	5.1%	1.2%	6.9%	6.0%	4.3%	5.1%	5.1%	6.4%	6.0%	7.3%	9.9%

①

Lander et al., *Nature*, **409**: 860-921, 2001

Contents

- Introduction、出現頻度解析（k=2）、出現頻度解析（k=1）
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図（例題10）、作図（例題11）、作図（例題12）
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ（例題32）、被覆率（coverage）、基本的な考え方(例題7)
  - 例題8（k=2)、例題9（k=3）、1,000塩基の仮想ゲノム（サンプルデータの例題33）
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

確率の話1

確率をおさらい。ゲノム中にA,C,G,Tの4種類しかなく、①それぞれの塩基の出現確率が等しい場合(= 1/4 = 0.25)、②任意の2連続塩基の出現確率は(0.25)^2で表される。つまり、AAの出現確率もCGの出現確率も全て1/16 = 0.0625だということ。③この条件はゲノム中のGC含量(GC contents)が50%だということと等価。

③

仮想ゲノムのGC含量を50%とすると…

(1) 1連続塩基の出現確率

A or Tの場合: 0.250

C or Gの場合: 0.250

①

(2) 2連続塩基の出現確率

AとTのみから構成される場合(AA, AT, TA, TT): 0.250×0.250 = 6.25%

CとGのみから構成される場合(CC, CG, GC, GG): 0.250×0.250 = 6.25%

それ以外(AC, AG, CA, CT, GA, GT, TC, TG): 0.250×0.250 = 6.25%

②

確率の話2

④

ヒトゲノムのGC含量は約41%なので…

(1) 1連続塩基の出現確率

A or Tの場合: 0.295

C or Gの場合: 0.205

⑤

(2) 2連続塩基の出現確率

AとTのみから構成される場合(AA, AT, TA, TT): 0.295×0.295 = 8.70%

CとGのみから構成される場合(CC, CG, GC, GG): 0.205×0.205 = 4.20%

それ以外(AC, AG, CA, CT, GA, GT, TC, TG): 0.205×0.295 = 6.05%

⑥

現実のゲノムはぴったりGC含量 = 50%というわけではない。④例えばヒトゲノムの場合は約41%なので、⑤C or Gの出現確率は0.41/2 = 0.205となる。必然的に残りのA or Tの出現確率は(1.00 − 0.41)/2 = 0.295となる。したがって、2連続塩基の出現確率(期待値)も、⑥構成塩基によって異なる。GC含量が50%よりも低いと、2連続塩基の出現確率は「CとGのみから構成される場合」に最も低くなる。

確率の話3

先ほどのヒトゲノムの染色体ごとの2連続塩基の出現確率(観測値)を、CとGを赤字にして再掲。

	AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
1	9.5%	5.0%	7.1%	7.4%	7.3%	5.4%	1.0%	7.1%	6.0%	4.4%	5.4%	5.0%	6.3%	6.0%	7.3%	9.6%
2	10.0%	5.0%	7.0%	7.9%	7.2%	5.0%	0.9%	7.0%	5.9%	4.1%	5.0%	5.0%	6.7%	5.9%	7.2%	10.0%
3	10.1%	5.0%	6.9%	8.0%	7.2%	4.9%	0.8%	6.9%	5.9%	4.0%	4.9%	5.0%	6.9%	5.9%	7.2%	10.2%
4	10.6%	5.0%	6.7%	8.5%	7.1%	4.5%	0.8%	6.7%	5.9%	3.8%	4.5%	5.0%	7.3%	5.8%	7.1%	10.6%
5	10.2%	5.0%	6.9%	8.1%	7.2%	4.8%	0.9%	6.9%	5.9%	4.0%	4.8%	5.1%	6.9%	5.9%	7.2%	10.3%
6	10.2%	5.0%	6.9%	8.1%	7.2%	4.8%	0.9%	6.9%	5.9%	4.0%	4.9%	5.0%	6.9%	5.9%	7.2%	10.2%
7	9.8%	5.0%	7.0%	7.7%	7.3%	5.1%	1.0%	7.0%	6.0%	4.2%	5.1%	5.1%	6.5%	5.9%	7.3%	10.0%
8	10.0%	5.1%	6.9%	7.9%	7.2%	5.0%	0.9%	6.9%	6.0%	4.1%	5.0%	5.0%	6.7%	5.9%	7.2%	10.0%
9	9.7%	5.1%	7.0%	7.6%	7.3%	5.3%	1.0%	7.0%	6.0%	4.3%	5.3%	5.0%	6.4%	6.0%	7.3%	9.7%
10	9.6%	5.0%	7.1%	7.5%	7.3%	5.3%	1.0%	7.1%	6.0%	4.4%	5.3%	5.1%	6.3%	6.0%	7.4%	9.7%
11	9.5%	5.1%	7.1%	7.5%	7.3%	5.3%	1.0%	7.1%	6.1%	4.3%	5.4%	5.0%	6.3%	6.0%	7.3%	9.6%
12	9.8%	5.0%	7.0%	7.7%	7.2%	5.1%	1.0%	7.0%	6.0%	4.2%	5.2%	5.1%	6.6%	6.0%	7.2%	9.9%
13	10.5%	5.0%	6.8%	8.4%	7.1%	4.5%	0.9%	6.7%	5.9%	3.8%	4.6%	5.0%	7.2%	5.8%	7.1%	10.6%
14	9.7%	5.0%	7.0%	7.7%	7.2%	5.1%	1.0%	7.0%	6.0%	4.2%	5.2%	5.1%	6.6%	5.9%	7.3%	9.9%
15	9.4%	5.1%	7.1%	7.3%	7.3%	5.4%	1.1%	7.1%	6.0%	4.5%	5.5%	5.1%	6.1%	6.0%	7.4%	9.5%
16	8.6%	5.1%	7.3%	6.7%	7.5%	6.1%	1.4%	7.2%	6.1%	5.0%	6.1%	5.1%	5.4%	6.1%	7.6%	8.8%
17	8.5%	5.1%	7.3%	6.4%	7.4%	6.3%	1.5%	7.4%	6.2%	5.1%	6.4%	5.0%	5.2%	6.1%	7.5%	8.6%
18	10.1%	5.1%	7.0%	7.9%	7.2%	4.7%	0.9%	6.9%	6.1%	4.0%	4.9%	5.1%	6.7%	5.9%	7.3%	10.3%
19	7.7%	5.1%	7.5%	5.7%	7.5%	7.0%	1.9%	7.4%	6.2%	5.6%	7.1%	5.2%	4.5%	6.2%	7.6%	7.9%
20	8.8%	5.0%	7.3%	6.8%	7.5%	5.8%	1.2%	7.3%	6.2%	4.8%	6.0%	5.1%	5.5%	6.1%	7.6%	9.1%
21	9.8%	5.1%	6.9%	7.7%	7.3%	5.1%	1.2%	6.9%	6.0%	4.3%	5.1%	5.1%	6.4%	6.0%	7.3%	9.9%

①

Lander et al., *Nature*, **409**: 860-921, 2001

確率の話4

先ほどのヒトゲノムの染色体ごとの2連続塩基の出現確率(観測値)を、CとGを赤字にして再掲。さらに構成塩基別にソート後。

	AA	AT	TA	TT	AC	AG	CA	CT	GA	GT	TC	TG	CC	CG	GC	GG
1	9.5%	7.4%	6.3%	9.6%	5.0%	7.1%	7.3%	7.1%	6.0%	5.0%	6.0%	7.3%	5.4%	1.0%	4.4%	5.4%
2	10.0%	7.9%	6.7%	10.0%	5.0%	7.0%	7.2%	7.0%	5.9%	5.0%	5.9%	7.2%	5.0%	0.9%	4.1%	5.0%
3	10.1%	8.0%	6.9%	10.2%	5.0%	6.9%	7.2%	6.9%	5.9%	5.0%	5.9%	7.2%	4.9%	0.8%	4.0%	4.9%
4	10.6%	8.5%	7.3%	10.6%	5.0%	6.7%	7.1%	6.7%	5.9%	5.0%	5.8%	7.1%	4.5%	0.8%	3.8%	4.5%
5	10.2%	8.1%	6.9%	10.3%	5.0%	6.9%	7.2%	6.9%	5.9%	5.1%	5.9%	7.2%	4.8%	0.9%	4.0%	4.8%
6	10.2%	8.1%	6.9%	10.2%	5.0%	6.9%	7.2%	6.9%	5.9%	5.0%	5.9%	7.2%	4.8%	0.9%	4.0%	4.9%
7	9.8%	7.7%	6.5%	10.0%	5.0%	7.0%	7.3%	7.0%	6.0%	5.1%	5.9%	7.3%	5.1%	1.0%	4.2%	5.1%
8	10.0%	7.9%	6.7%	10.0%	5.1%	6.9%	7.2%	6.9%	6.0%	5.0%	5.9%	7.2%	5.0%	0.9%	4.1%	5.0%
9	9.7%	7.6%	6.4%	9.7%	5.1%	7.0%	7.3%	7.0%	6.0%	5.0%	6.0%	7.3%	5.3%	1.0%	4.3%	5.3%
10	9.6%	7.5%	6.3%	9.7%	5.0%	7.1%	7.3%	7.1%	6.0%	5.1%	6.0%	7.4%	5.3%	1.0%	4.4%	5.3%
11	9.5%	7.5%	6.3%	9.6%	5.1%	7.1%	7.3%	7.1%	6.1%	5.0%	6.0%	7.3%	5.3%	1.0%	4.3%	5.4%
12	9.8%	7.7%	6.6%	9.9%	5.0%	7.0%	7.2%	7.0%	6.0%	5.1%	6.0%	7.2%	5.1%	1.0%	4.2%	5.2%
13	10.5%	8.4%	7.2%	10.6%	5.0%	6.8%	7.1%	6.7%	5.9%	5.0%	5.8%	7.1%	4.5%	0.9%	3.8%	4.6%
14	9.7%	7.7%	6.6%	9.9%	5.0%	7.0%	7.2%	7.0%	6.0%	5.1%	5.9%	7.3%	5.1%	1.0%	4.2%	5.2%
15	9.4%	7.3%	6.1%	9.5%	5.1%	7.1%	7.3%	7.1%	6.0%	5.1%	6.0%	7.4%	5.4%	1.1%	4.5%	5.5%
16	8.6%	6.7%	5.4%	8.8%	5.1%	7.3%	7.5%	7.2%	6.1%	5.1%	6.1%	7.6%	6.1%	1.4%	5.0%	6.1%
17	8.5%	6.4%	5.2%	8.6%	5.1%	7.3%	7.4%	7.4%	6.2%	5.0%	6.1%	7.5%	6.3%	1.5%	5.1%	6.4%
18	10.1%	7.9%	6.7%	10.3%	5.1%	7.0%	7.2%	6.9%	6.1%	5.1%	5.9%	7.3%	4.7%	0.9%	4.0%	4.9%
19	7.7%	5.7%	4.5%	7.9%	5.1%	7.5%	7.5%	7.4%	6.2%	5.2%	6.2%	7.6%	7.0%	1.9%	5.6%	7.1%
20	8.8%	6.8%	5.5%	9.1%	5.0%	7.3%	7.5%	7.3%	6.2%	5.1%	6.1%	7.6%	5.8%	1.2%	4.8%	6.0%
21	9.8%	7.7%	6.4%	9.9%	5.1%	6.9%	7.3%	6.9%	6.0%	5.1%	6.0%	7.3%	5.1%	1.2%	4.3%	5.1%

確率の話5

①AとTのみから構成される場合は8.70%、
②CとGのみから構成される場合は4.20%、
③それ以外は6.05%、というのが期待値。

①

③

②

	AA	AT	TA	TT	AC	AG	CA	CT	GA	GT	TC	TG	CC	CG	GC	GG
1	9.5%	7.4%	6.3%	9.6%	5.0%	7.1%	7.3%	7.1%	6.0%	5.0%	6.0%	7.3%	5.4%	1.0%	4.4%	5.4%
2	10.0%	7.9%	6.7%	10.0%	5.0%	7.0%	7.2%	7.0%	5.9%	5.0%	5.9%	7.2%	5.0%	0.9%	4.1%	5.0%
3	10.1%	8.0%	6.9%	10.2%	5.0%	6.9%	7.2%	6.9%	5.9%	5.0%	5.9%	7.2%	4.9%	0.8%	4.0%	4.9%
4	10.6%	8.5%	7.3%	10.6%	5.0%	6.7%	7.1%	6.7%	5.9%	5.0%	5.8%	7.1%	4.5%	0.8%	3.8%	4.5%
5	10.2%	8.1%	6.9%	10.3%	5.0%	6.9%	7.2%	6.9%	5.9%	5.1%	5.9%	7.2%	4.8%	0.9%	4.0%	4.8%
6	10.2%	8.1%	6.9%	10.2%	5.0%	6.9%	7.2%	6.9%	5.9%	5.0%	5.9%	7.2%	4.8%	0.9%	4.0%	4.9%
7	9.8%	7.7%	6.5%	10.0%	5.0%	7.0%	7.3%	7.0%	6.0%	5.1%	5.9%	7.3%	5.1%	1.0%	4.2%	5.1%
8	10.0%	7.9%	6.7%	10.0%	5.1%	6.9%	7.2%	6.9%	6.0%	5.0%	5.9%	7.2%	5.0%	0.9%	4.1%	5.0%
9	9.7%	7.6%	6.4%	9.7%	5.1%	7.0%	7.3%	7.0%	6.0%	5.0%	6.0%	7.3%	5.3%	1.0%	4.3%	5.3%
10	9.6%	7.5%	6.3%	9.7%	5.0%	7.1%	7.3%	7.1%	6.0%	5.1%	6.0%	7.4%	5.3%	1.0%	4.4%	5.3%
11	9.5%	7.5%	6.3%	9.6%	5.1%	7.1%	7.3%	7.1%	6.1%	5.0%	6.0%	7.3%	5.3%	1.0%	4.3%	5.4%
12	9.8%	7.7%	6.6%	9.9%	5.0%	7.0%	7.2%	7.0%	6.0%	5.1%	6.0%	7.2%	5.1%	1.0%	4.2%	5.2%
13	10.5%	8.4%	7.2%	10.6%	5.0%	6.8%	7.1%	6.7%	5.9%	5.0%	5.8%	7.1%	4.5%	0.9%	3.8%	4.6%
14	9.7%	7.7%	6.6%	9.9%	5.0%	7.0%	7.2%	7.0%	6.0%	5.1%	5.9%	7.3%	5.1%	1.0%	4.2%	5.2%
15	9.4%	7.3%	6.1%	9.5%	5.1%	7.1%	7.3%	7.1%	6.0%	5.1%	6.0%	7.4%	5.4%	1.1%	4.5%	5.5%
16	8.6%	6.7%	5.4%	8.8%	5.1%	7.3%	7.5%	7.2%	6.1%	5.1%	6.1%	7.6%	6.1%	1.4%	5.0%	6.1%
17	8.5%	6.4%	5.2%	8.6%	5.1%	7.3%	7.4%	7.4%	6.2%	5.0%	6.1%	7.5%	6.3%	1.5%	5.1%	6.4%
18	10.1%	7.9%	6.7%	10.3%	5.1%	7.0%	7.2%	6.9%	6.1%	5.1%	5.9%	7.3%	4.7%	0.9%	4.0%	4.9%
19	7.7%	5.7%	4.5%	7.9%	5.1%	7.5%	7.5%	7.4%	6.2%	5.2%	6.2%	7.6%	7.0%	1.9%	5.6%	7.1%
20	8.8%	6.8%	5.5%	9.1%	5.0%	7.3%	7.5%	7.3%	6.2%	5.1%	6.1%	7.6%	5.8%	1.2%	4.8%	6.0%
21	9.8%	7.7%	6.4%	9.9%	5.1%	6.9%	7.3%	6.9%	6.0%	5.1%	6.0%	7.3%	5.1%	1.2%	4.3%	5.1%

確率の話6

④CGの観測値（約1%）は、その期待値（4.20%）より非常に低い！

	AA	AT	TA	TT	AC	AG	CA	CT	GA	GT	TC	TG	CC	CG	GC	GG
1	9.5%	7.4%	6.3%	9.6%	5.0%	7.1%	7.3%	7.1%	6.0%	5.0%	6.0%	7.3%	5.4%	1.0%	4.4%	5.4%
2	10.0%	7.9%	6.7%	10.0%	5.0%	7.0%	7.2%	7.0%	5.9%	5.0%	5.9%	7.2%	5.0%	0.9%	4.1%	5.0%
3	10.1%	8.0%	6.9%	10.2%	5.0%	6.9%	7.2%	6.9%	5.9%	5.0%	5.9%	7.2%	4.9%	0.8%	4.0%	4.9%
4	10.6%	8.5%	7.3%	10.6%	5.0%	6.7%	7.1%	6.7%	5.9%	5.0%	5.8%	7.1%	4.5%	0.8%	3.8%	4.5%
5	10.2%	8.1%	6.9%	10.3%	5.0%	6.9%	7.2%	6.9%	5.9%	5.1%	5.9%	7.2%	4.8%	0.9%	4.0%	4.8%
6	10.2%	8.1%	6.9%	10.2%	5.0%	6.9%	7.2%	6.9%	5.9%	5.0%	5.9%	7.2%	4.8%	0.9%	4.0%	4.9%
7	9.8%	7.7%	6.5%	10.0%	5.0%	7.0%	7.3%	7.0%	6.0%	5.1%	5.9%	7.3%	5.1%	1.0%	4.2%	5.1%
8	10.0%	7.9%	6.7%	10.0%	5.1%	6.9%	7.2%	6.9%	6.0%	5.0%	5.9%	7.2%	5.0%	0.9%	4.1%	5.0%
9	9.7%	7.6%	6.4%	9.7%	5.1%	7.0%	7.3%	7.0%	6.0%	5.0%	6.0%	7.3%	5.3%	1.0%	4.3%	5.3%
10	9.6%	7.5%	6.3%	9.7%	5.0%	7.1%	7.3%	7.1%	6.0%	5.1%	6.0%	7.4%	5.3%	1.0%	4.4%	5.3%
11	9.5%	7.5%	6.3%	9.6%	5.1%	7.1%	7.3%	7.1%	6.1%	5.0%	6.0%	7.3%	5.3%	1.0%	4.3%	5.4%
12	9.8%	7.7%	6.6%	9.9%	5.0%	7.0%	7.2%	7.0%	6.0%	5.1%	6.0%	7.2%	5.1%	1.0%	4.2%	5.2%
13	10.5%	8.4%	7.2%	10.6%	5.0%	6.8%	7.1%	6.7%	5.9%	5.0%	5.8%	7.1%	4.5%	0.9%	3.8%	4.6%
14	9.7%	7.7%	6.6%	9.9%	5.0%	7.0%	7.2%	7.0%	6.0%	5.1%	5.9%	7.3%	5.1%	1.0%	4.2%	5.2%
15	9.4%	7.3%	6.1%	9.5%	5.1%	7.1%	7.3%	7.1%	6.0%	5.1%	6.0%	7.4%	5.4%	1.1%	4.5%	5.5%
16	8.6%	6.7%	5.4%	8.8%	5.1%	7.3%	7.5%	7.2%	6.1%	5.1%	6.1%	7.6%	6.1%	1.4%	5.0%	6.1%
17	8.5%	6.4%	5.2%	8.6%	5.1%	7.3%	7.4%	7.4%	6.2%	5.0%	6.1%	7.5%	6.3%	1.5%	5.1%	6.4%
18	10.1%	7.9%	6.7%	10.3%	5.1%	7.0%	7.2%	6.9%	6.1%	5.1%	5.9%	7.3%	4.7%	0.9%	4.0%	4.9%
19	7.7%	5.7%	4.5%	7.9%	5.1%	7.5%	7.5%	7.4%	6.2%	5.2%	6.2%	7.6%	7.0%	1.9%	5.6%	7.1%
20	8.8%	6.8%	5.5%	9.1%	5.0%	7.3%	7.5%	7.3%	6.2%	5.1%	6.1%	7.6%	5.8%	1.2%	4.8%	6.0%
21	9.8%	7.7%	6.4%	9.9%	5.1%	6.9%	7.3%	6.9%	6.0%	5.1%	6.0%	7.3%	5.1%	1.2%	4.3%	5.1%

確率の話7

④CGの観測値（約1%）は、その期待値（4.20%）より非常に低い！⑤原著論文中の、⑥の項目中の赤下線部分を実際に確認したことに相当します。

	AA	AT	TA	TT	AC	AG	CA	CT	GA	GT	TC	TG	CC	CG	GC	GG
1	9.5%	7.4%	6.3%	9.6%	5.0%	7.1%	7.3%	7.1%	6.0%	5.0%	6.0%	7.3%	5.4%	1.0%	4.4%	5.4%
2	10.0%	7.9%	6.7%	10.0%	5.0%	7.0%	7.2%	7.0%	5.9%	5.0%	5.9%	7.2%	5.0%	0.9%	4.1%	5.0%
3	10.1%	8.0%	6.9%	10.2%	5.0%	6.9%	7.2%	6.9%	5.9%	5.0%	5.9%	7.2%	4.9%	0.8%	4.0%	4.9%
4	10.6%	8.5%	7.3%	10.6%	5.0%	6.7%	7.1%	6.7%	5.9%	5.0%	5.8%	7.1%	4.5%	0.8%	3.8%	4.5%
5	10.2%	8.1%	6.9%	10.3%	5.0%	6.9%	7.2%	6.9%	5.9%	5.1%	5.9%	7.2%	4.8%	0.9%	4.0%	4.8%
														0.9%	4.0%	4.9%
														1.0%	4.2%	5.1%
														0.9%	4.1%	5.0%
														1.0%	4.3%	5.3%
														1.0%	4.4%	5.3%
														1.0%	4.3%	5.4%
														1.0%	4.2%	5.2%
														0.9%	3.8%	4.6%
														1.0%	4.2%	5.2%
														1.1%	4.5%	5.5%
														1.4%	5.0%	6.1%
17	8.5%	6.4%	5.2%	8.6%	5.1%	7.3%	7.4%	7.4%	6.2%	5.0%	6.1%	7.5%	6.3%	1.5%	5.1%	6.4%
18	10.1%	7.9%	6.7%	10.3%	5.1%	7.0%	7.2%	6.9%	6.1%	5.1%	5.9%	7.3%	4.7%	0.9%	4.0%	4.9%
19	7.7%	5.7%	4.5%	7.9%	5.1%	7.5%	7.5%	7.4%	6.2%	5.2%	6.2%	7.6%	7.0%	1.9%	5.6%	7.1%
20	8.8%	6.8%	5.5%	9.1%	5.0%	7.3%	7.5%	7.3%	6.2%	5.1%	6.1%	7.6%	5.8%	1.2%	4.8%	6.0%
21	9.8%	7.7%	6.4%	9.9%	5.1%	6.9%	7.3%	6.9%	6.0%	5.1%	6.0%	7.3%	5.1%	1.2%	4.3%	5.1%

⑥

CpG islands

A related topic is the distribution of so-called CpG islands across the genome. The dinucleotide CpG is notable because it is greatly under-represented in human DNA, occurring at only about one-fifth of the roughly 4% frequency that would be expected by simply multiplying the typical fraction of Cs and Gs (0.21 × 0.21). The deficit occurs because most

⑤

Lander et al., *Nature*, **409**: 860-921, 2001

確率の話8

④CGの観測値（約1%）は、その期待値（4.20%）より非常に低い！⑤原著論文中の、⑥の項目中の赤下線部分を実際に確認したことに相当します。ゲノム中でCとGの割合である⑦0.21を掛け合わせた結果から期待される、⑧約4%という値が…

	AA	AT	TA	TT	AC	AG	CA	CT	GA	GT	TC	T				
1	9.5%	7.4%	6.3%	9.6%	5.0%	7.1%	7.3%	7.1%	6.0%	5.0%	6.0%	7				
2	10.0%	7.9%	6.7%	10.0%	5.0%	7.0%	7.2%	7.0%	5.9%	5.0%	5.9%	7				
3	10.1%	8.0%	6.9%	10.2%	5.0%	6.9%	7.2%	6.9%	5.9%	5.0%	5.9%	7.2%	4.9%	0.8%	4.0%	4.9%
4	10.6%	8.5%	7.3%	10.6%	5.0%	6.7%	7.1%	6.7%	5.9%	5.0%	5.8%	7.1%	4.5%	0.8%	3.8%	4.5%
5	10.2%	8.1%	6.9%	10.3%	5.0%	6.9%	7.2%	6.9%	5.9%	5.1%	5.9%	7.2%	4.8%	0.9%	4.0%	4.8%
														0.9%	4.0%	4.9%
														1.0%	4.2%	5.1%
														0.9%	4.1%	5.0%
														1.0%	4.3%	5.3%
														1.0%	4.4%	5.3%
														1.0%	4.3%	5.4%
														1.0%	4.2%	5.2%
														0.9%	3.8%	4.6%
														1.0%	4.2%	5.2%
														1.1%	4.5%	5.5%
														1.4%	5.0%	6.1%
17	8.5%	6.4%	5.2%	8.6%	5.1%	7.3%	7.4%	7.4%	6.2%	5.0%	6.1%	7.5%	6.3%	1.5%	5.1%	6.4%
18	10.1%	7.9%	6.7%	10.3%	5.1%	7.0%	7.2%	6.9%	6.1%	5.1%	5.9%	7.3%	4.7%	0.9%	4.0%	4.9%
19	7.7%	5.7%	4.5%	7.9%	5.1%	7.5%	7.5%	7.4%	6.2%	5.2%	6.2%	7.6%	7.0%	1.9%	5.6%	7.1%
20	8.8%	6.8%	5.5%	9.1%	5.0%	7.3%	7.5%	7.3%	6.2%	5.1%	6.1%	7.6%	5.8%	1.2%	4.8%	6.0%
21	9.8%	7.7%	6.4%	9.9%	5.1%	6.9%	7.3%	6.9%	6.0%	5.1%	6.0%	7.3%	5.1%	1.2%	4.3%	5.1%

CpG islands

A related topic is the distribution of so-called CpG islands across the genome. The dinucleotide CpG is notable because it is greatly under-represented in human DNA, occurring at only about one-fifth of the roughly 4% frequency that would be expected by simply multiplying the typical fraction of Cs and Gs (0.21 × 0.21). The deficit occurs because most

⑧

⑦

確率の話9

④CGの観測値(約1%)は、その期待値(4.20%)より非常に低い！⑤原著論文中の、⑥の項目中の赤下線部分を実際に確認したことに相当します。ゲノム中でCとGの割合である⑦0.21を掛け合わせた結果から期待される、⑧約4%という値が、⑨0.205を掛け合わせた結果から期待される、⑩4.2%という数値に対応します。確率を考える場合はGC含量によって期待値が異なる点に注意すべし、ということ。

ヒトゲノムのGC含量は約41%なので…

(1) 1連続塩基の出現確率

A or Tの場合: 0.295

C or Gの場合: 0.205 ⑨

(2) 2連続塩基の出現確率

AとTのみから構成される場合(AA, AT, TA, TT): 0.295×0.295 = 8.70%

CとGのみから構成される場合(CC, CG, GC, GG): 0.205×0.205 = 4.20% ⑩

それ以外(AC, AG, CA, CT, GA, GT, TC, TG): 0.205×0.295 = 6.05%

Contents

作図(例題10)1

①例題10。②でも書かれていますが、さきほどの例題7と基本的に同じで、4^2 = 16種類の2連続塩基の出現確率の箱ひげ図(box plot)を追加で出力しています。

(Rで)塩基配列解析

保護されていない通信 | iu.a.u-tokyo.ac.jp/~kadota/r_seq.html#intro_general_kmer_2_bios...

ゲスト

①

10. ヒトゲノム配列パッケージ(BSgenome.Hsapiens.NCBI.GRCh38)の場合：

②

7.と基本的に同じですが、box plotのPNGファイルも出力しています。

```
out_f1 <- "hoge10.txt"                  #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge10.png"                  #出力ファイル名を指定してout_f2に格納
param_bsgenome <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定(BSgenome系のゲノムパッケージ
param_fig <- c(700, 400)                #ファイル出力時の横幅と縦幅を指定(単位はピクセル)

#必要なパッケージをロード
library(Biostrings)                     #パッケージの読み込み
library(param_bsgenome, character.only=T)#指定したパッケージの読み込み

#前処理(指定したパッケージ中のオブジェクト名をgenomeに統一)
tmp <- ls(paste("package", param_bsgenome, sep=":"))#指定したパッケージで利用可能なオブジェクト名を
genome <- eval(parse(text=tmp))         #文字列tmpをRオブジェクトとしてgenomeに格納(パッケージ中にオ
fasta <- getSeq(genome)                 #ゲノム塩基配列情報を抽出した結果をfastaに格納
names(fasta) <- seqnames(genome)        #description情報を追加している
fasta                                   #確認してるだけです

#本番
out <- dinucleotideFrequency(fasta, as.prob=T)#連続塩基の出現確率情報をoutに格納
```

トップページへ

作図(例題10)2

①例題10。②でも書かれていますが、さきほどの例題7と基本的に同じで、4^2 = 16種類の2連続塩基の出現確率の箱ひげ図(box plot)を追加で出力しています。③box plotはhoge10.pngに保存され、④その大きさは700×400と指定しています。

(Rで)塩基配列解析

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

10. ヒトゲノム配列パッケージ(BSgenome.Hsapiens.NCBI.GRCh38)の場合：

7.と基本的に同じですが、box plotのPNGファイルも出力しています。

```
out_f1 <- "hoge10.txt"                  #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge10.png"                  #出力ファイル名を指定してout_f2に格納
param_bsgenome <- "BSgen[illegible]Hsapiens.NCBI.GRCh38"#パッケージ名を指定(BSgenome系のゲノムパッケージ
param_fig <- c(700, 400)                #ファイル出力時の横幅と縦幅を指定(単位はピクセル)

#必要なパッケージをロード
library(Biostrings)                     #パッケージの読み込み
library(param_bsgenome, character.only=T)#指定したパッケージの読み込み

#前処理(指定したパッケージ中のオブジェクト名をgenomeに統一)
tmp <- ls(paste("package", param_bsgenome, sep=":"))#指定したパッケージで利用可能なオブジェクト名を
genome <- eval(parse(text=tmp))         #文字列tmpをRオブジェクトとしてgenomeに格納(パッケージ中にオ
fasta <- getSeq(genome)                 #ゲノム塩基配列情報を抽出した結果をfastaに格納
names(fasta) <- seqnames(genome)        #description情報を追加している
fasta                                   #確認してるだけです

#本番
out <- dinucleotideFrequency(fasta, as.prob=T)#連続塩基の出現確率情報をoutに格納
```

トップページへ

作図(例題10)3

①例題10。②でも書かれていますが、さきほどの例題7と基本的に同じで、4^2 = 16種類の2連続塩基の出現確率の箱ひげ図(box plot)を追加で出力しています。③box plotはhoge10.pngに保存され、④その大きさは700×400と指定しています。
⑤コード下部を表示。

(Rで)塩基配列解析

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

```
fasta <- getSeq(genome)                    #ゲノム塩基配列情報を抽出し
names(fasta) <- seqnames(genome)           #description情報を追加し
fasta                                      #確認してるだけです

#本番
out <- dinucleotideFrequency(fasta, as.prob=T)#連続塩基の出現確率情報をoutに格納

#ファイルに保存(テキストファイル)
tmp <- cbind(names(fasta), out)            #保存したい情報をtmpに格納
write.table(tmp, out_f1, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定したファイル名

#ファイルに保存(pngファイル)
png(out_f2, pointsize=13, width=param_fig[1], height=param_fig[2])#出力ファイルの各種パラメータを
boxplot(out, ylab="Probability")           #描画
grid(col="gray", lty="dotted")             #指定したパラメータでグリッドを表示
dev.off()                                  #おまじない
```

⑤

11. ヒトゲノム配列パッケージ(BSgenome.Hsapiens.NCBI.GRCh38)の場合：

トップページへ

10.と基本的に同じですが、連続塩基の種類ごとの期待値とボックスプロット(box plot)上での色情報を含むファイル(human_2mer.txt)を入力として利用し、色情報のみを取り出して利用しています。

作図(例題10)4

①の赤枠内が、染色体ごとの2連続塩基の出現確率情報を、②out_f1で指定したファイル名(hoge10.txt)で出力している部分。

(Rで)塩基配列解析

保護されていない通信 | iu.a.u-tokyo.ac.jp/~kadota/r_seq.html#intro_general_kmer_2_bios...

ゲスト

```
fasta <- getSeq(genome)                 #ゲノム塩基配列情報を抽出した結果をfastaに格納
names(fasta) <- seqnames(genome)        #description情報を追加している
fasta                                   #確認してるだけです

#本番
out <- dinucleotideFrequency(fasta, as.prob=T)#連続塩基の出現確率情報をoutに格納

#ファイルに保存(テキストファイル)
tmp <- cbind(names(fasta), out)         #保存したい情報をtmpに格納
write.table(tmp, out_f1, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定したファイル名

#ファイルに保存(pngファイル)
png(out_f2, pointsize=13, width=param_fig[1], height=param_fig[2])#出力ファイルの各種パラメータを
boxplot(out, ylab="Probability")        #描画
grid(col="gray", lty="dotted")          #指定したパラメータでグリッドを表示
dev.off()                               #おまじない
```

①

②

11. ヒトゲノム配列パッケージ(BSgenome.Hsapiens.NCBI.GRCh38)の場合：

トップページへ

10.と基本的に同じですが、連続塩基の種類ごとの期待値とボックスプロット(box plot)上での色情報を含むファイル(human_2mer.txt)を入力として利用し、色情報のみを取り出して利用しています

作図(例題10)5

①の赤枠内が、染色体ごとの2連続塩基の出現確率情報を、②out_f1で指定したファイル名(hoge10.txt)で出力している部分。③の赤枠内が、箱ひげ図を、④out_f2で指定したファイル名(hoge10.png)で出力している部分。

(Rで)塩基配列解析

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

```
fasta <- getSeq(genome)                    #ゲノム塩基配列情報を抽出した結果をfastaに格納
names(fasta) <- seqnames(genome)           #description情報を追加している
fasta                                      #確認してるだけです

#本番
out <- dinucleotideFrequency(fasta, as.prob=T)#連続塩基の出現確率情報をoutに格納

#ファイルに保存(テキストファイル)
tmp <- cbind(names(fasta), out)            #保存したい情報をtmpに格納
write.table(tmp, out_f1, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定したファイル名

#ファイルに保存(pngファイル)
png(out_f2, pointsize=13, width=param_fig[1], height=param_fig[2])#出力ファイルの各種パラメータを
boxplot(out, ylab="Probability")           #描画
grid(col="gray", lty="dotted")             #指定したパラメータでグリッドを表示
dev.off()                                  #おまじない
```

③ ④

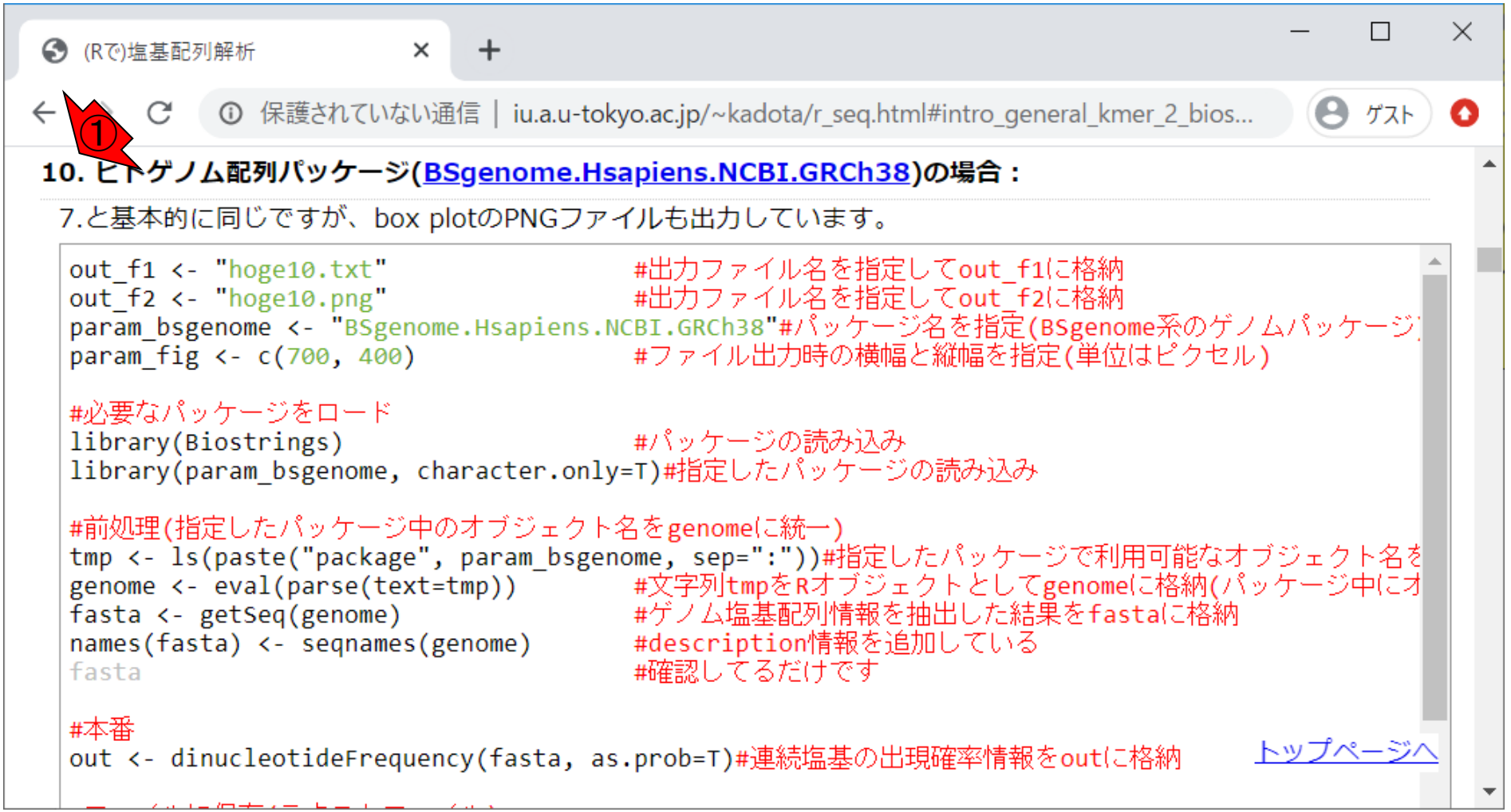
11. ヒトゲノム配列パッケージ(BSgenome.Hsapiens.NCBI.GRCh38)の場合：

トップページへ

10.と基本的に同じですが、連続塩基の種類ごとの期待値とボックスプロット(box plot)上での色情報を含むファイル(human_2mer.txt)を入力として利用し、色情報のみを取り出して利用しています

作図(例題10)6

①例題10のコード全体をコピペ実行。例題7と同じく、数分かかります。



10. ヒトゲノム配列パッケージ(BSgenome.Hsapiens.NCBI.GRCh38)の場合：

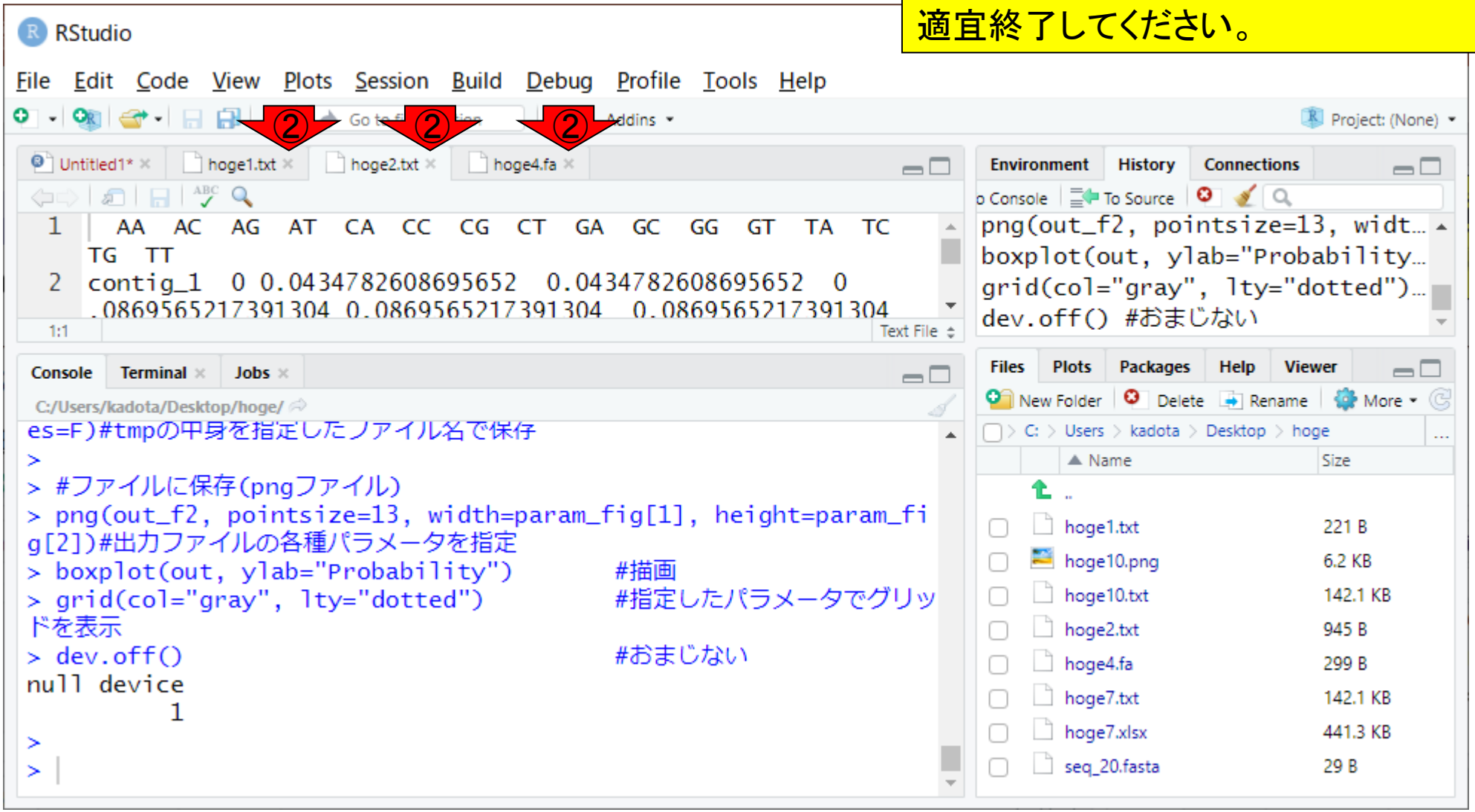
7.と基本的に同じですが、box plotのPNGファイルも出力しています。

```
out_f1 <- "hoge10.txt"                  #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge10.png"                  #出力ファイル名を指定してout_f2に格納
param_bsgenome <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定(BSgenome系のゲノムパッケージ
param_fig <- c(700, 400)                #ファイル出力時の横幅と縦幅を指定(単位はピクセル)

#必要なパッケージをロード
library(Biostrings)                     #パッケージの読み込み
library(param_bsgenome, character.only=T)#指定したパッケージの読み込み

#前処理(指定したパッケージ中のオブジェクト名をgenomeに統一)
tmp <- ls(paste("package", param_bsgenome, sep=":"))#指定したパッケージで利用可能なオブジェクト名を
genome <- eval(parse(text=tmp))         #文字列tmpをRオブジェクトとしてgenomeに格納(パッケージ中にオ
fasta <- getSeq(genome)                 #ゲノム塩基配列情報を抽出した結果をfastaに格納
names(fasta) <- seqnames(genome)        #description情報を追加している
fasta                                   #確認してるだけです

#本番
out <- dinucleotideFrequency(fasta, as.prob=T)#連続塩基の出現確率情報をoutに格納
```

トップページへ

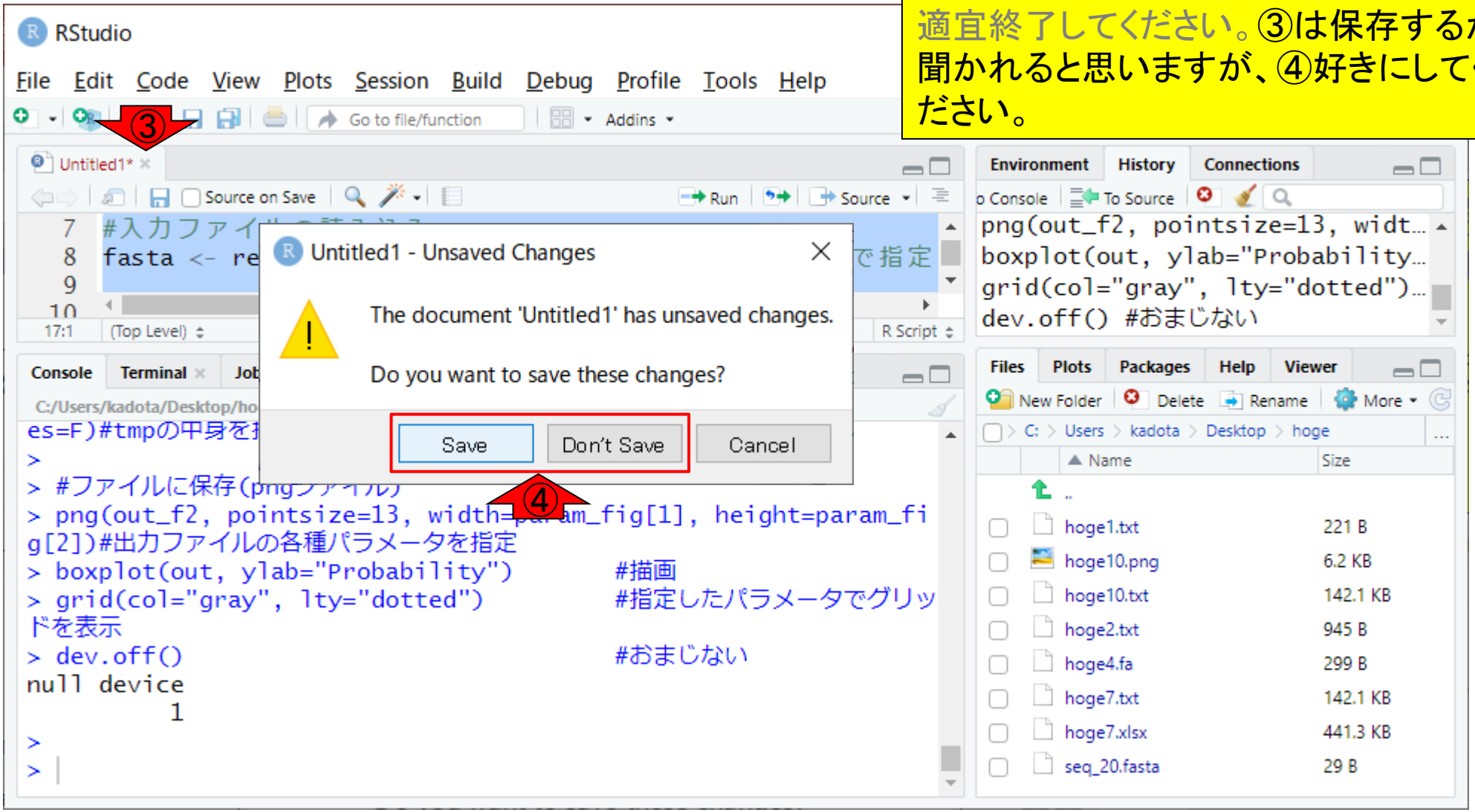
作図(例題10)7

①例題10のコード全体をコピペ実行。例題7と同じく、数分かかります。実行完了後。②エディタ画面上にあるこれらは、適宜終了してください。



作図(例題10)8

①例題10のコード全体をコピペ実行。例題7と同じく、数分かかります。実行完了後。②エディタ画面上にあるこれらは、適宜終了してください。③は保存するか聞かれると思いますが、④好きにしてください。



作図（例題10）9

出力ファイルをパワポに貼り付けると、確かに②横幅700 pixels、③縦幅400 pixelsっぽくなっていることがわかる。

(Rで)塩基配列解析

保護されていない通信 | iu.a.u-tokyo.ac.jp/~kadota/r_seq.html#intro_general_kmer_2_bios...

ゲスト

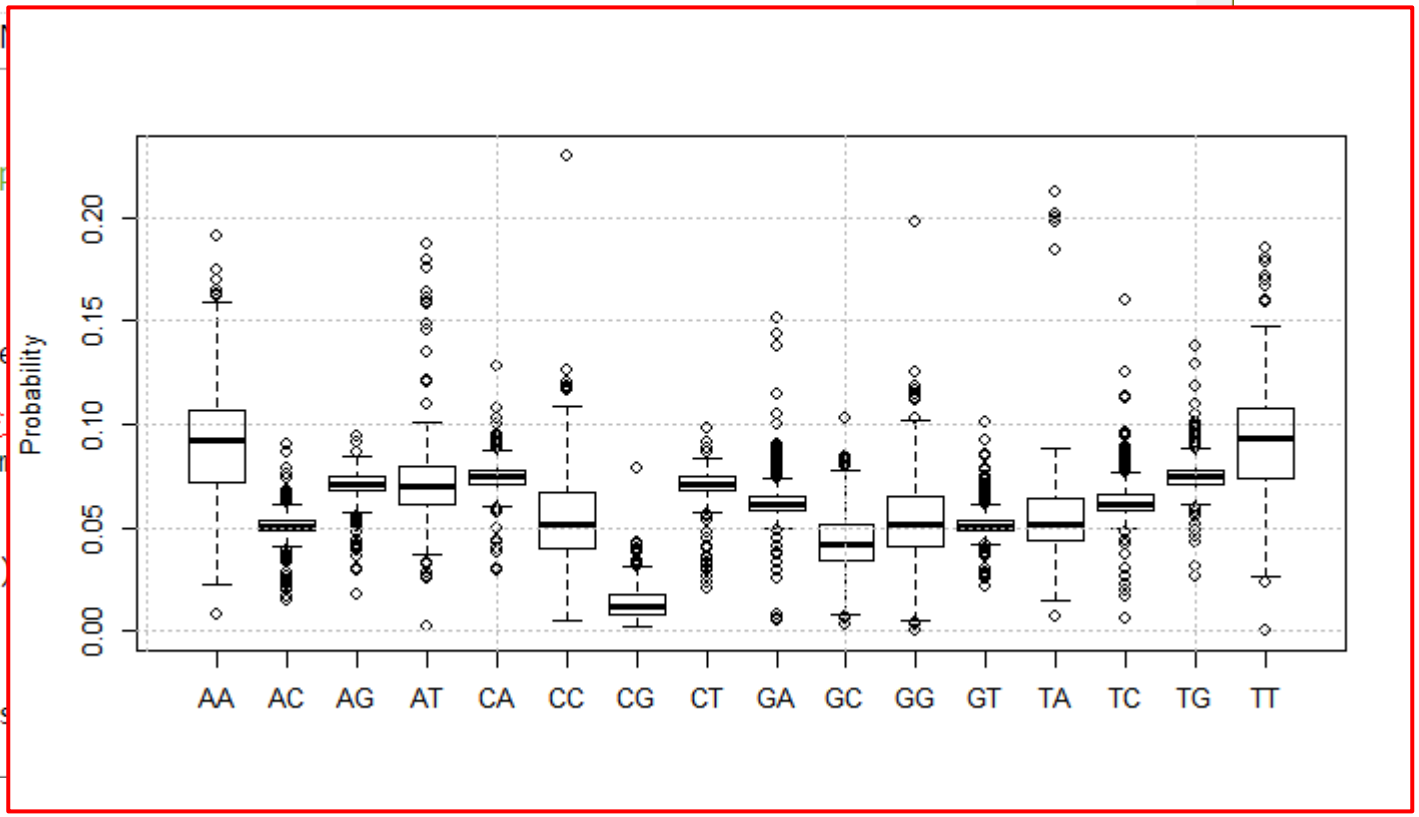
10. ヒトゲノム配列パッケージ(BSgenome.Hsapiens.NCBI.GRCh38)の場合：

7.と基本的に同じですが、box plotのP

```
out_f1 <- "hoge10.txt"
out_f2 <- "hoge10.png"
param_bsgenome <- "BSgen me.Hsap
param_fig <- c(700, 400)

#必要なパッケー
library(Biostrings)
library(param_bsgenome, characte

#前処理(指定したパッケージ中のオブ
tmp <- ls(paste("package", param
genome <- eval(parse(text=tmp))
fasta <- getSeq(genome)
names(fasta) <- seqnames(genome)
fasta

#本番
out <- dinucleotideFrequency(fas
```

①

②③



作図(例題10)10

出力ファイルをパワポに貼り付けると、確かに②横幅700 pixels、③縦幅400 pixelsっぽくなっていることがわかる。④AとTのみから構成される場合の観測値は、各塩基の出現確率(0.295)を掛けた期待値(8.7%)に比較的近い位置にプロットされていることがわかる。

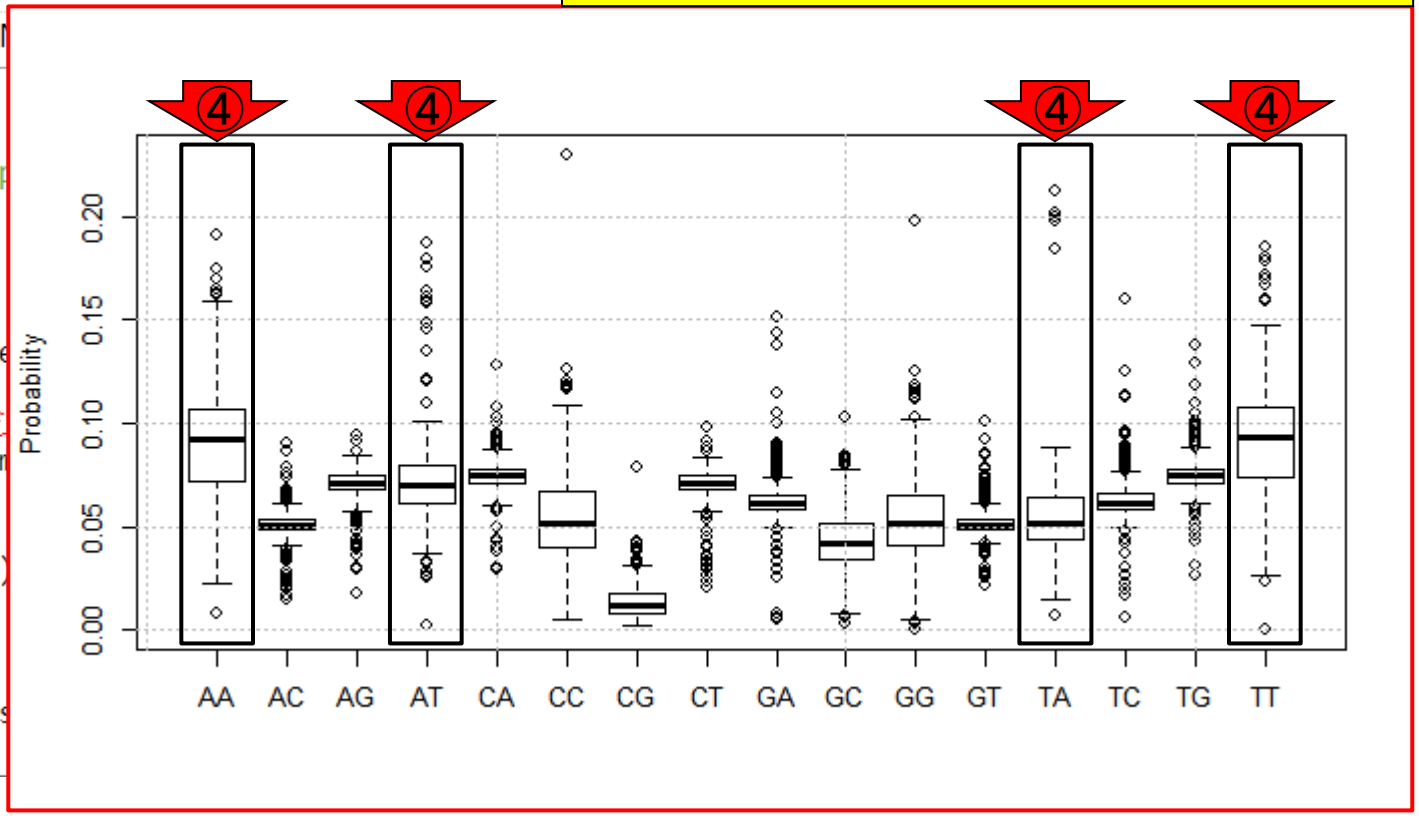
(Rで)塩基配列解析

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

10. ヒトゲノム配列パッケージ(BSgenome.Hsapiens.NCBI.GRCh38)の場

7.と基本的に同じですが、box plotのPl

```
out_f1 <- "hoge10.txt"
out_f2 <- "hoge10.png"
param_bsgenome <- "BSgenome.Hsap
param_fig <- c(700, 400)

#必要なパッケージをロード
library(Biostrings)
library(param_bsgenome, characte

#前処理(指定したパッケージ中のオブジ
tmp <- ls(paste("package", param
genome <- eval(parse(text=tmp))
fasta <- getSeq(genome)
names(fasta) <- seqnames(genome)
fasta

#本番
out <- dinucleotideFrequency(fas
```



作図(例題10)11

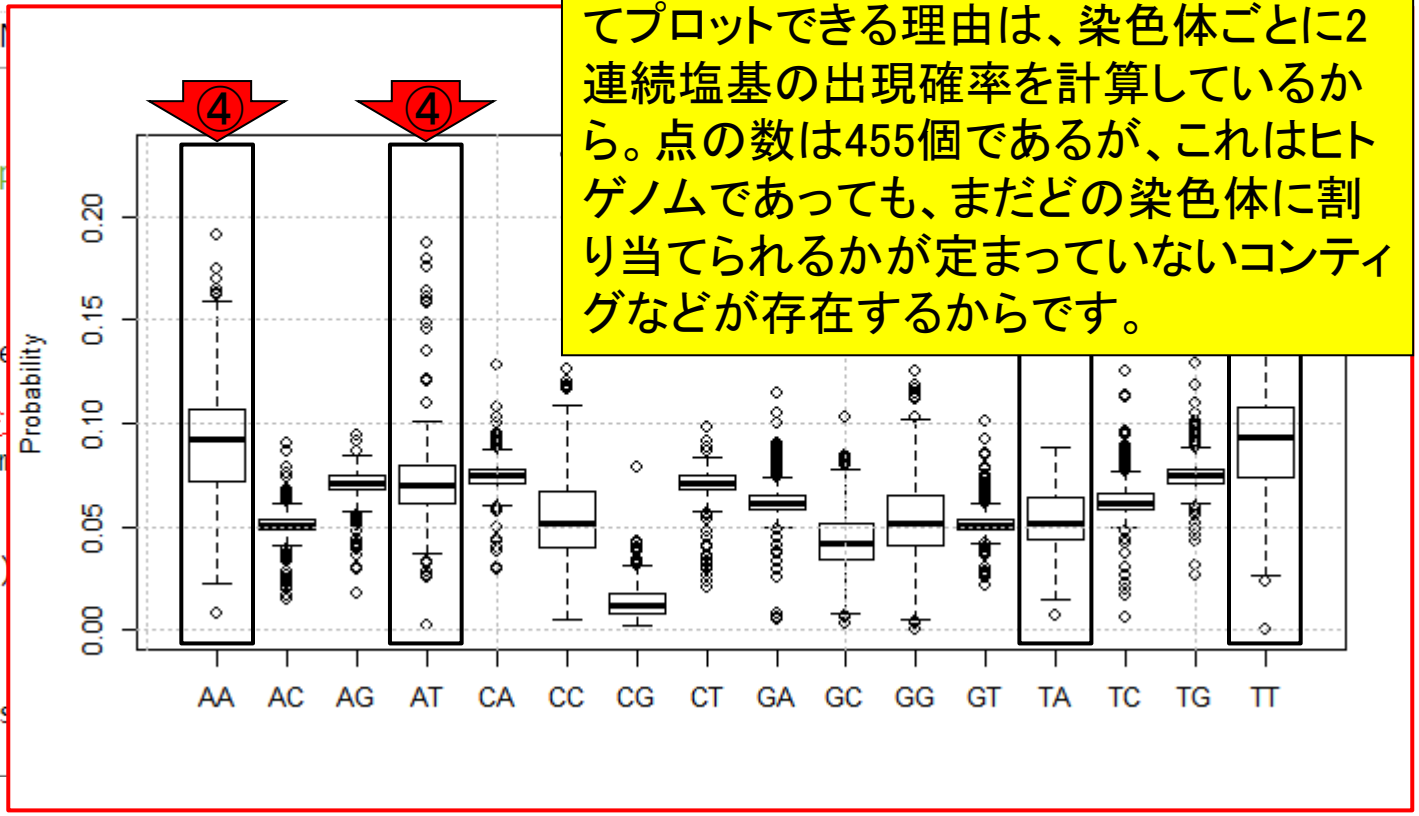
(Rで)塩基配列解析

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

10. ヒトゲノム配列パッケージ(BSgenome.Hsapiens.NCBI.GRCh38)の場

7.と基本的に同じですが、box plotのPl

```
out_f1 <- "hoge10.txt"
out_f2 <- "hoge10.png"
param_bsgenome <- "BSgenome.Hsap
param_fig <- c(700, 400)

#必要なパッケージをロード
library(Biostrings)
library(param_bsgenome, characte

#前処理(指定したパッケージ中のオブジ
tmp <- ls(paste("package", param
genome <- eval(parse(text=tmp))
fasta <- getSeq(genome)
names(fasta) <- seqnames(genome)
fasta

#本番
out <- dinucleotideFrequency(fas
```



出力ファイルをパワポに貼り付けると、確かに②横幅700 pixels、③縦幅400 pixelsっぽくなっていることがわかる。④AとTのみから構成される場合の観測値は、各塩基の出現確率(0.295)を掛けた期待値(8.7%)に比較的近い位置にプロットされていることがわかる。尚、箱ひげ図としてプロットできる理由は、染色体ごとに2連続塩基の出現確率を計算しているから。点の数は455個であるが、これはヒトゲノムであっても、まだどの染色体に割り当てられるかが定まっていないコンティグなどが存在するからです。

Contents

- Introduction、出現頻度解析（k=2）、出現頻度解析（k=1）
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図（例題10）、作図（例題11）、作図（例題12）
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ（例題32）、被覆率（coverage）、基本的な考え方(例題7)
  - 例題8（k=2)、例題9（k=3）、1,000塩基の仮想ゲノム（サンプルデータの例題33）
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

参考

①例題11。これは2連続塩基ごとに任意の色で表示させる際に利用するコード。

作図(例題11)1

(Rで)塩基配列解析

保護されていない通信 | iu.a.u-tokyo.ac.jp/~kadota/r_seq.html#intro_general_kmer_2_bios...

ゲスト

①

11. ヒトゲノム配列パッケージ(BSgenome.Hsapiens.NCBI.GRCh38)の場合：

10.と基本的に同じですが、連続塩基の種類ごとの期待値とボックスプロット(box plot)上での色情報を含むファイル (human_2mer.txt)を入力として利用し、色情報のみを取り出して利用しています。

```
in_f <- "human_2mer.txt"                    #入力ファイル名を指定してin_fに格納
out_f1 <- "hoge11.txt"                      #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge11.png"                      #出力ファイル名を指定してout_f2に格納
param_bsgenome <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定(BSgenome系のゲノムパッケージ
param_fig <- c(700, 400)                    #ファイル出力時の横幅と縦幅を指定(単位はピクセル)

#必要なパッケージをロード
library(Biostrings)                         #パッケージの読み込み
library(param_bsgenome, character.only=T)#指定したパッケージの読み込み

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

#前処理(指定したパッケージ中のオブジェクト名をgenomeに統一)
tmp <- ls(paste("package", param_bsgenome, sep=":"))#指定したパッケージで利用可能なオブジェクト名を
genome <- eval(parse(text=tmp))             #文字列tmpをRオブジェクトとしてgenomeに格納(パッケージ中にオ
fasta <- getSeq(genome)                     #ゲノム塩基配列情報を抽出した結果をfastaに格納
names(fasta) <- seqnames(genome)            #description情報を追加している
```

トップページへ

作図(例題11)2

①例題11。これは2連続塩基ごとに任意の色で表示させる際に利用するコード。
例題10との違いは、②を入力ファイルとして与えている点。赤枠が中身。③color列で任意の2連続塩基の色を与えている。

(Rで)塩基配列解析

保護されていない通信 | iu.a.u-tokyo.ac.jp/~kadota/r_seq.html#intro_general_kmer_2_bios...

ゲスト

①

11. ヒトゲノム配列パッケージ(BSgenome.Hsapiens.NCBI.GRCh38)の場合：

10.と基本的に同じですが、連続塩基の種類ごとの期待値とボックスプロット(box plot)上での色情
ル (human_2mer.txt)を入力として利用し、色情報のみを取り出して利用しています。

②

```
in_f <- "human_2mer.txt"                    #入力ファイル名を指定してin_fに格納
out_f1 <- "hoge11.txt"                      #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge11.png"                      #出力ファイル名を指定してout_f2に格納
param_bsgenome <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定(BSgenome系の
param_fig <- c(700, 400)                    #ファイル出力時の横幅と縦幅を指定(単位はピク

#必要なパッケージをロード
library(Biostrings)                         #パッケージの読み込み
library(param_bsgenome, character.only=T)#指定したパッケージの読み込み

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

#前処理(指定したパッケージ中のオブジェクト名をgenomeに統一)
tmp <- ls(paste("package", param_bsgenome, sep=":"))#指定したパッケージで利用可能な
genome <- eval(parse(text=tmp))             #文字列tmpをRオブジェクトとしてgenomeに格納(
fasta <- getSeq(genome)                     #ゲノム塩基配列情報を抽出した結果をfastaに格
names(fasta) <- seqnames(genome)            #description情報を追加している
```

③

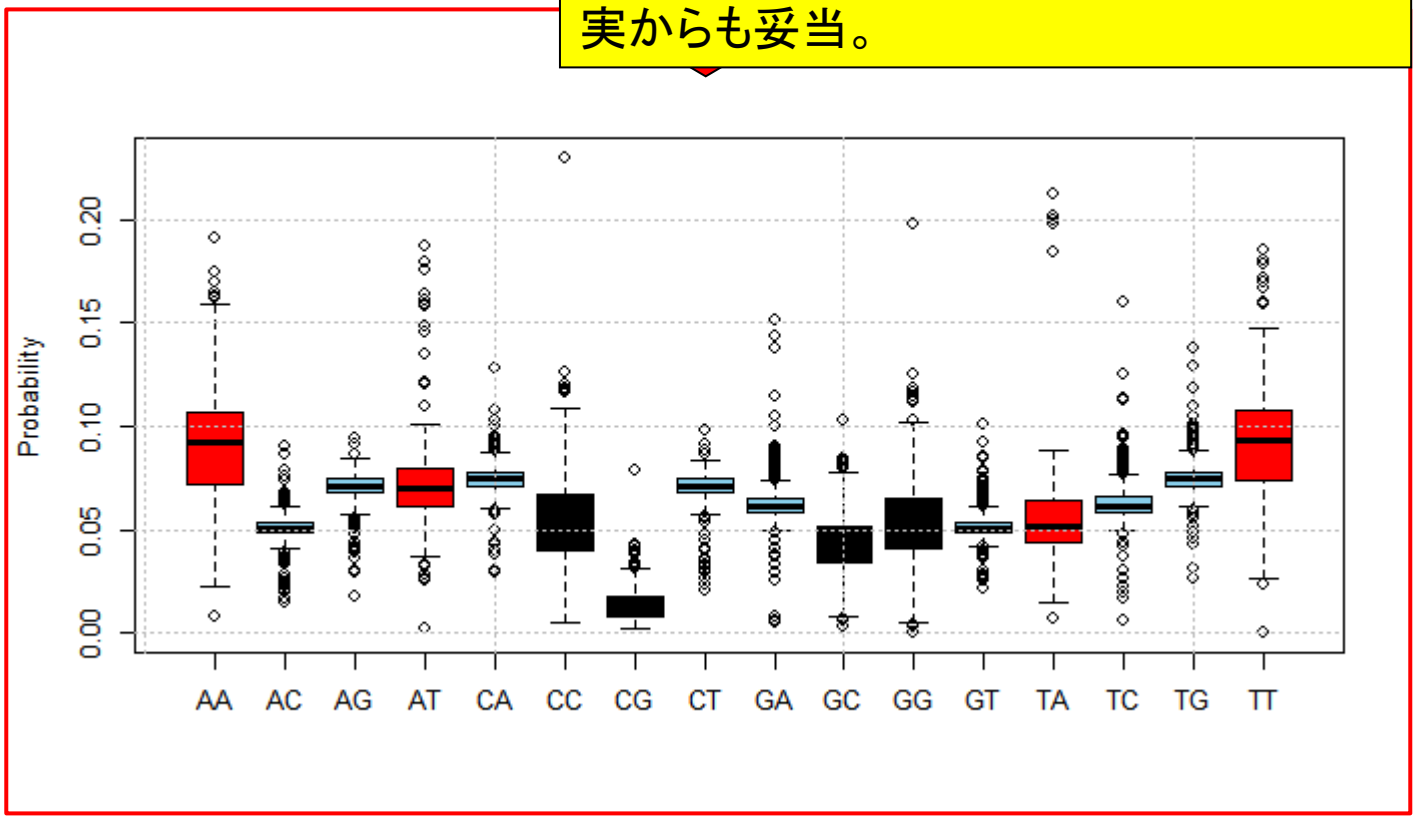
type	expected	color
AA	0.087025	red
AC	0.060475	skyblue
AG	0.060475	skyblue
AT	0.087025	red
CA	0.060475	skyblue
CC	0.042025	black
CG	0.042025	black
CT	0.060475	skyblue
GA	0.060475	skyblue
GC	0.042025	black
GG	0.042025	black
GT	0.060475	skyblue
TA	0.087025	red
TC	0.060475	skyblue
TG	0.060475	skyblue
TT	0.087025	red

作図(例題11)3

②

type	expected	color
AA	0.087025	red
AC	0.060475	skyblue
AG	0.060475	skyblue
AT	0.087025	red
CA	0.060475	skyblue
CC	0.042025	black
CG	0.042025	black
CT	0.060475	skyblue
GA	0.060475	skyblue
GC	0.042025	black
GG	0.042025	black
GT	0.060475	skyblue
TA	0.087025	red
TC	0.060475	skyblue
TG	0.060475	skyblue
TT	0.087025	red

①例題11の出力ファイル(hoge11.png)。確かに②で指定した通りの色になっていることがわかる。全体を眺めると、A or Tで構成されるもの(赤色)が全体的に高い値となり、C or Gで構成されるもの(黒色)が全体的に低い値となっていることがわかる。これはGC含量が41%という事実からも妥当。

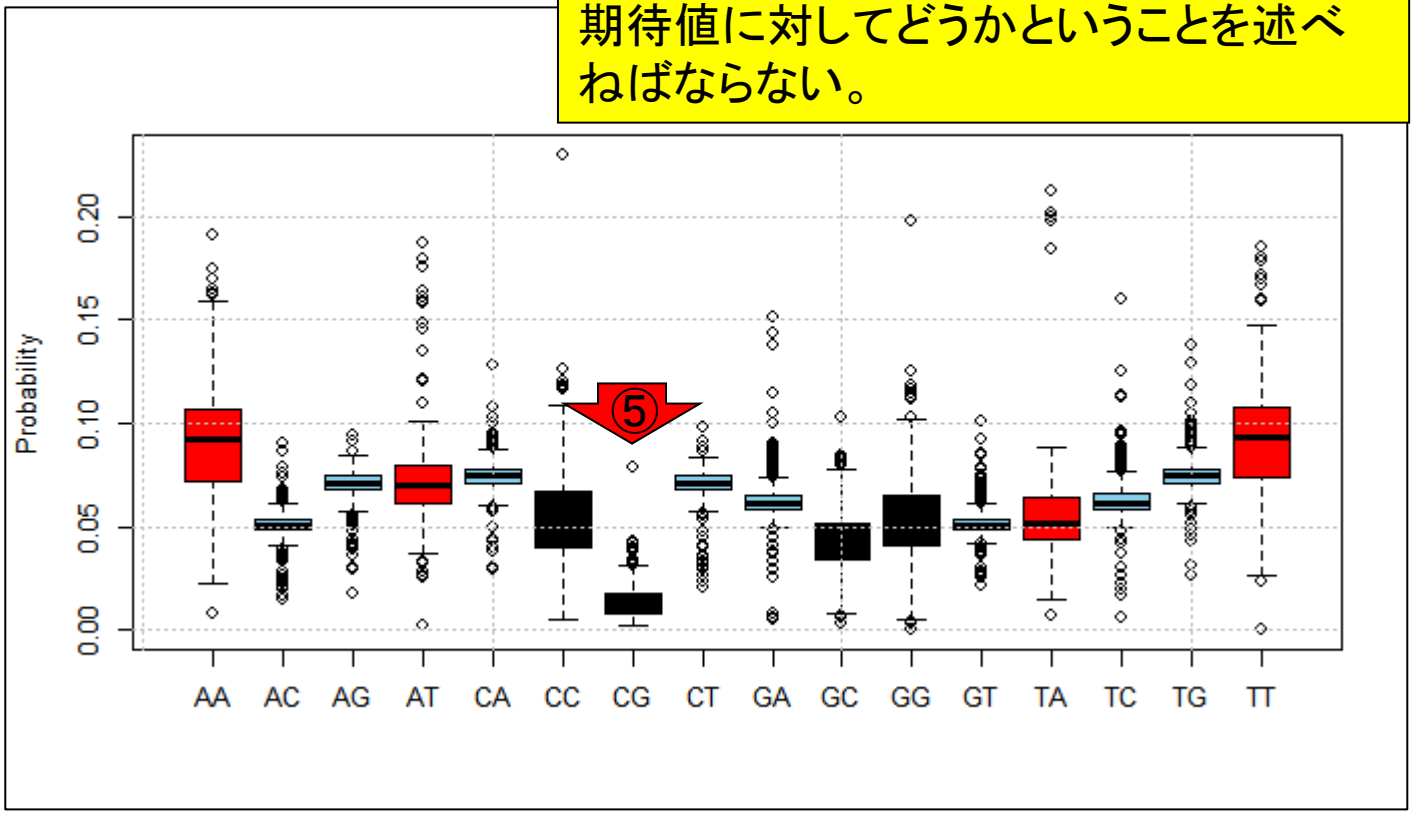


作図(例題11)4

それゆえ、③原著論文でも書かれていたように、ゲノム中の各塩基の出現確率を算出しておき、④「連続塩基の種類ごとに各塩基の出現確率を掛け合わせた期待値(expected)」を考慮に入れねばならない。⑤CGという連続塩基の出現確率がただ低いと主張するのではなく、期待値に対してどうかということを述べねばならない。

④

type	expected	color
AA	0.087025	red
AC	0.060475	skyblue
AG	0.060475	skyblue
AT	0.087025	red
CA	0.060475	skyblue
CC	0.042025	black
CG	0.042025	black
CT	0.060475	skyblue
GA	0.060475	skyblue
GC	0.042025	black
GG	0.042025	black
GT	0.060475	skyblue
TA	0.087025	red
TC	0.060475	skyblue
TG	0.060475	skyblue
TT	0.087025	red



③

Lander et al., *Nature*, **409**: 860-921, 2001

Contents

- Introduction、出現頻度解析（k=2）、出現頻度解析（k=1）
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図（例題10）、作図（例題11）、作図（例題12）
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ（例題32）、被覆率（coverage）、基本的な考え方(例題7)
  - 例題8（k=2)、例題9（k=3）、1,000塩基の仮想ゲノム（サンプルデータの例題33）
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

 参考

①例題12は、期待値を考慮に入れてプロットするやり方。

作図(例題12)1

(Rで)塩基配列解析

保護されていない通信 | iu.a.u-tokyo.ac.jp/~kadota/r_seq.html#intro_general_kmer_2_bios...

ゲスト

①

12. ヒトゲノム配列パッケージ(BSgenome.Hsapiens.NCBI.GRCh38)の場合：

11.と基本的に同じですが、human_2mer.txtというファイルを入力として与えて、 連続塩基の種類ごとの期待値とボックスプロット(box plot)上での色情報を利用しています。 また、重要なのは期待値からの差分であり、「プロットも期待値(expected)と同程度の観測値(observed)であればゼロ、 観測値のほうが大きければプラス、観測値のほうが小さければマイナス」といった具合で表現したほうがスマートです。 それゆえ、box plotの縦軸をlog(observed/expected)として表現しています。 CG以外の連続塩基は縦軸上でが0近辺に位置していることが分かります。

```
in_f <- "human_2mer.txt"                    #入力ファイル名を指定してin_fに格納
out_f1 <- "hoge12.txt"                      #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge12.png"                      #出力ファイル名を指定してout_f2に格納
param_bsgenome <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定(BSgenome系のゲノムパッケージ
param_fig <- c(700, 400)                    #ファイル出力時の横幅と縦幅を指定(単位はピクセル)

#必要なパッケージをロード
library(Biostrings)                         #パッケージの読み込み
library(param_bsgenome, character.only=T)#指定したパッケージの読み込み

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

#前処理(指定したパッケージ中のオブジェクト名をgenomeに統一)
```

トップページへ

参考

作図(例題12)2

①例題12は、期待値を考慮に入れてプロットするやり方。縦軸を「観測値」ではなく「log(観測値/期待値)」とすることで、②のような解釈ができるようにしています。このような期待値で割ってlogをとるのはよくやられる戦略なので感覚をつかんでおくとよいでしょう。

(Rで)塩基配列解析

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

①

12. ヒトゲノム配列パッケージ(BSgenome.Hsapiens.NCBI.GRCh38)の場

11.と基本的に同じですが、human_2mer.txtというファイルを入力として与えて、 連続塩基の種類ごとの期待値とボックスプロット(box plot)上での色情報を利用しています。 また、重要なのは期待値からの差分であり、「プロットも期待値(expected)と同程度の観測値(observed)であればゼロ、 観測値のほうが大きければプラス、観測値のほうが小さければマイナス」といった具合で表現したほうがスマートです。 それゆえ、box plotの縦軸をlog(observed/expected)として表現しています。 CG以外の連続塩基は縦軸上でが0近辺に位置していることが分かります。

②

```
in_f <- "human_2mer.txt"                    #入力ファイル名を指定してin_fに格納
out_f1 <- "hoge12.txt"                      #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge12.png"                      #出力ファイル名を指定してout_f2に格納
param_bsgenome <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定(BSgenome系のゲノムパッケージ
param_fig <- c(700, 400)                    #ファイル出力時の横幅と縦幅を指定(単位はピクセル)

#必要なパッケージをロード
library(Biostrings)                         #パッケージの読み込み
library(param_bsgenome, character.only=T)#指定したパッケージの読み込み

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

#前処理(指定したパッケージ中のオブジェクト名をgenomeに統一)
```

トップページへ

作図(例題12)3

例題12のコピペ実行結果の、①Console画面。②警告メッセージがでており、③その理由として外れ値(-Inf)が存在するようだ。-Infというのは「マイナス無限大(infinity)」のこと。log(観測値/期待値)で計算しているので、logの特性をある程度理解していれば、「観測値が0のものがあるのだろう」と簡単に原因の解読ができる。実際に出力ファイル(hoge12.txt)を眺めると、「TT」の列において、「HSCHRUN_RANDOM_146」というコンティグが-Infになっていることが確認できる。ヒトゲノムの場合は、21番染色体(xとyを含めてもよい)までで同様の計算を行えば、このような警告は出ない筈。

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

① Console | Terminal | Jobs

C:/Users/kadota/Desktop/hoge/

```
> tmp <- cbind(names(fasta), logratio)    #保存したい情報をtmpに格
> write.table(tmp, out_f1, sep="\t", append=F, quote=F, row.n
es=F)#tmpの中身を指定したファイル名で保存
>
> #ファイルに保存(pngファイル)
> png(out_f2, pointsize=13, width=param_fig[1], height=param_
g[2])#出力ファイルの各種パラメータを指定
> boxplot(lo    atio, ylab="log2(observed/expected)",#描画
+         co    .character(data$color))   #描画
 警告メッセージ:
 bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$ou
[z$group ==   で:
   外れ値 (-Inf) は描かれません 箱型図 16 中では
> grid(col="gr    , lty="dotted")          #指定したパラメータでグリッ
ドを表示
> dev.off()                               #おまじない
null device
          1
>
>
```

② ③

hoge10.png	6.2 KB
hoge10.txt	142.1 KB
hoge2.txt	945 B
hoge4.fa	299 B
hoge7.txt	142.1 KB
hoge7.xlsx	441.3 KB
seq_20.fasta	29 B

参考 ①例題12の、②出力結果(hoge12.png)。

作図(例題12)4

(Rで)塩基配列解析

保護されていない通信 | iu.a.u-tokyo.ac.jp/~kadota/r_seq.html#intro_general_kmer_2_bios...

ゲスト

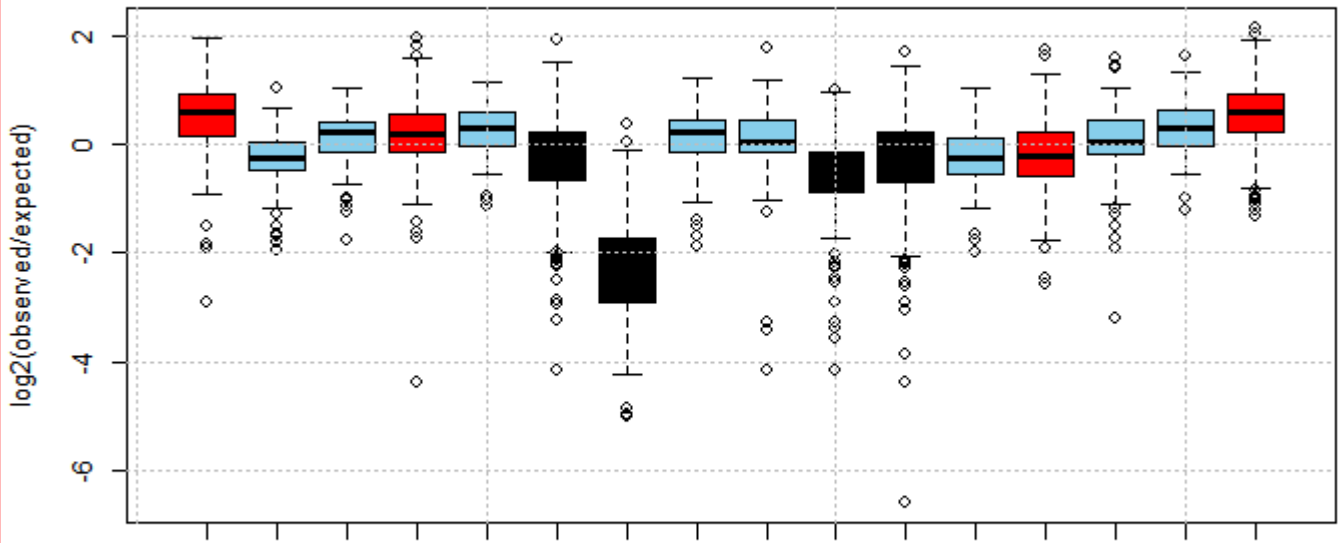
①

12. ヒトゲノム配列パッケージ(BSgenome.Hsapiens.NCBI.GRCh38)の場合：

11.と基本的に同じですが、human_2n
ボックスプロット(box plot)上での色情
ットも期待値(expected)と同程度の観測
ほうが小さければマイナス」といった具
log(observed/expected)として表現し
ります。

```
in_f <- "human_2mer.txt"
out_f1 <- "hoge12.txt"
out_f2 <- "hoge12.png"
param_bsgenome <- "BSge me.Hsap
param_fig <- c(700, 400)

#必要なパッケージをロード
library(Biostrings)
library(param_bsgenome, characte

#入力ファイルの読み込み
data <- read.table(in_f, header=

#前処理(指定したパッケージ中のオブジ
```

②



参考

作図(例題12)5

①例題12の、②出力結果(hoge12.png)。
③観測値(実際の2連続塩基の出現確率)と期待値(塩基ごとの出現確率を掛け合わせたもの)が似ている場合は、0付近にプロットされる。

(Rで)塩基配列解析

保護されていない通信 | iu.a.u-tokyo.ac.jp/~kadota/r_seq.html#intro_general_kmer_2_bios...

ゲスト

①

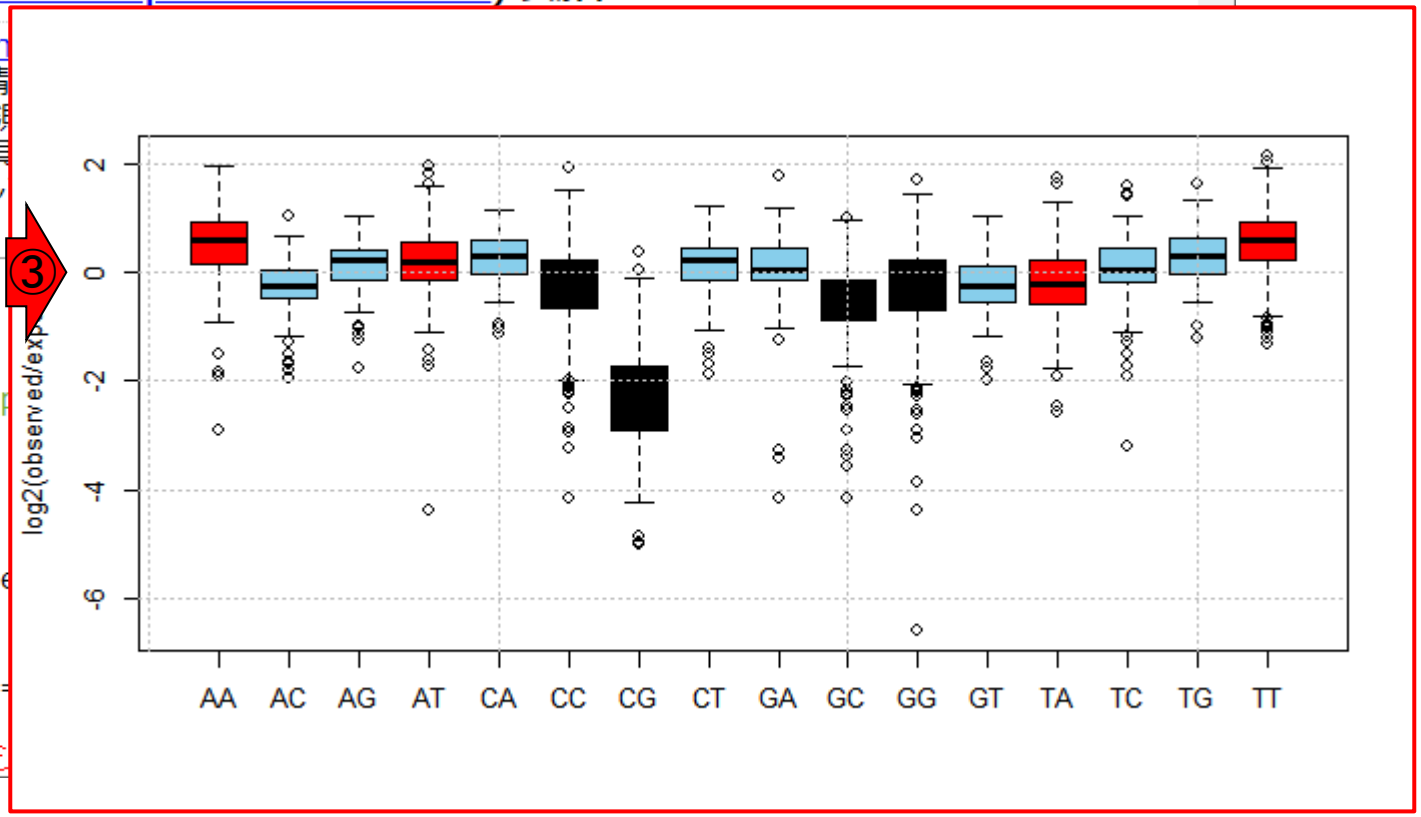
12. ヒトゲノム配列パッケージ(BSgenome.Hsapiens.NCBI.GRCh38)の場合：

11.と基本的に同じですが、human_2m
ボックスプロット(box plot)上での色情
ットも期待値(expected)と同程度の観測
ほうが小さければマイナス」といった具
log(observed/expected)として表現し
ります。

```
in_f <- "human_2mer.txt"
out_f1 <- "hoge12.txt"
out_f2 <- "hoge12.png"
param_bsgenome <- "BSge me.Hsap
param_fig <- c(700, 400)

#必要なパッケージをロード
library(Biostrings)
library(param_bsgenome, characte

#入力ファイルの読み込み
data <- read.table(in_f, header=

#前処理(指定したパッケージ中のオブジ
```

②

③



作図(例題12)6

①例題12の、②出力結果(hoge12.png)。③観測値(実際の2連続塩基の出現確率)と期待値(塩基ごとの出現確率を掛け合わせたもの)が似ている場合は、0付近にプロットされる。logの底は2で計算しているので、④縦軸が2に相当するのが、観測値のほうが期待値の2^2 = 4倍出現確率が高いことを意味する。

(Rで)塩基配列解析

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

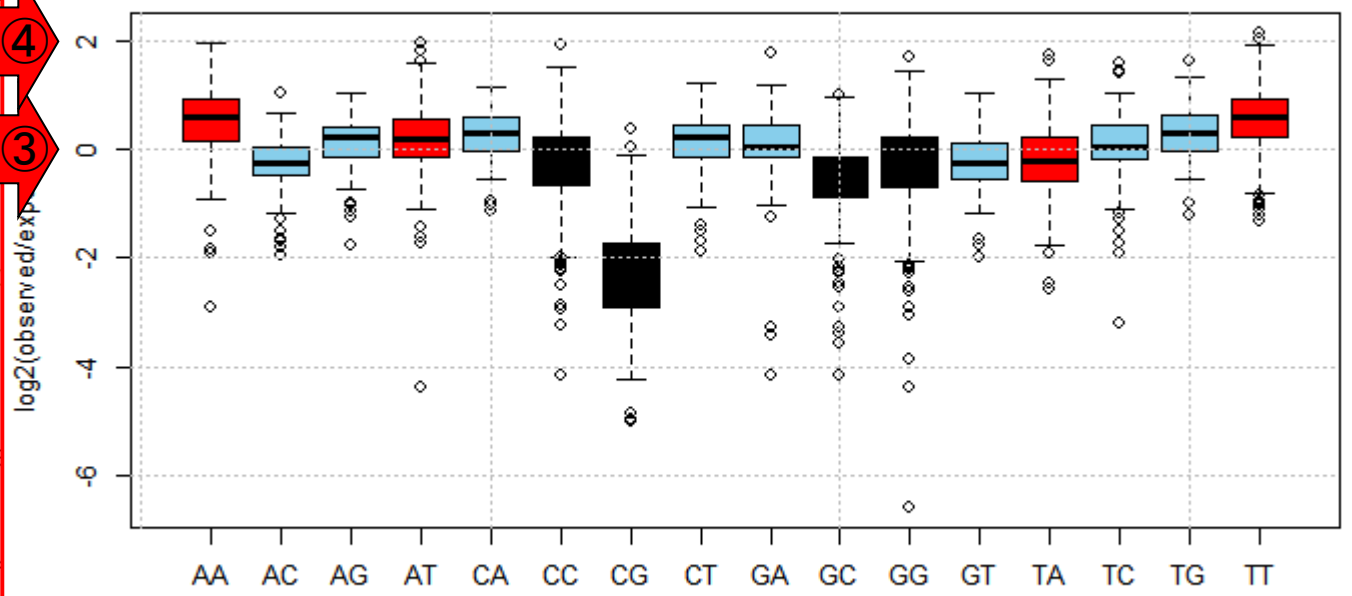
① ②

12. ヒトゲノム配列パッケージ(BSgenome.Hsapiens.NCBI.GRCh38)の場

11.と基本的に同じですが、human_2n
ボックスプロット(box plot)上での色情
ットも期待値(expected)と同程度の観測
ほうが小さければマイナス」といった具
log(observed/expected)として表現し
ります。

```
in_f <- "human_2mer.txt"
out_f1 <- "hoge12.txt"
out_f2 <- "hoge12.png"
param_bsgenome <- "BSge me.Hsap
param_fig <- c(700, 400)

#必要なパッケージをロード
library(Biostrings)
library(param_bsgenome, characte

#入力ファイルの読み込み
data <- read.table(in_f, header=

#前処理(指定したパッケージ中のオブジ
```

④ ③



参考

作図(例題12)7

①例題12の、②出力結果(hoge12.png)。③観測値(実際の2連続塩基の出現確率)と期待値(塩基ごとの出現確率を掛け合わせたもの)が似ている場合は、0付近にプロットされる。logの底は2で計算しているので、④縦軸が2に相当するのが、観測値のほうが期待値の2^2 = 4倍出現確率が高いことを意味する。同様にして、⑤縦軸が−2に相当するのが、観測値のほうが期待値の2^(−2) = 1/4倍出現確率が高い(つまり4倍出現確率が低い)ことを意味する。

(Rで)塩基配列解析

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

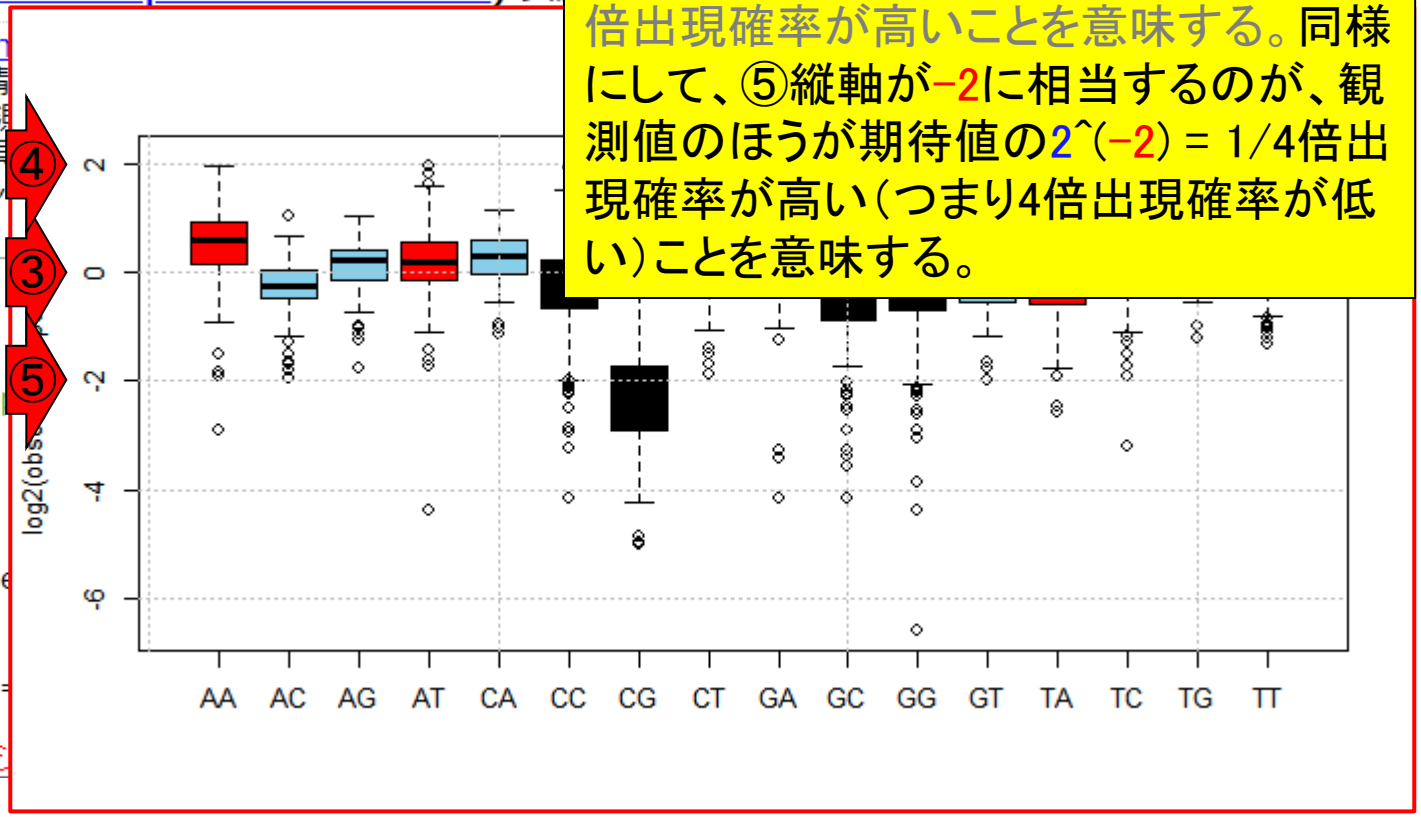
① ②

12. ヒトゲノム配列パッケージ(BSgenome.Hsapiens.NCBI.GRCh38)の場

11.と基本的に同じですが、human_2n
ボックスプロット(box plot)上での色情
ットも期待値(expected)と同程度の観測
ほうが小さければマイナス」といった具
log(observed/expected)として表現し
ります。

```
in_f <- "human_2mer.txt"
out_f1 <- "hoge12.txt"
out_f2 <- "hoge12.png"
param_bsgenome <- "BSge me.Hsa
param_fig <- c(700, 400)

#必要なパッケージをロード
library(Biostrings)
library(param_bsgenome, characte

#入力ファイルの読み込み
data <- read.table(in_f, header=

#前処理(指定したパッケージ中のオブジ
```

④ ③ ⑤



作図(例題12)8

このように期待値の違いを補正した結果で眺めると、確かに⑥CGの出現確率が期待値よりも大幅に低いことがよくわかりますね。

(Rで)塩基配列解析

保護されていない通信 | iu.a.u-tokyo.ac.jp/~kadota/r_seq.html#intro_general_kmer_2_bios...

ゲスト

①

12. ヒトゲノム配列パッケージ(BSgenome.Hsapiens.NCBI.GRCh38)の場合：

11.と基本的に同じですが、human_2n
ボックスプロット(box plot)上での色情
ットも期待値(expected)と同程度の観測
ほうが小さければマイナス」といった具
log(observed/expected)として表現し
ります。

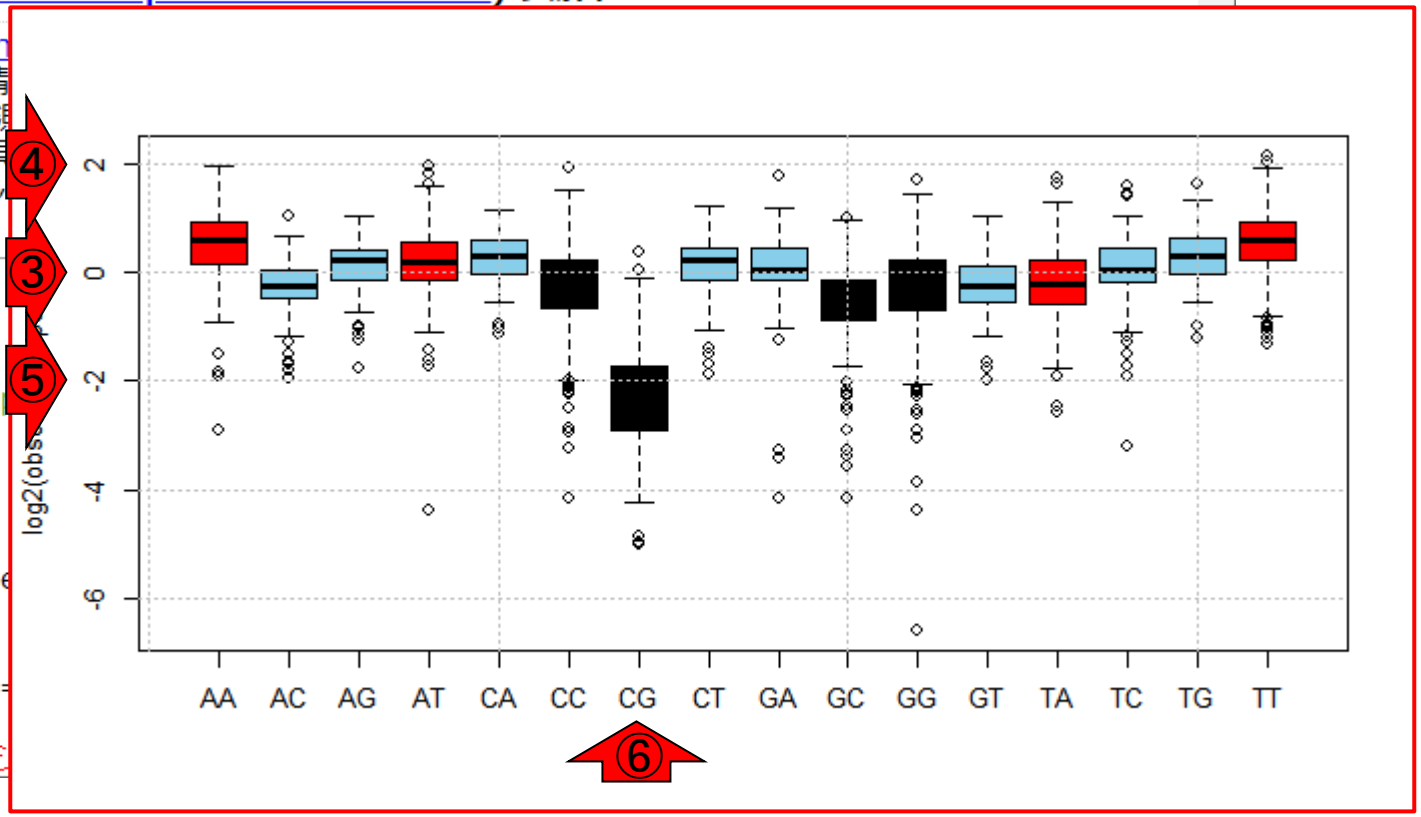
```
in_f <- "human_2mer.txt"
out_f1 <- "hoge12.txt"
out_f2 <- "hoge12.png"
param_bsgenome <- "BSge me.Hsa
param_fig <- c(700, 400)

#必要なパッケージをロード
library(Biostrings)
library(param_bsgenome, characte

#入力ファイルの読み込み
data <- read.table(in_f, header=

#前処理(指定したパッケージ中のオブジ
```

②

④

③

⑤



⑥

Contents

- Introduction、出現頻度解析（k=2）、出現頻度解析（k=1）
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図（例題10）、作図（例題11）、作図（例題12）
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ（例題32）、被覆率（coverage）、基本的な考え方(例題7)
  - 例題8（k=2)、例題9（k=3）、1,000塩基の仮想ゲノム（サンプルデータの例題33）
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

塩基配列解析の基礎

①では塩基配列解析の基礎的な項目を多く含んでいる。先ほどまで解説したGC含量の違いによる出現確率の期待値の違いなどがよくわからなかったヒトも、この後挙げるいくつかの項目を参考にして、仮想データを作成して解析してみると理解が深まるでしょう。

GC含量1

①では塩基配列解析の基礎的な項目を多く含んでいる。先ほどまで解説したGC含量の違いによる出現確率の期待値の違いなどがよくわからなかったヒトも、この後挙げるいくつかの項目を参考にして、仮想データを作成して解析してみると理解が深まるでしょう。例えば、②は配列ごとのGC含量を返す項目。

GC含量2

①では塩基配列解析の基礎的な項目を多く含んでいる。先ほどまで解説したGC含量の違いによる出現確率の期待値の違いなどがよくわからなかったヒトも、この後挙げるいくつかの項目を参考にして、仮想データを作成して解析してみると理解が深まるでしょう。例えば、②は配列ごとのGC含量を返す項目。配列ごとではなく、入力ファイル全体のGC含量を得たい場合は、③の例題3が便利。

GC含量3

③

①では塩基配列解析の基礎的な項目を多く含んでいる。先ほどまで解説したGC含量の違いによる出現確率の期待値の違いなどがよくわからなかったヒトも、この後挙げるいくつかの項目を参考にして、仮想データを作成して解析してみると理解が深まるでしょう。例えば、②は配列ごとのGC含量を返す項目。配列ごとではなく、入力ファイル全体のGC含量を得たい場合は、③の例題3が便利。③の、④例題3。⑤全配列をまとめて…と書かれているのがわかります。まずはこれをコピペ実行。

(Rで)塩基配列解析

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

④

3. イントロ | 一般 | ランダムな塩基配列を作成の4.を実行して得られたmult

配列ごとではなく、全配列をまとめて出現頻度をカウントした結果を返すやり

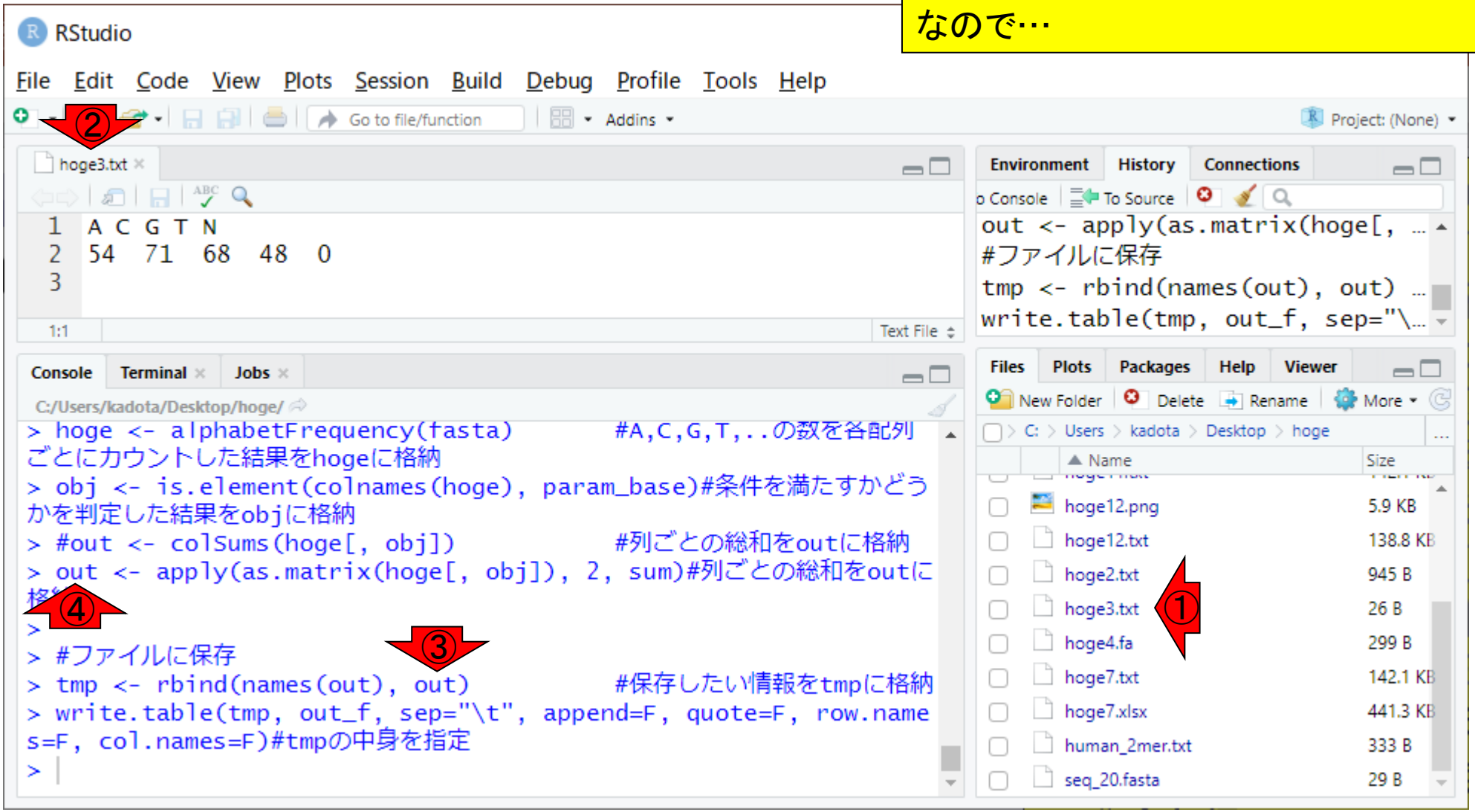
⑤

```
in_f <- "hoge4.fa"                    #入力ファイル名を指定して
out_f <- "hoge3.txt"                  #出力ファイル名を指定して
param_base <- c("A", "C", "G", "T", "N")#出力させたい塩基を指定

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

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

#本番
hoge <- alphabetFrequency(fasta)      #A,C,G,T,..の数を各配列ごとにカウントした結果をhogeに格納
obj <- is.element(colnames(hoge), param_base)#条件を満たすかどうかを判定した結果をobjに格納
#out <- colSums(hoge[, obj])          #列ごとの総和をoutに格納
out <- apply(as.matrix(hoge[, obj]), 2, sum)#列ごとの総和をoutに格納

#ファイルに保存
tmp <- rbind(names(out), out)         #保存したい情報をtmpに格納
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F, col.names=F)#tmpの中身を指定した
```

トップページへ

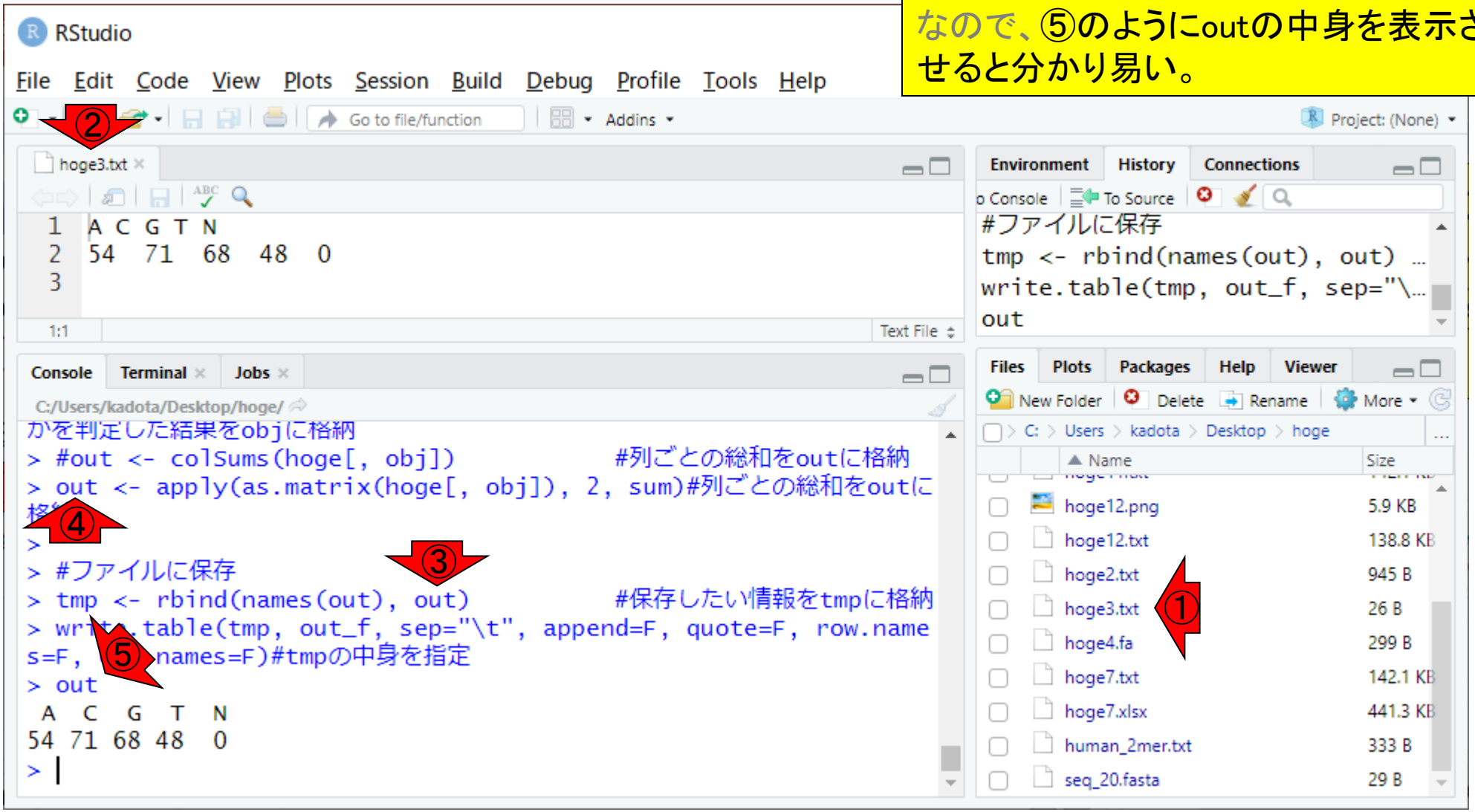
GC含量4

コピペ実行後に、①出力ファイル（hoge3.txt）を、②エディタで概観。③出力ファイルの中身は、④で作成したものなので…



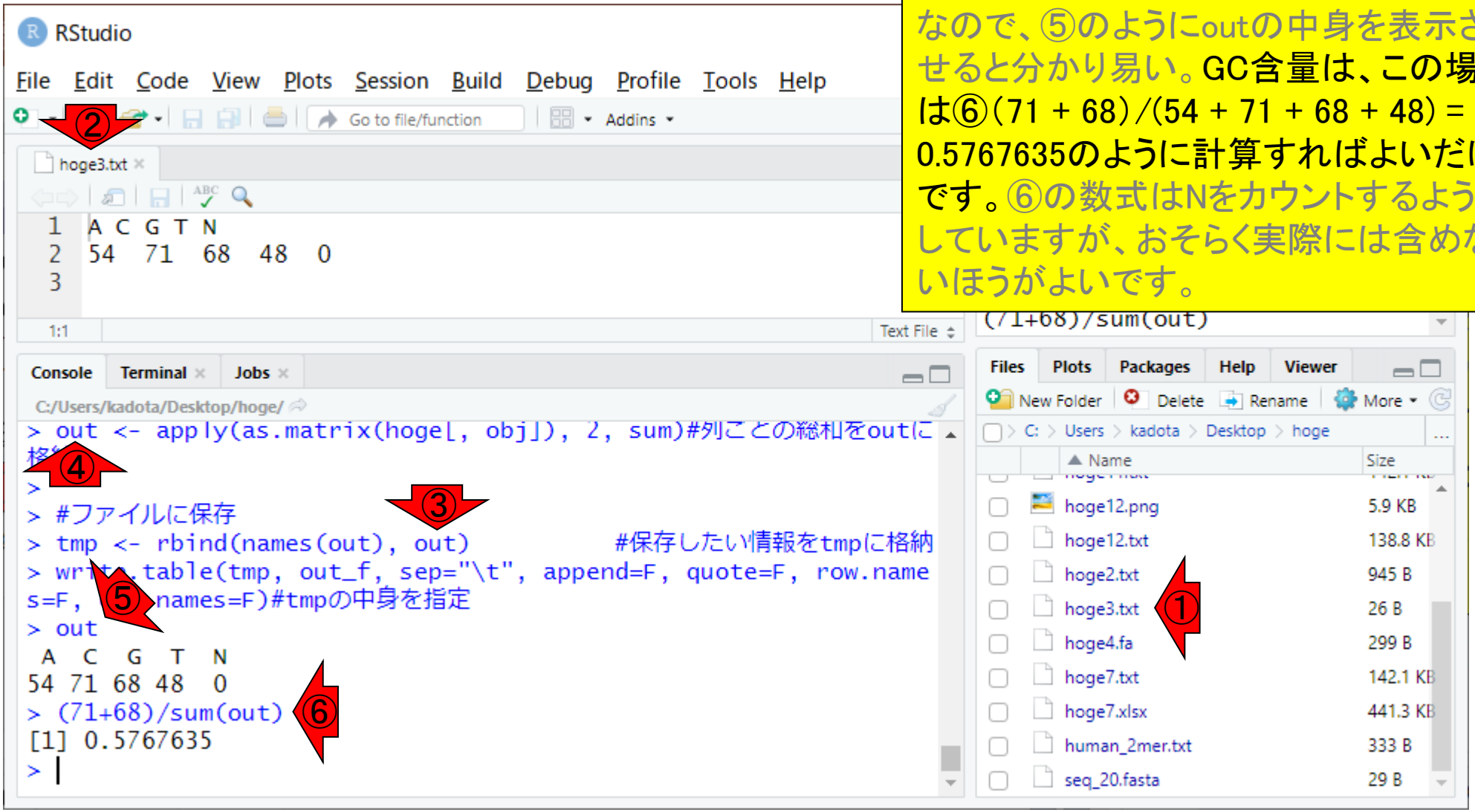
GC含量5

コピペ実行後に、①出力ファイル（hoge3.txt）を、②エディタで概観。③出力ファイルの中身は、④で作成したものなので、⑤のようにoutの中身を表示させると分かり易い。



GC含量6

コピペ実行後に、①出力ファイル（hoge3.txt）を、②エディタで概観。③出力ファイルの中身は、④で作成したものなので、⑤のようにoutの中身を表示させると分かり易い。GC含量は、この場合は⑥(71 + 68)/(54 + 71 + 68 + 48) = 0.5767635のように計算すればよいだけです。⑥の数式はNをカウントするようにしていますが、おそらく実際には含めないほうがよいです。



GC含量7

コピペ実行後に、①出力ファイル（hoge3.txt）を、②エディタで概観。③出力ファイルの中身は、④で作成したものなので、⑤のようにoutの中身を表示させると分かり易い。GC含量は、この場合は⑥(71 + 68)/(54 + 71 + 68 + 48) = 0.5767635のように計算すればよいだけです。⑥の数式はNをカウントするようにしていますが、おそらく実際には含めないほうがよいです。⑦こんな感じで入力すれば、塩基の種類ごとの出現確率情報も取得可能。

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

Console | Terminal | Jobs

C:/Users/kadota/Desktop/hoge/

```
ごとにカウントした結果をhogeに格納
> obj <- is.element(colnames(hoge), param_base)#条件を満たすかど
かを判定した結果をobjに格納
> #out <- colSums(hoge[, obj])              #列ごとの総和をoutに格納
> out <- apply(as.matrix(hoge[, obj]), 2, sum)#列ごとの総和をout
格納
>
> #ファイルに保存
> tmp <- rbind(names(out), out)             #保存したい情報をtmpに格納
> write.table(tmp, out_f, sep="\t", append=F, quote=F, row.name
s=F, col.names=F)#tmpの中身を指定
> out
 A  C  G  T  N
54 71 68 48  0
> (71+68)/sum(out)
[1] 0.5767635
> out/sum(out)
        A         C         G         T         N
0.2240664 0.2946058 0.2821577 0.1991701 0.0000000
>
```

④ ③ ⑤ ⑥ ⑦

C: > Users > kadota > Desktop > hoge

Name	Size
hoge12.png	5.9 KB
hoge12.txt	138.8 KB
hoge2.txt	945 B
hoge3.txt	26 B
hoge4.fa	299 B
hoge7.txt	142.1 KB
hoge7.xlsx	441.3 KB
human_2mer.txt	333 B
seq_20.fasta	29 B

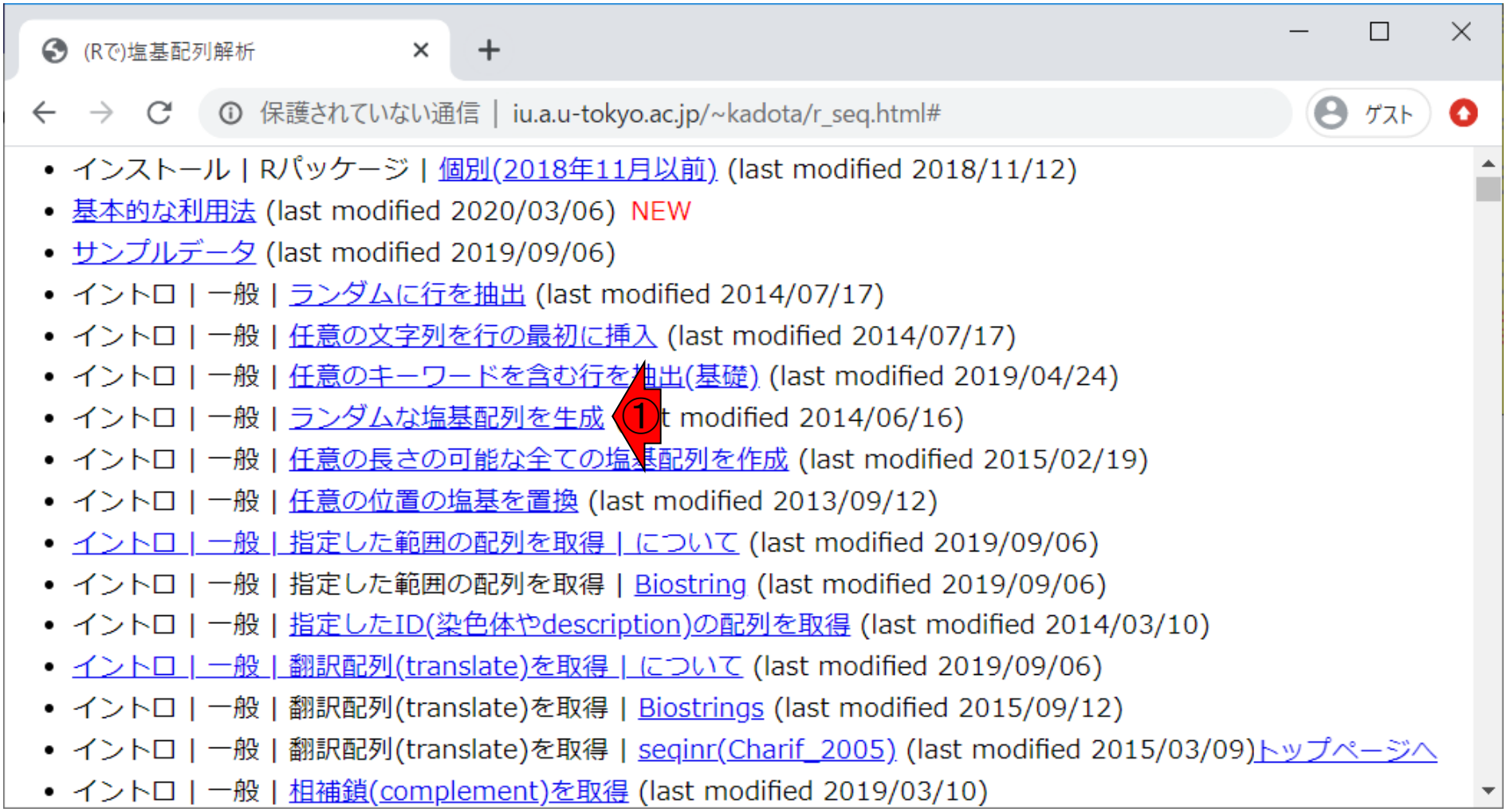
①

Contents

- Introduction、出現頻度解析（k=2）、出現頻度解析（k=1）
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図（例題10）、作図（例題11）、作図（例題12）
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ（例題32）、被覆率（coverage）、基本的な考え方(例題7)
  - 例題8（k=2)、例題9（k=3）、1,000塩基の仮想ゲノム（サンプルデータの例題33）
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

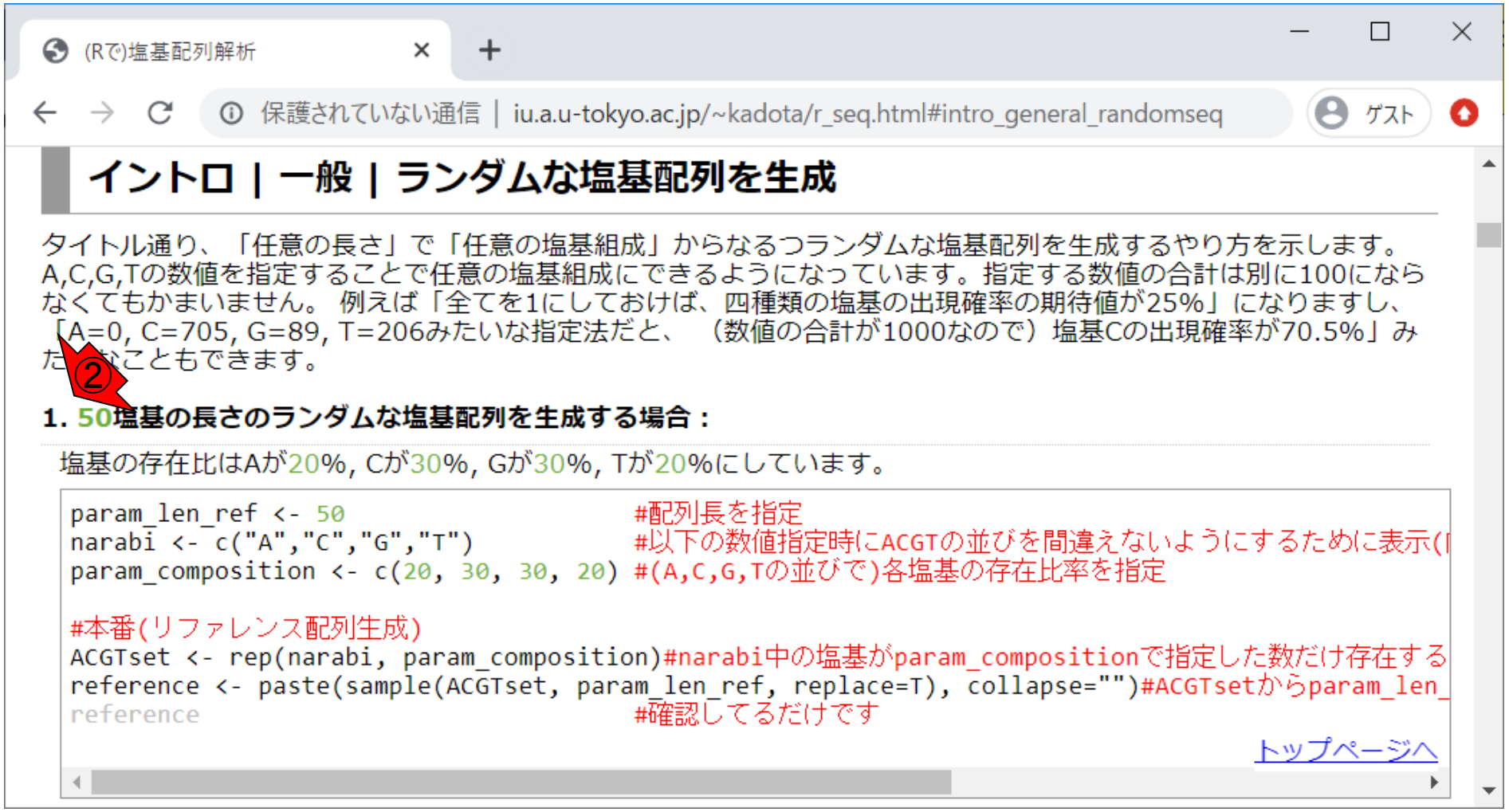
参考 ①任意の存在比を与えて、任意の長さの塩基配列を生成する項目。

ランダム配列を生成1



参考 ①任意の存在比を与えて、任意の長さの塩基配列を生成する項目。②例題1。

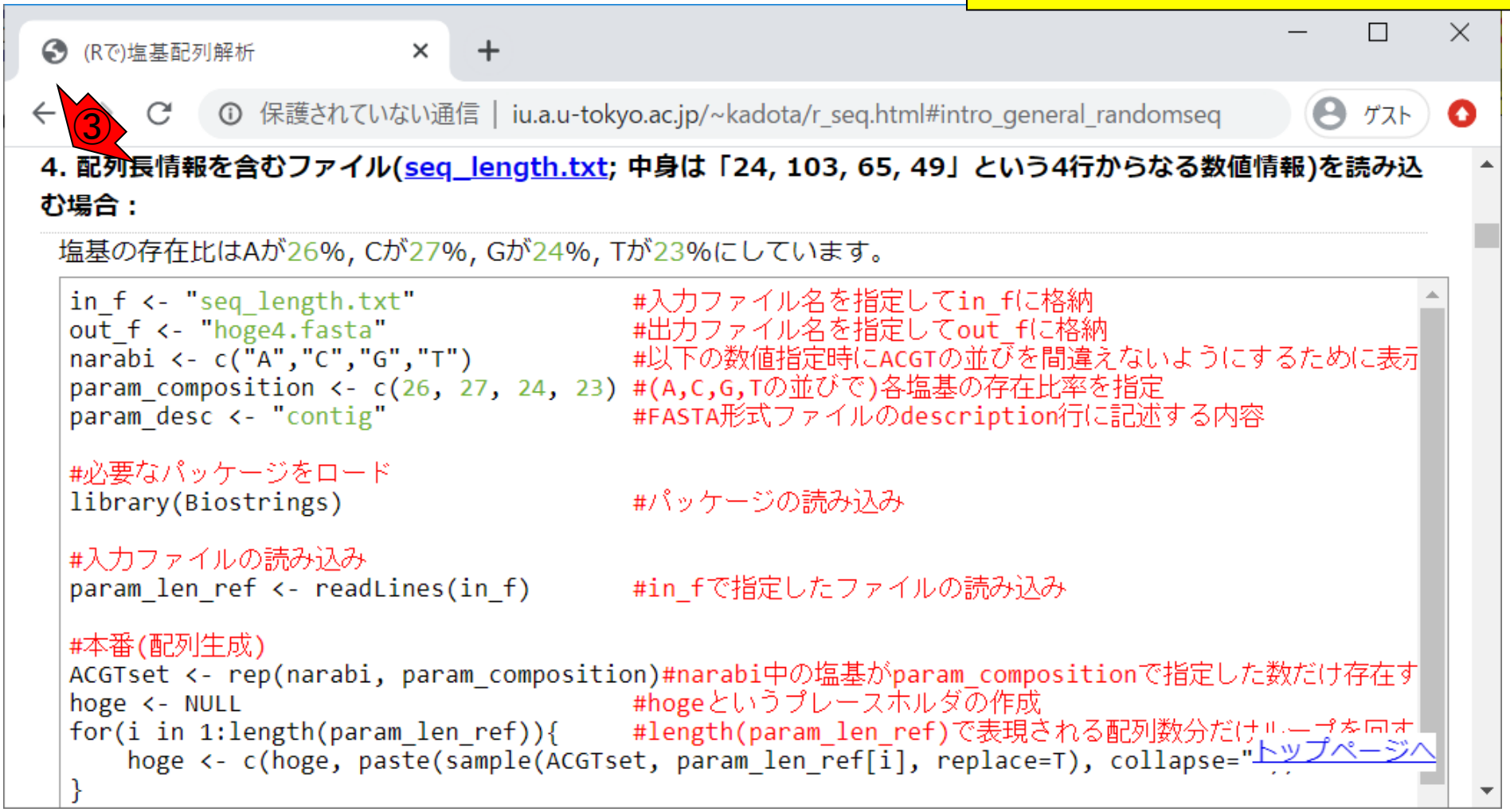
ランダム配列を生成2



参考

①任意の存在比を与えて、任意の長さの塩基配列を生成する項目。②例題1。③例題4。

ランダム配列を生成3



4. 配列長情報を含むファイル(seq_length.txt; 中身は「24, 103, 65, 49」という4行からなる数値情報)を読み込む場合：

塩基の存在比はAが26%, Cが27%, Gが24%, Tが23%にしています。

```
in_f <- "seq_length.txt"                #入力ファイル名を指定してin_fに格納
out_f <- "hoge4.fasta"                  #出力ファイル名を指定してout_fに格納
narabi <- c("A","C","G","T")            #以下の数値指定時にACGTの並びを間違えないようにするために表示
param_composition <- c(26, 27, 24, 23) #(A,C,G,Tの並びで)各塩基の存在比率を指定
param_desc <- "contig"                  #FASTA形式ファイルのdescription行に記述する内容

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

#入力ファイルの読み込み
param_len_ref <- readLines(in_f)        #in_fで指定したファイルの読み込み

#本番(配列生成)
ACGTset <- rep(narabi, param_composition)#narabi中の塩基がparam_compositionで指定した数だけ存在す
hoge <- NULL                            #hogeというプレースホルダの作成
for(i in 1:length(param_len_ref)){      #length(param_len_ref)で表現される配列数分だけループを回す
    hoge <- c(hoge, paste(sample(ACGTset, param_len_ref[i], replace=T), collapse="
}
```

トップページへ

参考

ランダム配列を生成4

①任意の存在比を与えて、任意の長さの塩基配列を生成する項目。②例題1。③例題4。(A, C, G, Tの並びで)④で与えた各塩基の存在比率(GC含量だと51%)にして、⑤の長さをもつ計4配列からなるランダムな塩基配列を生成し、⑥hoge4.fastaというファイル名で保存するスクリプト。

参考

ランダム配列を生成5

①任意の存在比を与えて、任意の長さの塩基配列を生成する項目。②例題1。③例題4。(A, C, G, Tの並びで)④で与えた各塩基の存在比率(GC含量だと51%)にして、⑤の長さをもつ計4配列からなるランダムな塩基配列を生成し、⑥hoge4.fastaというファイル名で保存するスクリプト。今回多くの入力ファイルとして使っているhoge4.faはこの項目の③例題4をコピペ実行して得たものです。

参考

ランダム配列を生成6

①任意の存在比を与えて、任意の長さの塩基配列を生成する項目。②例題1。③例題4。(A, C, G, Tの並びで)④で与えた各塩基の存在比率(GC含量だと51%)にして、⑤の長さをもつ計4配列からなるランダムな塩基配列を生成し、⑥hoge4.fastaというファイル名で保存するスクリプト。今回多くの入力ファイルとして使っているhoge4.faはこの項目の③例題4をコピペ実行して得たものです。この例題では常に同じ乱数を発生させるような操作を行っていない(set.seedという関数を利用していない)ため、⑥の出力ファイルとhoge4.faの塩基組成は異なりますが、⑤配列数と配列長は同じです。

参考

①の項目の、②例題4を実行して、ある試行でたまたま得た結果ファイルがhoge4.faということ。それが、例えば…

ランダム配列を生成7

①

②

(Rで)塩基配列解析

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

ゲスト

4. 配列長情報を含むファイル(seq_length.txt; 中身は「24, 103, 65, 49」という4行からなる数値情報)を読み込む場合：

塩基の存在比はAが26%, Cが27%, Gが24%, Tが23%にしています。

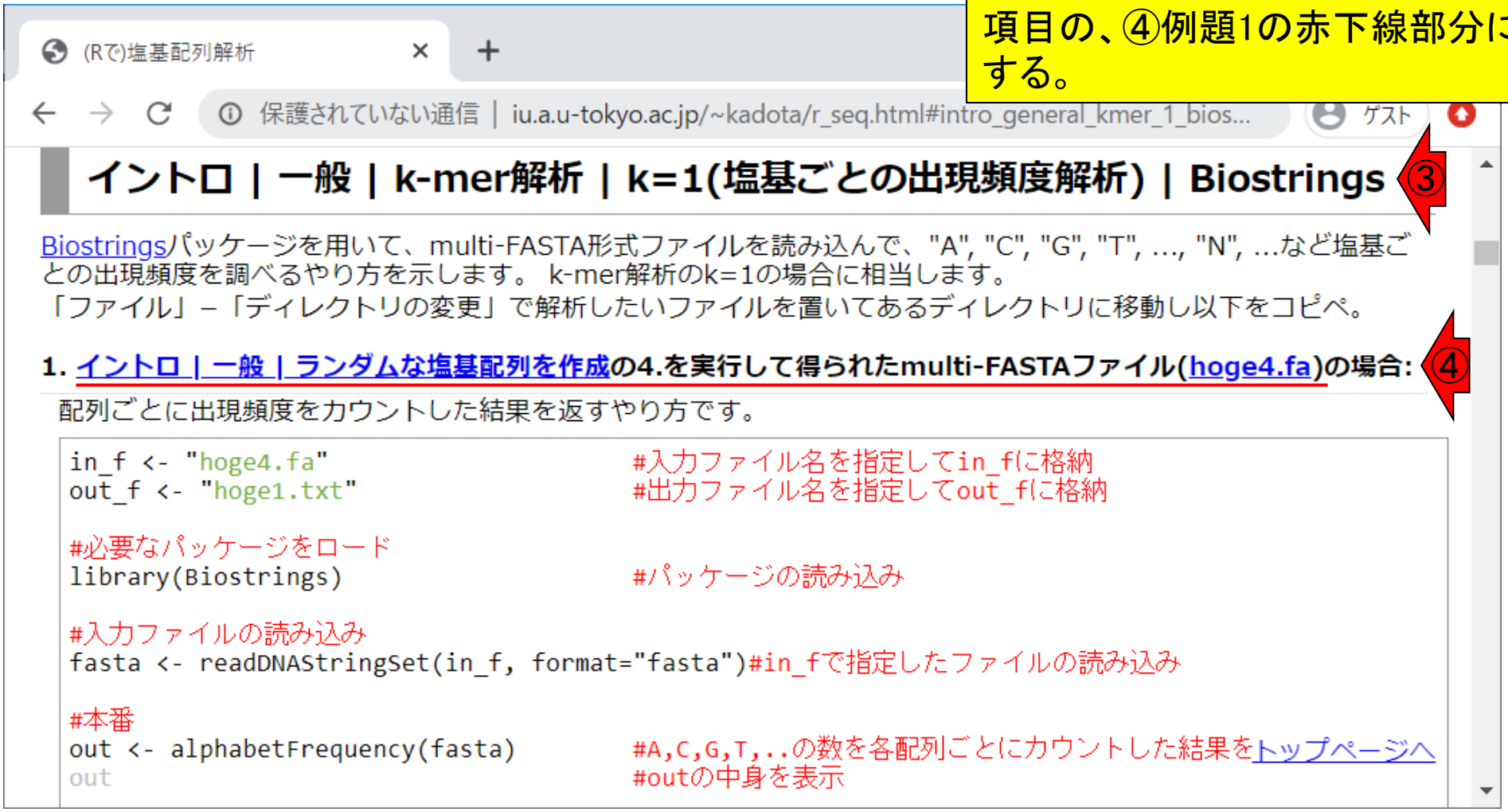
```
in_f <- "seq_length.txt"                #入力ファイル名を指定してin_fに格納
out_f <- "hoge4.fasta"                  #出力ファイル名を指定してout_fに格納
narabi <- c("A","C","G","T")            #以下の数値指定時にACGTの並びを間違えないようにするために表示
param_composition <- c(26, 27, 24, 23) #(A,C,G,Tの並びで)各塩基の存在比率を指定
param_desc <- "contig"                  #FASTA形式ファイルのdescription行に記述する内容

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

#入力ファイルの読み込み
param_len_ref <- readLines(in_f)        #in_fで指定したファイルの読み込み

#本番(配列生成)
ACGTset <- rep(narabi, param_composition)#narabi中の塩基がparam_compositionで指定した数だけ存在す
hoge <- NULL                            #hogeというプレースホルダの作成
for(i in 1:length(param_len_ref)){      #length(param_len_ref)で表現される配列数分だけループを回す
    hoge <- c(hoge, paste(sample(ACGTset, param_len_ref[i], replace=T), collapse="
}
```

トップページへ

• イントロ | 一般 | k-mer解析 | k=1(塩基ごとの出現頻度解析) | Biostrings 参考

ランダム配列を生成8

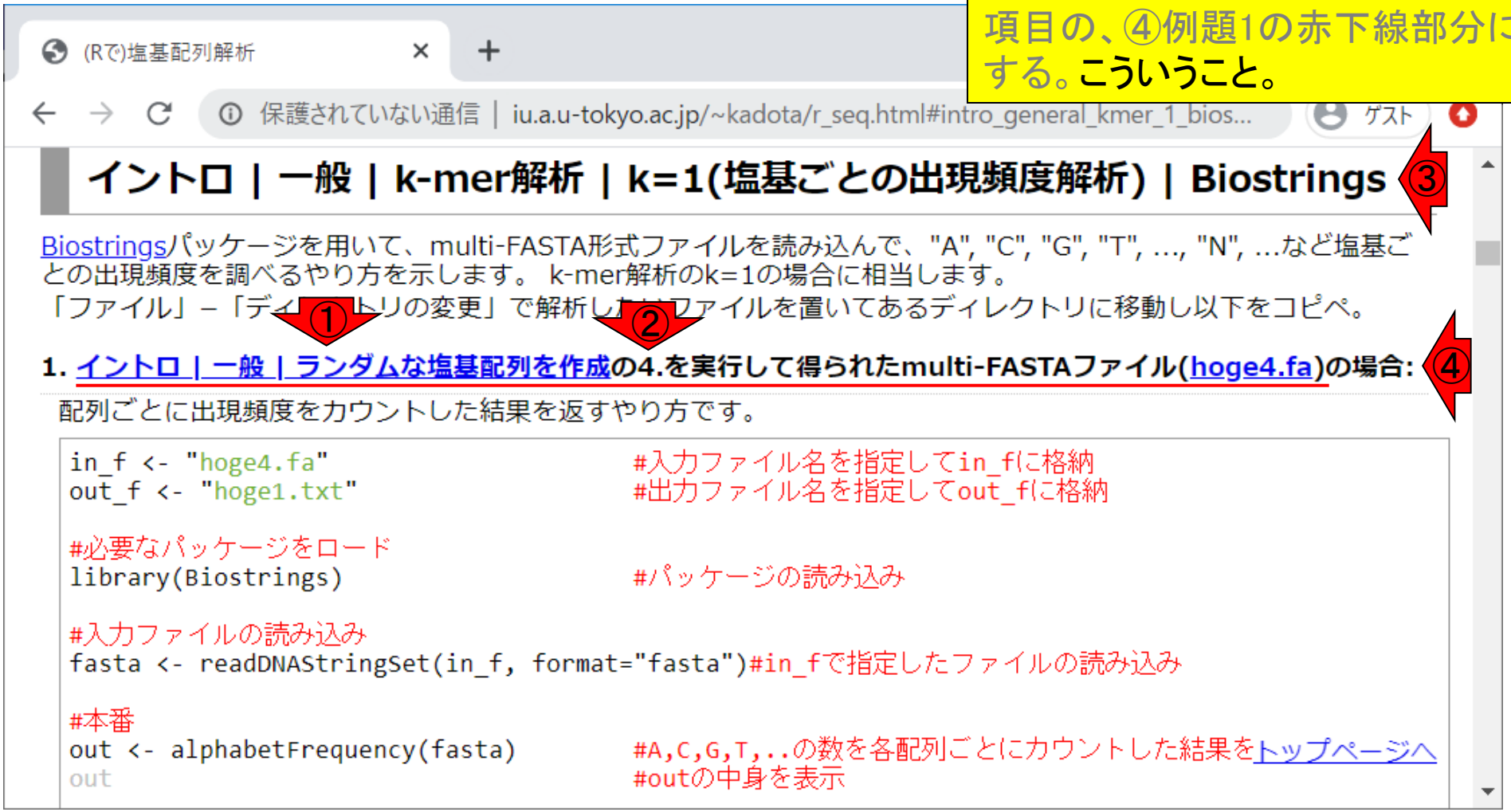
①の項目の、②例題4を実行して、ある試行でたまたま得た結果ファイルがhoge4.faということ。それが、例えば③の項目の、④例題1の赤下線部分に対応する。



• イントロ | 一般 | k-mer解析 | k=1(塩基ごとの出現頻度解析) | Biostrings 参考

ランダム配列を生成9

①の項目の、②例題4を実行して、ある試行でたまたま得た結果ファイルがhoge4.faということ。それが、例えば③の項目の、④例題1の赤下線部分に対応する。こういうこと。



Contents

- Introduction、出現頻度解析（k=2）、出現頻度解析（k=1）
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図（例題10）、作図（例題11）、作図（例題12）
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ（例題32）、被覆率（coverage）、基本的な考え方(例題7)
  - 例題8（k=2)、例題9（k=3）、1,000塩基の仮想ゲノム（サンプルデータの例題33）
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

参考

部分配列の切り出し1

さきほどまでの①で、任意の塩基組成からなる、任意の長さの仮想ゲノム配列を作成可能になった。次は、②で仮想ゲノム配列から、任意の範囲の部分配列を切り出すやり方を示します。②をクリック。

部分配列の切り出し2

さきほどまでの①で、任意の塩基組成からなる、任意の長さの仮想ゲノム配列を作成可能になった。次は、②で仮想ゲノム配列から、任意の範囲の部分配列を切り出すやり方を示します。②をクリック。こんな感じになります。③例題1の、④入力ファイルの、⑤中身。

(Rで)塩基配列解析

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

イントロ | 一般 | 指定した範囲の配列を取得 | Biost

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

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 ペ。

⑤

>kadota↓
AGTGACGGTCTT

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

任意の範囲 (始点が3, 終点が9)の配列を抽出するやり方です。

④

```
in_f <- "sample1.fasta"          #入力ファイル名を指定してin_fに格納
out_f <- "hoge1.fasta"           #出力ファイル名を指定してout_fに格納
param <- c(3, 9)                 #抽出したい範囲の始点と終点を指定

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

トップページへ

```
#入力ファイルの読み込み
fasta <- readDNAStringSet(in_f, format="fasta")#in_fで指定したファイルの読み込み
```

参考

部分配列の切り出し3

さきほどまでの①で、任意の塩基組成からなる、任意の長さの仮想ゲノム配列を作成可能になった。次は、②で仮想ゲノム配列から、任意の範囲の部分配列を切り出すやり方を示します。②をクリック。こんな感じになります。③例題1の、④入力ファイルの、⑤中身。⑥で指定した3−9番目の、⑦部分塩基配列を切り出した結果が…

(Rで)塩基配列解析

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

イントロ | 一般 | 指定した範囲の配列を取得 | Biost

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

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

任意の範囲 (始点が3, 終点が9)の配列を抽出するやり方です。

```
in_f <- "sample1.fasta"                #入力ファイル名を指定してin_fに格納
out_f <- "hoge1.fasta"                 #出力ファイル名を指定してout_fに格納
param <- c(3, 9)                       #抽出したい範囲の始点と終点を指定

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

                                                    トップページへ
#入力ファイルの読み込み
fasta <- readDNAStringSet(in_f, format="fasta")#in_fで指定したファイルの読み込み
```

⑥

>kadota↓
AGTGACGGTCTT
⑦

参考

部分配列の切り出し4

さきほどまでの①で、任意の塩基組成からなる、任意の長さの仮想ゲノム配列を作成可能になった。次は、②で仮想ゲノム配列から、任意の範囲の部分配列を切り出すやり方を示します。②をクリック。こんな感じになります。③例題1の、④入力ファイルの、⑤中身。⑥で指定した3-9番目の、⑦部分塩基配列を切り出した結果が、⑧の出力ファイル（hoge1.fasta）。

(Rで)塩基配列解析

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

イントロ | 一般 | 指定した範囲の配列を取得 | Biost

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

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

任意の範囲 (始点が3, 終点が9)の配列を抽出するやり方です。

```
in_f <- "sample1.fasta"                    #入力ファイル名を指定してin_fに格納
out_f <- "hoge1.fasta"                     #出力ファイル名を
param <- c(3, 9)                           #抽出したい範囲の

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

#入力ファイルの読み込み
fasta <- readDNAStringSet(in_f, format="fasta")#in_fで指定したファイルの読み込み
```

⑧

>kadota↓
AGTGACGGTCTT

⑧

>kadota
TGACGGT

トップページへ

参考

部分配列の切り出し5

さきほどまでの①で、任意の塩基組成からなる、任意の長さの仮想ゲノム配列を作成可能になった。次は、②で仮想ゲノム配列から、任意の範囲の部分配列を切り出すやり方を示します。②をクリック。こんな感じになります。③例題1の、④入力ファイルの、⑤中身。⑥で指定した3-9番目の、⑦部分塩基配列を切り出した結果が、⑧の出力ファイル（hoge1.fasta）。
RStudioの実行結果画面でも分かります。

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

Console | Terminal | Jobs

C:/Users/kadota/Desktop/hoge/

```
> fasta <- readDNAStringSet(in_f, format="fasta")#in_fで指定し
ファイルの読み込み
> fasta                                    #確認してるだけです
  A DNAStringSet instance of length 1
    width seq                              names
[1]    12 AGTGACGGTCTT                     kadota
>
> #本番
> fasta <- subseq(fasta, param[1], param[2])#paramで指
終点の範囲の配列を抽出した結果をfastaに格納
> fasta                                    #確認してるだけです
  A DNAStringSet instance of length 1
    width seq                              names
[1]     7 TGACGGT                          kadota
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta",
fastaの中身を指定したファイル名で保存
>
>
```

writeXStringSet(fasta, file=ou...

Help | Viewer

Rename | More

Desktop > hoge

Size
14.8 KB
221 B
6.2 KB
142.1 KB
6.3 KB
142.1 KB
5.9 KB
138.8 KB

.Rhistory
hoge1.txt
hoge10.png
hoge10.txt
hoge11.png
hoge11.txt
hoge12.png
hoge12.txt

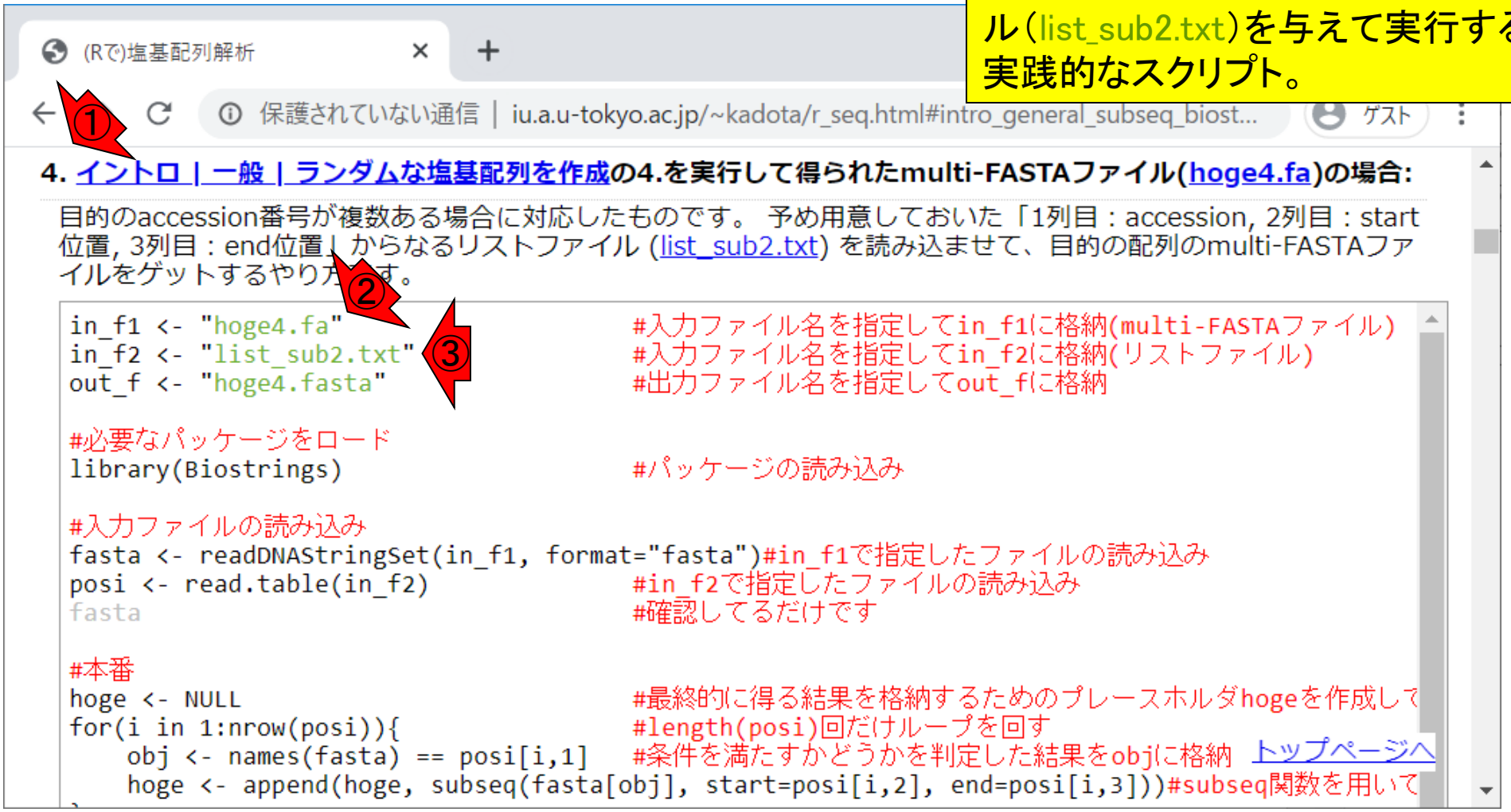
```
>kadota↓
AGTGACGGTCTT
```

```
>kadota
TGACGGT
```

参考

部分配列の切り出し6

①例題4が、②よく知っている入力ファイル（hoge4.fa）を使っており、③任意の配列の任意の領域情報を含むリストファイル（list_sub2.txt）を与えて実行する、より実践的なスクリプト。



4. イントロ | 一般 | ランダムな塩基配列を作成の4.を実行して得られたmulti-FASTAファイル(hoge4.fa)の場合:

目的のaccession番号が複数ある場合に対応したものです。 予め用意しておいた「1列目：accession, 2列目：start位置, 3列目：end位置」からなるリストファイル (list_sub2.txt) を読み込ませて、目的の配列のmulti-FASTAファイルをゲットするやり方です。

```
in_f1 <- "hoge4.fa"                    #入力ファイル名を指定してin_f1に格納(multi-FASTAファイル)
in_f2 <- "list_sub2.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 <- "hoge4.fasta"                 #出力ファイル名を指定してout_fに格納

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

#入力ファイルの読み込み
fasta <- readDNAStringSet(in_f1, format="fasta")#in_f1で指定したファイルの読み込み
posi <- read.table(in_f2)              #in_f2で指定したファイルの読み込み
fasta                                  #確認してるだけです

#本番
hoge <- NULL                           #最終的に得る結果を格納するためのプレースホルダhogeを作成して
for(i in 1:nrow(posi)){                #length(posi)回だけループを回す
    obj <- names(fasta) == posi[i,1]   #条件を満たすかどうかを判定した結果をobjに格納
    hoge <- append(hoge, subseq(fasta[obj], start=posi[i,2], end=posi[i,3]))#subseq関数を用いて
}
```

トップページへ

Contents

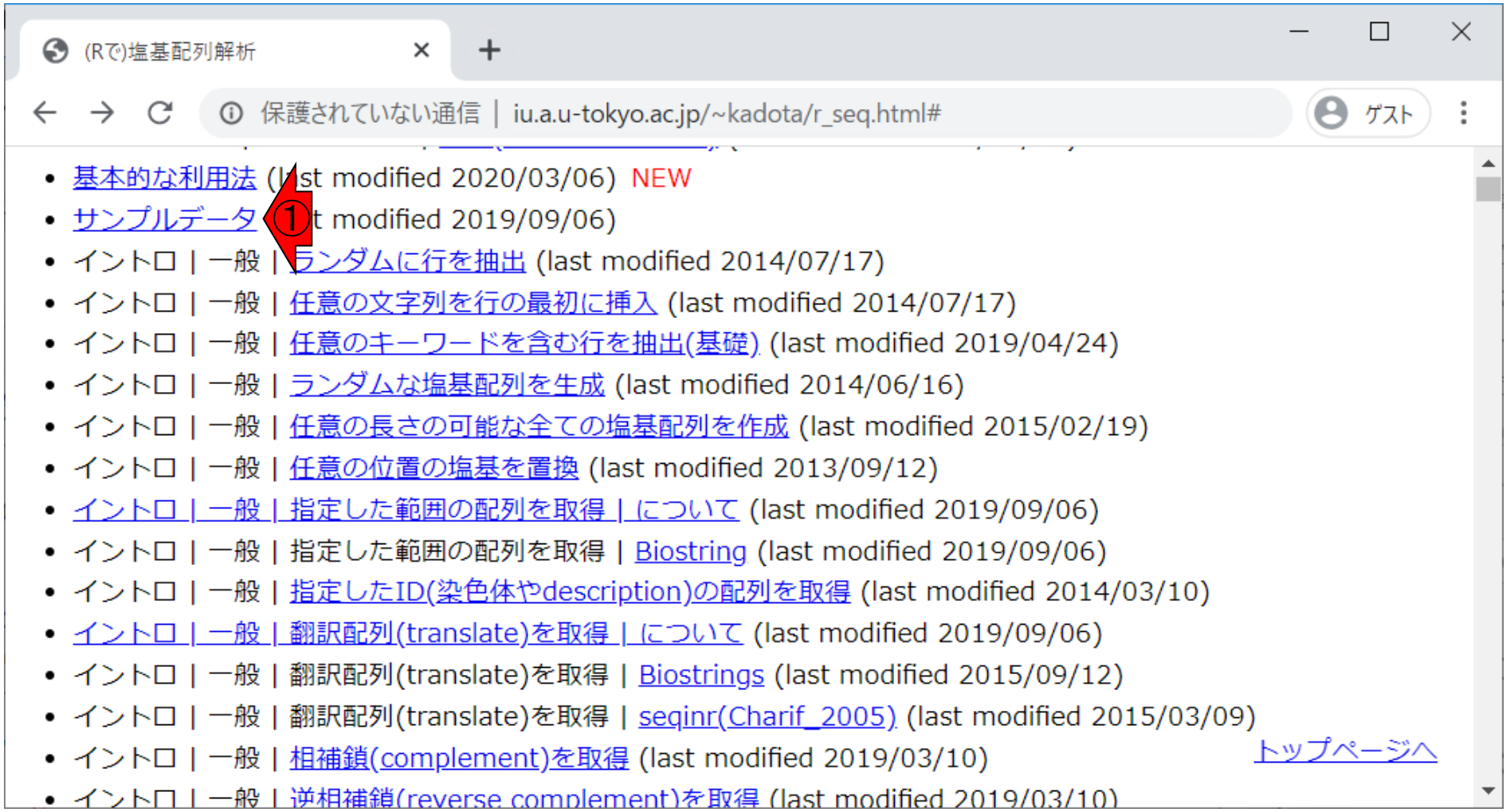
- Introduction、出現頻度解析(k=2)、出現頻
- k=1で実践、multi-FASTAファイル、他の例
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図(例題10)、作図(例題11)、作図(例題12)
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ(例題32)、被覆率(coverage)、基本的な考え方(例題7)
  - 例題8(k=2)、例題9(k=3)、1,000塩基の仮想ゲノム(サンプルデータの例題33)
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

K-mer解析の応用編で、NGSデータのみを入力として、ゲノムサイズ推定までできちゃうんですよ、という話です。基本的な考え方をRStudio上でシミュレーションデータで示していく内容で、「おお！確かにできそうだね！」と楽しんでもらうのが目的です。レポートとは無関係部分です。

①サンプルデータの…

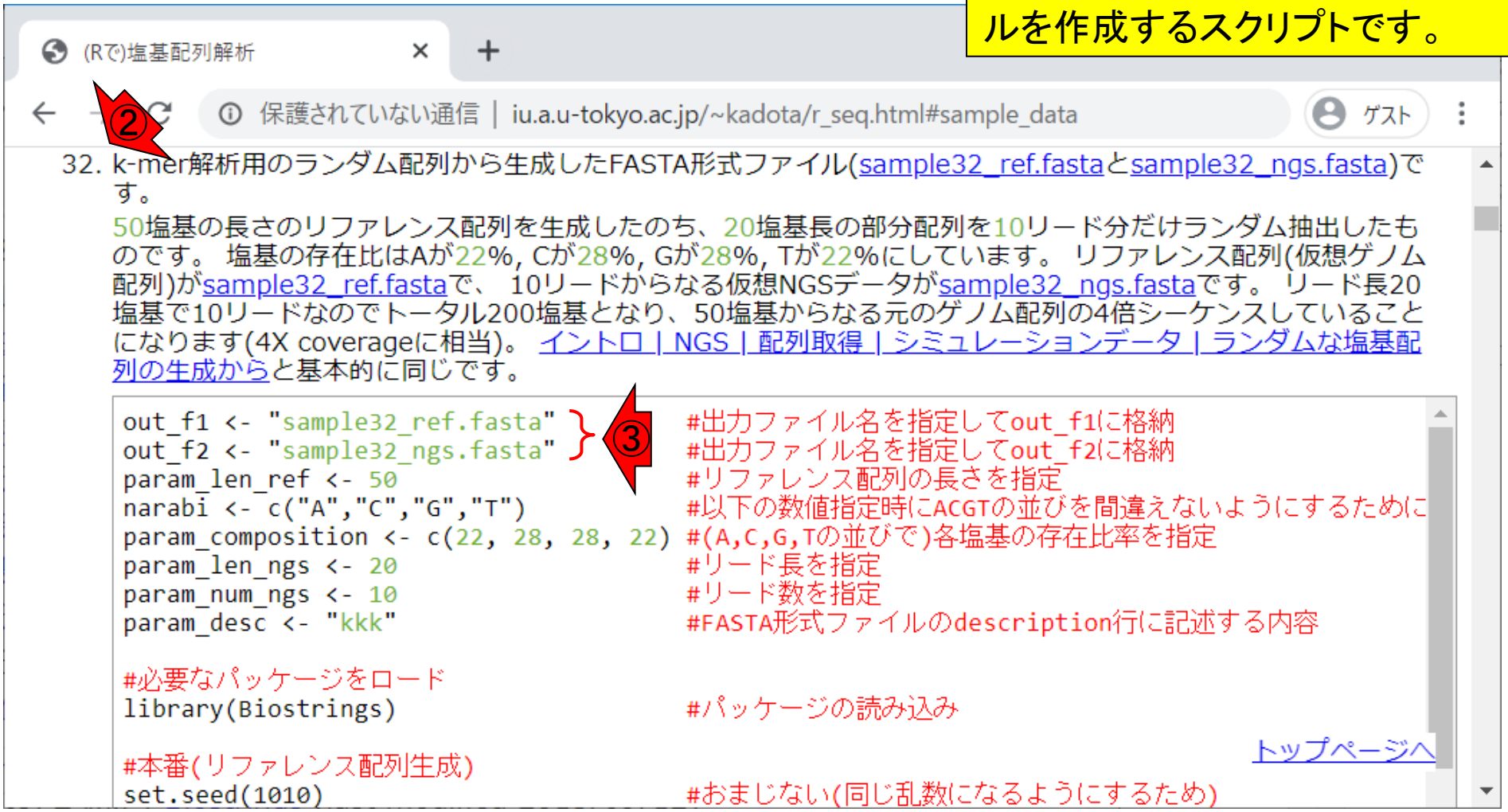
サンプルデータ（例題32）

ここは見るだけ



サンプルデータ(例題32)

①サンプルデータの、②例題32は、塩基配列解析の基本テクニックのみを用いて、③仮想ゲノムと仮想NGSリードのファイルを作成するスクリプトです。



32. k-mer解析用のランダム配列から生成したFASTA形式ファイル(sample32_ref.fastaとsample32_ngs.fasta)です。
50塩基の長さのリファレンス配列を生成したのち、20塩基長の部分配列を10リード分だけランダム抽出したものです。 塩基の存在比はAが22%, Cが28%, Gが28%, Tが22%にしています。 リファレンス配列(仮想ゲノム配列)がsample32_ref.fastaで、 10リードからなる仮想NGSデータがsample32_ngs.fastaです。 リード長20塩基で10リードなのでトータル200塩基となり、50塩基からなる元のゲノム配列の4倍シーケンスしていることになります(4X coverageに相当)。 イントロ | NGS | 配列取得 | シミュレーションデータ | ランダムな塩基配列の生成からと基本的に同じです。

```
out_f1 <- "sample32_ref.fasta"          #出力ファイル名を指定してout_f1に格納
out_f2 <- "sample32_ngs.fasta"          #出力ファイル名を指定してout_f2に格納
param_len_ref <- 50                     #リファレンス配列の長さを指定
narabi <- c("A","C","G","T")            #以下の数値指定時にACGTの並びを間違えないようにするために
param_composition <- c(22, 28, 28, 22) #(A,C,G,Tの並びで)各塩基の存在比率を指定
param_len_ngs <- 20                     #リード長を指定
param_num_ngs <- 10                     #リード数を指定
param_desc <- "kkk"                     #FASTA形式ファイルのdescription行に記述する内容

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

                                                        トップページへ
#本番(リファレンス配列生成)
set.seed(1010)                          #おまじない(同じ乱数になるようにするため)
```

サンプルデータ(例題32)

①サンプルデータの、②例題32は、塩基配列解析の基本テクニックのみを用いて、③仮想ゲノムと仮想NGSリードのファイルを作成するスクリプトです。④で示すA, C, G, Tの存在比率で、⑤50塩基の長さの、⑥仮想ゲノム(sample32_ref.fasta)を、まず⑦の部分で作成しています。

(Rで)塩基配列解析

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

32. k-mer解析用のランダム配列から生成したFASTA形式ファイル(sample3
す。
50塩基の長さのリファレンス配列を生成したのち、20塩基長の部分配列を10リード分だけランダム抽出したものです。 塩基の存在比はAが22%, Cが28%, Gが28%, Tが22%にしています。 リファレンス配列(仮想ゲノム配列)がsample32_ref.fastaで、 10リードからなる仮想NGSデータがsample32_ngs.fastaです。 リード長20塩基で10リードなのでトータル200塩基となり、50塩基からなる元のゲノム配列の4倍シーケンスしていることになります(4X coverageに相当)。 イントロ | NGS | 配列取得 | シミュレーションデータ | ランダムな塩基配列の生成からと基本的に同じです。

```
out_f1 <- "sample32_ref.fasta"          #出力ファイル名を指定してout_f1に格納
out_f2 <- "sample32_ngs.fasta"          #出力ファイル名を指定してout_f2に格納
param_len_ref <- 50                     #リファレンス配列の長さを指定
narabi <- c("A","C","G","T")            #以下の数値指定時にACGTの並びを間違えないようにするために
param_composition <- c(22, 28, 28, 22)  #(A,C,G,Tの並びで)各塩基の存在比率を指定
param_len_ngs <- 20                     #リード長を指定
param_num_ngs <- 10                     #リード数を指定
param_desc <- "kkk"                     #FASTA形式ファイルのdescription行に記述する内容

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

#本番(リファレンス配列生成)
set.seed(1010)                          #おまじない(同じ乱数になるようにするため)
```

トップページへ

サンプルデータ(例題32)

①サンプルデータの、②例題32は、塩基配列解析の基本テクニックのみを用いて、③仮想ゲノムと仮想NGSリードのファイルを作成するスクリプトです。④で示すA, C, G, Tの存在比率で、⑤50塩基の長さの、⑥仮想ゲノム(sample32_ref.fasta)を、まず⑦の部分で作成しています。⑥の中身が、⑧。

(Rで)塩基配列解析

⑧

```
>kkk
GAAGTTGTCTTTTGAACTCACTACACCAGGACATGAAGACGCGGACGGCA
```

す。

50塩基の長さのリファレンス配列を生成したのち、20塩基長の部分配列
のです。 塩基の存在比はAが22%, Cが28%, Gが28%, Tが22%にしています。 リファレンス配列(仮想ゲノム配列)がsample32_ref.fastaで、 10リードからなる仮想NGSデータがsample32_ngs.fastaです。 リード長20塩基で10リードなのでトータル200塩基となり、50塩基からなる元のゲノム配列の4倍シーケンスしていることになります(4X coverageに相当)。 イントロ | NGS | 配列取得 | シミュレーションデータ | ランダムな塩基配列の生成からと基本的に同じです。

⑤ ⑥ ④

```
out_f1 <- "sample32_ref.fasta"          #出力ファイル名を指定してout_f1に格納
out_f2 <- "sample32_ngs.fasta"          #出力ファイル名を指定してout_f2に格納
param_len_ref <- 50                     #リファレンス配列の長さを指定
narabi <- c("A","C","G","T")            #以下の数値指定時にACGTの並びを間違えないようにするために
param_composition <- c(22, 28, 28, 22)  #(A,C,G,Tの並びで)各塩基の存在比率を指定
param_len_ngs <- 20                     #リード長を指定
param_num_ngs <- 10                     #リード数を指定
param_desc <- "kkk"                     #FASTA形式ファイルのdescription行に記述する内容

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

#本番(リファレンス配列生成)
set.seed(1010)                          #おまじない(同じ乱数になるようにするため)
```

⑦

トップページへ

サンプルデータ(例題32)

①サンプルデータの、②例題32は、塩基配列解析の基本テクニックのみを用いて、③仮想ゲノムと仮想NGSリードのファイルを作成するスクリプトです。④で示すA, C, G, Tの存在比率で、⑤50塩基の長さの、⑥仮想ゲノム(sample32_ref.fasta)を、まず⑦の部分で作成しています。⑥の中身が、⑧。次に、仮想ゲノム配列をリファレンスとして、⑨20塩基長の部分配列(リード)を、⑩10個分ランダム抽出した、⑪仮想リード(sample32_ngs.fasta)。

(Rで)塩基配列解析

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

32. k-mer解析用のランダム配列から生成したFASTA形式ファイル(sample3
す。
50塩基の長さのリファレンス配列を生成したのち、20塩基長の部分配列
のです。 塩基の存在比はAが22%, Cが28%, Gが28%, Tが22%にして
配列)がsample32_ref.fastaで、 10リードからなる仮想NGSデータがsa
塩基で10リードなのでトータル200塩基となり、50塩基からなる元のゲ
になります(4X coverageに相当)。 イントロ | NGS | 配列取得 | シミュ
列の生成からと基本的に同じです。

```
out_f1 <- "sample32_ref.fasta"          #出力ファイル名を指定してout_f1に格納
out_f2 <- "sample32_ngs.fasta"          #出力ファイル名を指定してout_f2に格納
param_len_ref <- 50                     #リファレンス配列の長さを指定
narabi <- c("A","C","G","T")            #以下の数値指定時にACGTの並びを間違えないようにするために
param_composition <- c(22, 28, 28, 22)  #(A,C,G,Tの並びで)各塩基の存在比率を指定
param_len_ngs <- 20                     #リード長を指定
param_num_ngs <- 10                     #リード数を指定
param_desc <- "kkk"                     #FASTA形式ファイルのdescription行に記述する内容

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

                                                        トップページへ
#本番(リファレンス配列生成)
set.seed(1010)                          #おまじない(同じ乱数になるようにするため)
```

サンプルデータ(例題32)

(Rで)塩基配列解析

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

32. k-mer解析用のランダム配列から生成したFASTA形式ファイル(sample3
す。
50塩基の長さのリファレンス配列を生成したのち、20塩基長の部分配列
のです。 塩基の存在比はAが22%, Cが28%, Gが28%, Tが22%にして
配列)がsample32_ref.fastaで、 10リードからなる仮想NGSデータがsa
塩基で10リードなのでトータル200塩基となり、50塩基からなる元のゲ
になります(4X coverageに相当)。 イントロ | NGS | 配列取得 | シミュ
列の生成からと基本的に同じです。

```
out_f1 <- "sample32_ref.fasta"          #出力ファイル名を指定
out_f2 <- "sample32_ngs.fasta"          #出力ファイル名を指定
param_len_ref <- 50                     #リファレンス配列の長
narabi <- c("A","C","G","T")            #以下の数値指定時にAC
param_composition <- c(22, 28, 28, 22) #(A,C,G,Tの並びで)各
param_len_ngs <- 20                     #リード長を指定
param_num_ngs <- 10                     #リード数を指定
param_desc <- "kkk"                     #FASTA形式ファイルのd

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

#本番(リファレンス配列生成)
set.seed(1010)                          #おまじない(同じ乱数になるようにするため)
```

トップページへ

⑫

①サンプルデータの、②例題32は、塩基配列解析の基本テクニックのみを用いて、③仮想ゲノムと仮想NGSリードのファイルを作成するスクリプトです。④で示すA, C, G, Tの存在比率で、⑤50塩基の長さの、⑥仮想ゲノム(sample32_ref.fasta)を、まず⑦の部分で作成しています。⑥の中身が、⑧。次に、仮想ゲノム配列をリファレンスとして、⑨20塩基長の部分配列(リード)を、⑩10個分ランダム抽出した、⑪仮想リード(sample32_ngs.fasta)。⑫毎回同じ乱数が生成されるようにしているので、おそらく同じ結果が得られる。ただし、R本体のバージョンアップ後に再度実行すると結果が異なります。おそらくですが、R本体のバージョンが同じ場合のみ有効なのだろうと思います。

サンプルデータ(例題32)

⑨20塩基長の部分配列(リード)を、⑩10個分ランダム抽出した、⑪仮想リード(sample32_ngs.fasta)の中身。

サンプルデータ(例題32)

①仮想リード中の、②description部分を見れば、③そのリードが、④仮想ゲノム配列中のどの領域由来かがわかる。

(Rで)塩基配列解析

```
>kkk
GAAGTTGTCTTTTGAACTCACTACACCAGGACATGAAGACGCGGACGGCA
```

ample_data

ゲスト

32_ref.fastaとsample32_ngs.fasta)で
す。
50塩基の長さのリファレンス配列を生成し 0リード分だけランダム抽出したも
のです。 塩基の存在比はAが22%, Cが28% す。 リファレンス配列(仮想ゲノム
配列)がsample32_ref.fastaで、 10リード 32_ngs.fastaです。 リード長20
塩基で10リードなのでトータル200塩基と 列の4倍シーケンスしていること
になります(4X coverageに相当)。 イントロ ーションデータ | ランダムな塩基配
列の生成からと基本的に同じです。

```
out_f1 <- "sample32_ref.fasta"
out_f2 <- "sample32_ngs.fasta"
param_len_ref <- 50
narabi <- c("A","C","G","T")
param_composition <- c(22, 28, 28,
param_len_ngs <- 20
param_num_ngs <- 10
param_desc <- "kkk"

#必要なパッケージをロード
library(Biostrings)

#本番(リファレンス配列生成)
set.seed(1010)
```

out_f1に格納
out_f2に格納
指定
並びを間違えないようにするために
の存在比率を指定

ription行に記述する内容

トップページへ

るようにするため)

```
>kkk_24_48
CACCAGGACA AAGACGCG
>kkk_28_47
AGGACATGAAGACGCGGACG
>kkk_12_31
TTGAACTCACTACACCAGGA
>kkk_4_23
GTTGTCTTTTGAACTCACTA
>kkk_5_24
TTGTCTTTTGAACTCACTAC
>kkk_7_26
GTCTTTTGAACTCACTACAC
>kkk_12_31
TTGAACTCACTACACCAGGA
>kkk_23_42
ACACCAGGACATGAAGACGC
>kkk_1_20
GAAGTTGTCTTTTGAACTCA
>kkk_3_22
AGTTGTCTTTTGAACTCACT
```

サンプルデータ(例題32)

①仮想リード中の、②description部分を見れば、③そのリードが、④仮想ゲノム配列中のどの領域由来かがわかる。他の例。

(Rで)塩基配列解析

```
>kkk
GAAGTTGTCTTTTGAACTCACTACACCAGGACATGAAGACGCGGACGGCA
```

ample_data　ゲスト

32_ref.fastaとsample32_ngs.fasta)で
す。
50塩基の長さのリファレンス配列を生成し
のです。 塩基の存在比はAが22%, Cが28%
配列)がsample32_ref.fastaで、 10リード
塩基で10リードなのでトータル200塩基と
になります(4X coverageに相当)。 イントロ
列の生成からと基本的に同じです。

0リード分だけランダム抽出したも
す。 リファレンス配列(仮想ゲノム
le32_ngs.fastaです。 リード長20
配列の4倍シーケンスしていること
ーションデータ | ランダムな塩基配

```
out_f1 <- "sample32_ref.fasta"
out_f2 <- "sample32_ngs.fasta"
param_len_ref <- 50
narabi <- c("A","C","G","T")
param_composition <- c(22, 28, 28,
param_len_ngs <- 20
param_num_ngs <- 10
param_desc <- "kkk"

#必要なパッケージをロード
library(Biostrings)

#本番(リファレンス配列生成)
set.seed(1010)
```

out_f1に格納
out_f2に格納
指定
並びを間違えないようにするために
の存在比率を指定

ription行に記述する内容

トップページへ

るようにするため)

```
>kkk_24_43
CACCAGGACATGAAGACGCG
>kkk_28_47
AGGACATGAAGACGCGGACG
>kkk_12_31
TTGAACTCACTACACCAGGA
>kkk_4_23
GTTGTCTTTTGAACTCACTA
>kkk_5_24
TTGTCTTTTGAACTCACTAC
>kkk_7_26
GTCTTTTGAACTCACTACAC
>kkk_12_31
TTGAACTCACTACACCAGGA
>kkk_23_42
ACACCAGGACATGAAGACGC
>kkk_1_20
GAAGTTGTCTTTTGAACTCA
>kkk_3_22
AGTTGTCTTTTGAACTCACT
```

④ ① ② ③

Contents

- Introduction、出現頻度解析（k=2）、出現頻度解析（k=1）
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図（例題10）、作図（例題11）、作図（例題12）
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ（例題32）、被覆率（coverage）、基本的な考え方(例題7)
  - 例題8（k=2)、例題9（k=3）、1,000塩基の仮想ゲノム（サンプルデータの例題33）
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

被覆率（coverage）1

おさらい。①仮想ゲノムの配列長は、②50塩基。③仮想リードは、④10個で、⑤リード長は20塩基。従って、リード側の総塩基数は10×20＝200塩基。この場合、仮想ゲノムの200／50＝4倍分塩基配列決定（sequencing）がなされているので、被覆率（coverage）は4X coverageみたいな表現をします。

(Rで)塩基配列解析

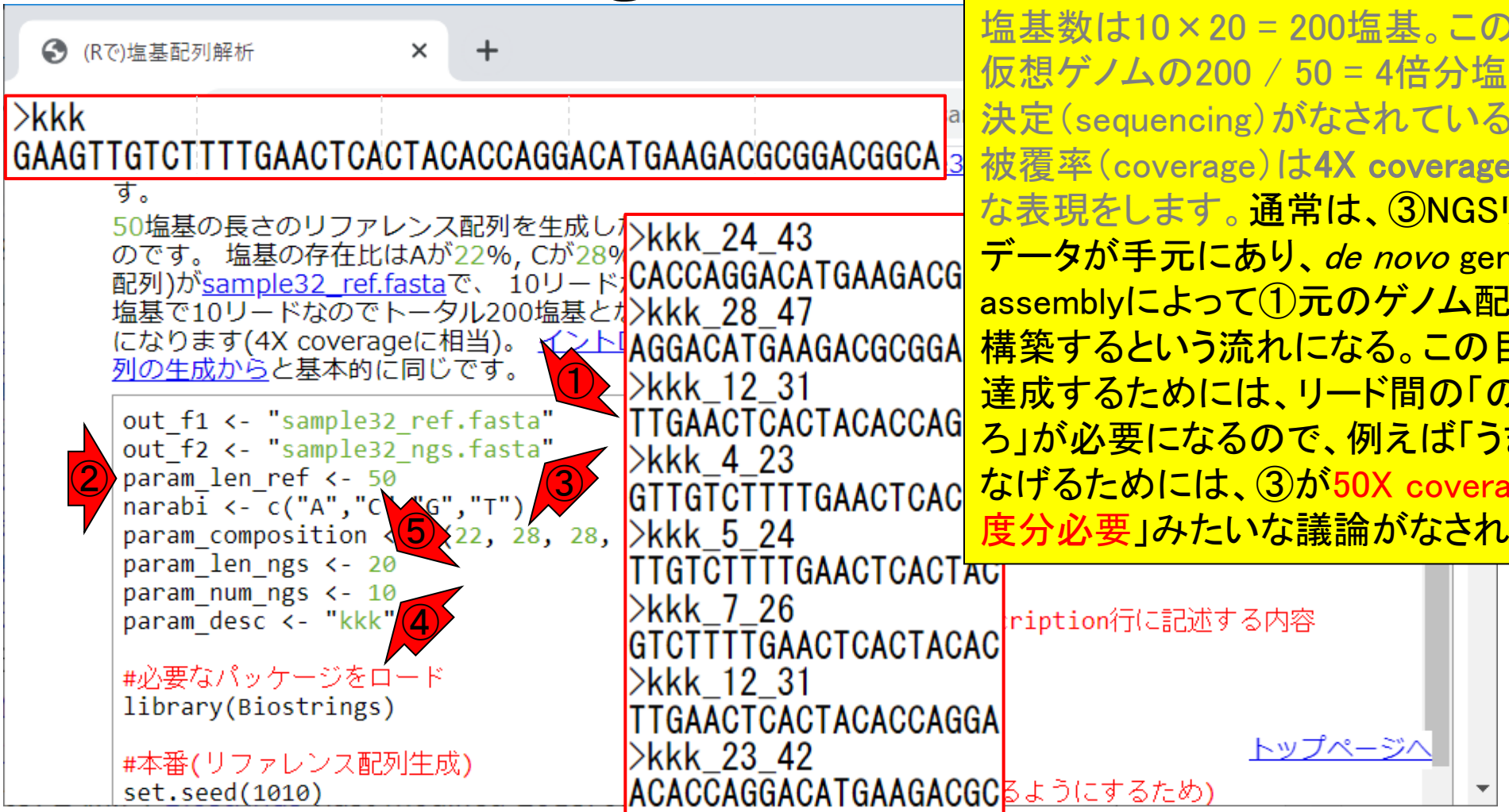
```
>kkk
GAAGTTGTCTTTTGAACTCACTACACCAGGACATGAAGACGCGGACGGCA
```

す。
50塩基の長さのリファレンス配列を生成し
のです。 塩基の存在比はAが22%, Cが28%
配列)がsample32_ref.fastaで、 10リード
塩基で10リードなのでトータル200塩基と
になります(4X coverageに相当)。 イントロ
列の生成からと基本的に同じです。

```
out_f1 <- "sample32_ref.fasta"
out_f2 <- "sample32_ngs.fasta"
param_len_ref <- 50
narabi <- c("A","C","G","T")
param_composition <- c(22, 28, 28,
param_len_ngs <- 20
param_num_ngs <- 10
param_desc <- "kkk"

#必要なパッケージをロード
library(Biostrings)

#本番(リファレンス配列生成)
set.seed(1010)
```

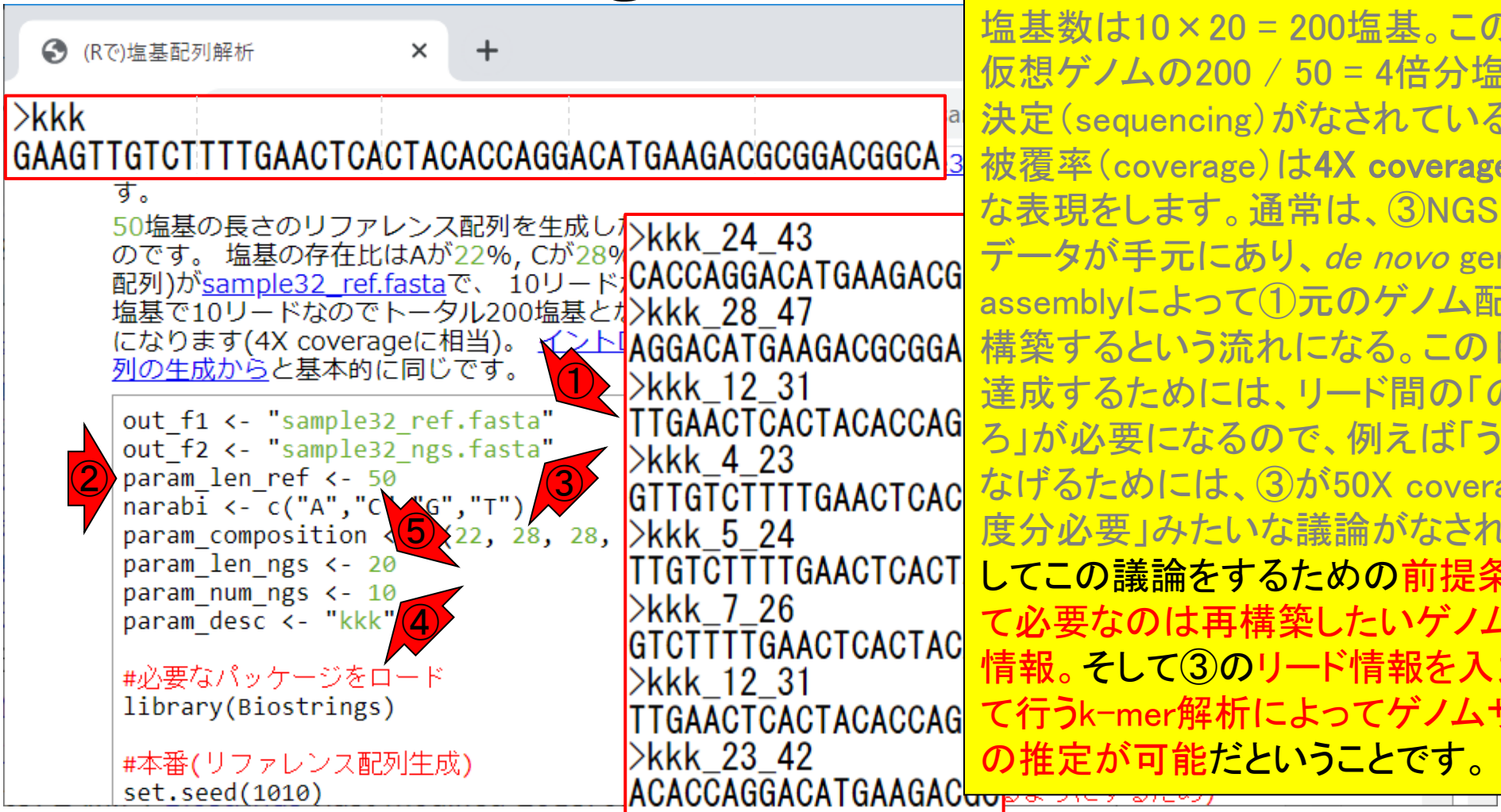
```
>kkk_24_43
CACCAGGACATGAAGACGCG
>kkk_28_47
AGGACATGAAGACGCGGACG
>kkk_12_31
TTGAACTCACTACACCAGGA
>kkk_4_23
GTTGTCTTTTGAACTCACTA
>kkk_5_24
TTGTCTTTTGAACTCACTAC
>kkk_7_26
GTCTTTTGAACTCACTACAC
>kkk_12_31
TTGAACTCACTACACCAGGA
>kkk_23_42
ACACCAGGACATGAAGACGC
>kkk_1_20
GAAGTTGTCTTTTGAACTCA
>kkk_3_22
AGTTGTCTTTTGAACTCACT
```

被覆率（coverage）2



おさらい。①仮想ゲノムの配列長は、②50塩基。③仮想リードは、④10個で、⑤リード長は20塩基。従って、リード側の総塩基数は10×20 = 200塩基。この場合、仮想ゲノムの200 / 50 = 4倍分塩基配列決定（sequencing）がなされているので、被覆率（coverage）は**4X coverage**みたいな表現をします。通常は、③NGSリードデータが手元にあり、*de novo* genome assemblyによって①元のゲノム配列を再構築するという流れになる。この目的を達成するためには、リード間の「のりしろ」が必要になるので、例えば「うまくつなげるためには、③が50X coverage程度分必要」みたいな議論がなされる。

被覆率（coverage）3



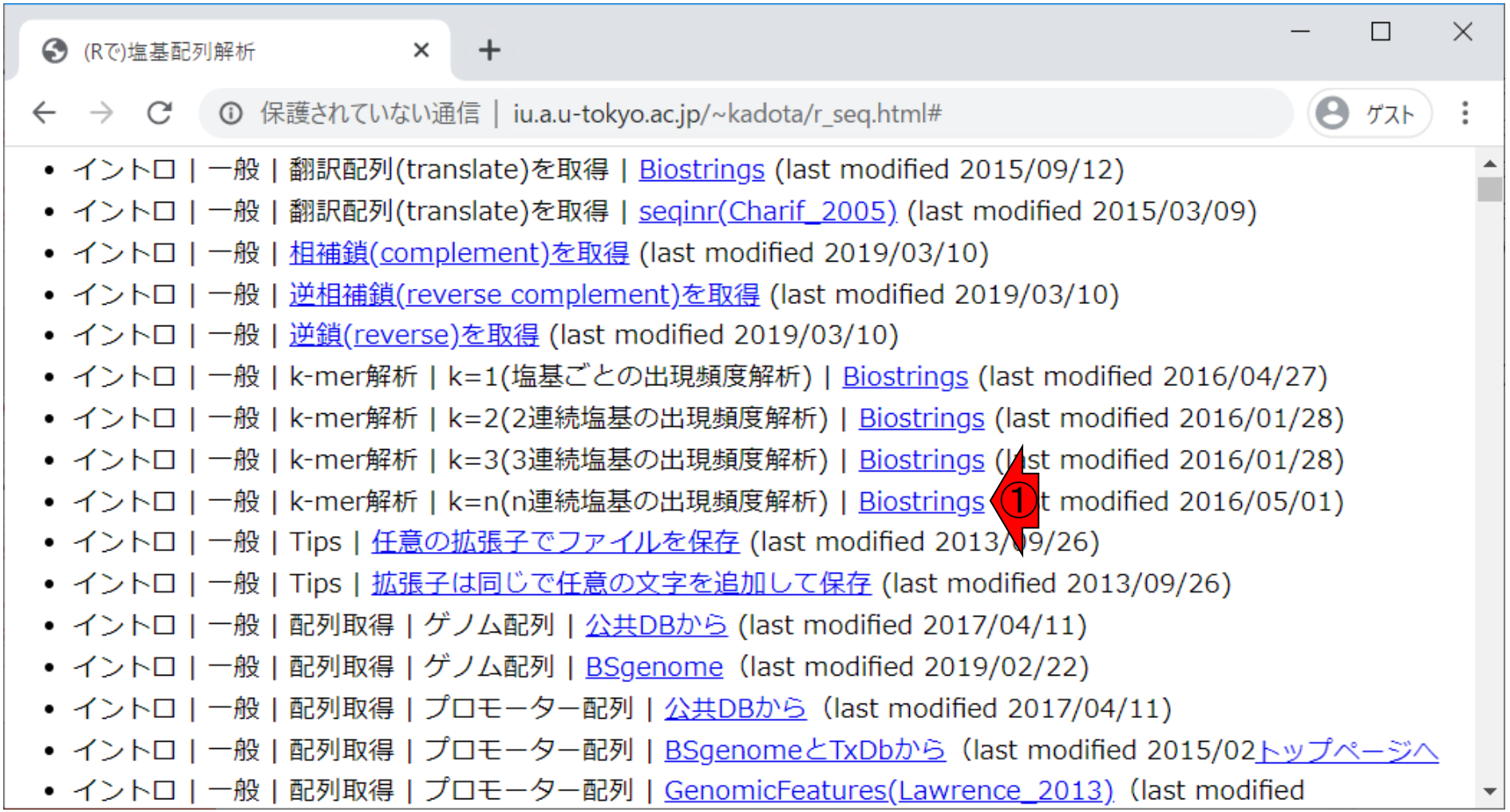
おさらい。①仮想ゲノムの配列長は、②50塩基。③仮想リードは、④10個で、⑤リード長は20塩基。従って、リード側の総塩基数は10×20 = 200塩基。この場合、仮想ゲノムの200 / 50 = 4倍分塩基配列決定（sequencing）がなされているので、被覆率（coverage）は**4X coverage**みたいな表現をします。通常は、③NGSリードデータが手元にあり、*de novo* genome assemblyによって①元のゲノム配列を再構築するという流れになる。この目的を達成するためには、リード間の「のりしろ」が必要になるので、例えば「うまくつなげるためには、③が50X coverage程度分必要」みたいな議論がなされる。そしてこの議論をするための前提条件として必要なのは再構築したいゲノムサイズ情報。そして③のリード情報を入力として行うk-mer解析によってゲノムサイズの推定が可能だということです。

Contents

- Introduction、出現頻度解析（k=2）、出現頻度解析（k=1）
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図（例題10）、作図（例題11）、作図（例題12）
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ（例題32）、被覆率（coverage）、基本的な考え方(例題7)
  - 例題8（k=2)、例題9（k=3）、1,000塩基の仮想ゲノム（サンプルデータの例題33）
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

基本的な考え方(例題7)1

①「n連続塩基」のほうです。項目名に注意。



基本的な考え方(例題7)2

①「n連続塩基」のほうです。項目名に注意。②例題7。

(Rで)塩基配列解析

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

ゲスト

②

7. サンプルデータの例題32を実行して得られたmulti-FASTAファイル(sample32_ngs.fasta)の場合:

2連続塩基（k=2）の出現頻度情報を得るやり方です。4^2 = 16通りのk-merの出現頻度を計算することになります。 リード毎に出現頻度を算出しています。

```
in_f <- "sample32_ngs.fasta"            #入力ファイル名を指定してin_fに格納
out_f <- "hoge7.txt"                    #出力ファイル名を指定してout_fに格納
param_kmer <- 2                         #k-merのkの値を指定

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

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

#本番
out <- oligonucleotideFrequency(fasta, width=param_kmer)#k連続塩基の出現頻度情報をoutに格納

#ファイルに保存
tmp <- cbind(names(fasta), out)         #最初の列にID情報、そのあとに出現頻度情報のoutを結合したtmpを作
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定したファイル名で保
```

トップページへ

基本的な考え方(例題7)3

①「n連続塩基」のほうです。項目名に注意。②例題7。③入力ファイルの中身。これの元のゲノム配列は50塩基であり、③の総塩基数は10リード×20塩基 = 200塩基。ゲノムサイズの4倍の被覆率(4X coverage)であることが分かっている。

(Rで)塩基配列解析

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

②

7. サンプルデータの例題32を実行して得られたmulti-FASTAファイル(sample32_ngs.fasta)の場合:

2連続塩基（k=2）の出現頻度情報を得るやり方[illegible]の出現頻度を計算することになります。 リード毎に出現頻度を算出しています。

③

```
in_f <- "sample32_ngs.fasta"
out_f <- "hoge7.txt"
param_kmer <- 2

#必要なパッケージをロード
library(Biostrings)

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

#本番
out <- oligonucleotideFrequency(fasta,

#ファイルに保存
tmp <- cbind(names(fasta), out)
write.table(tmp, out_f, sep="\t", apper
```

fに格納
fに格納
イルの読み込み
の出現頻度情報をoutに格納
出現頻度情報のoutを結合したtmpを作
#tmpの中身を指定したファイル名で保
トップページへ

```
>kkk_24_43
CACCAGGACATGAAGACGCG
>kkk_28_47
AGGACATGAAGACGCGGACG
>kkk_12_31
TTGAACTCACTACACCAGGA
>kkk_4_23
GTTGTCTTTTGAACTCACTA
>kkk_5_24
TTGTCTTTTGAACTCACTAC
>kkk_7_26
GTCTTTTGAACTCACTACAC
>kkk_12_31
TTGAACTCACTACACCAGGA
>kkk_23_42
ACACCAGGACATGAAGACGC
>kkk_1_20
GAAGTTGTCTTTTGAACTCA
>kkk_3_22
AGTTGTCTTTTGAACTCACT
```

基本的な考え方(例題7)4

①「n連続塩基」のほうです。項目名に注意。②例題7。③入力ファイルの中身。これの元のゲノム配列は50塩基であり、③の総塩基数は10リード×20塩基 = 200塩基。ゲノムサイズの4倍の被覆率（4X coverage）であることが分かっている。④で与えているのはk-mer解析のkの値。③入力ファイルをダウンロードして、コピペ実行。

(Rで)塩基配列解析

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

②

7. サンプルデータの例題32を実行して得られたmulti-FASTAファイル(samp

2連続塩基（k=2）の出現頻度情報を得るやり方です。4^2 = 16通りのk-me
す。 リード毎に出現頻度を算出しています。

③

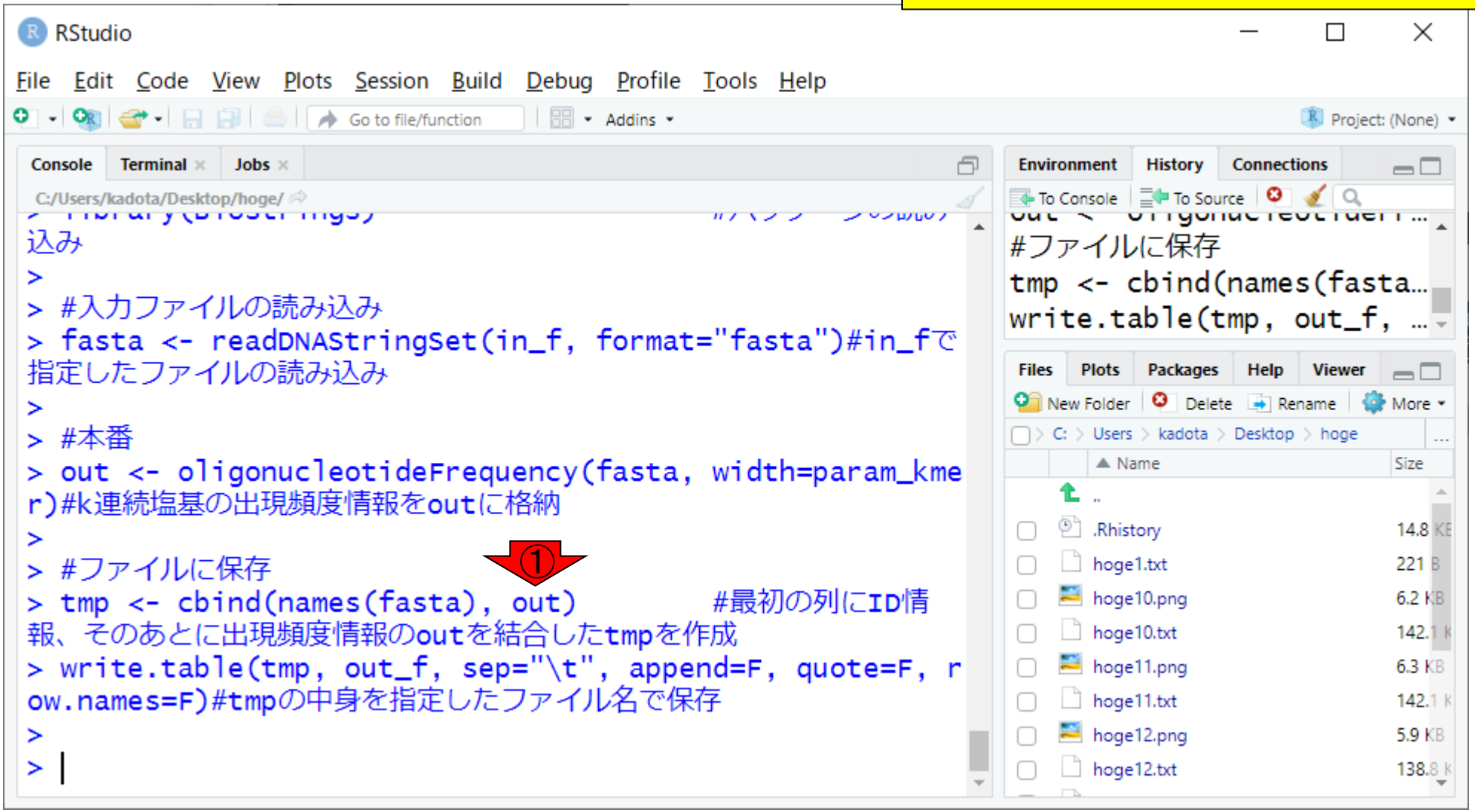
④

```
in_f <- "sample32_ngs.fasta"          #入力ファイル名を指定してin_fに格納
out_f <- "hoge7.txt"                  #出力ファイル名を指定してout_fに格納
param_kmer <- 2                       #k-merのkの値を指定

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

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

#本番
out <- oligonucleotideFrequency(fasta, width=param_kmer)#k連続塩基の出現頻度情報をoutに格納

#ファイルに保存
tmp <- cbind(names(fasta), out)       #最初の列にID情報、そのあとに出現頻度情報のoutを結合したtmpを作
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定したファイル名で保
```

トップページへ

基本的な考え方(例題7)5

実行結果。出力ファイルの中身は、実質的に①outオブジェクトの中身と同じなので…



基本的な考え方(例題7)6

実行結果。出力ファイルの中身は、実質的に①outオブジェクトの中身と同じなので、②outと打ち込んで表示。

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Project: (None)

Console | Terminal | Jobs

C:/Users/kadota/Desktop/hoge/

```
報、そのあとに出現頻度情報のoutを結合したtmpを作成
> write.table(tmp, out_f, sep="\t", append=F, quote=F, r
ow.names=F)#tmpの中身を指定したファイル名で保存
>
> out
      AA AC AG AT CA CC CG CT GA GC GG GT TA TC TG TT
 [1,]  1  3  2  1  3  1  2  0  3  1  1  0  0  0  1  0
 [2,]  1  3  2  1  1  0  3  0  4  1  2  0  0  0  1  0
 [3,]  1  4  1  0  3  1  0  2  2  0  1  0  1  1  1  1
 [4,]  1  2  0  0  1  0  0  3  1  0  0  2  1  2  2  4
 [5,]  1  3  0  0  1  0  0  3  1  0  0  1  1  2  2  4
 [6,]  1  4  0  0  2  0  0  3  1  0  0  1  1  2  1  3
 [7,]  1  4  1  0  3  1  0  2  2  0  1  0  1  1  1  1
 [8,]  1  4  2  1  3  1  1  0  3  1  1  0  0  0  1  0
 [9,]  2  1  1  0  1  0  0  2  2  0  0  2  0  2  2  4
[10,]  1  2  1  0  1  0  0  3  1  0  0  2  0  2  2  4
> |
```

②

Environment | History | Connections

To Console | To Source

```
tmp <- cbind(names(fasta…
write.table(tmp, out_f, …
out
```

Files | Plots | Packages | Help | Viewer

New Folder | Delete | Rename | More

C: > Users > kadota > Desktop > hoge

Name	Size
..	
.Rhistory	14.8 KB
hoge1.txt	221 B
hoge10.png	6.2 KB
hoge10.txt	142.1 K
hoge11.png	6.3 KB
hoge11.txt	142.1 K
hoge12.png	5.9 KB
hoge12.txt	138.8 K

基本的な考え方(例題7)7

実行結果。出力ファイルの中身は、実質的に①outオブジェクトの中身と同じなので、②outと打ち込んで表示。③全部で10行分ある理由は、入力ファイルが10リードだから。

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Project: (None)

Console | Terminal | Jobs

C:/Users/kadota/Desktop/hoge/

```
報、そのあとに出現頻度情報のoutを結合したtmpを作成
> write.table(tmp, out_f, sep="\t", append=F, quote=F, r
ow.names=F)#tmpの中身を指定したファイル名で保存
>
> out
      AA AC AG AT CA CC CG CT GA GC GG GT TA TC TG TT
 [1,]  1  3  2  1  3  1  2  0  3  1  1  0  0  0  1  0
 [2,]  1  3  2  1  1  0  3  0  4  1  2  0  0  0  1  0
 [3,]  1  4  1  0  3  1  0  2  2  0  1  0  1  1  1  1
 [4,]  1  2  0  0  1  0  0  3  1  0  0  2  1  2  2  4
 [5,]  1  3  0  0  1  0  0  3  1  0  0  1  1  2  2  4
 [6,]  1  4  0  0  2  0  0  3  1  0  0  1  1  2  1  3
 [7,]  1  4  1  0  3  1  0  2  2  0  1  0  1  1  1  1
 [8,]  1  4  2  1  3  1  1  0  3  1  1  0  0  0  1  0
 [9,]  2  1  1  0  1  0  0  2  2  0  0  2  0  2  2  4
[10,]  1  2  1  0  1  0  0  3  1  0  0  2  0  2  2  4
> |
```

②

③

Environment | History | Connections

More

Size

14.8 KB
221 B
6.2 KB
142.1 KB
6.3 KB
142.1 KB
5.9 KB
138.8 KB

```
>kkk_24_43
CACCAGGACATGAAGACGCG
>kkk_28_47
AGGACATGAAGACGCGGACG
>kkk_12_31
TTGAACTCACTACACCAGGA
>kkk_4_23
GTTGTCTTTTGAACTCACTA
>kkk_5_24
TTGTCTTTTGAACTCACTAC
>kkk_7_26
GTCTTTTGAACTCACTACAC
>kkk_12_31
TTGAACTCACTACACCAGGA
>kkk_23_42
ACACCAGGACATGAAGACGC
>kkk_1_20
GAAGTTGTCTTTTGAACTCA
>kkk_3_22
AGTTGTCTTTTGAACTCACT
```

基本的な考え方(例題7)8

実行結果。出力ファイルの中身は、実質的に①outオブジェクトの中身と同じなので、②outと打ち込んで表示。③全部で10行分ある理由は、入力ファイルが10リードだから。例えば、④3番目のリードに対して行われた2連続塩基の出現頻度解析結果が、⑤3行目の結果に相当。

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

Console | Terminal | Jobs

C:/Users/kadota/Desktop/hoge/

```
報、そのあとに出現頻度情報のoutを結合したtmpを作成
> write.table(tmp, out_f, sep="\t", append=F, quote=F, r
ow.names=F)#tmpの中身を指定したファイル名で保存
>
> out
      AA AC AG AT CA CC CG CT GA GC GG GT TA TC TG TT
 [1,]  1  3  2  1  3  1  2  0  3  1  1  0  0  0  1  0
 [2,]  1  3  2  1  1  0  3  0  4  1  2  0  0  0  1  0
 [3,]  1  4  1  0  3  1  0  2  2  0  1  0  1  1  1  1
 [4,]  1  2  0  0  1  0  0  3  1  0  0  2  1  2  2  4
 [5,]  1  3  0  0  1  0  0  3  1  0  0  1  1  2  2  4
 [6,]  1  4  0  0  2  0  0  3  1  0  0  1  1  2  1  3
 [7,]  1  4  1  0  3  1  0  2  2  0  1  0  1  1  1  1
 [8,]  1  4  2  1  3  1  1  0  3  1  1  0  0  0  1  0
 [9,]  2  1  1  0  1  0  0  2  2  0  0  2  0  2  2  4
[10,]  1  2  1  0  1  0  0  3  1  0  0  2  0  2  2  4
> |
```

②

③

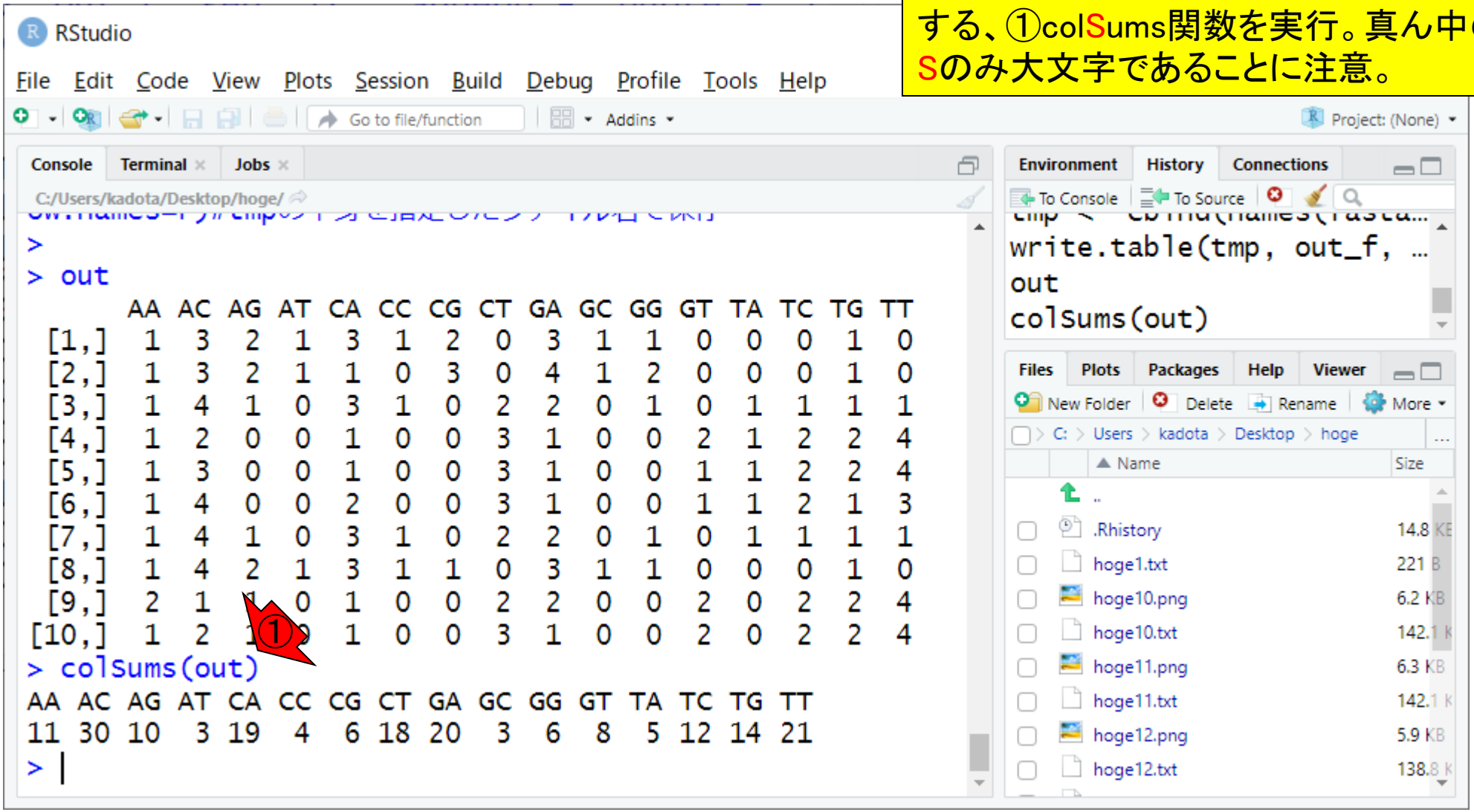
⑤

```
>kkk_24_43
CACCAGGACATGAAGACGCG
>kkk_28_47
AGGACATGAAGACGCGGACG
>kkk_12_31
TTGAACTCACTACACCAGGA
>kkk_4_23
GTTGTCTTTTGAACTCACTA
>kkk_5_24
TTGTCTTTTGAACTCACTAC
>kkk_7_26
GTCTTTTGAACTCACTACAC
>kkk_12_31
TTGAACTCACTACACCAGGA
>kkk_23_42
ACACCAGGACATGAAGACGC
>kkk_1_20
GAAGTTGTCTTTTGAACTCA
>kkk_3_22
AGTTGTCTTTTGAACTCACT
```

④

基本的な考え方(例題7)9

ゲノムサイズ推定の場合は、リードごとではなく入力ファイル全体で考える。列(columns)ごとの総和(summation)を計算する、①colSums関数を実行。真ん中のSのみ大文字であることに注意。



基本的な考え方(例題7)10

ゲノムサイズ推定の場合は、リードごとではなく入力ファイル全体で考える。列(columns)ごとの総和(summation)を計算する、①colSums関数を実行。真ん中のSのみ大文字であることに注意。②例えばTTという2連続塩基は、③全部で21回出現したと解釈する。

基本的な考え方(例題7)11

ゲノムサイズ推定の場合は、リードごとではなく入力ファイル全体で考える。列(columns)ごとの総和(summation)を計算する、①colSums関数を実行。真ん中のSのみ大文字であることに注意。②例えばTTという2連続塩基は、③全部で21回出現したと解釈する。今は例題7の結果に対して、①colSumsを追加で実行した。これを最初からやっているのが例題8。

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

Console | Terminal | Jobs

C:/Users/kadota/Desktop/hoge/

```
> 
> out
      AA AC AG AT CA CC CG CT GA GC GG GT TA TC TG TT
 [1,]  1  3  2  1  3  1  2  0  3  1  1  0  0  0  1  0
 [2,]  1  3  2  1  1  0  3  0  4  1  2  0  0  0  1  0
 [3,]  1  4  1  0  3  1  0  2  2  0  1  0  1  1  1  1
 [4,]  1  2  0  0  1  0  0  3  1  0  0  2  1  2  2  4
 [5,]  1  3  0  0  1  0  0  3  1  0  0  1  1  2  2  4
 [6,]  1  4  0  0  2  0  0  3  1  0  0  1  1  2  1  3
 [7,]  1  4  1  0  3  1  0  2  2  0  1  0  1  1  1  1
 [8,]  1  4  2  1  3  1  1  0  3  1  1  0  0  0  1  0
 [9,]  2  1  1  0  1  0  0  2  2  0  0  2  0  2  2  4
[10,]  1  2  1  0  1  0  0  3  1  0  0  2  0  2  2  4
> colSums(out)
AA AC AG AT CA CC CG CT GA GC GG GT TA TC TG TT
11 30 10  3 19  4  6 18 20  3  6  8  5 12 14 21
> 
```

①

```
out
colSums(out)
```

Files | Plots | Packages | Help | Viewer

New Folder | Delete | Rename | More

C: > Users > kadota > Desktop > hoge

Name	Size
..	
.Rhistory	14.8 KB
hoge1.txt	221 B
hoge10.png	6.2 KB
hoge10.txt	142.1 KB
hoge11.png	6.3 KB
hoge11.txt	142.1 KB
hoge12.png	5.9 KB
hoge12.txt	138.8 KB

Contents

- Introduction、出現頻度解析（k=2）、出現頻度解析（k=1）
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図（例題10）、作図（例題11）、作図（例題12）
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ（例題32）、被覆率（coverage）、基本的な考え方(例題7)
  - 例題8（k=2)、例題9（k=3）、1,000塩基の仮想ゲノム（サンプルデータの例題33）
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

例題8（k=2）1

ゲノムサイズ推定の場合は、リードごとではなく入力ファイル全体で考える。列(columns)ごとの総和(summation)を計算する、①colSums関数を実行。真ん中のSのみ大文字であることに注意。②例えばTTという2連続塩基は、③全部で21回出現したと解釈する。今は例題7の結果に対して、①colSumsを追加で実行した。これを最初から行っているのが、④例題8。⑤colSums関数はここで使っています。コピペ実行。

(Rで)塩基配列解析

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

④

8. サンプルデータの例題32を実行して得られたmulti-FASTAファイル(samp

2連続塩基（k=2）の出現頻度情報を得るやり方です。4^2 = 16通りのk-me
す。 全リードを合算した出現頻度を出力するやり方です。

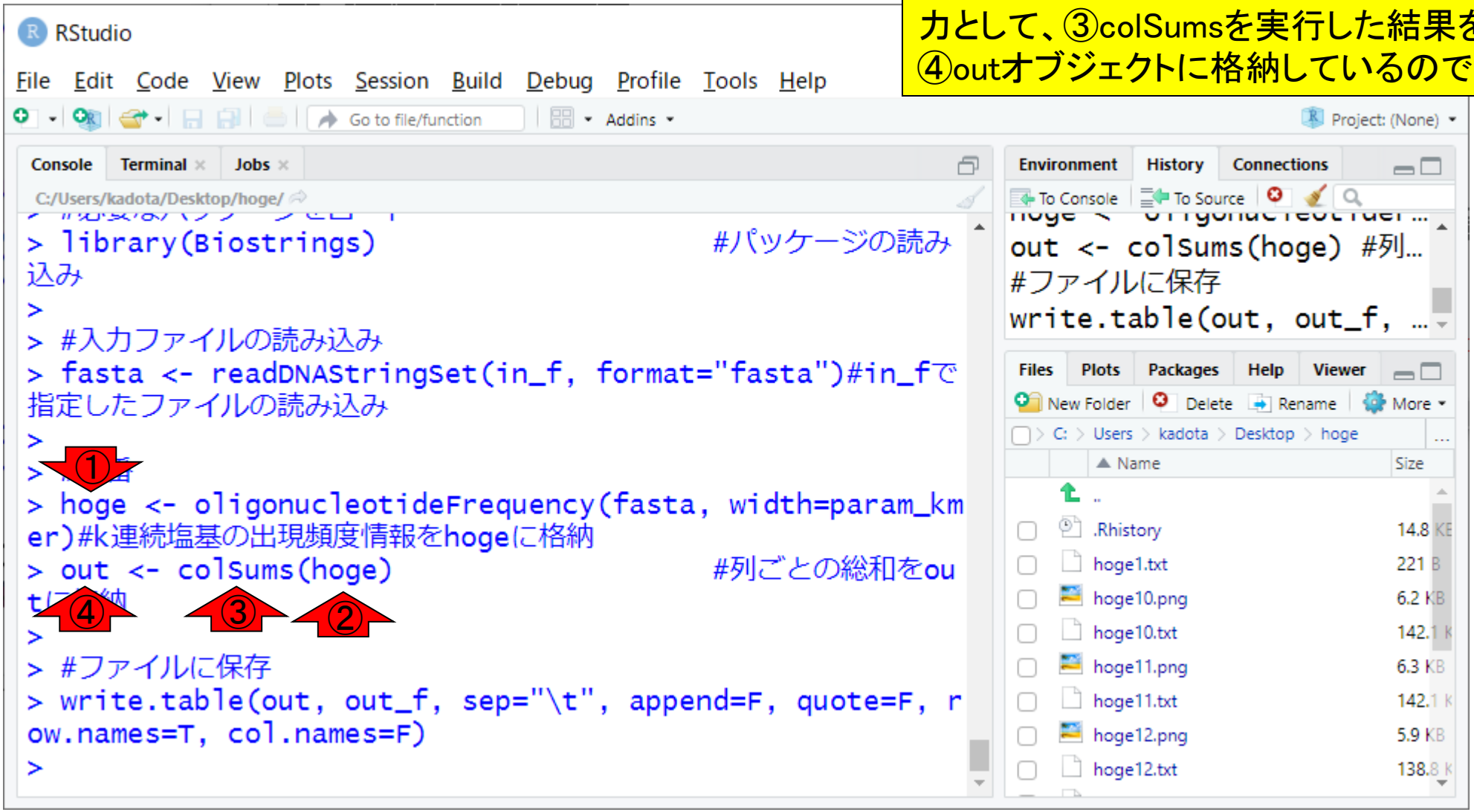
```
in_f <- "sample32_ngs.fasta"              #入力ファイル名を指定して
out_f <- "hoge8.txt"                      #出力ファイル名を指定して
param_kmer <- 2                           #k-merのkの値を指定

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

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

#本番
hoge <- oligonucleotideFrequency(fasta, width=param_kmer)#k連続塩基の出現頻度情報をhogeに格納
out <- colSums(hoge)                      #列ごとの総和をoutに格納

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

⑤

トップページへ

例題8（k=2）2

コピペ実行結果。例題7と異なり、例題8では①2連続塩基の出現頻度解析結果をhogeオブジェクトに格納し、②それを入力として、③colSumsを実行した結果を、④outオブジェクトに格納しているので…



例題8（k=2）3

コピペ実行結果。例題7と異なり、例題8では①2連続塩基の出現頻度解析結果をhogeオブジェクトに格納し、②それを入力として、③colSumsを実行した結果を、④outオブジェクトに格納しているので、⑤outの中身はこんな感じになります。

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

Console | Terminal | Jobs

C:/Users/kadota/Desktop/hoge/

```
> #入力ファイルの読み込み
> fasta <- readDNAStringSet(in_f, format="fasta")#in_fで指定したファイルの読み込み
>
> #①省略
> hoge <- oligonucleotideFrequency(fasta, width=param_kmer)#k連続塩基の出現頻度情報をhogeに格納
> out <- colSums(hoge)                    #列ごとの総和をoutに格納
>
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F, quote=F, row.names=T, col.names=F)
> out
AA AC AG AT CA CC CG CT GA GC GG GT TA TC TG TT
11 30 10  3 19  4  6 18 20  3  6  8  5 12 14 21
>
```

④ ③ ② ⑤

Environment | History | Connections

To Console | To Source

```
#ファイルに保存
write.table(out, out_f, ...
out
```

Files | Plots | Packages | Help | Viewer

New Folder | Delete | Rename | More

C: > Users > kadota > Desktop > hoge

Name	Size
..	
.Rhistory	14.8 KB
hoge1.txt	221 B
hoge10.png	6.2 KB
hoge10.txt	142.1 KB
hoge11.png	6.3 KB
hoge11.txt	142.1 KB
hoge12.png	5.9 KB
hoge12.txt	138.8 KB

例題8（k=2）4

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

Console | Terminal | Jobs

C:/Users/kadota/Desktop/hoge/

```
> #入力ファイルの読み込み
> fasta <- readDNAStringSet(in_f, format="fasta")#in_
指定したファイルの読み込み
>
> #本番
> hoge <- oligonucleotideFrequency(fasta, width=param_km
er)#k連続塩基の出現頻度情報をhogeに格納
> out <- colSums(hoge)                    #列ごとの総和をou
tに格納
>
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F, quote=F, r
ow.names=T, col.names=F)
> out
AA AC AG AT CA CC CG CT GA GC GG GT TA TC TG TT
11 30 10  3 19  4  6 18 20  3  6  8  5 12 14 21
>
```

⑥

Files | Plots | Packages | Help | Viewer

New Folder | Delete | Rename | More

C: > Users > kadota > Desktop > hoge

Name	Size
..	
.Rhistory	14.8 KB
hoge1.txt	221 B
hoge10.png	6.2 KB
hoge10.txt	142.1 KB
hoge11.png	6.3 KB
hoge11.txt	142.1 KB
hoge12.png	5.9 KB
hoge12.txt	138.8 KB

コピペ実行結果。例題7と異なり、例題8では①2連続塩基の出現頻度解析結果をhogeオブジェクトに格納し、②それを入力として、③colSumsを実行した結果を、④outオブジェクトに格納しているので、⑤outの中身はこんな感じになります。⑥4^2 = 16通り全ての2連続塩基（2-mer）が、1回以上出現していることがわかる。ゲノムサイズ推定は、kの値を大きくして、1回以上出現しているk連続塩基(k-mer)の種類数をカウントするのが基本。

例題8（k=2）5

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

Console | Terminal | Jobs

C:/Users/kadota/Desktop/hoge/

```
> out <- colSums(hoge)                    #列ごとの総和
tに格納
>
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F, quote=F
ow.names=T, col.names=F)
> out
AA AC AG AT CA CC CG CT GA GC GG GT TA TC TG TT
11 30 10  3 19  4  6 18 20  3  6  8  5 12 14 21
> out > 0
   AA    AC    AG    AT    CA    CC    CG    CT    GA    GC
TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
   GG    GT    TA    TC    TG    TT
TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
> sum(out > 0)
[1] 16
> |
```

⑥

⑦

hoge12.txt 138.8 K

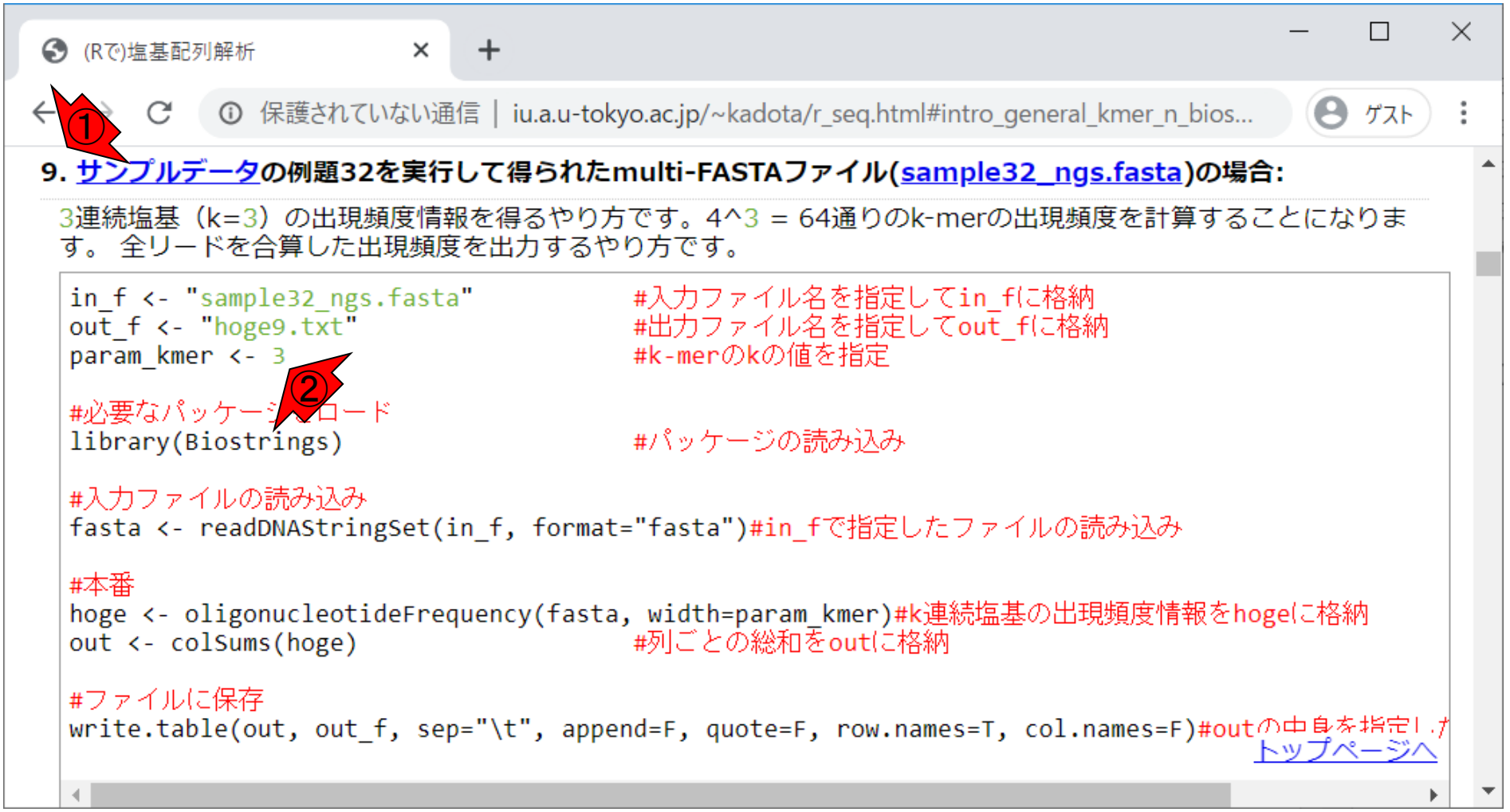
コピペ実行結果。例題7と異なり、例題8では①2連続塩基の出現頻度解析結果をhogeオブジェクトに格納し、②それを入力として、③colSumsを実行した結果を、④outオブジェクトに格納しているので、⑤outの中身はこんな感じになります。⑥4^2 = 16通り全ての2連続塩基（2-mer）が、1回以上出現していることがわかる。ゲノムサイズ推定は、kの値を大きくして、1回以上出現しているk連続塩基(k-mer)の種類数をカウントするのが基本。この場合はkの値が小さい（= 2）のでほぼ無意味だが、⑥の結果をベースとして考えると、この入力ファイル（sample32_ngs.fasta）の推定ゲノムサイズは、⑦16塩基ということになる（ちなみに正解は50塩基）。⑦のsum関数は、TRUEとなった要素数（1回以上出現したk-merの数）をカウントしていることに相当します。

Contents

- Introduction、出現頻度解析（k=2）、出現頻度解析（k=1）
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図（例題10）、作図（例題11）、作図（例題12）
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ（例題32）、被覆率（coverage）、基本的な考え方(例題7)
  - 例題8（k=2)、例題9（k=3）、1,000塩基の仮想ゲノム（サンプルデータの例題33）
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

例題9（k=3）1

①例題9。②k = 3になっただけです。コピペ実行。



9. サンプルデータの例題32を実行して得られたmulti-FASTAファイル(sample32_ngs.fasta)の場合:

3連続塩基（k=3）の出現頻度情報を得るやり方です。4^3 = 64通りのk-merの出現頻度を計算することになります。 全リードを合算した出現頻度を出力するやり方です。

```
in_f <- "sample32_ngs.fasta"            #入力ファイル名を指定してin_fに格納
out_f <- "hoge9.txt"                    #出力ファイル名を指定してout_fに格納
param_kmer <- 3                         #k-merのkの値を指定

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

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

#本番
hoge <- oligonucleotideFrequency(fasta, width=param_kmer)#k連続塩基の出現頻度情報をhogeに格納
out <- colSums(hoge)                    #列ごとの総和をoutに格納

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

トップページへ

例題9（k=3）2

①例題9。②k = 3になっただけです。コピペ実行。実行結果。

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Project: (None)

Console Terminal Jobs

C:/Users/kadota/Desktop/hoge/

```
> library(Biostrings)                    #パッケージの読み込み
>
> #入力ファイルの読み込み
> fasta <- readDNAStringSet(in_f, format="fasta")#in_fで指定したファイルの読み込み
>
> #本番
> hoge <- oligonucleotideFrequency(fasta, width=param_km
er)#k連続塩基の出現頻度情報をhogeに格納
> out <- colSums(hoge)                   #列ごとの総和をou
tに格納
>
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F, quote=F, r
ow.names=T, col.names=F)
>
```

Environment History Connections

To Console | To Source

```
out <- colSums(hoge) #列...
#ファイルに保存
write.table(out, out_f, ...
```

Files Plots Packages Help Viewer

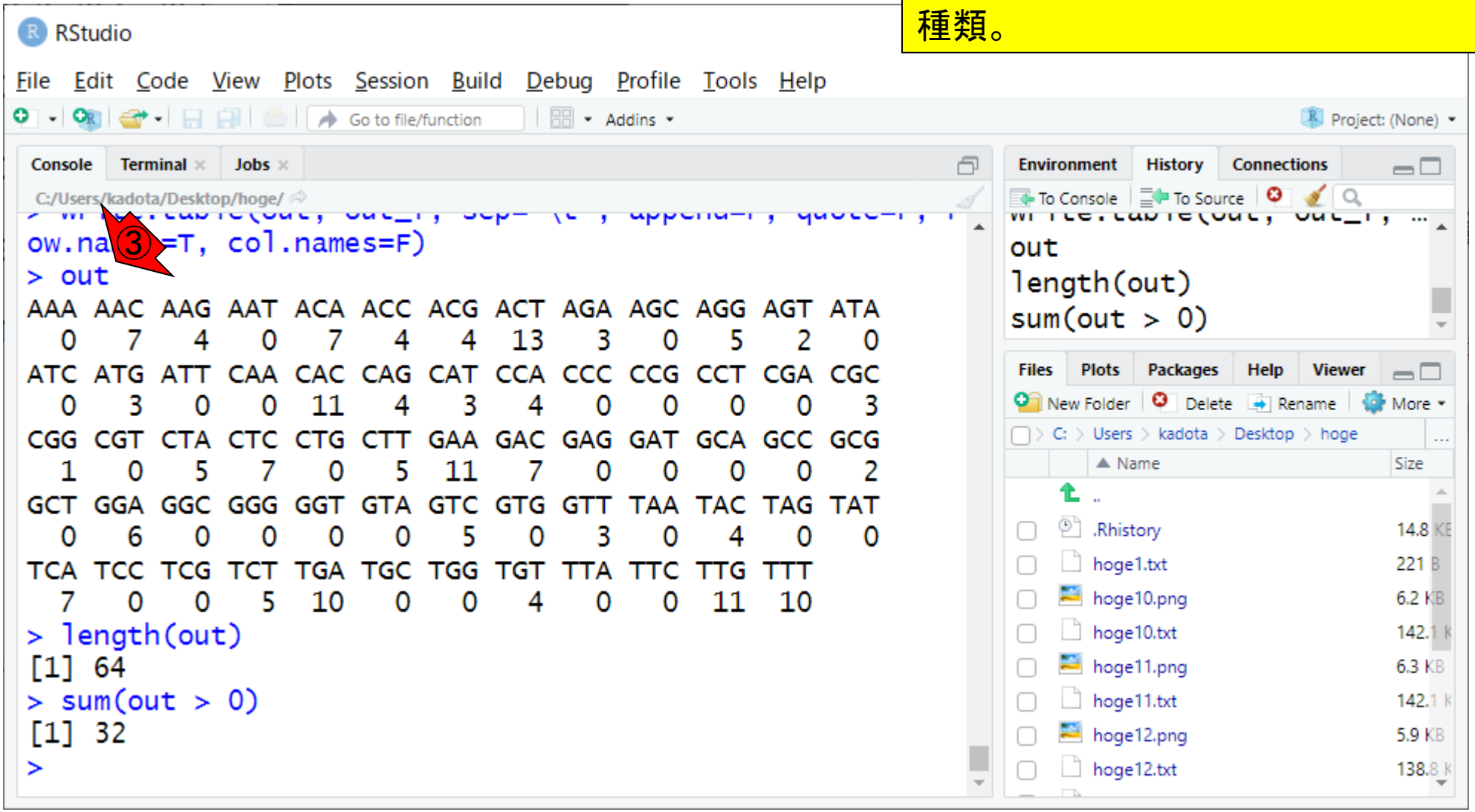
New Folder | Delete | Rename | More

C: > Users > kadota > Desktop > hoge

Name	Size
..	
.Rhistory	14.8 KB
hoge1.txt	221 B
hoge10.png	6.2 KB
hoge10.txt	142.1 KB
hoge11.png	6.3 KB
hoge11.txt	142.1 KB
hoge12.png	5.9 KB
hoge12.txt	138.8 KB

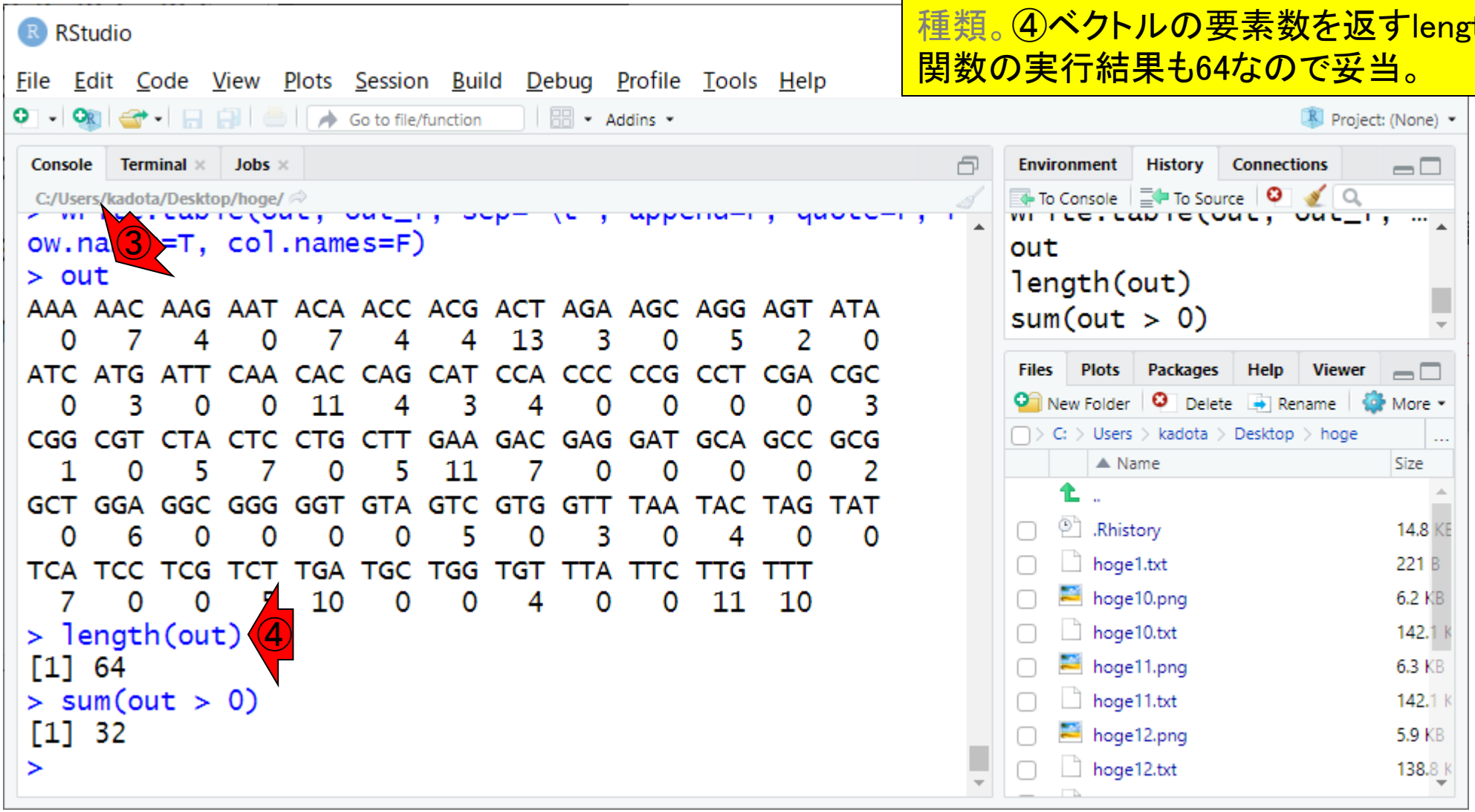
例題9（k=3）3

①例題9。②k = 3になっただけです。コピペ実行。実行結果。③outの中身を表示。4種類の塩基が3連続なので4^3 = 64種類。



例題9（k=3）4

①例題9。②k = 3になっただけです。コピペ実行。実行結果。③outの中身を表示。4種類の塩基が3連続なので4^3 = 64種類。④ベクトルの要素数を返すlength関数の実行結果も64なので妥当。



例題9（k=3）5

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

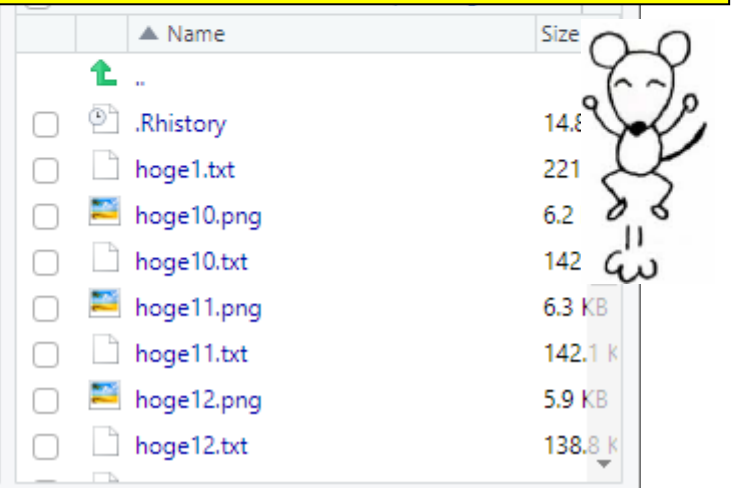
Go to file/function | Addins

Console | Terminal | Jobs

C:/Users/kadota/Desktop/hoge/

```
ow.na③=T, col.names=F)
> out
AAA AAC AAG AAT ACA ACC ACG ACT AGA AGC AGG AGT ATA
  0   7   4   0   7   4   4  13   3   0   5   2   0
ATC ATG ATT CAA CAC CAG CAT CCA CCC CCG CCT CGA CGC
  0   3   0   0  11   4   3   4   0   0   0   0   3
CGG CGT CTA CTC CTG CTT GAA GAC GAG GAT GCA GCC GCG
  1   0   5   7   0   5  11   7   0   0   0   0   2
GCT GGA GGC GGG GGT GTA GTC GTG GTT TAA TAC TAG TAT
  0   6   0   0   0   0   5   0   3   0   4   0   0
TCA TCC TCG TCT TGA TGC TGG TGT TTA TTC TTG TTT
  7   0   0   5  10   0   0   4   0   0  11  10
> length(out)④
[1] 64
> sum(out > 0)⑤
[1] 32
>
```

①例題9。②k = 3になっただけです。コピペ実行。実行結果。③outの中身を表示。4種類の塩基が3連続なので4^3 = 64種類。④ベクトルの要素数を返すlength関数の実行結果も64なので妥当。⑤1回以上出現したk-merの数は32。これはk=3でのゲノムサイズ推定値であり、k=2のときの推定値（16塩基）のときよりも正解（50塩基）に近づいていることがわかる。こんな感じで簡単なテクニックさえあれば、シミュレーションデータを自在に作成して自分で検証して納得することができるのがよいですね。

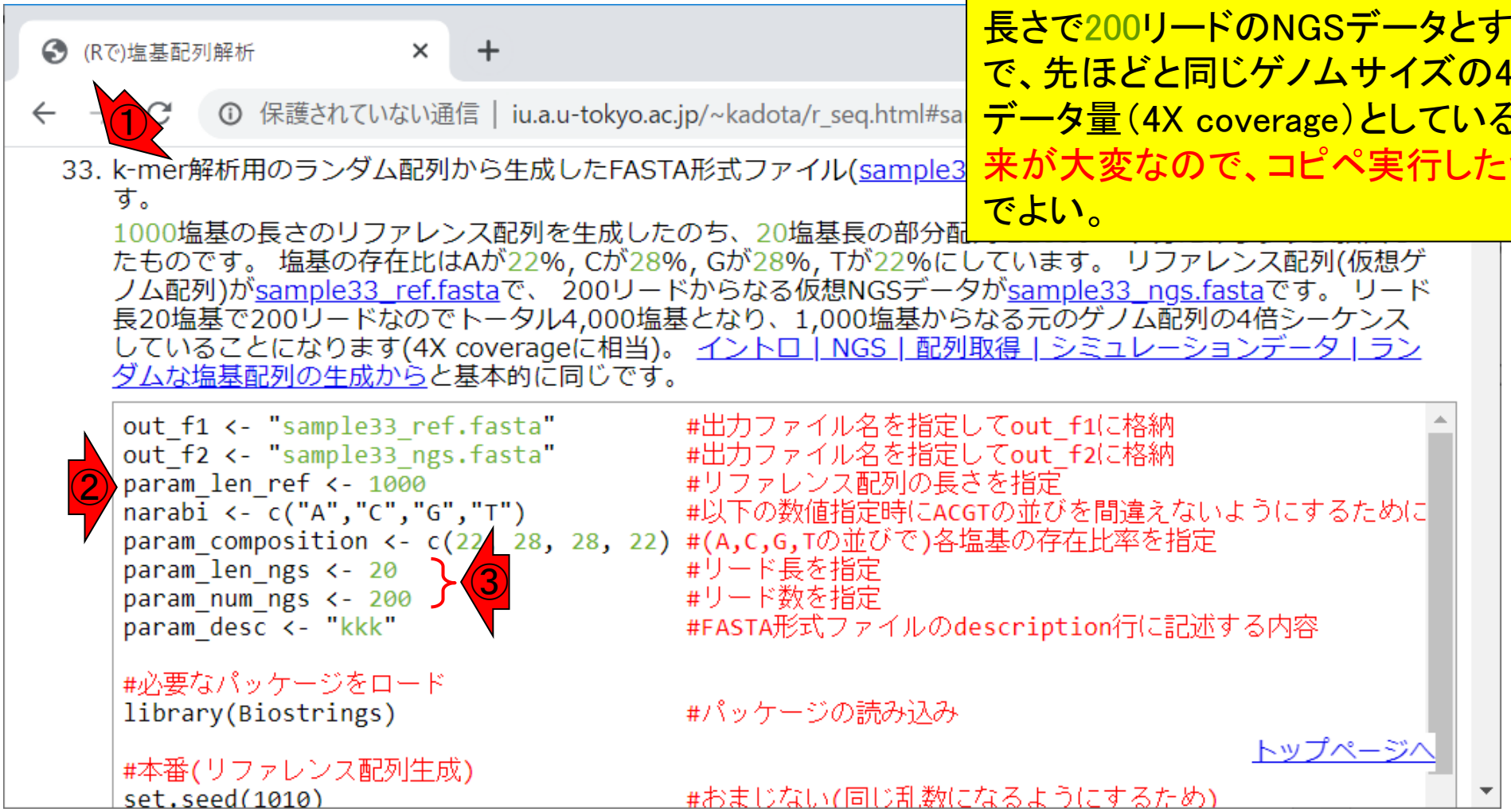


Contents

- Introduction、出現頻度解析（k=2）、出現頻度解析（k=1）
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図（例題10）、作図（例題11）、作図（例題12）
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ（例題32）、被覆率（coverage）、基本的な考え方(例題7)
  - 例題8（k=2)、例題9（k=3）、1,000塩基の仮想ゲノム（サンプルデータの例題33）
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

サンプルデータ(例題33)

①サンプルデータの例題33。より現実の解析に近づけるべく、②1000塩基からなる仮想ゲノム配列を作成。③20塩基の長さで200リードのNGSデータとすることで、先ほどと同じゲノムサイズの4倍のデータ量(4X coverage)としている。行き来が大変なので、コピペ実行したつもりでよい。



33. k-mer解析用のランダム配列から生成したFASTA形式ファイル(sample3
す。
1000塩基の長さのリファレンス配列を生成したのち、20塩基長の部分配
たものです。 塩基の存在比はAが22%, Cが28%, Gが28%, Tが22%にしています。 リファレンス配列(仮想ゲノム配列)がsample33_ref.fastaで、 200リードからなる仮想NGSデータがsample33_ngs.fastaです。 リード長20塩基で200リードなのでトータル4,000塩基となり、1,000塩基からなる元のゲノム配列の4倍シーケンスしていることになります(4X coverageに相当)。 イントロ | NGS | 配列取得 | シミュレーションデータ | ランダムな塩基配列の生成からと基本的に同じです。

```
out_f1 <- "sample33_ref.fasta"          #出力ファイル名を指定してout_f1に格納
out_f2 <- "sample33_ngs.fasta"          #出力ファイル名を指定してout_f2に格納
param_len_ref <- 1000                   #リファレンス配列の長さを指定
narabi <- c("A","C","G","T")            #以下の数値指定時にACGTの並びを間違えないようにするために
param_composition <- c(22, 28, 28, 22) #(A,C,G,Tの並びで)各塩基の存在比率を指定
param_len_ngs <- 20                     #リード長を指定
param_num_ngs <- 200                    #リード数を指定
param_desc <- "kkk"                     #FASTA形式ファイルのdescription行に記述する内容

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

トップページへ

```
#本番(リファレンス配列生成)
set.seed(1010)                          #おまじない(同じ乱数になるようにするため)
```

サンプルデータ(例題33)

①サンプルデータの例題33。より現実の解析に近づけるべく、②1000塩基からなる仮想ゲノム配列を作成。③20塩基の長さで200リードのNGSデータとすることで、先ほどと同じゲノムサイズの4倍のデータ量(4X coverage)としている。行き来が大変なので、コピペ実行したつもりでよい。**重要なのは、④のファイル(sample33_ngs.fasta)を入力としてゲノムサイズ推定を行ったときの正解は1,000塩基だということ。**

(Rで)塩基配列解析

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

33. k-mer解析用のランダム配列から生成したFASTA形式ファイル(sample3
す。
1000塩基の長さのリファレンス配列を生成したのち、20塩基長の部分配
たものです。 塩基の存在比はAが22%, Cが28%, Gが28%, Tが22%に
ノム配列)がsample33_ref.fastaで、 200リードからなる仮想NGSデー
長20塩基で200リードなのでトータル4,000塩基となり、1,000塩基から
していることになります(4X coverageに相当)。 イントロ | NGS | 配列
ダムな塩基配列の生成からと基本的に同じです。

```
out_f1 <- "sample33_ref.fasta"          #出力ファイル名を指定してout_f1に格納
out_f2 <- "sample33_ngs.fasta"          #出力ファイル名を指定してout_f2に格納
param_len_ref <- 1000                   #リファレンス配列の長さを指定
narabi <- c("A","C","G","T")            #以下の数値指定時にACGTの並びを間違えないようにする
param_composition <- c(22, 28, 28, 22)  #(A,C,G,Tの並びで)各塩基の存在比率を指定
param_len_ngs <- 20                     #リード長を指定
param_num_ngs <- 200                    #リード数を指定
param_desc <- "kkk"                     #FASTA形式ファイルのdescription行に記述する内容

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

                                                    トップページへ
#本番(リファレンス配列生成)
set.seed(1010)                          #おまじない(同じ乱数になるようにするため)
```

サンプルデータ（例題33）

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

Console | Terminal | Jobs

C:/Users/ /Desktop/hoge/

⑤

```
> 4^3
[1] 64
> 4^4
[1] 256
> 4^5
[1] 1024
> 4^6
[1] 4096
> 4^7
[1] 16384
> 4^8
[1] 65536
> 4^9
[1] 262144
> 4^10
[1] 1048576
>
```

⑥

hoge12.txt 138.8 K

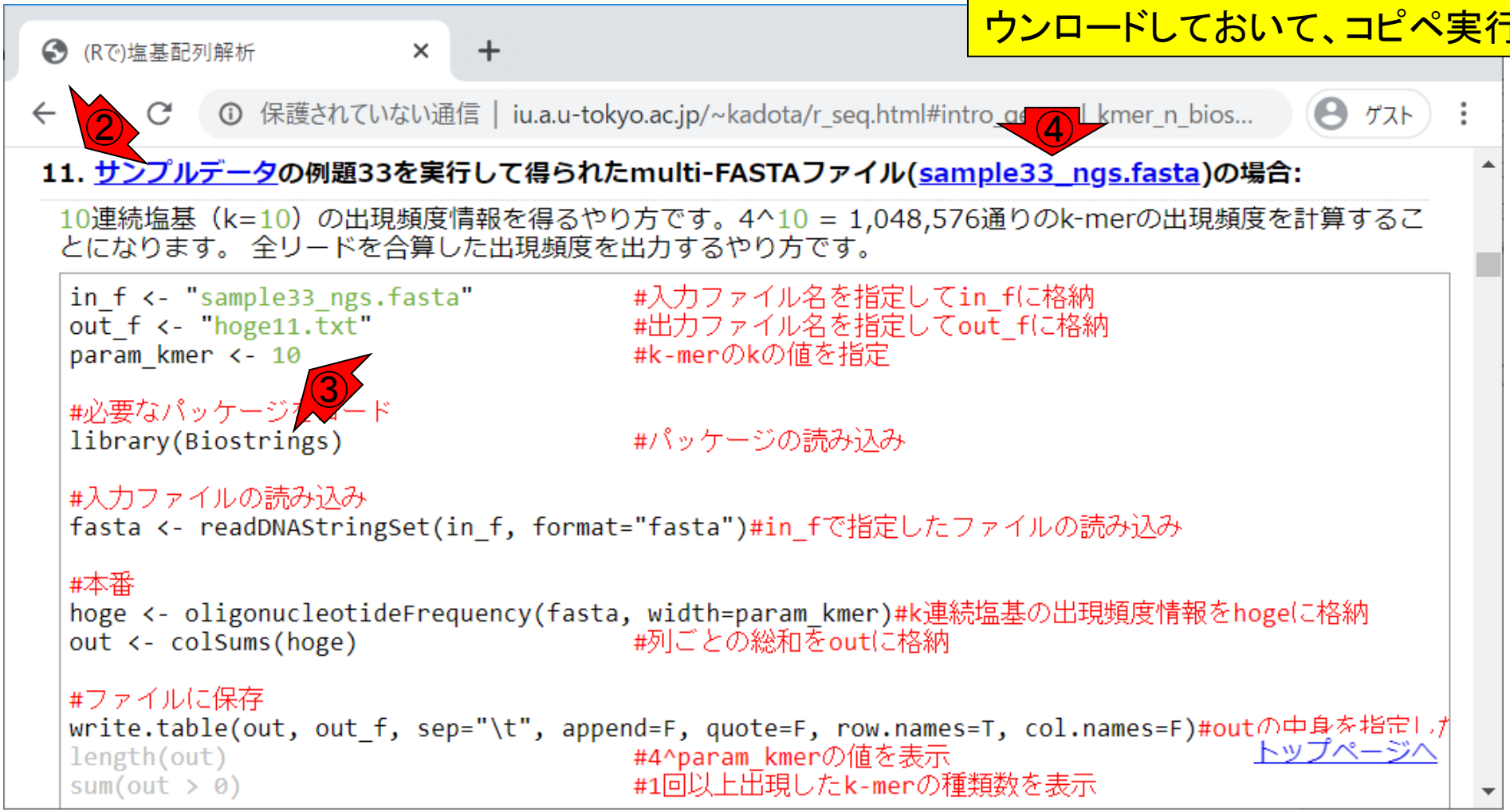
①サンプルデータの例題33。より現実の解析に近づけるべく、②1000塩基からなる仮想ゲノム配列を作成。③20塩基の長さで200リードのNGSデータとすることで、先ほどと同じゲノムサイズの4倍のデータ量（4X coverage）としている。行き来が大変なので、コピペ実行したつもりでよい。重要なのは、④のファイル（sample33_ngs.fasta）を入力としてゲノムサイズ推定を行ったときの正解は1,000塩基だということ。それゆえ、⑤k=3で1回以上出現したk-merの数を調べたとしても、最大値でも4^3 = 64にしかならないので、1,000塩基のゲノムサイズ推定を行うのは原理的に不可能という事実を正しく理解しておかねばならない。1,000塩基のゲノムサイズ推定目的の場合は、最低でも⑥k=5は必要だが、現実にはシークエンスエラーなども含むのでもっと大きなk値を選択します。

Contents

- Introduction、出現頻度解析（k=2）、出現頻度解析（k=1）
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図（例題10）、作図（例題11）、作図（例題12）
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ（例題32）、被覆率（coverage）、基本的な考え方(例題7)
  - 例題8（k=2)、例題9（k=3）、1,000塩基の仮想ゲノム（サンプルデータの例題33）
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

例題11（k=10）1

①の、②例題11。③k=10で、④正解ゲノムサイズ1,000塩基だということが分かっているファイル（sample33_ngs.fasta）をダウンロードしておいて、コピペ実行。



11. サンプルデータの例題33を実行して得られたmulti-FASTAファイル(sample33_ngs.fasta)の場合:

10連続塩基（k=10）の出現頻度情報を得るやり方です。4^10 = 1,048,576通りのk-merの出現頻度を計算することになります。 全リードを合算した出現頻度を出力するやり方です。

```
in_f <- "sample33_ngs.fasta"            #入力ファイル名を指定してin_fに格納
out_f <- "hoge11.txt"                   #出力ファイル名を指定してout_fに格納
param_kmer <- 10                        #k-merのkの値を指定

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

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

#本番
hoge <- oligonucleotideFrequency(fasta, width=param_kmer)#k連続塩基の出現頻度情報をhogeに格納
out <- colSums(hoge)                    #列ごとの総和をoutに格納

#ファイルに保存
write.table(out, out_f, sep="\t", append=F, quote=F, row.names=T, col.names=F)#outの中身を指定した
length(out)                             #4^param_kmerの値を表示
sum(out > 0)                            #1回以上出現したk-merの種類数を表示
```

トップページへ

例題11（k=10）2

①の、②例題11。③k=10で、④正解ゲノムサイズ1,000塩基だということが分かっているファイル（sample33_ngs.fasta）をダウンロードしておいて、コピペ実行。実行結果。⑤推定ゲノムサイズは904。真の値に近いといえば近いが、若干小さめともいえる。この主な理由は、入力ファイルのデータ量が少なめだから（4X coverage）。データ量を10X coverageに増やしたものが例題12。

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

Console Terminal Jobs

R 4.2.0 · C:/Users/kadota/Desktop/hogo/

```
> #必要なパッケージをロード
> library(Biostrings)                    #パッケージの読み込み
>
> #入力ファイルの読み込み
> fasta <- readDNAStringSet(in_f, format="fasta")#in_fで指定したファイルの読み込み
>
> #本番
> hoge <- oligonucleotideFrequency(fasta, width=param_kmer)#k連続塩基の出現頻度情報をhogeに格納
> out <- colSums(hoge)                   #列ごとの総和をoutに格納
>
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F, quote=F, row.names=T, col.names=F)#outの中身を指定したファイル名で保存
> length(out)                            #4^param_kmerの値を表示
[1] 1048576
> sum(out > 0)                           #1回以上出現したk-merの種類数を表示
[1] 904
>
>
```

⑤

out_f2	"sample32_ngs.fast…
param_…	chr [1:5] "A" "C" …
param_…	num [1:4] 22 28 28…
param_…	"kkk"
param_…	10

Files Plots Packages Help Viewer

C: > Users > kadota > Desktop > hogo

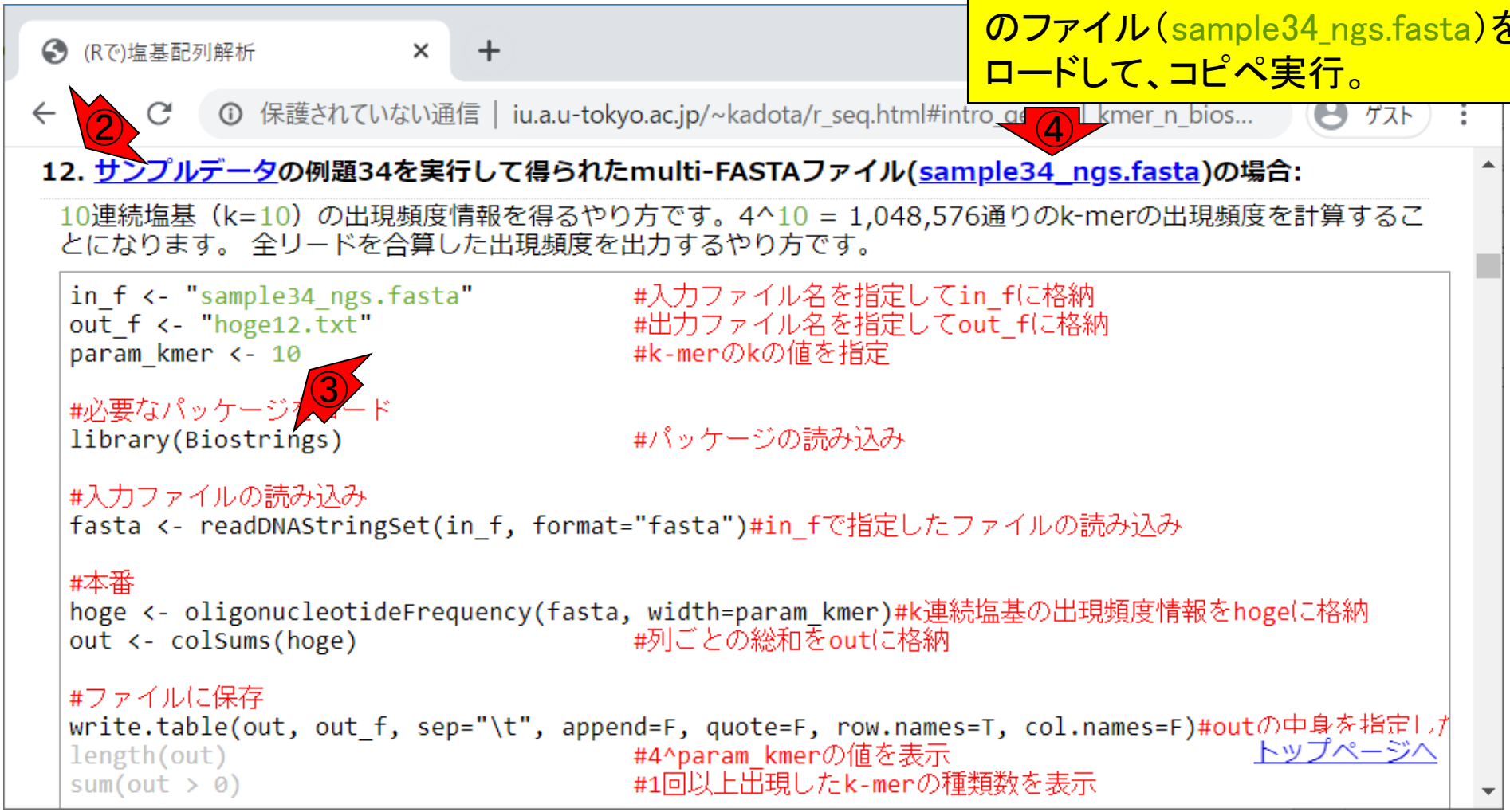
Name	Size
..	
seq_20.fasta	29 B
hoge4.fa	299 B
hoge2.txt	104 B
hoge1.txt	221 B

Contents

- Introduction、出現頻度解析（k=2）、出現頻度解析（k=1）
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図（例題10）、作図（例題11）、作図（例題12）
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ（例題32）、被覆率（coverage）、基本的な考え方(例題7)
  - 例題8（k=2)、例題9（k=3）、1,000塩基の仮想ゲノム（サンプルデータの例題33）
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

例題12（k=10）1

①の、②例題12。③k=10で、④正解ゲノムサイズ1,000塩基から20塩基長のリードを500個分生成した、総塩基数10,000のファイル（sample34_ngs.fasta）をダウンロードして、コピペ実行。



12. サンプルデータの例題34を実行して得られたmulti-FASTAファイル(sample34_ngs.fasta)の場合:

10連続塩基（k=10）の出現頻度情報を得るやり方です。4^10 = 1,048,576通りのk-merの出現頻度を計算することになります。 全リードを合算した出現頻度を出力するやり方です。

```
in_f <- "sample34_ngs.fasta"            #入力ファイル名を指定してin_fに格納
out_f <- "hoge12.txt"                   #出力ファイル名を指定してout_fに格納
param_kmer <- 10                        #k-merのkの値を指定

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

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

#本番
hoge <- oligonucleotideFrequency(fasta, width=param_kmer)#k連続塩基の出現頻度情報をhogeに格納
out <- colSums(hoge)                    #列ごとの総和をoutに格納

#ファイルに保存
write.table(out, out_f, sep="\t", append=F, quote=F, row.names=T, col.names=F)#outの中身を指定した
length(out)                             #4^param_kmerの値を表示
sum(out > 0)                            #1回以上出現したk-merの種類数を表示
```

トップページへ

例題12(k=10)2

①の、②例題13。③k=10で、④正解ゲノムサイズ1,000塩基から20塩基長のリードを500個分生成した、総塩基数10,000のファイル(sample34_ngs.fasta)をダウンロードして、コピペ実行。実行結果。⑤推定ゲノムサイズは985。ほぼ正解ですね。シークエンスエラーのないデータなら、10X程度のcoverageであれば十分だということがわかります。

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

Console | Terminal | Jobs

R 4.2.0 · C:/Users/kadota/Desktop/hogo/

```
> #必要なパッケージをロード
> library(Biostrings)                    #パッケージの読み込み
>
> #入力ファイルの読み込み
> fasta <- readDNAStringSet(in_f, format="fasta")#in_fで指定したファイルの読み込み
>
> #本番
> hoge <- oligonucleotideFrequency(fasta, width=param_kmer)#k連続塩基の出現頻度情報をhogeに格納
> out <- colSums(hoge)                   #列ごとの総和をoutに格納
>
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F, quote=F, row.names=T, col.names=F)#outの中身を指定したファイル名で保存
> length(out)                            #4^param_kmerの値を表示
[1] 1048576
> sum(out > 0)                           #1回以上出現したk-merの種類数を表示
[1] 985
>
> |
```

⑤

out_f	"hoge12.txt"
out_f1	"sample32_ref.fast…
out_f2	"sample32_ngs.fast…
param_…	chr [1:5] "A" "C" …
param_…	num [1:4] 22 28 28…
param_…	"kkk"
param_…	10

Files | Plots | Packages | Help | Viewer

C: > Users > kadota > Desktop > hogo

Name	Size
..	
seq_20.fasta	29 B
hoge4.fa	299 B
hoge2.txt	104 B
hoge1.txt	221 B

例題12(k=10)3

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

Console | Terminal | Jobs

R 4.2.0 · C:/Users/kadota/Desktop/hogo/

```
> #必要なパッケージをロード
> library(Biostrings)                    #パッケージの読み込み
>
> #入力ファイルの読み込み
> fasta <- readDNAStringSet(in_f, format="fasta")#in_fで指定したファイルの読み込み
>
> #本番
> hoge <- oligonucleotideFrequency(fasta, width=param_kmer)#k連続塩基の出現頻度情報をhogeに格納
> out <- colSums(hoge)                   #列ごとの総和をoutに格納
>
> #ファイルに保存
> write.table(out, out_f, sep="\t", append=F, quote=F, row.names=T, col.names=F)#outの中身を指定したファイル名で保存
> length(out)                            #4^param_kmerの値を表示
[1] 1048576
> sum(out > 0)                           #1回以上出現したk-merの種類数を表示
[1] 985
>
> |
```

⑤

Name | Size

..	
seq_20.fasta	29 B
hoge4.fa	299 B
hoge2.txt	104 B
hoge1.txt	221 B

①の、②例題13。③k=10で、④正解ゲノムサイズ1,000塩基から20塩基長のリードを500個分生成した、総塩基数10,000のファイル(sample34_ngs.fasta)をダウンロードして、コピペ実行。実行結果。⑤推定ゲノムサイズは988。ほぼ正解ですね。シークエンスエラーのないデータなら、10X程度のcoverageであれば十分だということがわかります。ちなみに最初のほうのスライドでも解説しているが、「長さ1,000の塩基配列を10-merで分割すると、(1,000 − 10 + 1)個のk-merを生成可能」でした。従って、この場合の「1回以上出現したk-merの数」をカウントするやり方で得られるパーフェクトの値は991だという点は念頭においておくとよい。

Contents

- Introduction、出現頻度解析（k=2）、出現頻度解析（k=1）
- k=1で実践、multi-FASTAファイル、他の例題を実行
- k=2で実践、関数マニュアル、例題2を実行、例題7を実行
- 確率の話、作図（例題10）、作図（例題11）、作図（例題12）
- 塩基配列解析の基礎
  - GC含量、ランダム配列を生成、部分配列の切り出し
- ゲノムサイズ推定
  - サンプルデータ（例題32）、被覆率（coverage）、基本的な考え方(例題7)
  - 例題8（k=2)、例題9（k=3）、1,000塩基の仮想ゲノム（サンプルデータの例題33）
  - 例題11(k=10)、例題12(k=10)、シークエンスエラーを含む場合

シークエンスエラー1

実際には、シークエンスエラーを含むデータからゲノムサイズ推定が行われます。①で基本戦略についての解説がなされています。

参考図書

藤博幸・編、**よくわかるバイオインフォマティクス入門**、講談社、2018
坊農秀雅 著、**Dr.Bonoの生命科学データ解析**、MEDSi、2017
坊農秀雅 著、**RNA-Seqデータ解析 WETラボのための鉄板レシピ**、羊土社、2019

講義日程（2020年度）

1. 2020年05月12日(谷澤)
2. 2020年05月19日(門田)
   講義資料PDF(最終更新：2020.03.16)
   (Rで)塩基配列解析
   seq_20.fasta
   NGSハンズオン講習会の2016年7月20日の講義資料(スライド114-) ①
   課題1：**kadai1.fasta**
   課題3：**kadai3.fasta**
   課題4：**kadai4_2mer.txt**, **kadai4.R**
   課題5：**kadai5.fasta**
3. 2020年05月26日(門田)
4. 2020年06月02日(門田)

シークエンスエラー2

実際には、シークエンスエラーを含むデータからゲノムサイズ推定が行われます。①で基本戦略についての解説がなされています。②のスライド114以降。

ゲノム解析、塩基配列解析

門田 幸二

- NGS解析手段、ウェブツール(DDBJ Pipeline)との連携
- k-mer解析（k個の連続塩基に基づく各種解析）の基礎と応用
- 塩基ごとの出現頻度解析(k=1)、2連続塩基の出現頻度解析(k=2)
- 塩基配列解析を行うための基本スキルの復習や作図
- de novoアセンブリ時のエラー補正やゲノムサイズ推定の基本的な考え方

› 講義動画 › スライド(7.3MB) ②

› 予習項目 › 講義詳細 › 解析データ(ZIP:2.2MB)