2019.04.25版

ゲノム情報解析基礎：第4回

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

各講義科目へのアクセス

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

Contents

- 配列長でフィルタリング（ゲノムアセンブリ結果の後処理）
  - イントロ、本番
- ゲノムアノテーション
  - イントロ、参考文献
  - DFAST実行、課題1
- CDS（タンパク質コード領域の配列；coding sequenceの略）の取得
  - 比較ゲノム解析のイントロ、問題設定
  - 答え（cds.fna）を眺めておく、sequence logos
  - CDS配列取得の成功例、TxDbオブジェクト、CDS座標情報取得
  - CDS配列取得の失敗例、GFF3をExcelで概観
  - 基本テクを駆使して解決する基本戦略、CDSを含む行の抽出、補足説明
  - Excelで確認、リストファイルの作成、基本テクでCDS配列取得を実行
  - ストランド情報つきのリストファイルの作成、例題6のコード解説
  - ストランド情報反映戦略を練り、コードに修正を加えて実行（例題7）

①（Rで）塩基配列解析のサブ。

本科目のページ

講義日程（2019年度）

1. 2019年04月08日（PC使用）
   講義資料PDF(最終更新：2019.04.09)
   学会(国外)：ISCB
   学会(国内)：JSBi
   QAサイト：Biostar (Parnell et al., PLoS Comput Biol., 2011)
   QAサイト：SEQanswers (Li et al., Bioinformatics, 2012)
   学習教材：バイオインフォマティクス人材育成のための講習会(平成26-29年度)
   学習教材：(Rで)塩基配列解析
   学習教材：(Rで)塩基配列解析のサブ
   RStudio

2. 2019年04月15日（PC使用）
   講義資料PDF(最終更新：2019.04.15)
   (Rで)塩基配列解析
   (Rで)塩基配列解析のサブ

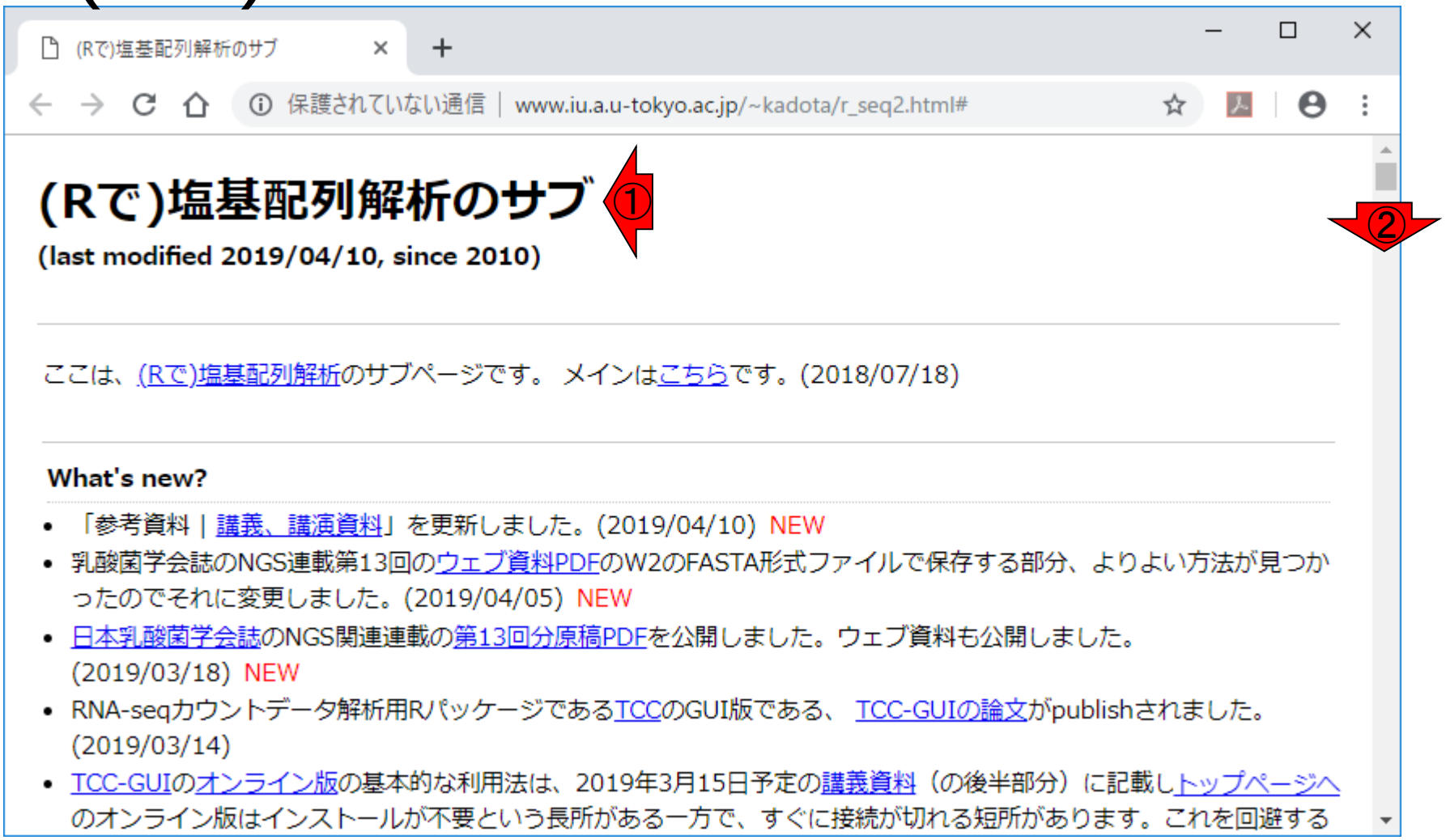
3. 2019年04月22日（PC使用）
   講義資料PDF
   課題：docx版とPDF版
   (Rで)塩基配列解析
   hoge8.fa (課題用)
   Bioconductor
   CRAN

4. 2019年05月13日（PC使用）
   講義資料PDF
   (Rで)塩基配列解析のサブ
   out_gapClosed.fa (約2.3MB)
   (Rで)塩基配列解析
   DFAST：Tanizawa et al., Bioinformatics, 2018
   DFAST実行結果：genome.fna
   DFAST実行結果：annotation.gff
   DFAST実行結果：cds.fna

4. 2019年05月13日（PC使用）
講義資料PDF
①
(Rで)塩基配列解析のサブ
out_gapClosed.fa (約2.3MB)
(Rで)塩基配列解析
DFAST：Tanizawa et al., Bioinformatics, 2018
DFAST実行結果：genome.fna
DFAST実行結果：annotation.gff
DFAST実行結果：cds.fna

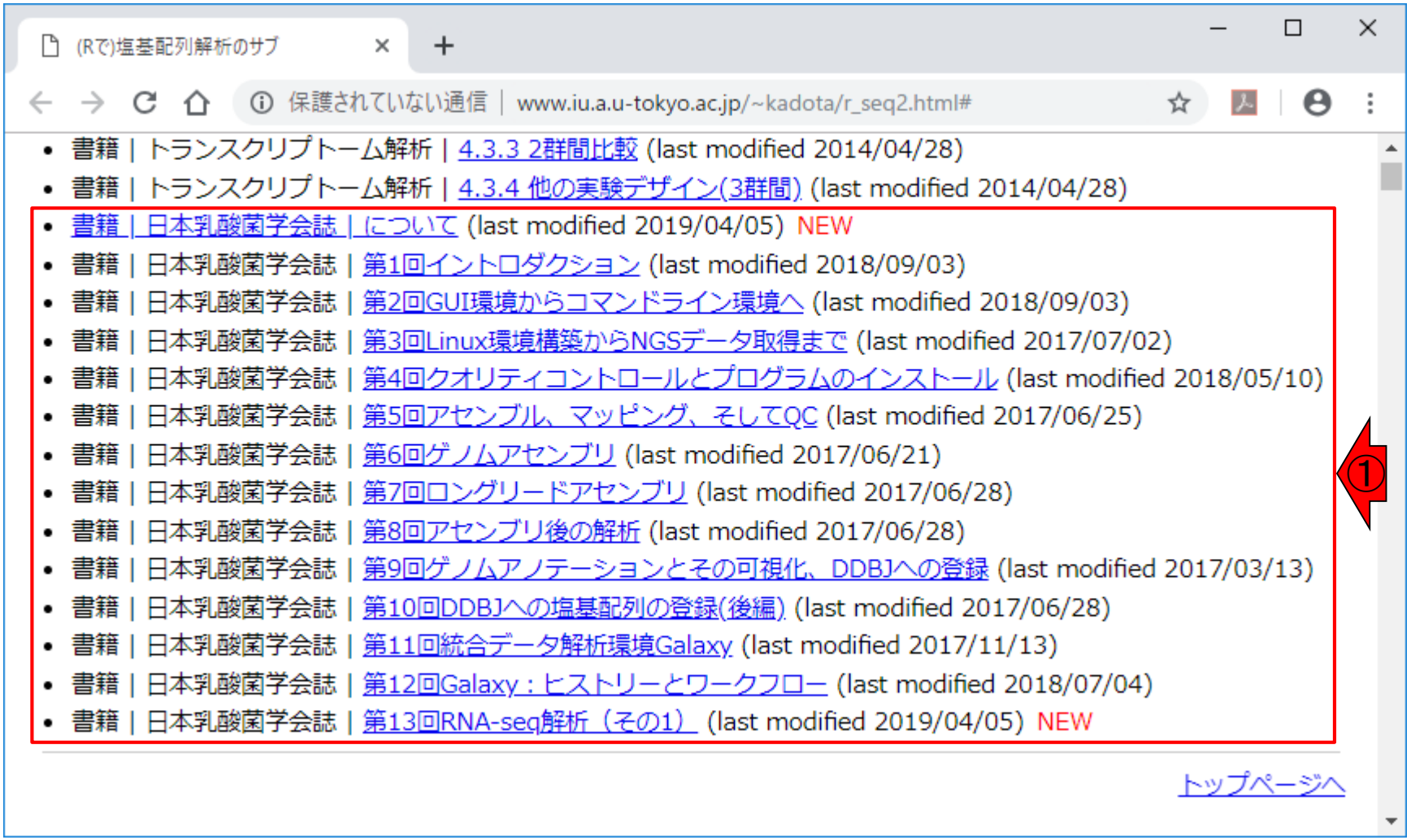
(Rで)塩基配列解析のサブ

①サブのほうのページです。②3ページ分ほど下に移動。



乳酸菌NGS連載

①日本乳酸菌学会誌のNGSデータ解析手法に関する連載（乳酸菌NGS連載）のところ。



乳酸菌NGS連載

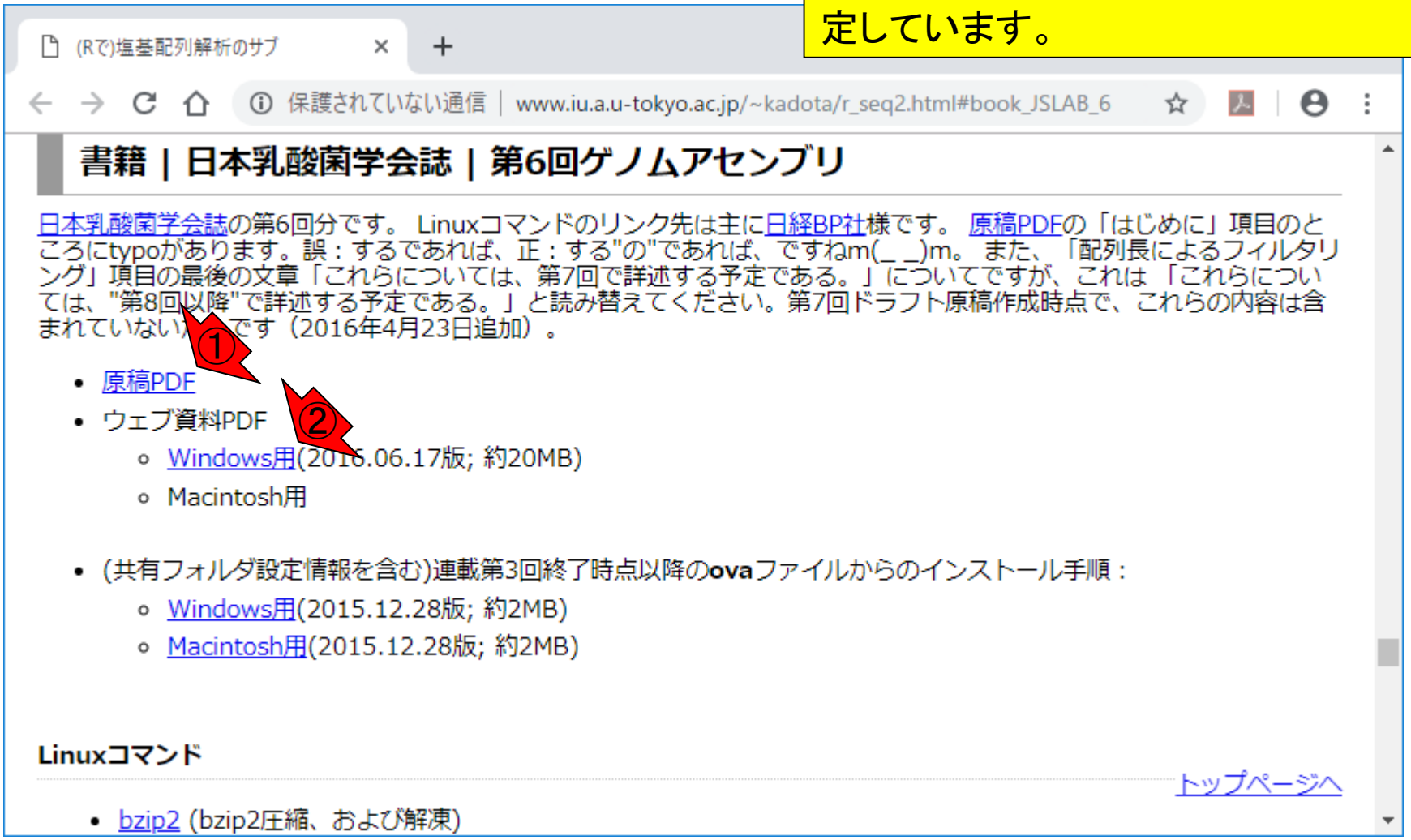
①2014年、②2016年、③2018年出版です。最終更新の日付から出版年が読み解けないのは、リンク切れなどの修正を随時行っているからです。

乳酸菌NGS連載

①では、2つの *de novo*ゲノムアセンブリプログラム（VelvetとPlatanus）の比較を行っています。*de novo* というのは、「最初から」という意味です。入力はNGS塩基配列データのFASTQ形式ファイル、出力はmulti-FASTA形式ファイルです。この場合の「最初から」というのは、他の補足情報を使わずにNGSデータのみを用いるという意味だと解釈すればよい。
①をクリック

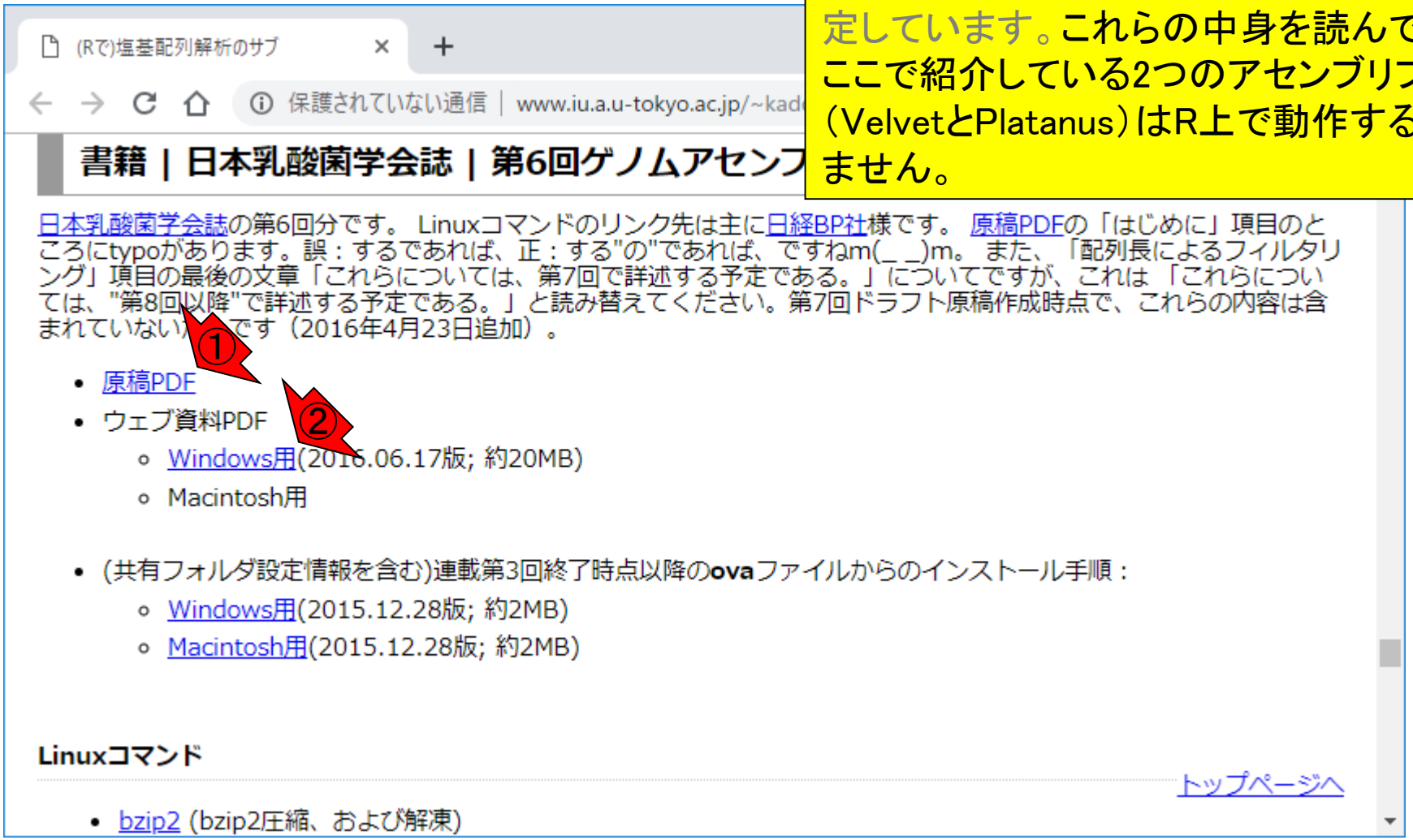
連載第6回

乳酸菌NGS連載の教材は、基本的にメインの①原稿PDFを読みながら、②ウェブ資料PDFでより詳細な情報を知る、という感じで利用してもらうことを想定しています。

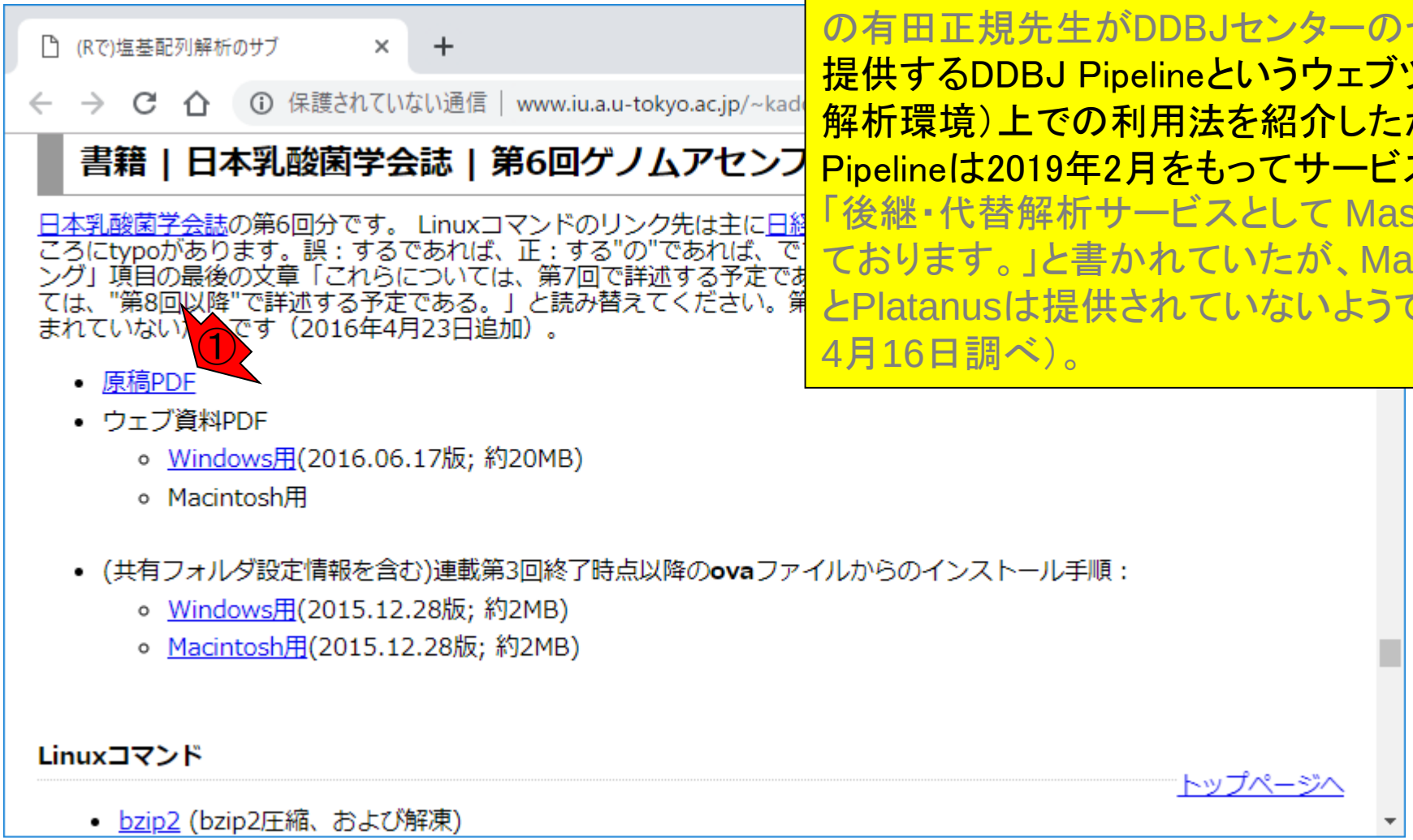


連載第6回

乳酸菌NGS連載の教材は、基本的にメインの①原稿PDFを読みながら、②ウェブ資料PDFでより詳細な情報を知る、という感じで利用してもらうことを想定しています。これらの中身を読んでもわかるが、ここで紹介している2つのアセンブリプログラム（VelvetとPlatanus）はR上で動作するものではありません。



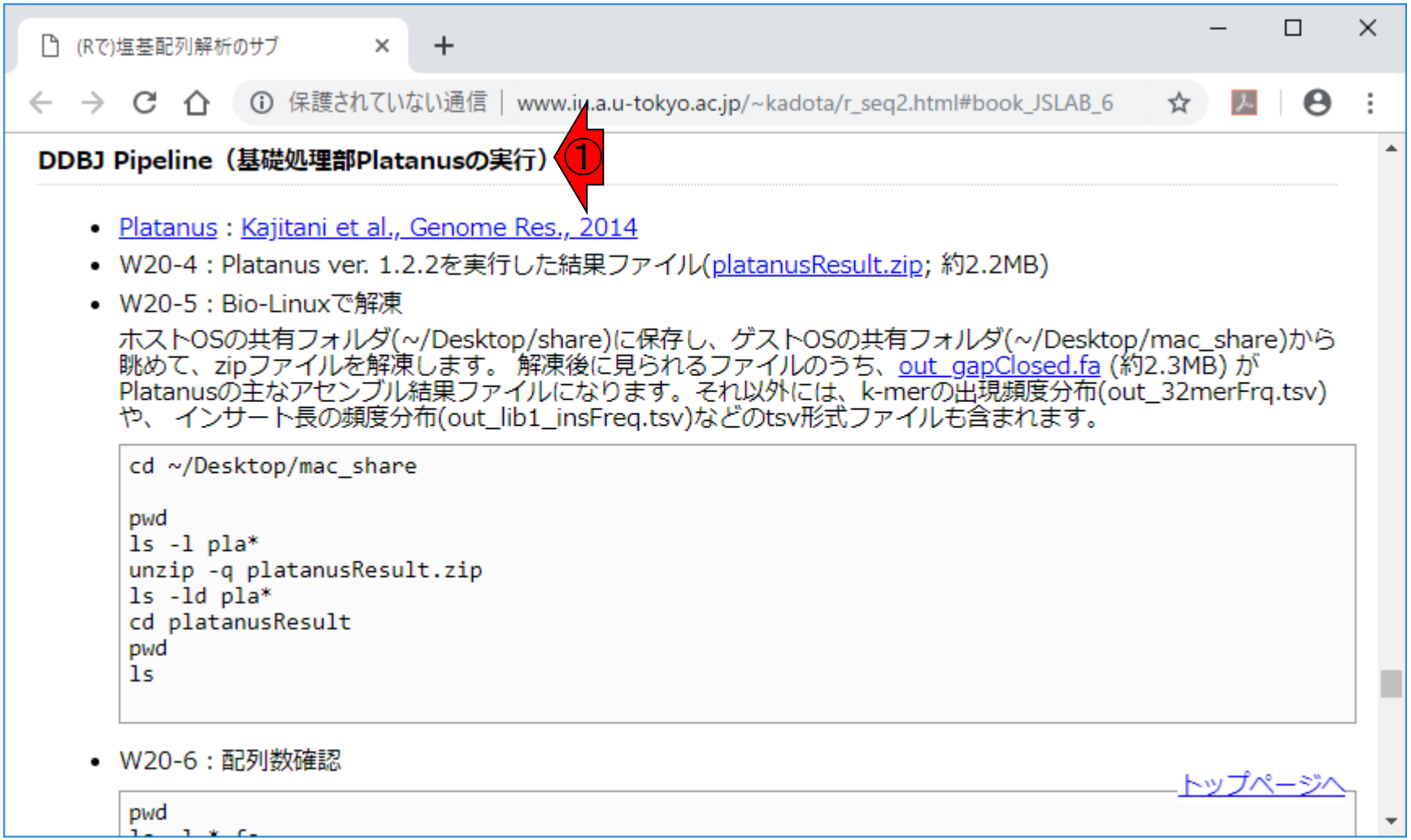
連載第6回



①第6回の原稿中では、DDBJ（塩基配列を登録する際にお世話になるところ。遺伝研内にあります。アグリバイオ講義科目「システム生物学概論」担当の有田正規先生がDDBJセンターのセンター長）が提供するDDBJ Pipelineというウェブツール（クラウド解析環境）上での利用法を紹介したが、DDBJ Pipelineは2019年2月をもってサービス終了しました。「後継・代替解析サービスとして Maser をご案内しております。」と書かれていたが、MaserではVelvetとPlatanusは提供されていないようで残念（2019年4月16日調べ）。

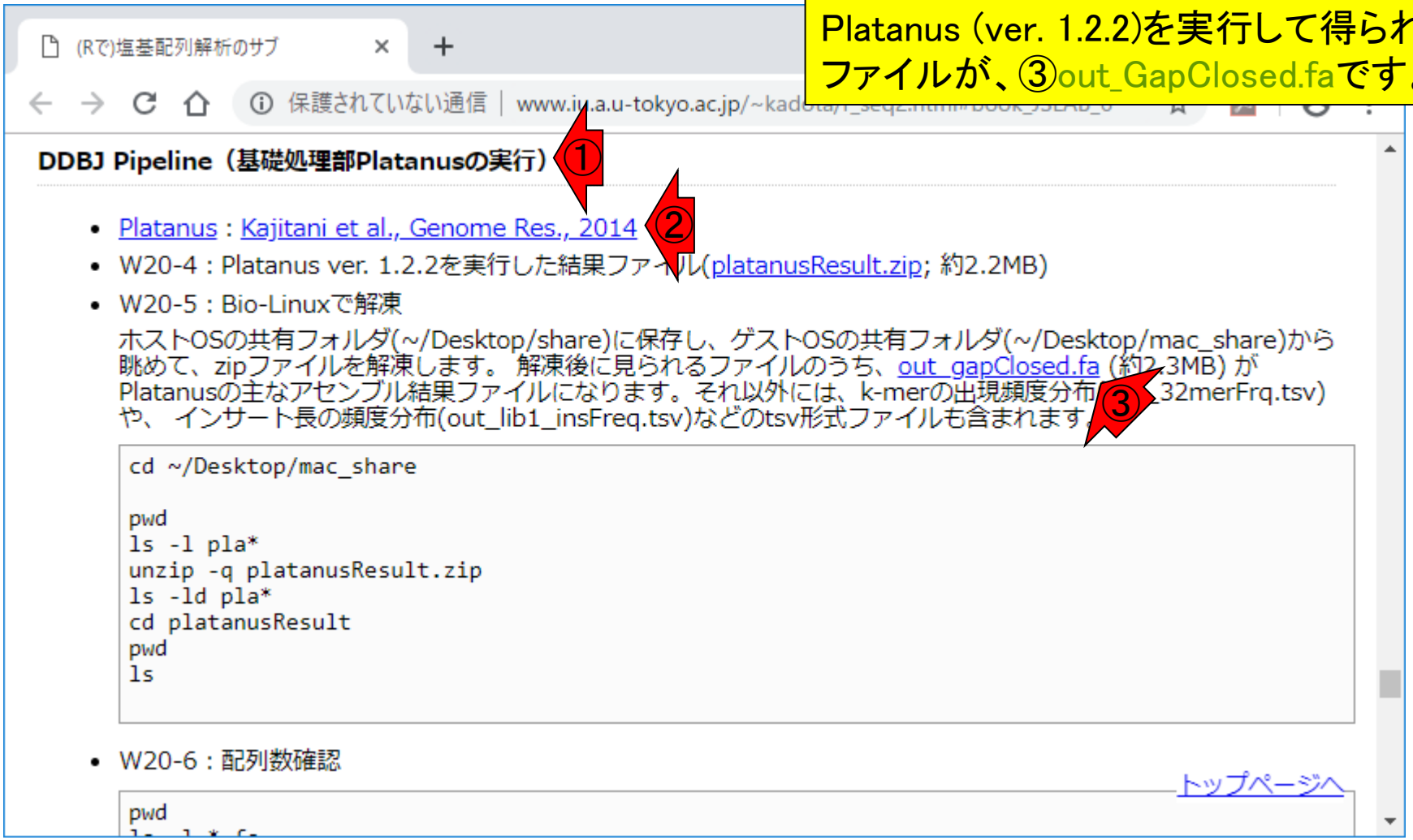
連載第6回

①のあたりまでページ下部に移動したところだが、
たどり着けなくてもよい。



連載第6回

①のあたりまでページ下部に移動したところだが、たどり着けなくてもよい。①DDBJ Pipelineというウェブツール上で、②ゲノムアセンブリプログラムPlatanus (ver. 1.2.2)を実行して得られた主な出力ファイルが、③out_GapClosed.faです。



連載第6回

①のあたりまでページ下部に移動したところだが、たどり着けなくてもよい。①DDBJ Pipelineというウェブツール上で、②ゲノムアセンブリプログラムPlatanus (ver. 1.2.2)を実行して得られた主な出力ファイルが、③out_GapClosed.faです。本科目のページの、④からも取得可能です。out_gapClosed.faをデスクトップにダウンロードしておきましょう。同じものなので、③でも④でもどちらでもよい。

講義日程（2019年度）

1. 2019年04月08日（PC使用）
   - 講義資料PDF(最終更新：2019.04.09)
   - 学会(国外)：ISCB
   - 学会(国内)：JSBi
   - QAサイト：Biostar (Parnell et al., PLoS Comput Biol., 2011)
   - QAサイト：SEQanswers (Li et al., Bioinformatics, 2012)
   - 学習教材：バイオインフォマティクス人材育成のための講習会(平成26-29年度)
   - 学習教材：(Rで)塩基配列解析
   - 学習教材：(Rで)塩基配列解析のサブ
   - RStudio
2. 2019年04月15日（PC使用）
   - 講義資料PDF(最終更新：2019.04.15)
   - (Rで)塩基配列解析
   - (Rで)塩基配列解析のサブ
3. 2019年04月22日（PC使用）
   - 講義資料PDF
   - 課題：docx版とPDF版
   - (Rで)塩基配列解析
   - hoge8.fa (課題用)
   - Bioconductor
   - CRAN
4. 2019年05月13日（PC使用）
   - 講義資料PDF
   - (Rで)塩基配列解析のサブ
   - out_gapClosed.fa (約2.3MB)
   - (Rで)塩基配列解析
   - DFAST：Tanizawa et al., Bioinformatics, 2018
   - DFAST実行結果：genome.fna
   - DFAST実行結果：annotation.gff
   - DFAST実行結果：cds.fna

.a.u-tokyo.ac.jp/~kad

2014

果ファイル(platanusResult.zip; 約2.2MB)

e)に保存し、ゲストOSの共有フォルダ(~/Desktop/mac_share)から
に見られるファイルのうち、out_gapClosed.fa (約2.3MB) が
なります。それ以外には、k-merの出現頻度分布(_32merFrq.tsv)
Freq.tsv)などのtsv形式ファイルも含まれます。

③

pwd

④

4. 2019年05月13日（PC使用）
   - **講義資料PDF**
   - **(Rで)塩基配列解析のサブ**
   - **out_gapClosed.fa** (約2.3MB)
   - **(Rで)塩基配列解析**
   - **DFAST：Tanizawa et al., Bioinformatics, 2018**
   - DFAST実行結果：**genome.fna**
   - DFAST実行結果：**annotation.gff**
   - DFAST実行結果：**cds.fna**

第6回の原稿PDF

①第6回の原稿PDFをクリックすると、赤枠のような感じの原稿が見られます。この中の…

(Rで)塩基配列解析のサブ

保護されていない通信 | www.iu.a.u-tokyo.ac.jp/~kadota/r_seq2.html#book_JSLAB_6

書籍 | 日本乳酸菌学会誌 | 第6回ゲノムアセンブリ

日本乳酸菌学会誌の第6回分です。 Linuxコマンドのリンク
ころにtypoがあります。誤：するであれば、正：する"の"
ング」項目の最後の文章「これらについては、第7回で詳述
ては、"第8回以降"で詳述する予定である。」と読み替えて
まれていない です（2016年4月23日追加）。

①

- 原稿PDF
- ウェブ資料PDF
  - Windows用(2016.06.17版; 約20MB)
  - Macintosh用
- (共有フォルダ設定情報を含む)連載第3回終了時点以
  - Windows用(2015.12.28版; 約2MB)
  - Macintosh用(2015.12.28版; 約2MB)

Linuxコマンド

- bzip2 (bzip2圧縮、および解凍)

解　説

次世代シーケンサーデータの解析手法
第6回 ゲノムアセンブリ

谷澤 靖洋[1,2]、神沼 英里[2*]、中村 保一[2]、清水 謙多郎[3]、門田 幸二[3*]

[1]東京大学大学院新領域創成科学研究科
[2]国立遺伝学研究所生命情報研究センター
[3]東京大学大学院農学生命科学研究科

ゲノムの *de novo* アセンブリ結果に影響を及ぼす主要なパラメータは、*k*-mer（任意の長さ *k* の連続塩基）の *k* 値である。第6回は、Bio-Linux にプレインストールされているゲノムアセンブリ用プログラム Velvet の基本的な利用法、make コマンドを用いたプログラムのインストール法、およびウェブツール DDBJ pipeline の利用法について述べる。複数の異なる *k*-mer で実行した乳酸菌ゲノム *de novo* アセンブリ結果の違い、用いたプログラム間の違い（Velvet vs. Platanus）などを述べる。また、ウェブサイト（R で）塩基配列解析（URL: http://www.iu.a.u-tokyo.ac.jp/~kadota/r_seq.html）中に本連載をまとめた項目（URL: http://www.iu.a.u-tokyo.ac.jp/~kadota/r_seq.html#about_book_JSLAB）が存在する。ウェブ資料（以下、W）や関連ウェブサイトなどのリンク先を効率的に活用してほしい。

第6回の原稿PDF

①第6回の原稿PDFをクリックすると、赤枠のような感じの原稿が見られます。この中の、②45ページの右下あたりに、②配列長によるフィルタリングの項目があります。フィルタリングの必要性については、ここを読んで各自で納得してもらうとして、実際の作業をR上で行います。

(Rで)塩基配列解析のサブ

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

書籍 | 日本乳酸菌学会誌 | 第6回ゲノムアセンブリ

日本乳酸菌学会誌の第6回分です。 Linuxコマンドのリンク先は主に日経BP社様です。 原稿PDFの「はじめに」項目のところにtypoがあります。誤：するであれば、正：する"の"であれば、ですねm(_ _)m。 また、「配列長によるフィルタリング」項目の最後の文章「これらについては、第7回で詳述する予定である。」についてですが、これは「これらについては、"第8回以降"で詳述する予定である。」と読み替えてください。第7回ドラフト原稿作成時点で、これらの内容は含まれていないためです（2016年

①

- 原稿PDF
- ウェブ資料PDF
  - Windows用(2016.
  - Macintosh用
- (共有フォルダ設定情報を
  - Windows用(2015.
  - Macintosh用(2015.

Linuxコマンド

- bzip2 (bzip2圧縮、およ

151, 171, 181, 191）で実行した結果を眺め、主に配列数の観点から、$k=171$ 周辺の結果が一番よさそうだと解釈する。もちろんこのデータの場合は、「真のゲノムサイズは約 2.4MB」だという答えがわかった状態でアセンブリ結果の評価を行っていることになるが、実際には近縁種との比較により妥当と考えられるゲノムサイズを検討する。ここではそのような情報が得られなかったと仮定して「ゲノムサイズ推定」を行い、アセンブリ結果の評価を行う。

ゲノムサイズ推定

ゲノムサイズの推定は、フローサイトメトリー（flow cytometry）という手法を用いて実験的に求めるやり方

②

—45—

配列長によるフィルタリング

③

比較的マイナーな事柄ではあるが、通常下記に示す3つの理由から、アセンブリ結果から短い配列を除外する：

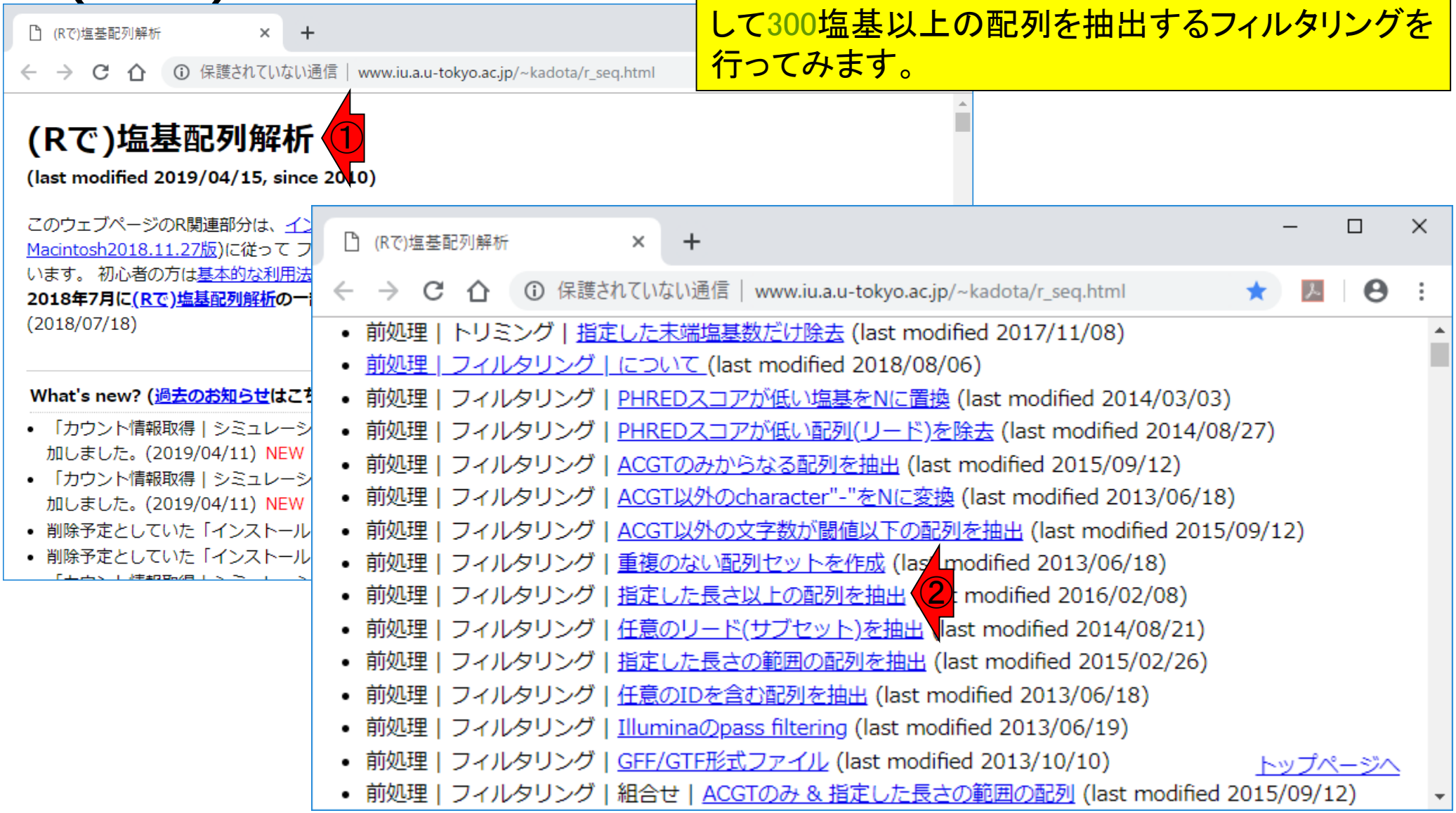
1. MiSeq を含むショートリードの *de novo* アセンブリでは、挿入配列（insertion sequence）やリボソーム RNA 遺伝子領域（rDNA）のような、ゲノム中に複数コピーが散在する反復領域（dispersed repeat）の再現は難しい。配列（コンティグ）がこれらの反復領域部分で分断されてしまうからである。アセンブリ結果に含まれる短いコンティグは、これらの反復領域の一部である場合が多く、その後の解析には大きな影響を及

Contents

- 配列長でフィルタリング（ゲノムアセンブリ結果の後処理）
  - イントロ、本番
- ゲノムアノテーション
  - イントロ、参考文献
  - DFAST実行、課題1
- CDS（タンパク質コード領域の配列；coding sequenceの略）の取得
  - 比較ゲノム解析のイントロ、問題設定
  - 答え（cds.fna）を眺めておく、sequence logos
  - CDS配列取得の成功例、TxDbオブジェクト、CDS座標情報取得
  - CDS配列取得の失敗例、GFF3をExcelで概観
  - 基本テクを駆使して解決する基本戦略、CDSを含む行の抽出、補足説明
  - Excelで確認、リストファイルの作成、基本テクでCDS配列取得を実行
  - ストランド情報つきのリストファイルの作成、例題6のコード解説
  - ストランド情報反映戦略を練り、コードに修正を加えて実行（例題7）

(Rで)塩基配列解析

①サブではなくメインのほうです。②「指定した長さ以上の配列を抽出」。ここの例題1をテンプレートとして、ダウンロード済みのout_gapClosed.faを入力として300塩基以上の配列を抽出するフィルタリングを行ってみます。



フィルタリング

①サブではなくメインのほうです。②「指定した長さ以上の配列を抽出」。ここの例題1をテンプレートとして、ダウンロード済みのout_gapClosed.faを入力として300塩基以上の配列を抽出するフィルタリングを行ってみます。

(Rで)塩基配列解析

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

前処理 | フィルタリング | 指定した長さ以上の配列を抽出

FASTA形式やFASTQ形式ファイルを入力として、指定した配列長以上の配列を抽出するやり方を示します。
「ファイル」－「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピペ。

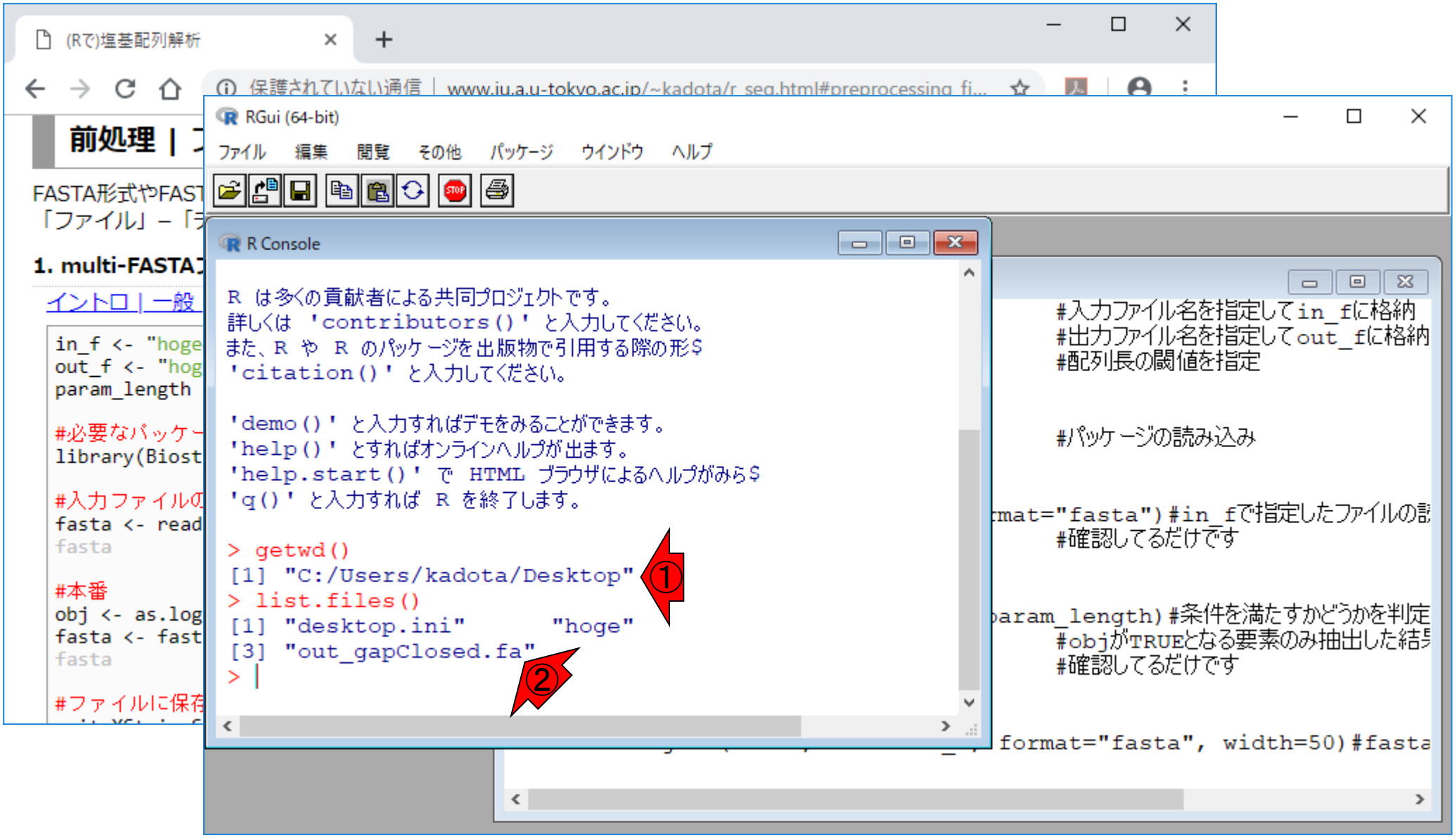
1. multi-FASTAファイル(hoge4.fa)の場合:

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ファイルです。

```
in_f <- "hoge4.fa"                          #入力ファイル名を指定してin_fに格納
out_f <- "hoge1.fasta"                      #出力ファイル名を指定してout_fに格納
param_length <- 50                          #配列長の閾値を指定

#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
fasta                                       #確認してるだけです

#本番
obj <- as.logical(width(fasta) >= param_length)#条件を満たすかどうかを判定した結果をobjに格納
fasta <- fasta[obj]                         #objがTRUEとなる要素のみ抽出した結果をfastaに格納
fasta                                       #確認してるだけです
                                                                    トップページへ
#ファイルに保存
```

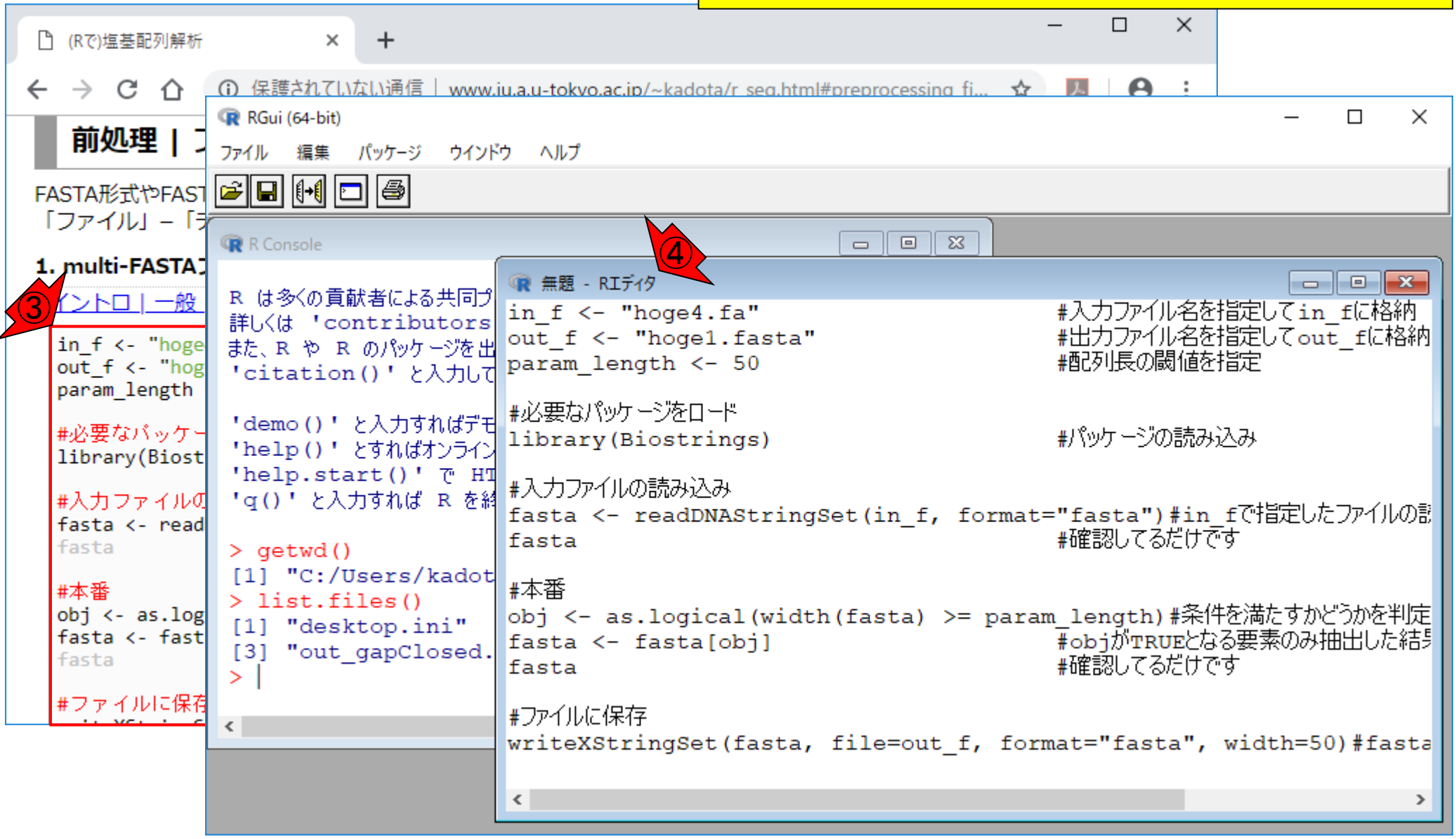
事前準備

①デスクトップ上に保存した、②out_gapClosed.faが見えていることを確認。



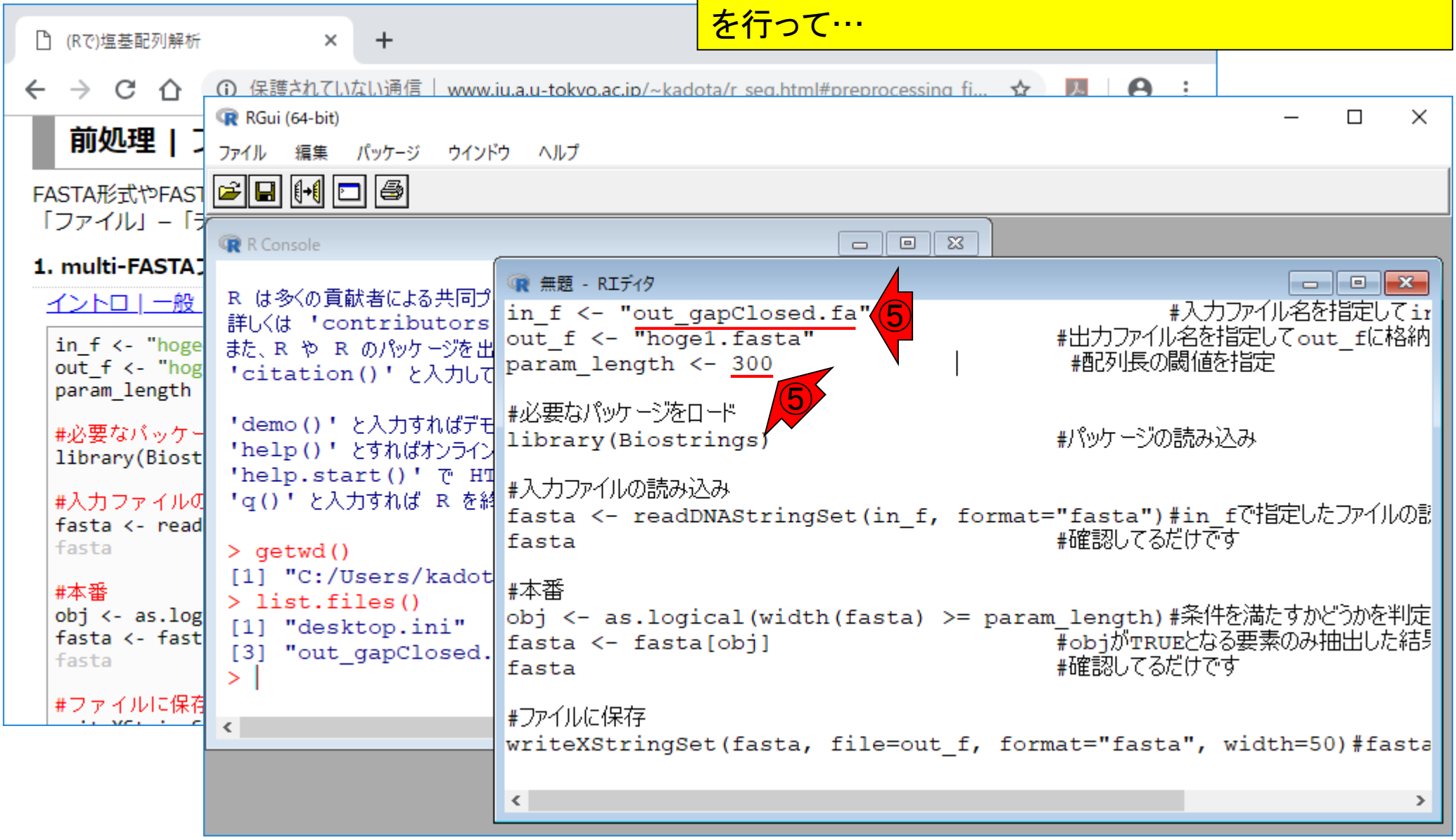
Rエディタで編集

①デスクトップ上に保存した、②out_gapClosed.faが見えていることを確認。③例題1のテンプレートコードを、④Rエディタ上にコピペ。



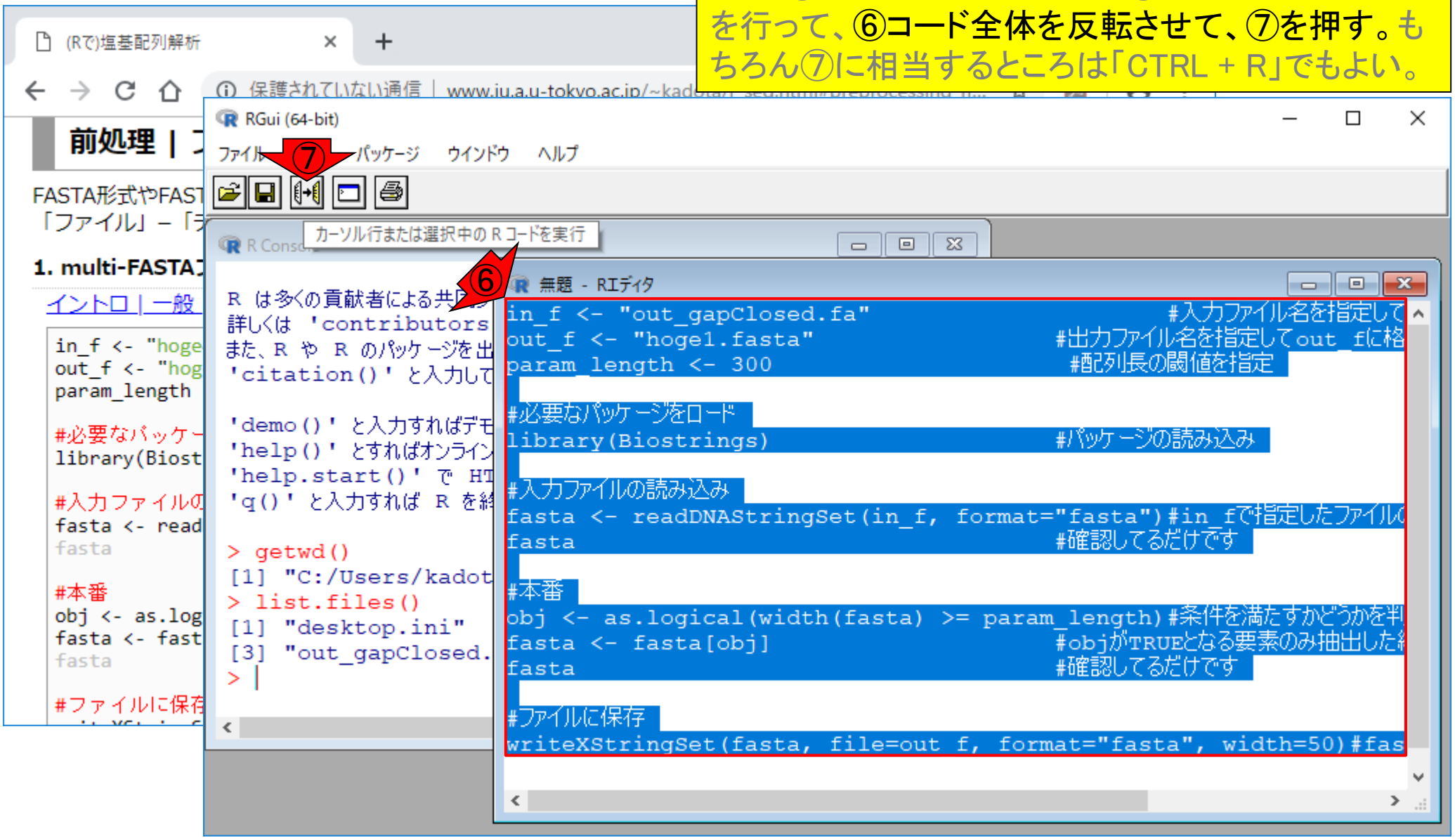
必要最小限の変更

①デスクトップ上に保存した、②out_gapClosed.faが見えていることを確認。③例題1のテンプレートコードを、④Rエディタ上にコピペ。⑤必要最小限の変更を行って…



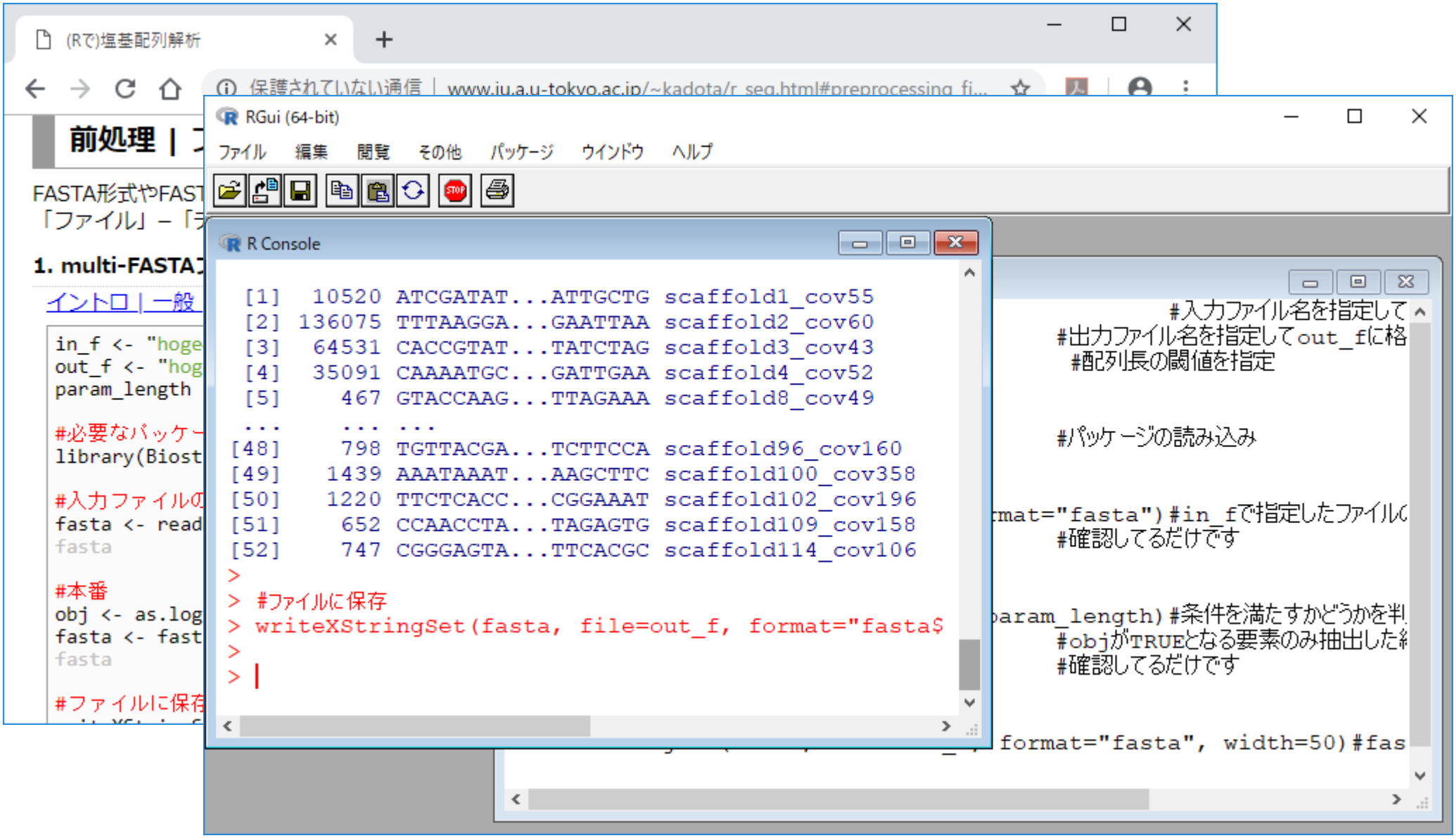
コードを実行

①デスクトップ上に保存した、②out_gapClosed.faが見えていることを確認。③例題1のテンプレートコードを、④Rエディタ上にコピペ。⑤必要最小限の変更を行って、⑥コード全体を反転させて、⑦を押す。もちろん⑦に相当するところは「CTRL + R」でもよい。



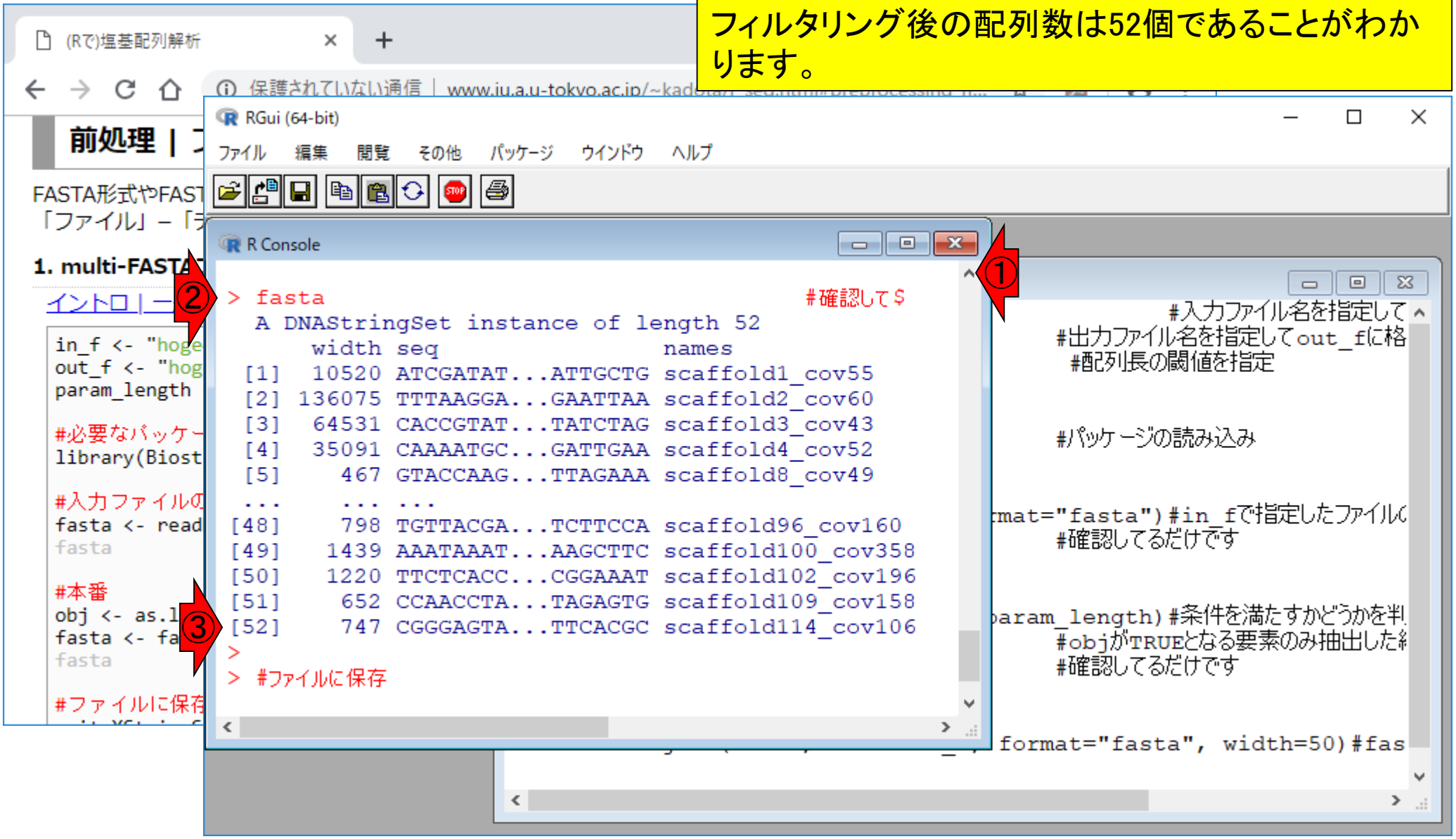
こんな感じになります。

実行結果



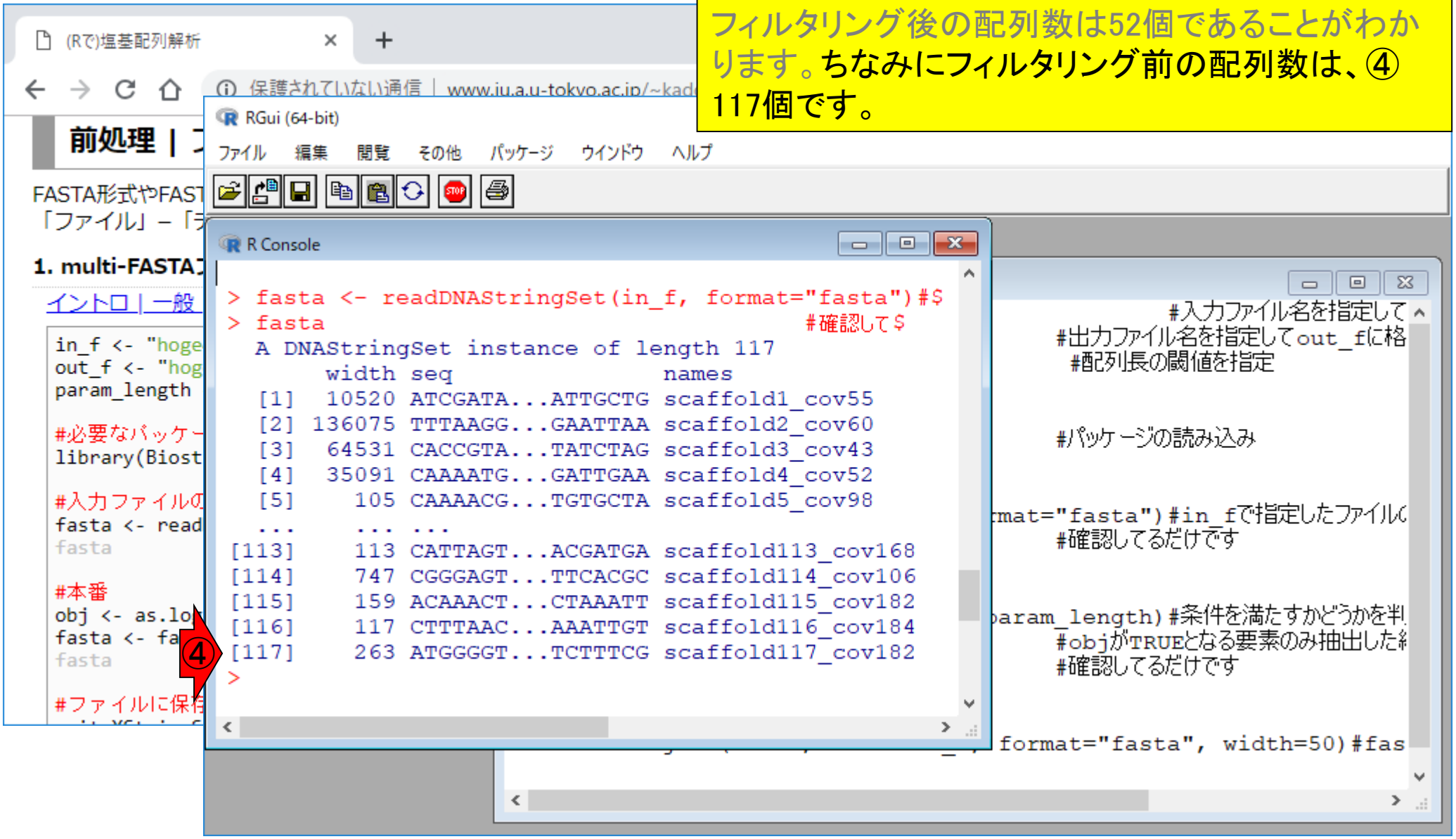
実行後は52配列

こんな感じになります。①を押していって、②fastaオブジェクトが見える位置にしただけです。これが、出力ファイル（hoge1.fasta）の中身に相当します。③フィルタリング後の配列数は52個であることがわかります。



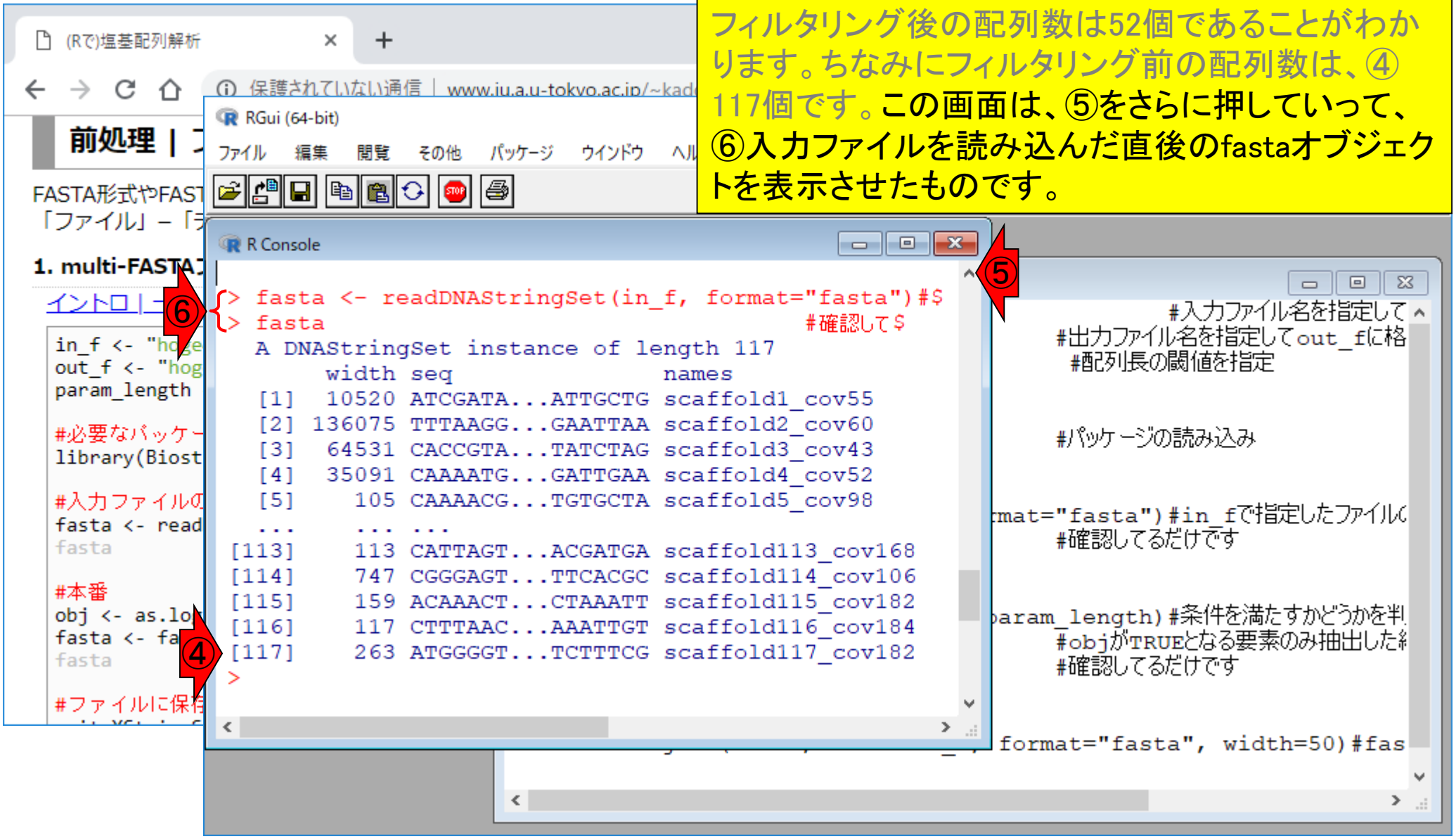
実行前は117配列

こんな感じになります。①を押していって、②fastaオブジェクトが見える位置にしただけです。これが、出力ファイル（hoge1.fasta）の中身に相当します。③フィルタリング後の配列数は52個であることがわかります。ちなみにフィルタリング前の配列数は、④117個です。



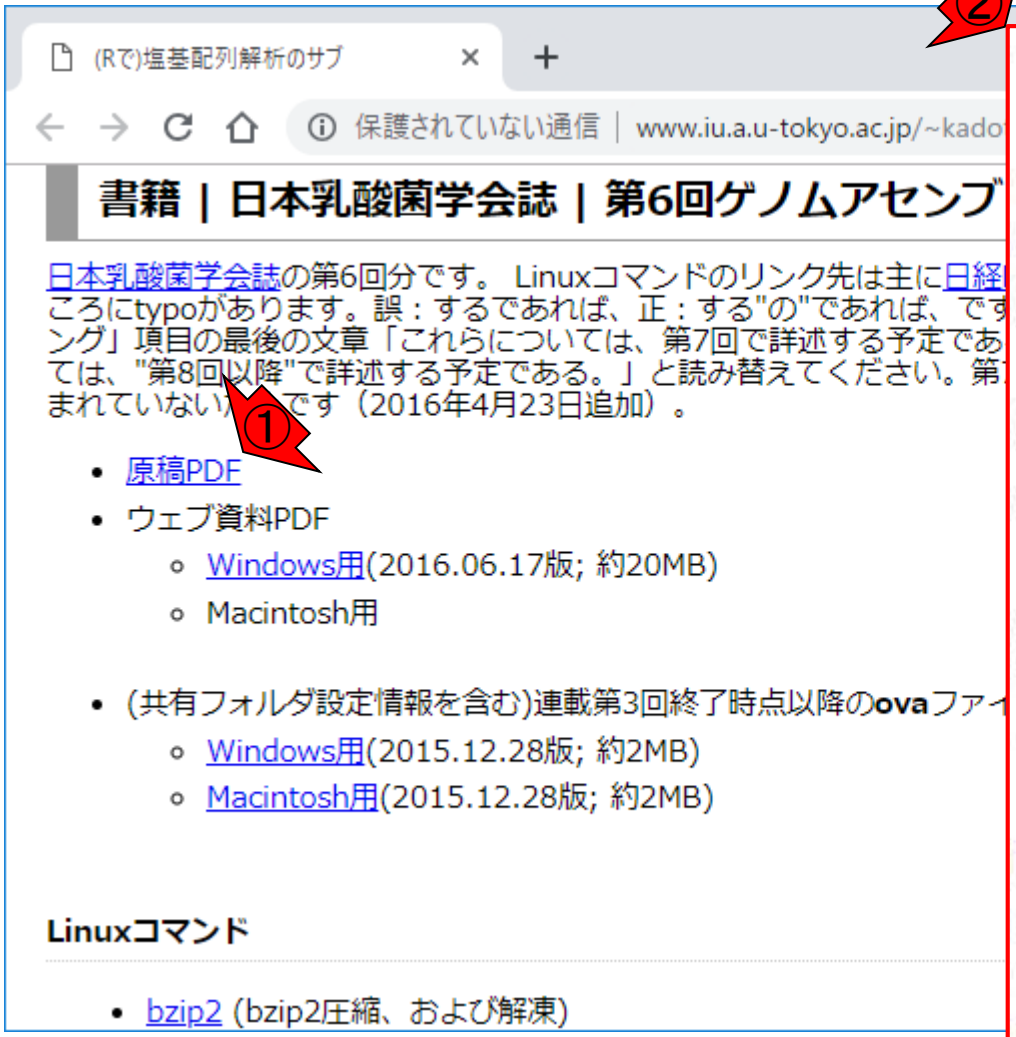
実行前は117配列

こんな感じになります。①を押していって、②fastaオブジェクトが見える位置にしただけです。これが、出力ファイル（hoge1.fasta）の中身に相当します。③フィルタリング後の配列数は52個であることがわかります。ちなみにフィルタリング前の配列数は、④117個です。この画面は、⑤をさらに押していって、⑥入力ファイルを読み込んだ直後のfastaオブジェクトを表示させたものです。



第6回の原稿PDF

①第6回の原稿PDFの、②p50の左上あたりのスクショ。

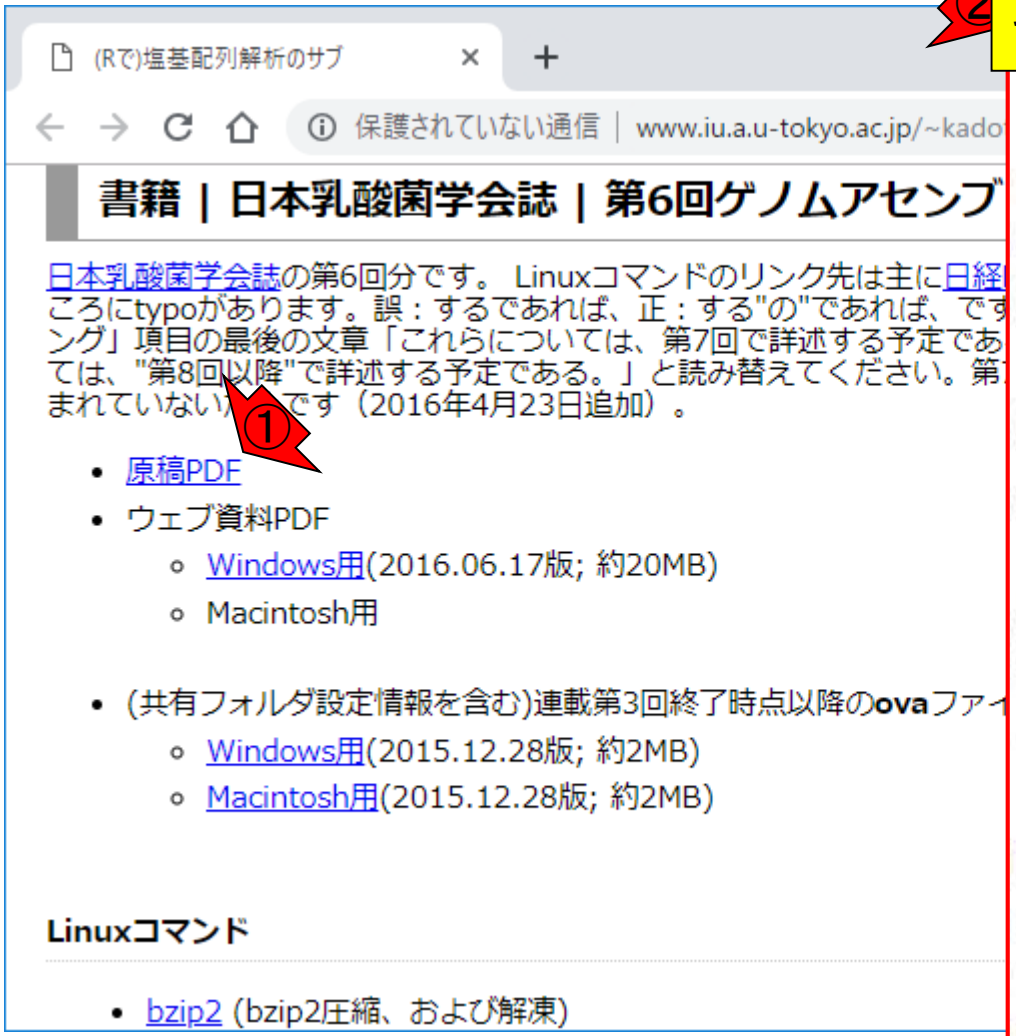


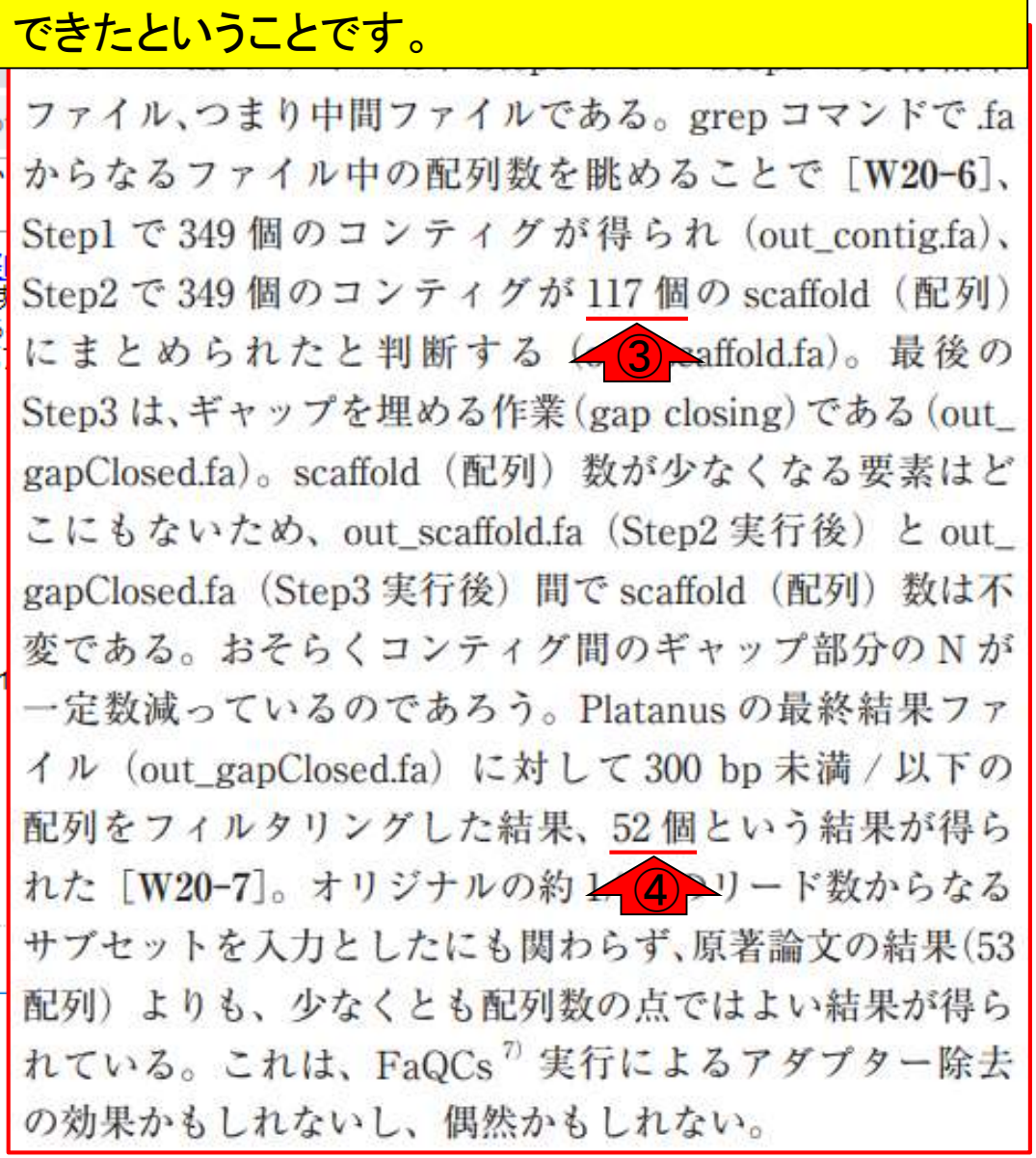
②

の 4 つの .fa ファイルは、Step1 および Step2 の実行結果ファイル、つまり中間ファイルである。grep コマンドで .fa からなるファイル中の配列数を眺めることで［**W20-6**］、Step1 で 349 個のコンティグが得られ（out_contig.fa）、Step2 で 349 個のコンティグが 117 個の scaffold（配列）にまとめられたと判断する（out_scaffold.fa）。最後の Step3 は、ギャップを埋める作業（gap closing）である（out_gapClosed.fa）。scaffold（配列）数が少なくなる要素はどこにもないため、out_scaffold.fa（Step2 実行後）と out_gapClosed.fa（Step3 実行後）間で scaffold（配列）数は不変である。おそらくコンティグ間のギャップ部分の N が一定数減っているのであろう。Platanus の最終結果ファイル（out_gapClosed.fa）に対して 300 bp 未満 / 以下の配列をフィルタリングした結果、52 個という結果が得られた［**W20-7**］。オリジナルの約 1/10 のリード数からなるサブセットを入力としたにも関わらず、原著論文の結果（53 配列）よりも、少なくとも配列数の点ではよい結果が得られている。これは、FaQCs[7] 実行によるアダプター除去の効果かもしれないし、偶然かもしれない。

第6回の原稿PDF

①第6回の原稿PDFの、②p50の左上あたりのスクショ。③117個の配列から、300塩基以上の配列のみ抽出すると④52個になるという結果を自力で再現できたということです。





ファイル、つまり中間ファイルである。grep コマンドで .fa からなるファイル中の配列数を眺めることで［W20-6］、Step1 で 349 個のコンティグが得られ（out_contig.fa）、Step2 で 349 個のコンティグが 117 個の scaffold（配列）にまとめられたと判断する（③affold.fa）。最後の Step3 は、ギャップを埋める作業（gap closing）である（out_gapClosed.fa）。scaffold（配列）数が少なくなる要素はどこにもないため、out_scaffold.fa（Step2 実行後）と out_gapClosed.fa（Step3 実行後）間で scaffold（配列）数は不変である。おそらくコンティグ間のギャップ部分の N が一定数減っているのであろう。Platanus の最終結果ファイル（out_gapClosed.fa）に対して 300 bp 未満 / 以下の配列をフィルタリングした結果、52 個という結果が得られた［W20-7］。オリジナルの約 1[illegible]リード数からなるサブセットを入力としたにも関わらず、原著論文の結果（53 配列）よりも、少なくとも配列数の点ではよい結果が得られている。これは、FaQCs[7] 実行によるアダプター除去の効果かもしれないし、偶然かもしれない。

④

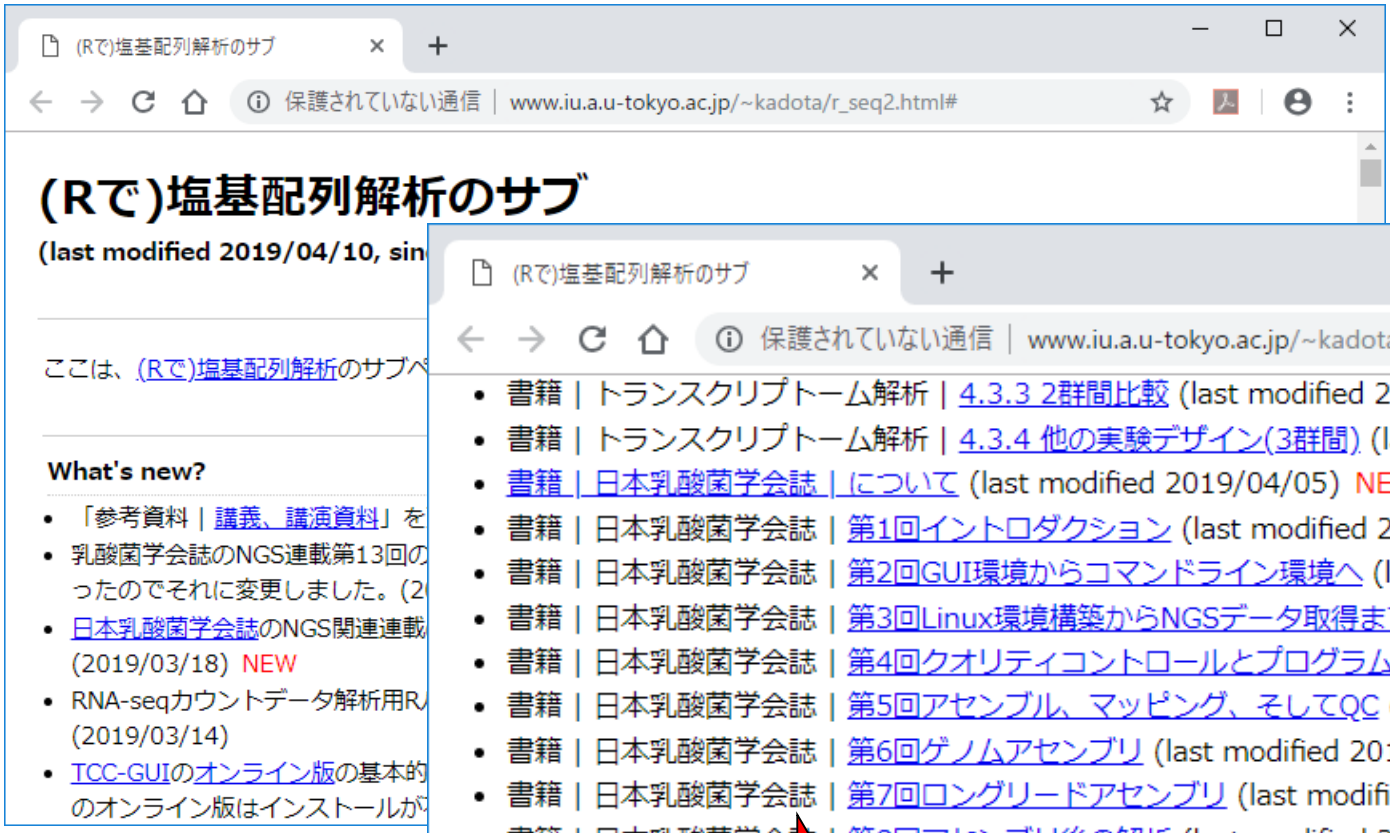
Contents

- 配列長でフィルタリング（ゲノムアセンブリ結果の後処理）
  - イントロ、本番
- ゲノムアノテーション
  - イントロ、参考文献
  - DFAST実行、課題1
- CDS（タンパク質コード領域の配列；coding sequenceの略）の取得
  - 比較ゲノム解析のイントロ、問題設定
  - 答え（cds.fna）を眺めておく、sequence logos
  - CDS配列取得の成功例、TxDbオブジェクト、CDS座標情報取得
  - CDS配列取得の失敗例、GFF3をExcelで概観
  - 基本テクを駆使して解決する基本戦略、CDSを含む行の抽出、補足説明
  - Excelで確認、リストファイルの作成、基本テクでCDS配列取得を実行
  - ストランド情報つきのリストファイルの作成、例題6のコード解説
  - ストランド情報反映戦略を練り、コードに修正を加えて実行（例題7）

スライドを見るだけ

ゲノムアノテーションについては、①第9回で述べています。

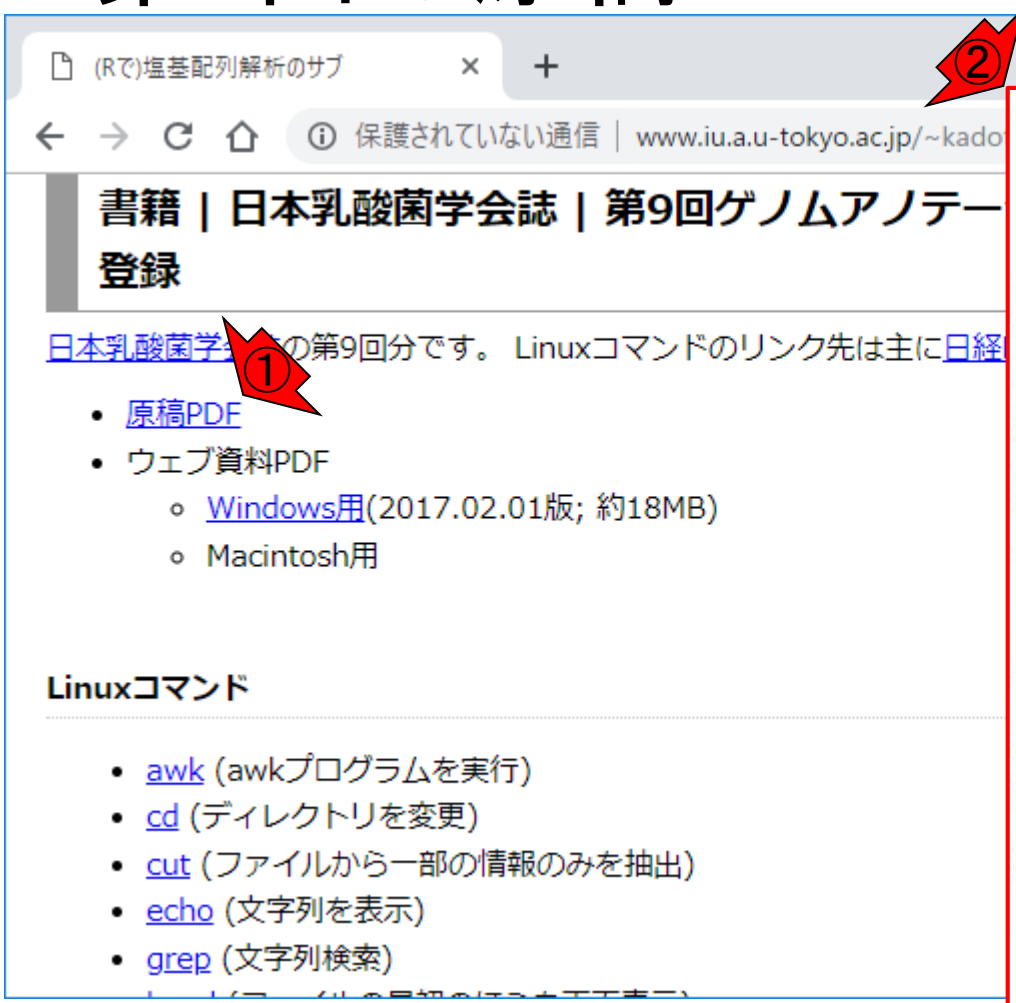
ゲノムアノテーション

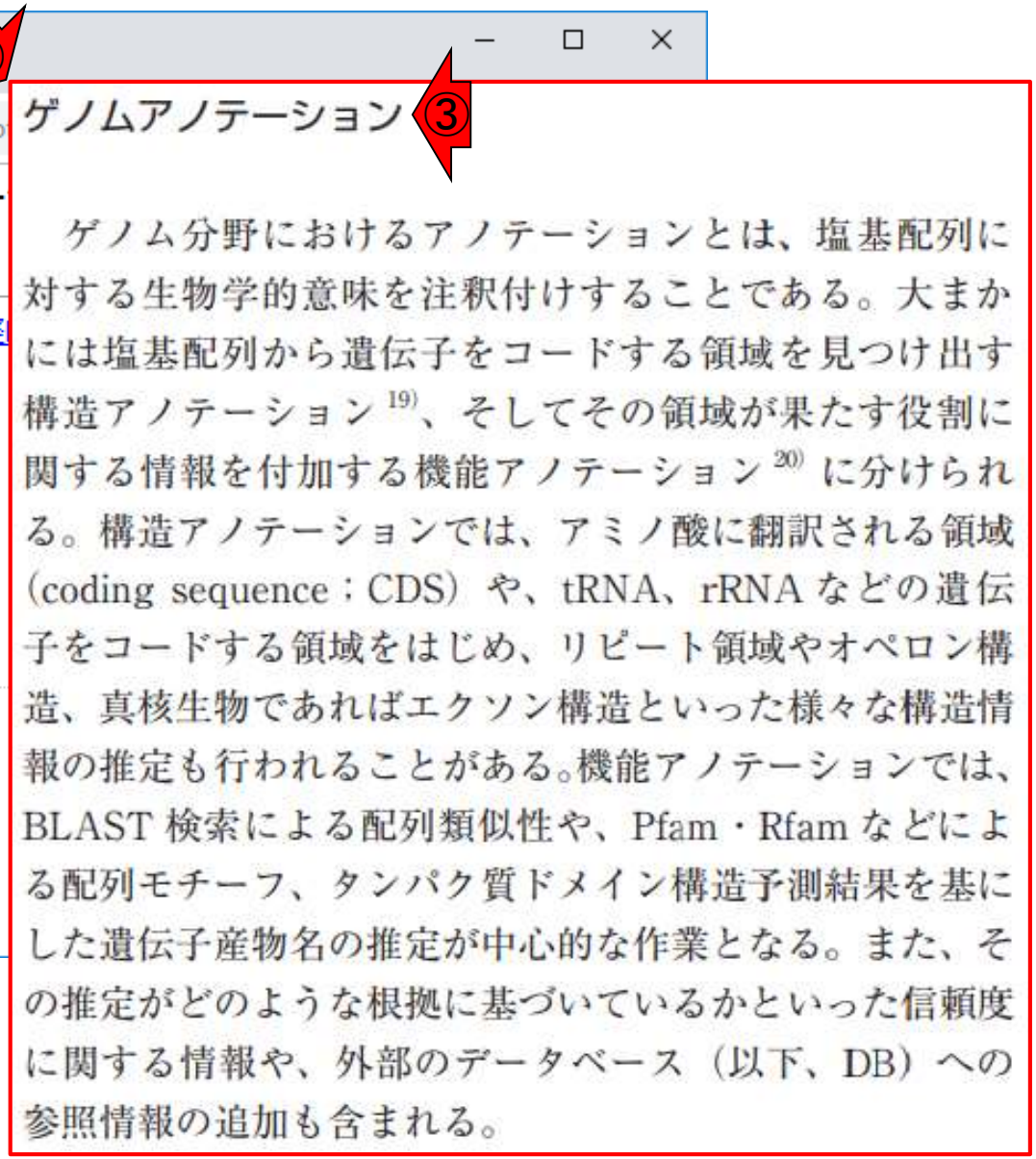


スライドを見るだけ

①第9回の原稿PDFの、②p4の右側のスクショ。③ゲノムアノテーションに関する解説があります。

第9回の原稿PDF

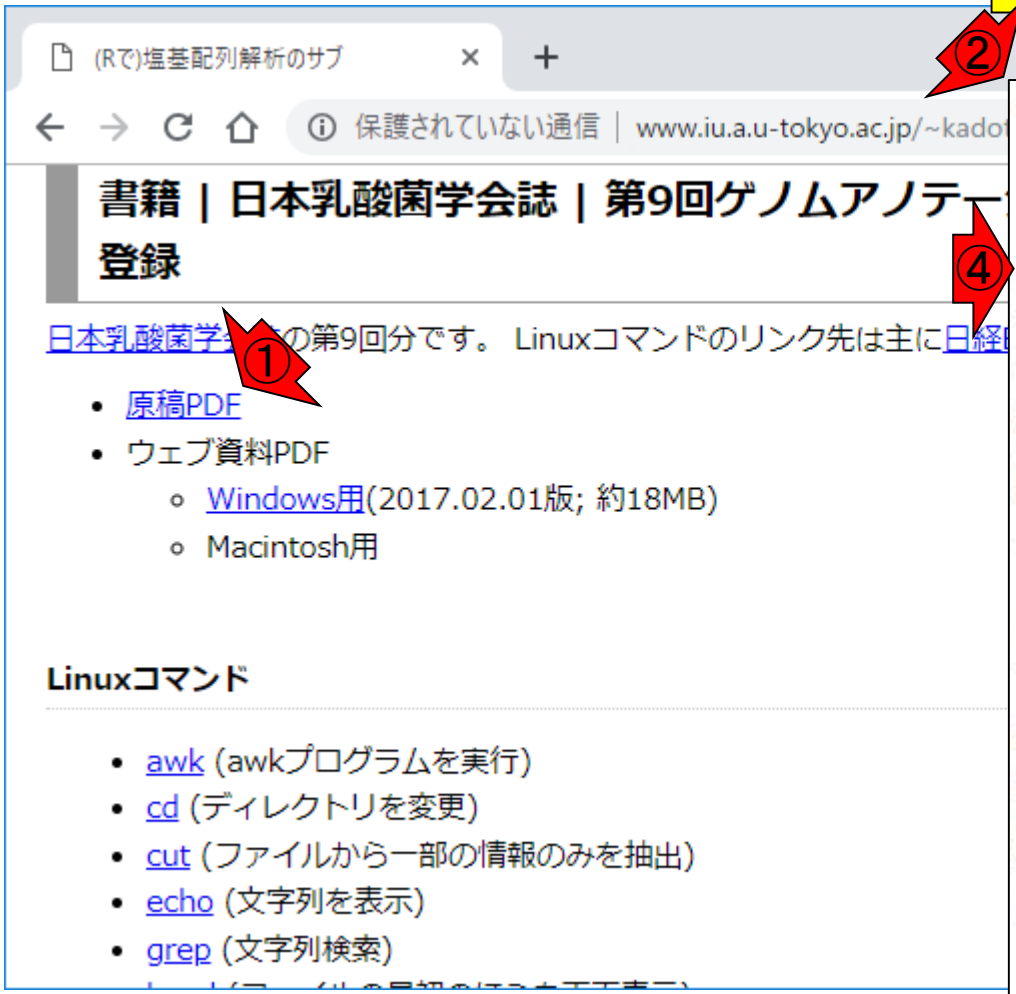


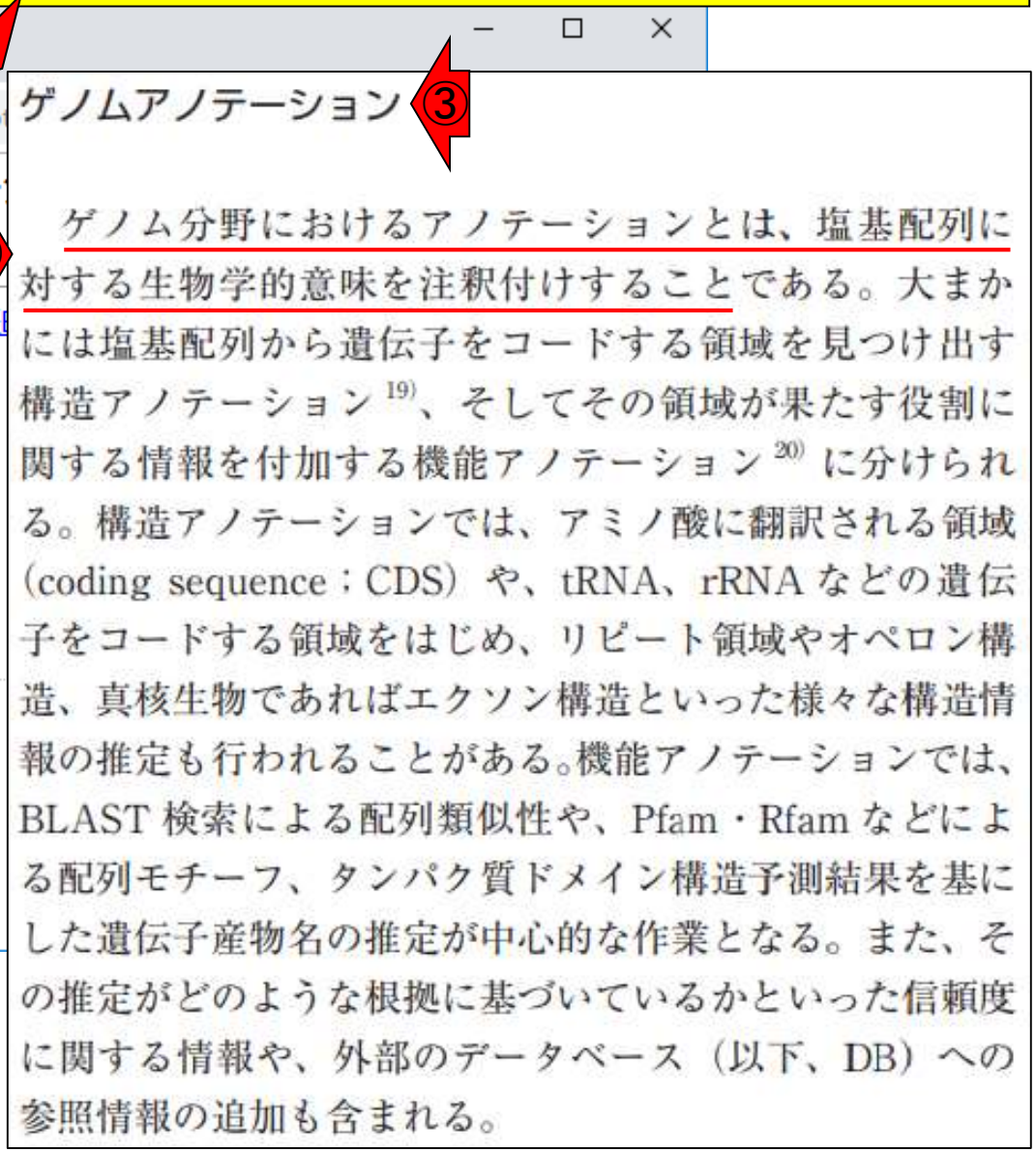


スライドを見るだけ

ゲノムアノテーション

①第9回の原稿PDFの、②p4の右側のスクショ。③ゲノムアノテーションに関する解説があります。④が概念的な説明。

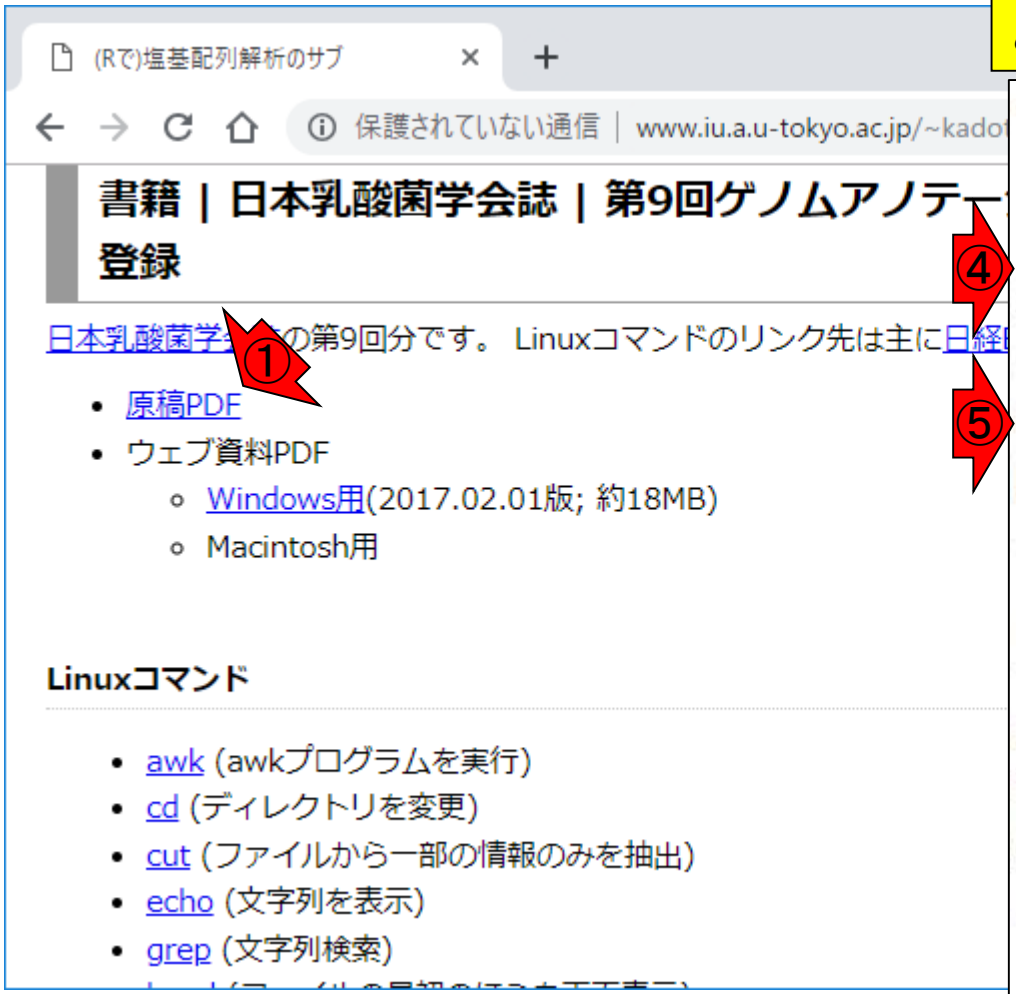




スライドを見るだけ

ゲノムアノテーション

①第9回の原稿PDFの、②p4の右側のスクショ。③ゲノムアノテーションに関する解説があります。④が概念的な説明。大まかには、⑤構造アノテーションと⑥機能アノテーションに分けられる。


(Rで)塩基配列解析のサブ

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

書籍 | 日本乳酸菌学会誌 | 第9回ゲノムアノテー
登録

日本乳酸菌学[illegible]の第9回分です。 Linuxコマンドのリンク先は主に日経[illegible]

- 原稿PDF
- ウェブ資料PDF
  - Windows用(2017.02.01版; 約18MB)
  - Macintosh用

Linuxコマンド

- awk (awkプログラムを実行)
- cd (ディレクトリを変更)
- cut (ファイルから一部の情報のみを抽出)
- echo (文字列を表示)
- grep (文字列検索)

①

④

⑤

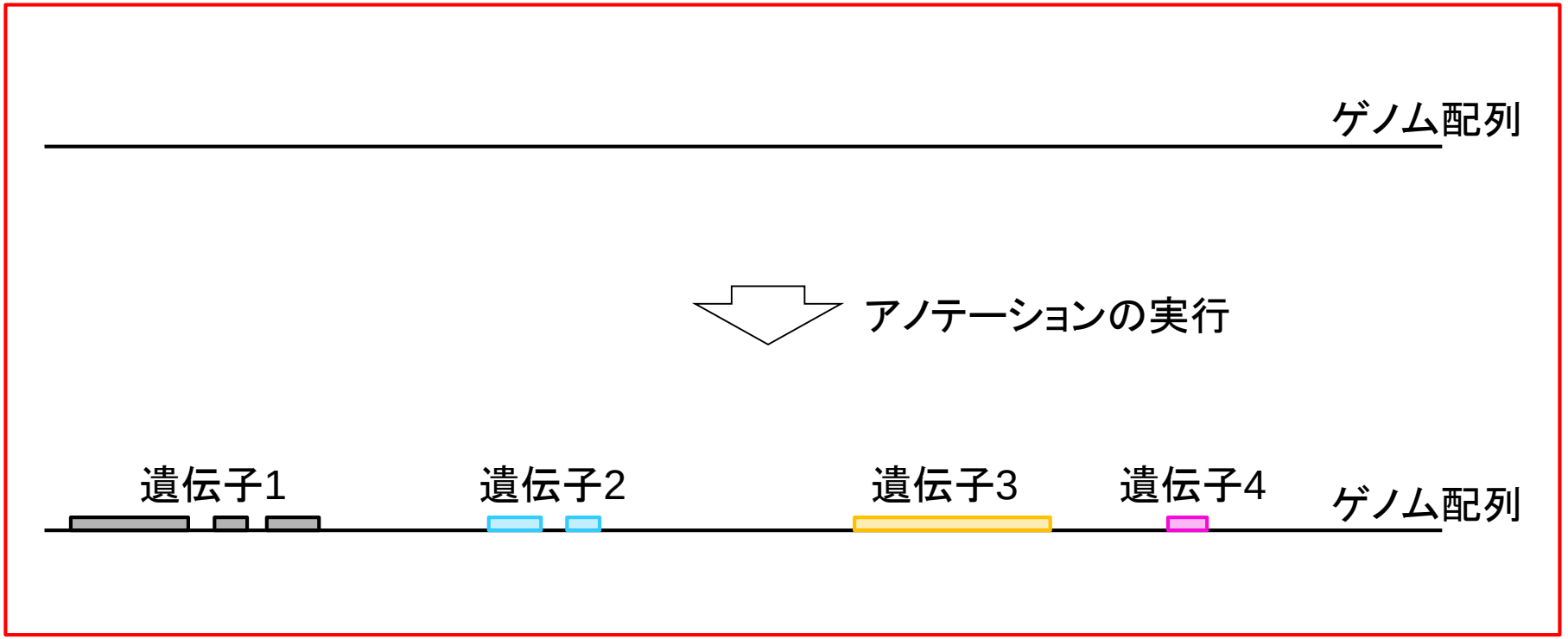
⑥

ゲノムアノテーション

ゲノム分野におけるアノテーションとは、塩基配列に対する生物学的意味を注釈付けすることである。大まかには塩基配列から遺伝子をコードする領域を見つけ出す構造アノテーション[19]、そしてその領域が果たす役割に関する情報を付加する機能アノテーション[20]に分けられる。構造アノテーションでは、アミノ酸に[illegible]される領域（coding sequence：CDS）や、tRNA、rRNA などの遺伝子をコードする領域をはじめ、リピート領域やオペロン構造、真核生物であればエクソン構造といった様々な構造情報の推定も行われることがある。機能アノテーションでは、BLAST 検索による配列類似性や、Pfam・Rfam などによる配列モチーフ、タンパク質ドメイン構造予測結果を基にした遺伝子産物名の推定が中心的な作業となる。また、その推定がどのような根拠に基づいているかといった信頼度に関する情報や、外部のデータベース（以下、DB）への参照情報の追加も含まれる。

構造アノテーション

構造アノテーションについては、大まかには赤枠をやるという理解でよい。実験データ（RNA-seqデータをゲノム上にマッピングした情報）や、遺伝子領域予測プログラムの利用など手段は多様。原核生物（prokaryotes）の場合は、主に後者だと思います。



Contents

- 配列長でフィルタリング（ゲノムアセンブリ結果の後処理）
  - イントロ、本番
- ゲノムアノテーション
  - イントロ、参考文献
  - DFAST実行、課題1
- CDS（タンパク質コード領域の配列；coding sequenceの略）の取得
  - 比較ゲノム解析のイントロ、問題設定
  - 答え（cds.fna）を眺めておく、sequence logos
  - CDS配列取得の成功例、TxDbオブジェクト、CDS座標情報取得
  - CDS配列取得の失敗例、GFF3をExcelで概観
  - 基本テクを駆使して解決する基本戦略、CDSを含む行の抽出、補足説明
  - Excelで確認、リストファイルの作成、基本テクでCDS配列取得を実行
  - ストランド情報つきのリストファイルの作成、例題6のコード解説
  - ストランド情報反映戦略を練り、コードに修正を加えて実行（例題7）

参考文献(全般)

ゲノムアノテーションに関するガイドライン系論文。様々な生物種に言及されているようです。PMID: 30943207

PLoS Comput Biol. 2019 Apr 3;15(4):e1006682. doi: 10.1371/journal.pcbi.1006682. eCollection 2019 Apr.

A quick guide for student-driven community genome annotation.

Hosmani PS[1], Shippy T[2], Miller S[2], Benoit JB[3], Munoz-Torres M[4,5], Flores-Gonzalez M[1], Mueller LA[1], Wiersma-Koch H[6], D'Elia T[6], Brown SJ[2], Saha S[1].

Author information

Abstract

High quality gene models are necessary to expand the molecular and genetic tools available for a target organism, but these are available for only a handful of model organisms that have undergone extensive curation and experimental validation over the course of many years. The majority of gene models present in biological databases today have been identified in draft genome assemblies using automated annotation pipelines that are frequently based on orthologs from distantly related model organisms and usually have minor or major errors. Manual curation is time consuming and often requires substantial expertise, but is instrumental in improving gene model structure and identification. Manual annotation may seem to be a daunting and cost-prohibitive task for small research communities but involving undergraduates in community genome annotation consortiums can be mutually beneficial for both education and improved genomic resources. We outline a workflow for efficient manual annotation driven by a team of primarily undergraduate annotators. This model can be scaled to large teams and includes quality control processes through incremental evaluation. Moreover, it gives students an opportunity to increase their understanding of genome biology and to participate in scientific research in collaboration with peers and senior researchers at multiple institutions.

参考文献(原核生物)

バクテリア用のプログラムDFAST。PMID: 29106469。
①著者はDDBJのヒト。

Bioinformatics. 2018 Mar 15;34(6):1037-1039. doi: 10.1093/bioinformatics/btx713.

DFAST: a flexible prokaryotic genome annotation pipeline for faster genome publication.

Tanizawa Y[1], Fujisawa T[1], Nakamura Y[1].

① ← Nakamura Y[1]

Author information

Abstract

SUMMARY: We developed a prokaryotic genome annotation pipeline, DFAST, that also supports genome submission to public sequence databases. DFAST was originally started as an on-line annotation server, and to date, over 7000 jobs have been processed since its first launch in 2016. Here, we present a newly implemented background annotation engine for DFAST, which is also available as a standalone command-line program. The new engine can annotate a typical-sized bacterial genome within 10 min, with rich information such as pseudogenes, translation exceptions and orthologous gene assignment between given reference genomes. In addition, the modular framework of DFAST allows users to customize the annotation workflow easily and will also facilitate extensions for new functions and incorporation of new tools in the future.

AVAILABILITY AND IMPLEMENTATION: The software is implemented in Python 3 and runs in both Python 2.7 and 3.4-on Macintosh and Linux systems. It is freely available at https://github.com/nigyta/dfast_core/under the GPLv3 license with external binaries bundled in the software distribution. An on-line version is also available at https://dfast.nig.ac.jp/.

CONTACT: yn@nig.ac.jp.

SUPPLEMENTARY INFORMATION: Supplementary data are available at Bioinformatics online.

参考文献(原核生物

バクテリア用のプログラムDFAST。PMID: 29106469。
①著者はDDBJのヒト。一般に、ゲノム配列決定はアノテーションまで行い、DDBJ/ENA/GenBankに登録しないと論文として受理されないが、②DFASTはDDBJへの登録もサポートしています。筆頭著者の谷澤先生は、2019年5月20日の2限(10:25-12:10)の大学院講義「情報生命工学」で話される予定です。

Bioinformatics. 2018 Mar 15;34(6):1037-1039. doi: 10.1093/bioinformatics/bt

DFAST: a flexible prokaryotic genome anno
publication.

Tanizawa Y[1], Fujisawa T[1], Nakamura Y[1].

① Author information

Abstract

② **SUMMARY:** We developed a prokaryotic genome annotation pipeline, DFAST, that also supports genome submission to public sequence databases. DFAST was originally started as an on-line annotation server, and to date, over 7000 jobs have been processed since its first launch in 2016. Here, we present a newly implemented background annotation engine for DFAST, which is also available as a standalone command-line program. The new engine can annotate a typical-sized bacterial genome within 10 min, with rich information such as pseudogenes, translation exceptions and orthologous gene assignment between given reference genomes. In addition, the modular framework of DFAST allows users to customize the annotation workflow easily and will also facilitate extensions for new functions and incorporation of new tools in the future.

AVAILABILITY AND IMPLEMENTATION: The software is implemented in Python 3 and runs in both Python 2.7 and 3.4-on Macintosh and Linux systems. It is freely available at https://github.com/nigyta/dfast_core/under the GPLv3 license with external binaries bundled in the software distribution. An on-line version is also available at https://dfast.nig.ac.jp/.

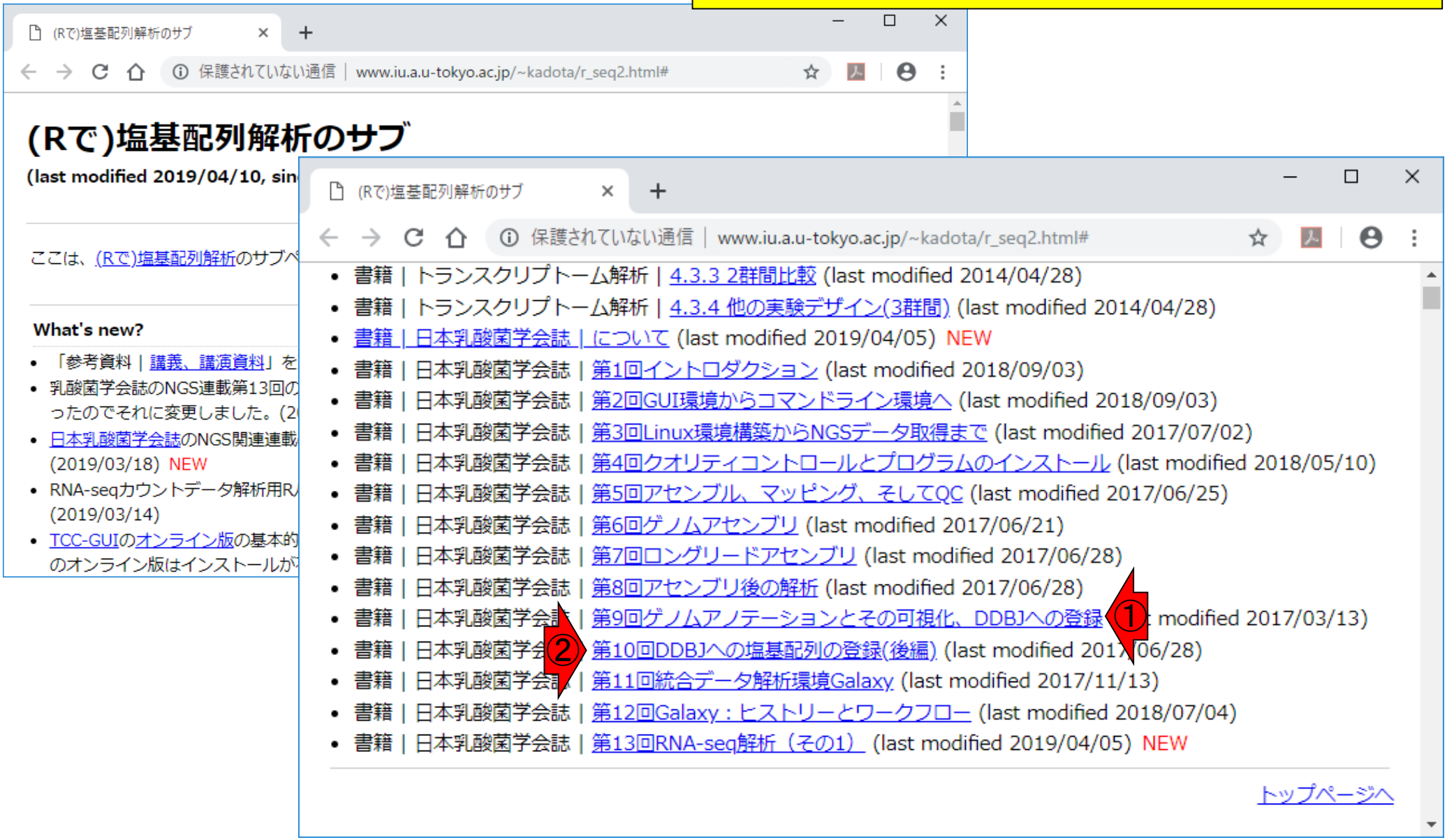
CONTACT: yn@nig.ac.jp.

SUPPLEMENTARY INFORMATION: Supplementary data are available at Bioinformatics online.

スライドを見るだけ

DFAST→DDBJ

①第9回でDFAST(の昔のバージョン)を実行して、
②第10回でその結果をDDBJに登録して公開されるまでを実際に私自身で行いましたが、楽でした。



スライドを見るだけ

参考文献(原核生物

①オンライン版があります。手元のアセンブリ結果ファイル(out_gapClosed.fa)を入力として、実際にアノテーションを行ってみます。

Bioinformatics. 2018 Mar 15;34(6):1037-1039. doi: 10.1093/bioinformatics/btx713.

DFAST: a flexible prokaryotic genome annotation pipeline for faster genome publication.

Tanizawa Y[1], Fujisawa T[1], Nakamura Y[1].

⊕ Author information

Abstract

SUMMARY: We developed a prokaryotic genome annotation pipeline, DFAST, that also supports genome submission to public sequence databases. DFAST was originally started as an on-line annotation server, and to date, over 7000 jobs have been processed since its first launch in 2016. Here, we present a newly implemented background annotation engine for DFAST, which is also available as a standalone command-line program. The new engine can annotate a typical-sized bacterial genome within 10 min, with rich information such as pseudogenes, translation exceptions and orthologous gene assignment between given reference genomes. In addition, the modular framework of DFAST allows users to customize the annotation workflow easily and will also facilitate extensions for new functions and incorporation of new tools in the future.

AVAILABILITY AND IMPLEMENTATION: The software is implemented in Python 3 and runs in both Python 2.7 and 3.4-on Macintosh and Linux systems. It is freely available at https://github.com/nigyta/dfast_core/under the GPLv3 license with external binaries bundled in the software distribution. An on-line version is also available at https://dfast.nig.ac.jp/.

①

CONTACT: yn@nig.ac.jp.

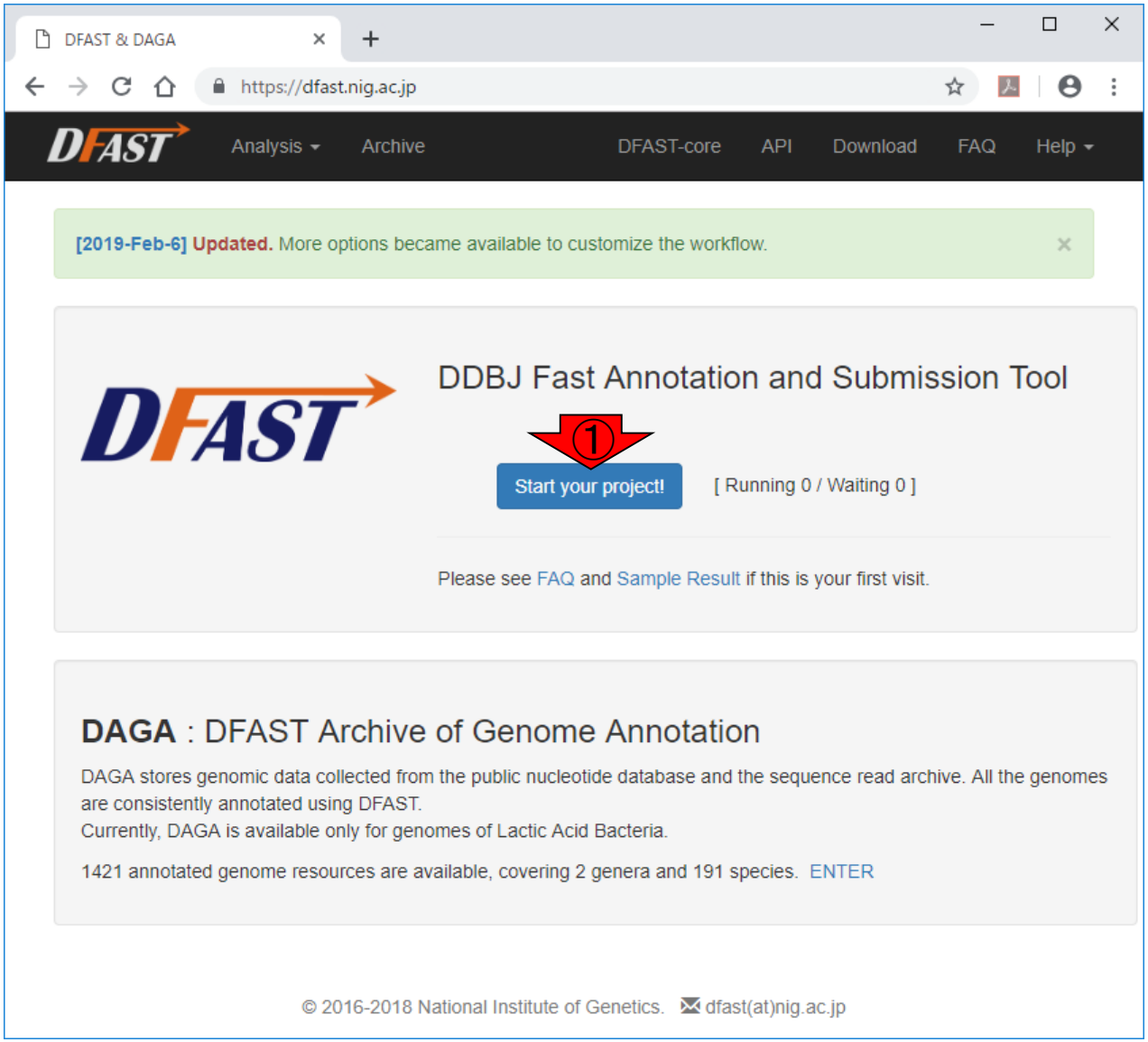
SUPPLEMENTARY INFORMATION: Supplementary data are available at Bioinformatics online.

Contents

- 配列長でフィルタリング（ゲノムアセンブリ結果の後処理）
  - イントロ、本番
- ゲノムアノテーション
  - イントロ、参考文献
  - DFAST実行、課題1
- CDS（タンパク質コード領域の配列；coding sequenceの略）の取得
  - 比較ゲノム解析のイントロ、問題設定
  - 答え（cds.fna）を眺めておく、sequence logos
  - CDS配列取得の成功例、TxDbオブジェクト、CDS座標情報取得
  - CDS配列取得の失敗例、GFF3をExcelで概観
  - 基本テクを駆使して解決する基本戦略、CDSを含む行の抽出、補足説明
  - Excelで確認、リストファイルの作成、基本テクでCDS配列取得を実行
  - ストランド情報つきのリストファイルの作成、例題6のコード解説
  - ストランド情報反映戦略を練り、コードに修正を加えて実行（例題7）

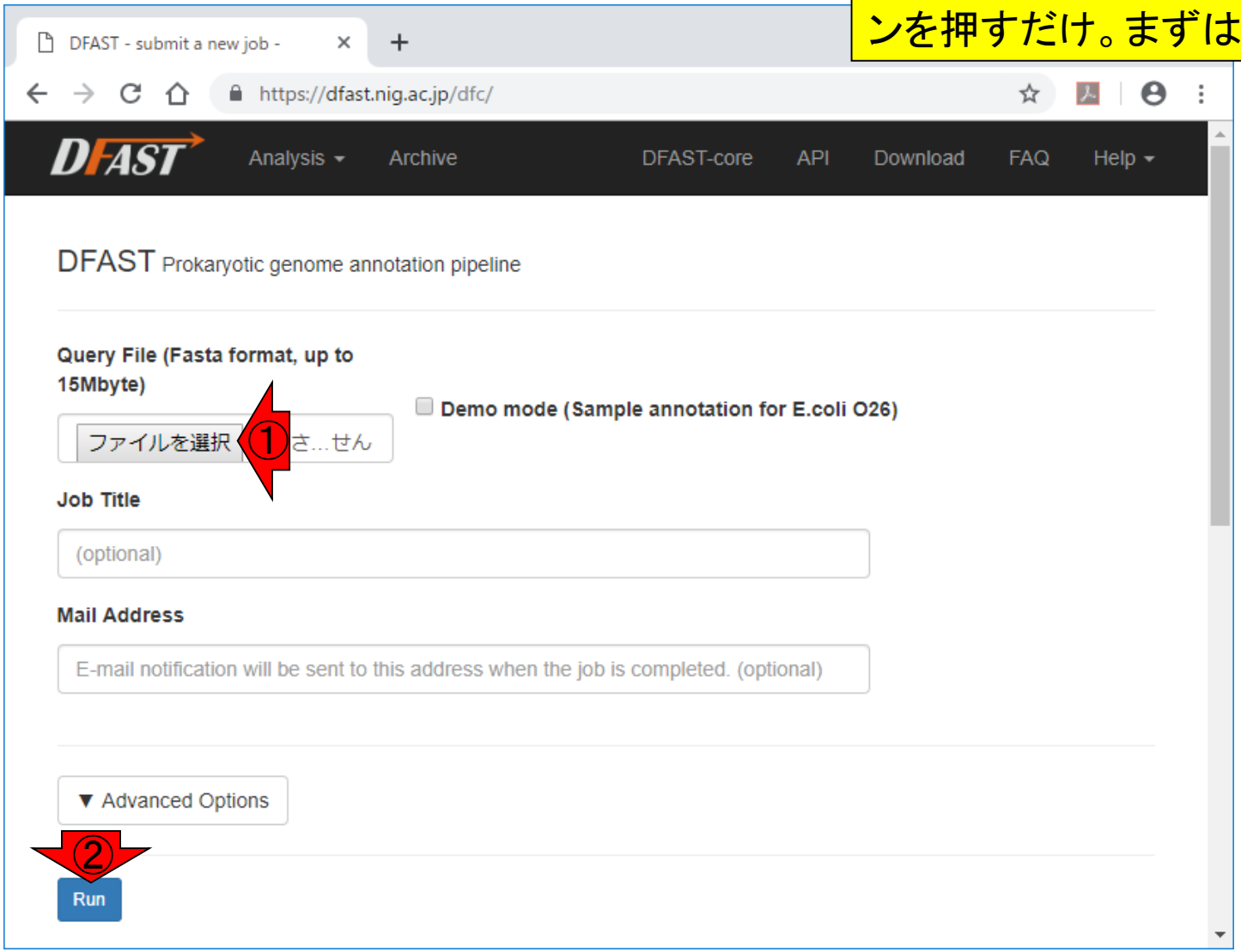
DFASTの初期画面。①をクリック。

DFAST実行



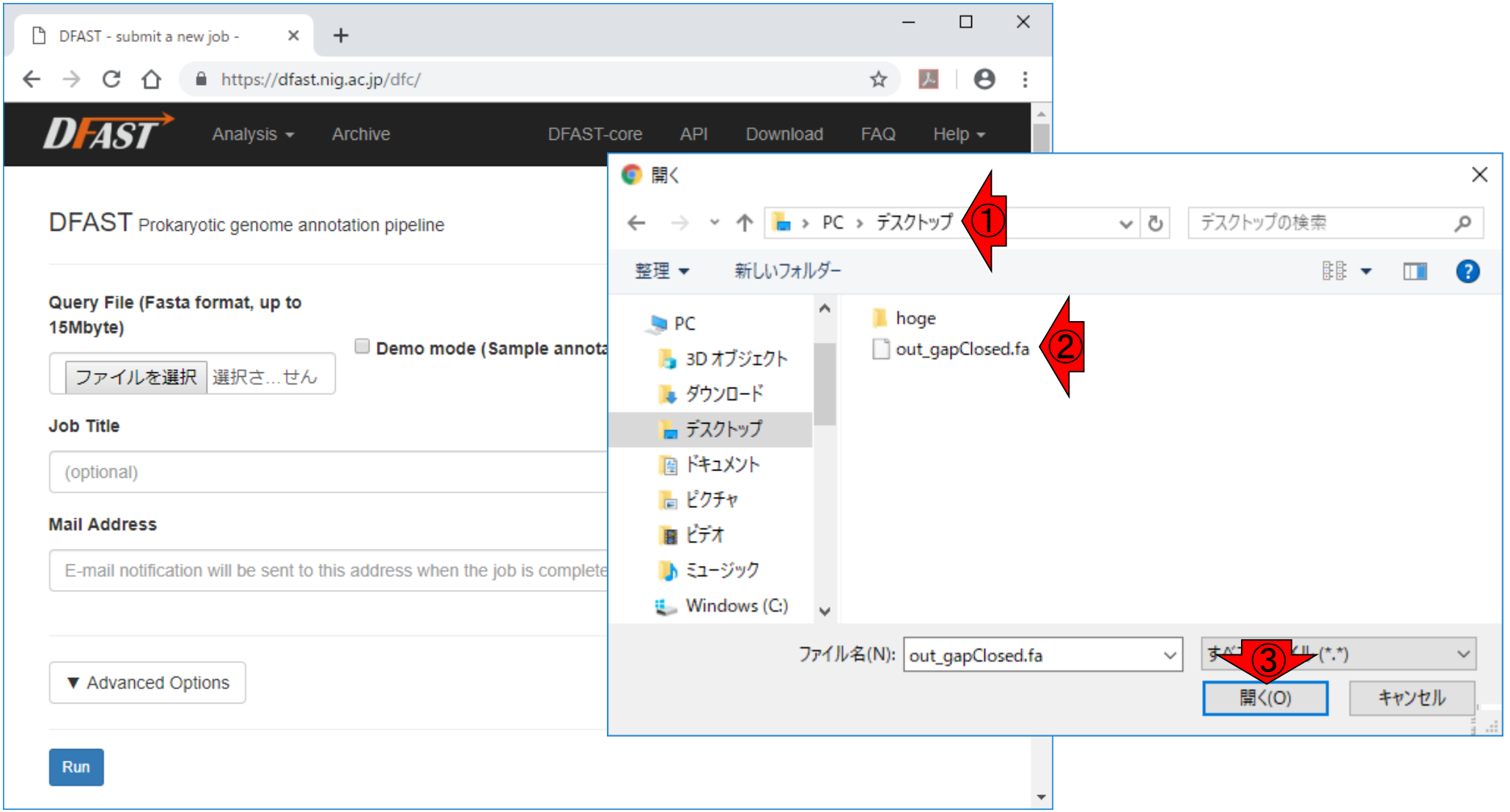
DFAST実行

こんな感じの画面になります。作業自体はとてもシンプル。①アノテーションしたいファイル（この場合はout_gapClosed.fa）をアップロードして、②Runボタンを押すだけ。まずは①をクリック。



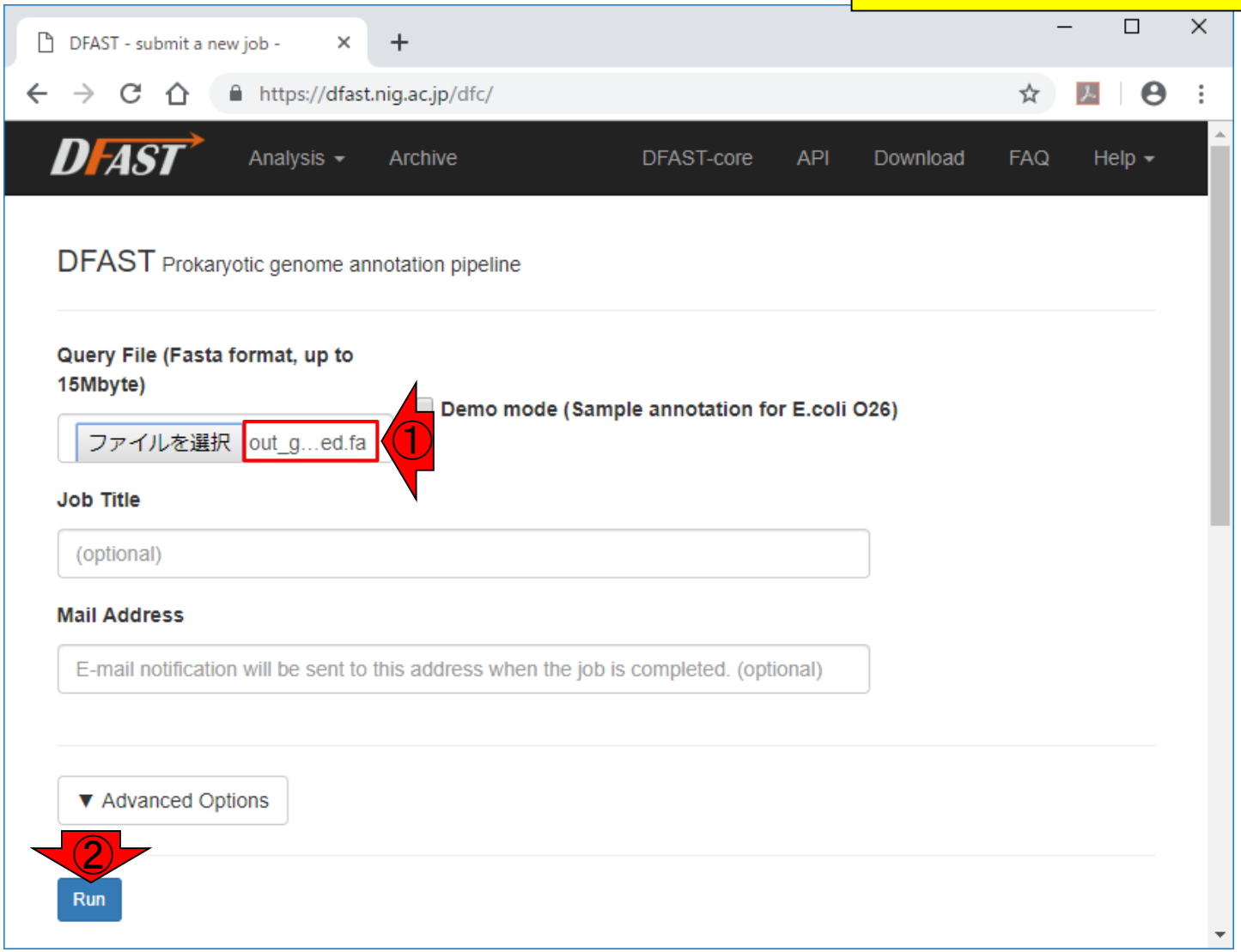
DFAST実行

①デスクトップ上にある、②アノテーションしたいファイル（out_gapClosed.fa）を選択して、③開く



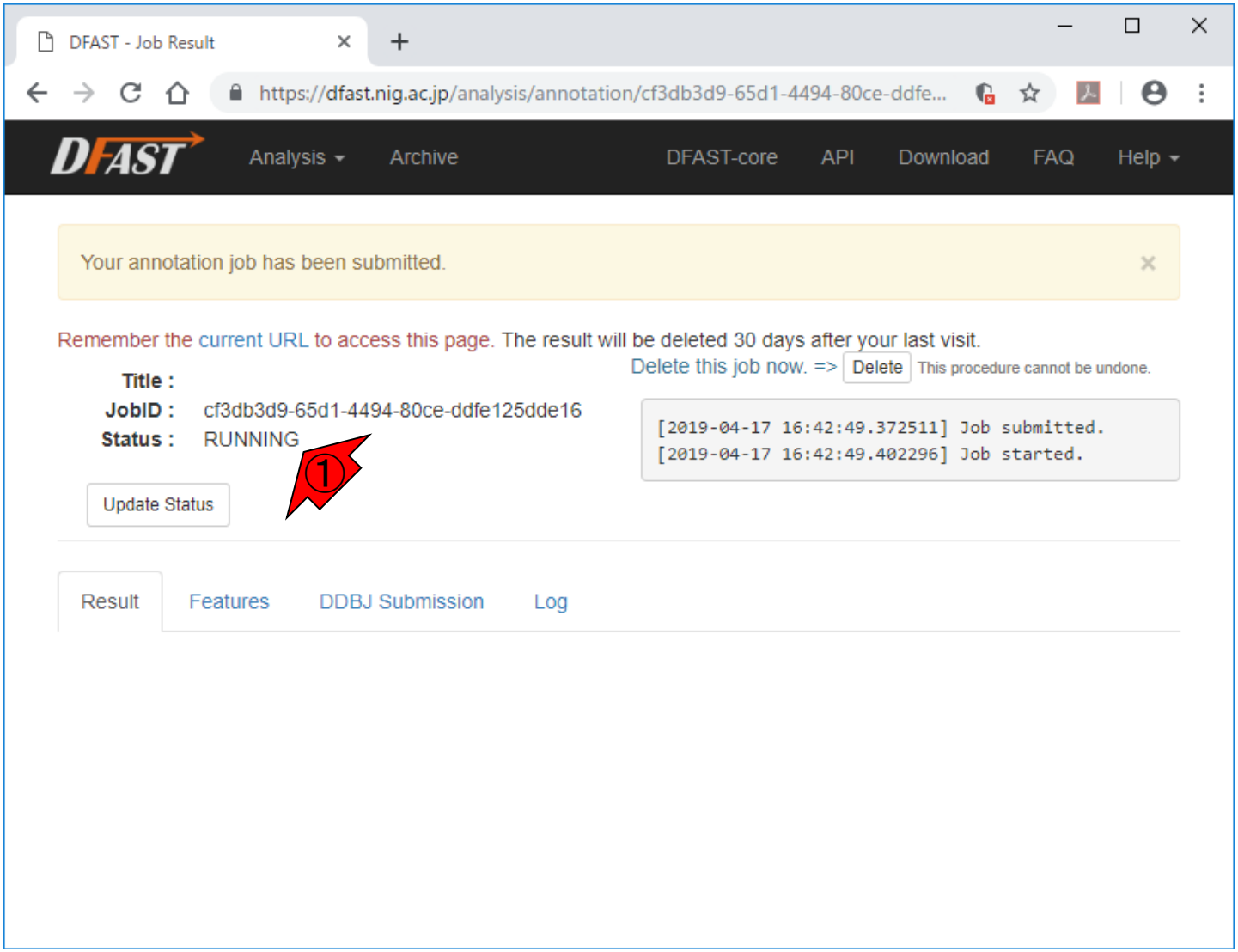
DFAST実行

①アップロードしたファイル名（out_gapClosed.fa）っぽくなったことがわかります。数分程度で終わるので、そのまま②Run。



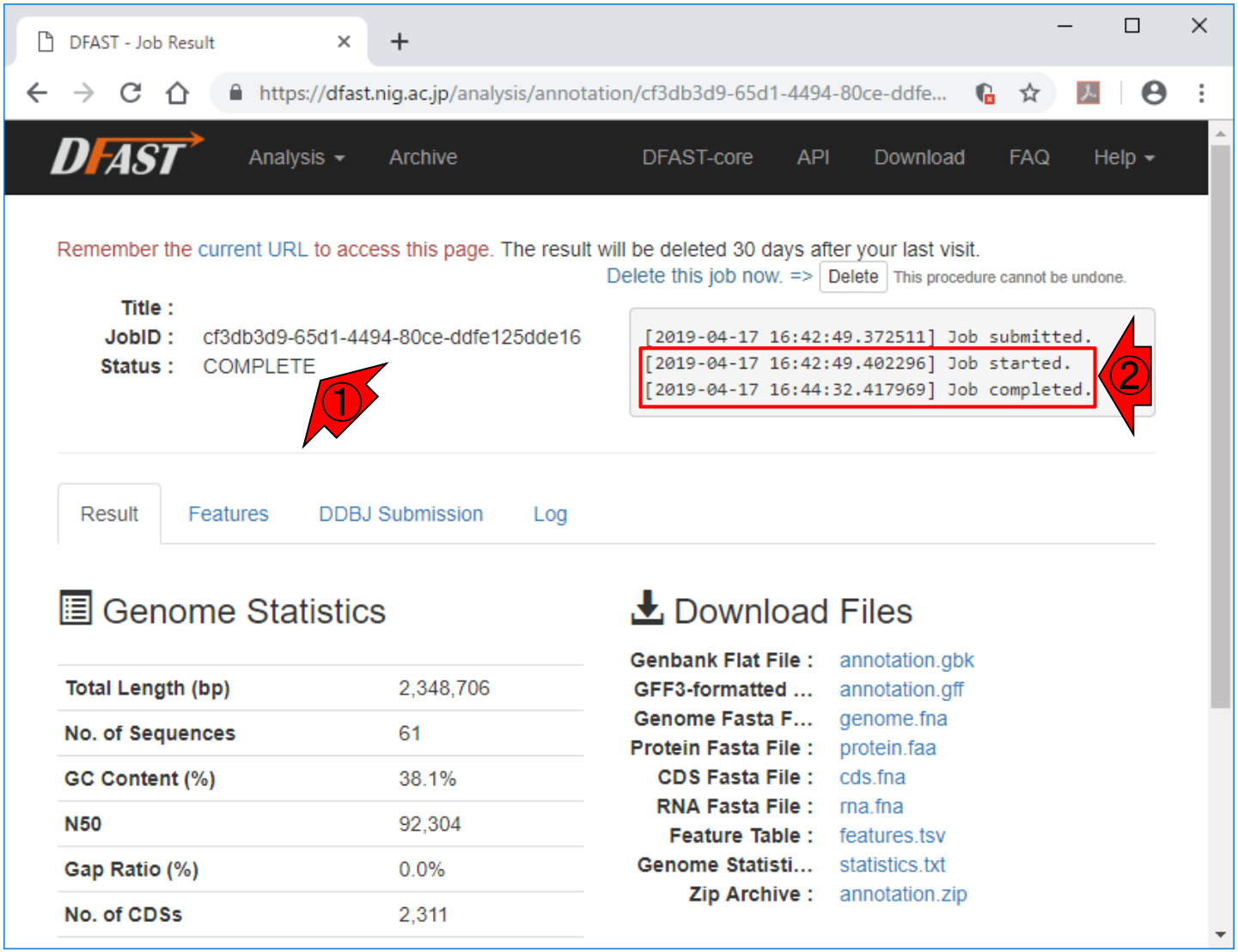
DFAST実行開始

こんな感じになります。①実行中（RUNNING）となってますね。



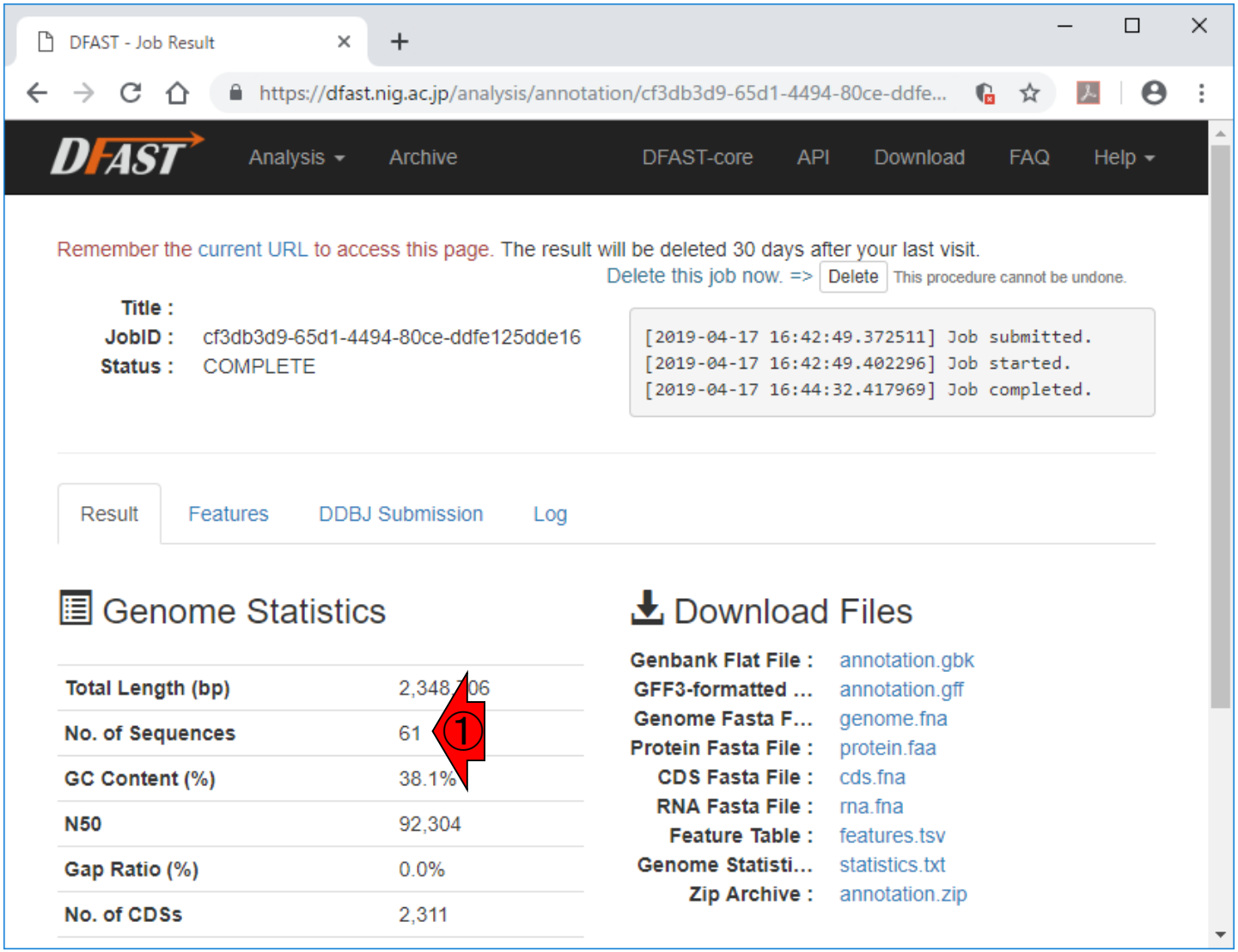
約2分で終了

①実行終了。②このときは2分足らずで終了していることがわかります。



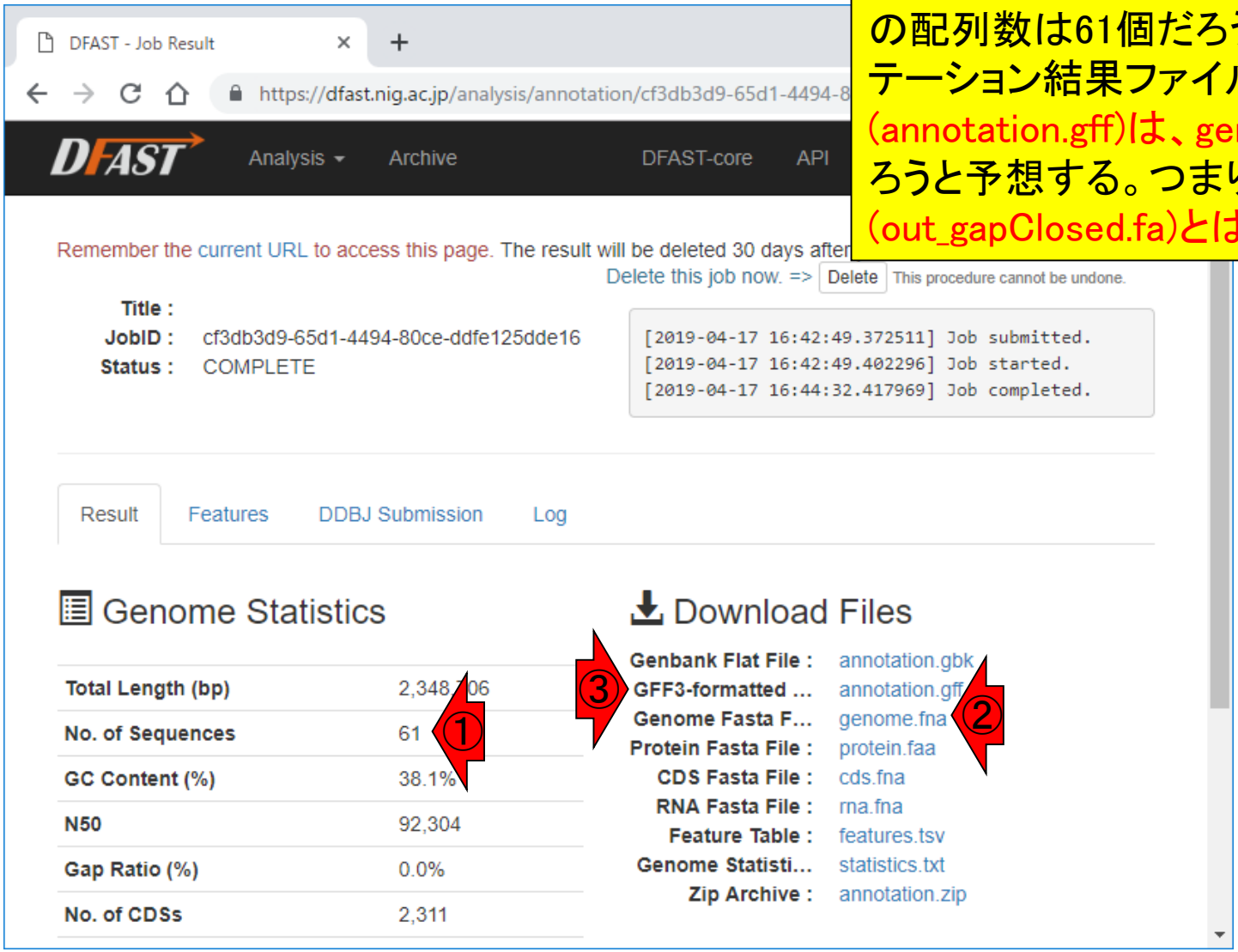
配列数に注目！

入力として与えたout_gapClosed.faの配列数は117個だったが、①で見えている配列数は61個！



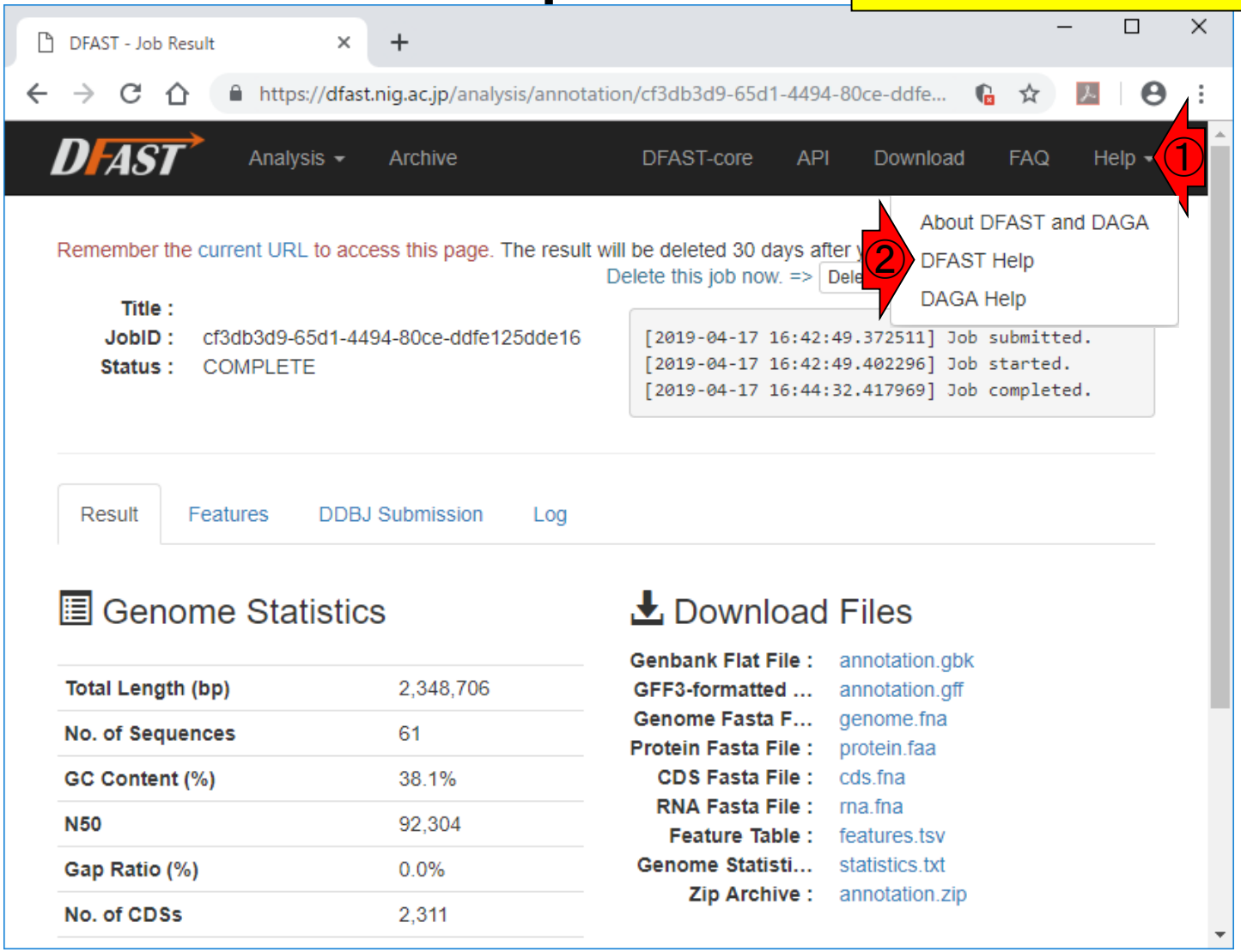
配列数に注目！

入力として与えたout_gapClosed.faの配列数は117個だったが、①で見えている配列数は61個！②で取得可能なゲノムのFASTAファイル(genome.fna)中の配列数は61個だろうと予想する。そして、主なアノテーション結果ファイルである③GFFファイル(annotation.gff)は、genome.fnaと対応しているのだろうと予想する。つまり、元の入力ファイル(out_gapClosed.fa)とは対応していないということ！



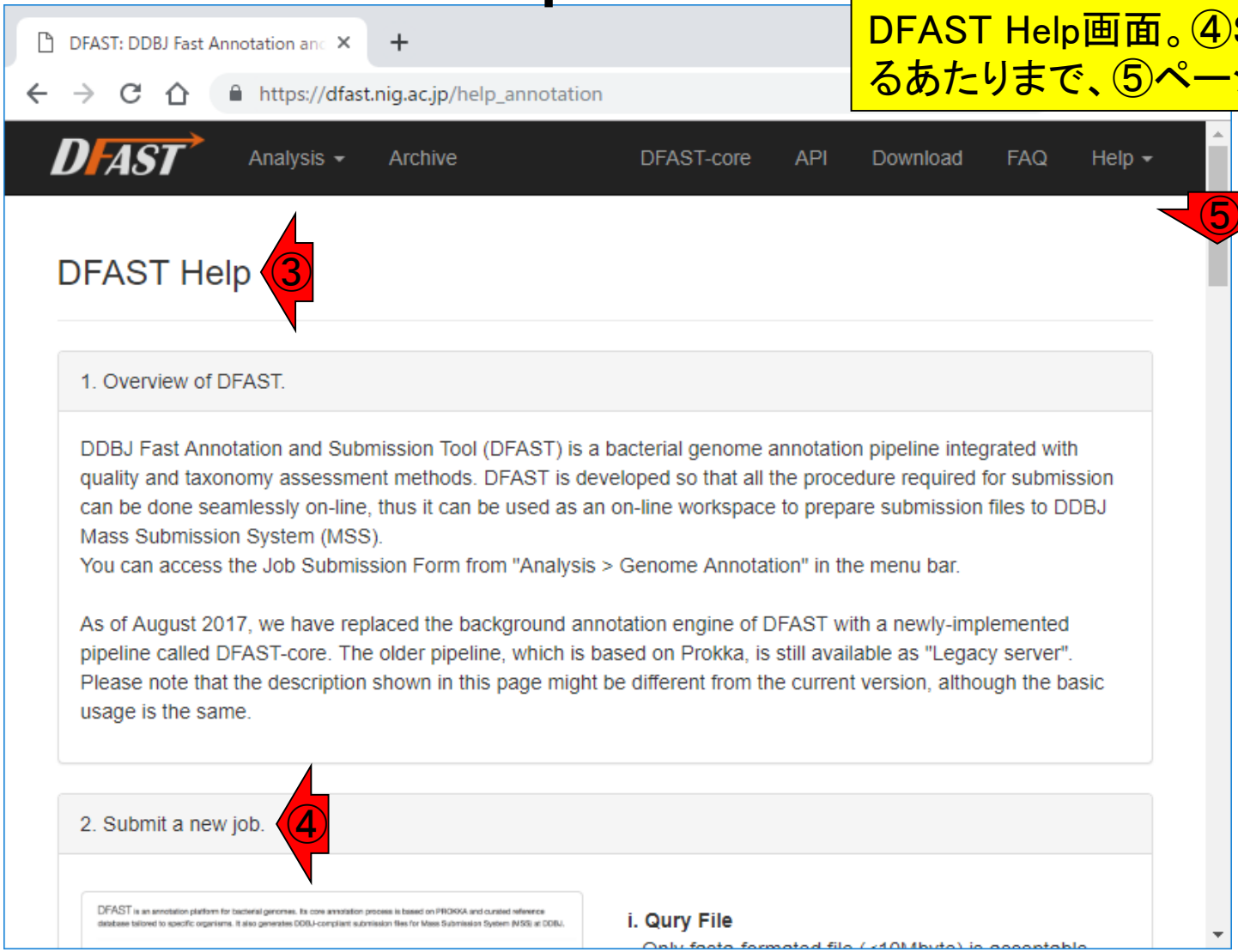
DFAST Help

ざっと調べた限り、原著論文中では入力配列のフィルタリングに関する言及はない。しかし、①Help、②DFAST Helpを眺めるとヒントが得られます。



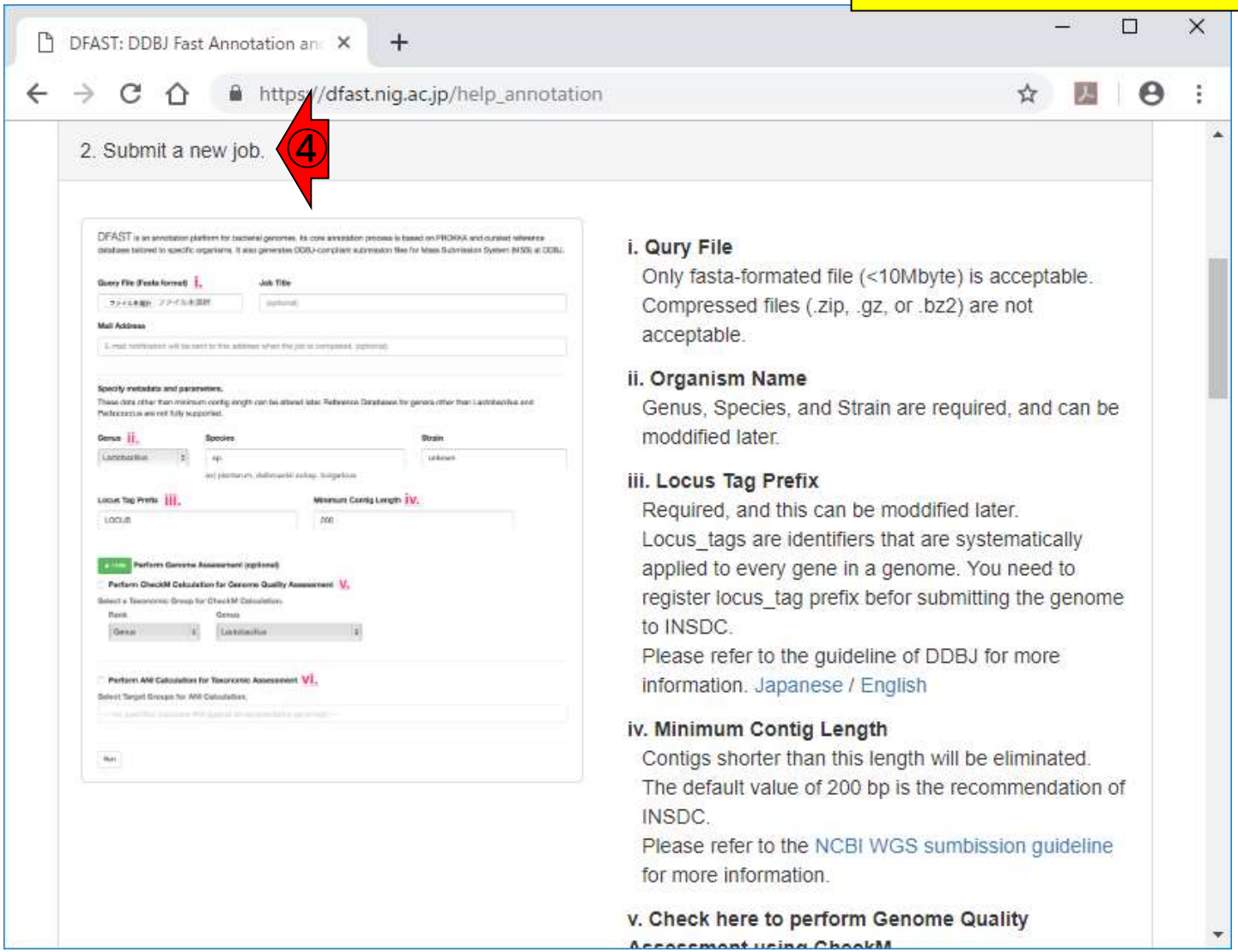
DFAST Help

ざっと調べた限り、原著論文中では入力配列のフィルタリングに関する言及はない。しかし、①Help、②DFAST Helpを眺めるとヒントが得られます。③DFAST Help画面。④Submit a new jobが一番上にくるあたりまで、⑤ページ下部に移動。



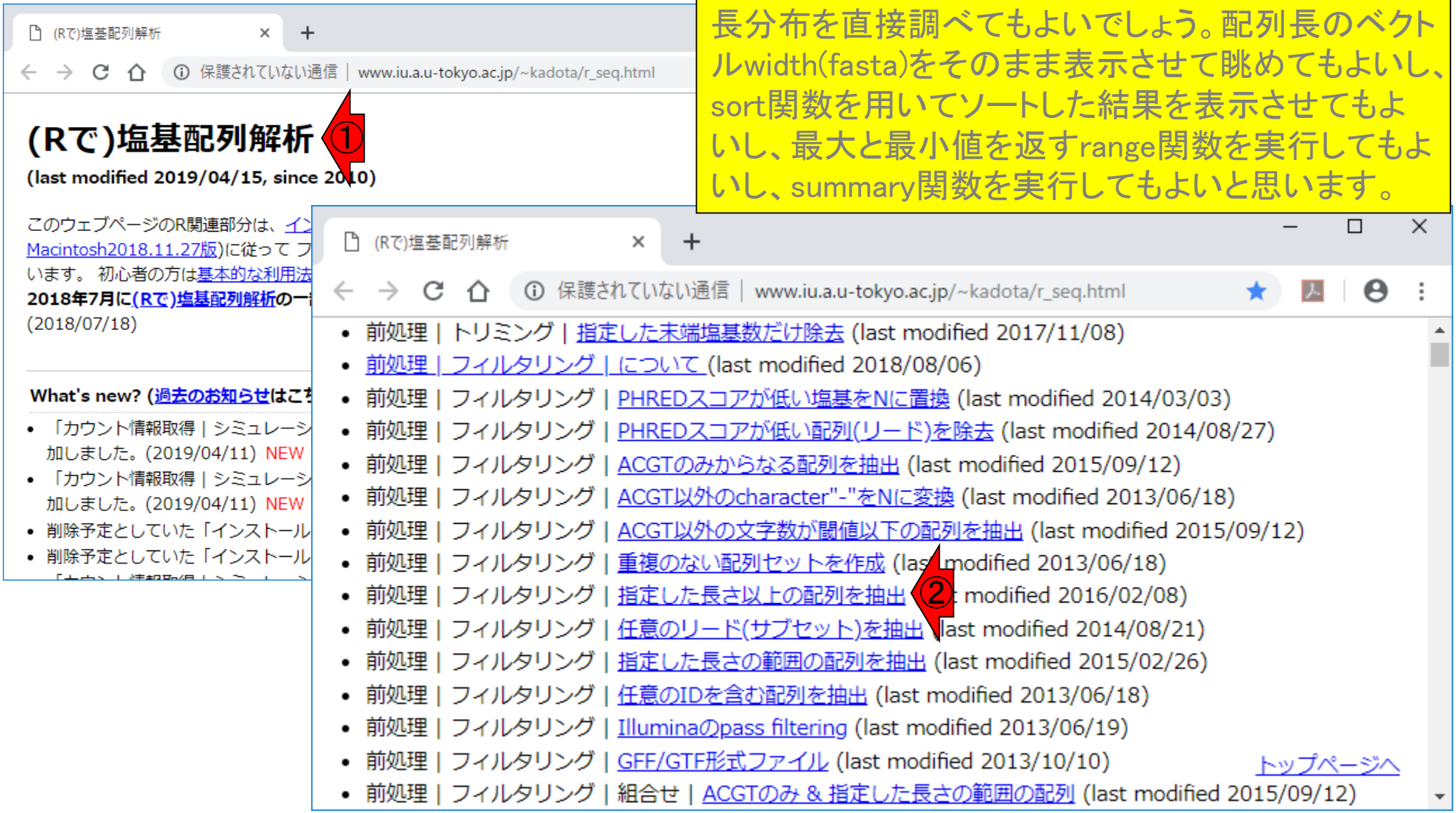
課題1

DFASTへの入力として与えたout_gapClosed.faの配列数は117個だったが、実行結果として配列数が61個となった。この理由について述べよ。



課題1のヒント

②の例題1をテンプレートとして、パラメータを変更しながら実行して確認してもよい。他の手段としては、FASTAファイルの読み込みまで行った段階で、配列長分布を直接調べてもよいでしょう。配列長のベクトルwidth(fasta)をそのまま表示させて眺めてもよいし、sort関数を用いてソートした結果を表示させてもよいし、最大と最小値を返すrange関数を実行してもよいし、summary関数を実行してもよいと思います。



DFAST結果ファイル

①が117配列からなるDFASTへの入力ファイル、②が61配列からなるDFAST実行結果のファイル。

講義日程（2019年度）

1. 2019年04月08日（PC使用）
   講義資料PDF(最終更新：2019.04.09)
   学会(国外)：ISCB
   学会(国内)：JSBi
   QAサイト：Biostar (Parnell et al., PLoS Comput Biol., 2011)
   QAサイト：SEQanswers (Li et al., Bioinformatics, 2012)
   学習教材：バイオインフォマティクス人材育成のための講習会(平成26-29年度)
   学習教材：(Rで)塩基配列解析
   学習教材：(Rで)塩基配列解析のサブ
   RStudio

2. 2019年04月15日（PC使用）
   講義資料PDF(最終更新：2019.04.15)
   (Rで)塩基配列解析
   (Rで)塩基配列解析のサブ

3. 2019年04月22日（PC使用）
   講義資料PDF
   課題：docx版とPDF版
   (Rで)塩基配列解析
   hoge8.fa (課題用)
   Bioconductor
   CRAN

4. 2019年05月13日（PC使用）
   講義資料PDF
   (Rで)塩基配列解析のサブ
   out_gapClosed.fa (約2.3MB)
   (Rで)塩基配列解析
   DFAST：Tanizawa et al., Bioinformatics, 2018
   DFAST実行結果：genome.fna
   DFAST実行結果：annotation.gff
   DFAST実行結果：cds.fna

Contents

- 配列長でフィルタリング（ゲノムアセンブリ結果の後処理）
  - イントロ、本番
- ゲノムアノテーション
  - イントロ、参考文献
  - DFAST実行、課題1
- CDS（タンパク質コード領域の配列；coding sequenceの略）の取得
  - 比較ゲノム解析のイントロ、問題設定
  - 答え（cds.fna）を眺めておく、sequence logos
  - CDS配列取得の成功例、TxDbオブジェクト、CDS座標情報取得
  - CDS配列取得の失敗例、GFF3をExcelで概観
  - 基本テクを駆使して解決する基本戦略、CDSを含む行の抽出、補足説明
  - Excelで確認、リストファイルの作成、基本テクでCDS配列取得を実行
  - ストランド情報つきのリストファイルの作成、例題6のコード解説
  - ストランド情報反映戦略を練り、コードに修正を加えて実行（例題7）

おさらい

これまでに行ったのは、①ゲノムアノテーション作業。その後の解析はおそらく多様であるが、ざっくり述べると比較ゲノム解析が行われる。

- ゲノム配列決定
  - NGSデータ(FASTQ形式ファイル)を入力として、ゲノムアセンブリを実行。
  - 結果として、multi-FASTA形式のアセンブリ結果ファイルが得られる。
  - 例：out_gapClosed.fa。数十万リードを入力として、117個のコンティグを得た。
  - (概要配列レベルではなく完全配列にする方向で頑張るヒトも一定数存在。)
- ゲノムアノテーション
  - 単に塩基配列が羅列された情報が得られただけではうれしくない。ゲノム上のどの領域にCDS(タンパク質コード領域の配列；coding sequenceの略)があるのかなどを知りたい。
  - アセンブリ結果ファイル(FASTA形式ファイル)を入力として、アノテーションを実行。
  - 主な結果として、GFF3形式のアノテーションファイルが得られる。

①

比較ゲノム解析

一口に比較ゲノム解析と言っても、中身は多様。代表的なものとしては、オーソログの同定が挙げられる。より原始的だが効果的な他の手段としては、ドットプロットが挙げられます。

- オーソログの同定
  - オーソログとは、異なる生物に存在する相同な機能を持った遺伝子群のこと。種分化により派生した相同遺伝子*。
  - 相同（homologous; ホモロガス）とは、共通祖先に由来するという意味。
  - 配列が似たタンパク質は機能も似ていることが経験的に分かっている。それを利用して、2つの生物種間で配列の類似性が高いタンパク質コード領域の配列（coding sequence; CDS）同士を対応づける作業が行われます。
  - 具体的な作業としては、まず比較したい2つの生物種のCDS領域を抽出したFASTAファイルをそれぞれ独立に用意する。そして、2つのゲノム間でCDSの総当たり検索を行った際、どちらのゲノムをクエリー（質問配列）にして他方を検索した場合でもベストヒットになるようなCDSの関係、すなわち「双方向ベストヒット（Reciprocal best hit）」の基準を用いて対応づけを行う* 。
  - この配列相同性検索手段としては、Blastの使用がデファクトスタンダード。

*：内山郁夫、生物物理、42(6), 266-269, 2002

①の作業の一部は…

比較ゲノム解析

- オーソログの同定
  - オーソログとは、異なる生物に存在する相同な機能を持った遺伝子群のこと。種分化により派生した相同遺伝子*。
  - 相同（homologous; ホモロガス）とは、共通祖先に由来するという意味。
  - 配列が似たタンパク質は機能も似ていることが経験的に分かっている。それを利用して、2つの生物種間で配列の類似性が高いタンパク質コード領域の配列（coding sequence; CDS）同士を対応づける作業が行われます。
  - 具体的な作業としては、まず比較したい2つの生物種のCDS領域を抽出したFASTAファイルをそれぞれ独立に用意する。そして、2つのゲノム間でCDSの総当たり検索を行った際、どちらのゲノムをクエリー（質問配列）にして他方を検索した場合でもベストヒットになるようなCDSの関係、すなわち「双方向ベストヒット（Reciprocal best hit）」の基準を用いて対応づけを行う*。
  - この配列相同性検索手段としては、Blastの使用がデファクトスタンダード。

比較ゲノム解析

①の作業の一部は、②アグリバイオ講義科目「生物配列解析基礎」の、③第1回（2019年4月10日）で行っています。③のスライド54とか。

1. 生物配列解析基礎 ②

授業の目標・概要

生命科学のためのデータベースの利用と基本的な解析手法について講義します。 配列データベースや機能データベースの使用法を紹介するとともに、ホモロジー検索、モチーフ解析、Perlプログラミング、系統解析などの基本的な手法について、実習形式で解説します。
バイオインフォマティクス関連の各種データベースにアクセスしたことのない人は、ぜひ本講義を受講して下さい。

担当教員

清水謙多郎 （東大・農・応用生命工学専攻 / 教授）
大島研郎 （法政大学生命科学部 / 教授）

お知らせ

ご自身のノートPCを利用される場合はこちらを参考にして必要なソフトウェアを予めインストールしておいてください。

講義日程（2019年度）

1. 2019年04月10日（PC使用）

講師：大島研郎
- 20190410.pdf ③
- kiso1
- Mgenitalium.faa
- Mpneumoniae.faa
- parse-blast7.pl
- test1.seq
- test2.seq
- test3.seq
- Ureaplasma.faa

する相同な機能を持った遺伝子群のこと。
。
、共通祖先に由来するという意味。
ていることが経験的に分かっている。それを
類似性が高いタンパク質コード領域の配列
づける作業が行われます。

たい2つの生物種のCDS領域を抽出した
用意する。そして、2つのゲノム間でCDSの
ゲノムをクエリー（質問配列）にして他方を
ようなCDSの関係、すなわち「双方向ベスト
を用いて対応づけを行う*。
、Blastの使用がデファクトスタンダード。

①

Contents

- 配列長でフィルタリング（ゲノムアセンブリ結果の後処理）
  - イントロ、本番
- ゲノムアノテーション
  - イントロ、参考文献
  - DFAST実行、課題1
- CDS（タンパク質コード領域の配列；coding sequenceの略）の取得
  - 比較ゲノム解析のイントロ、問題設定
  - 答え（cds.fna）を眺めておく、sequence logos
  - CDS配列取得の成功例、TxDbオブジェクト、CDS座標情報取得
  - CDS配列取得の失敗例、GFF3をExcelで概観
  - 基本テクを駆使して解決する基本戦略、CDSを含む行の抽出、補足説明
  - Excelで確認、リストファイルの作成、基本テクでCDS配列取得を実行
  - ストランド情報つきのリストファイルの作成、例題6のコード解説
  - ストランド情報反映戦略を練り、コードに修正を加えて実行（例題7）

①のファイルは…

ファイルの準備

- オーソログの同定
  - オーソログとは、異なる生物に存在する相同な機能を持った遺伝子群のこと。種分化により派生した相同遺伝子*。
  - 相同（homologous; ホモロガス）とは、共通祖先に由来するという意味。
  - 配列が似たタンパク質は機能も似ていることが経験的に分かっている。それを利用して、2つの生物種間で配列の類似性が高いタンパク質コード領域の配列（coding sequence; CDS)同士を対応づける作業が行われます。
  - 具体的な作業としては、まず比較したい2つの生物種のCDS領域を抽出したFASTAファイルをそれぞれ独立に用意する。そして、2つのゲノム間でCDSの総当たり検索を行った際、どちらのゲノムをクエリー（質問配列）にして他方を検索した場合でもベストヒットになるようなCDSの関係、すなわち「双方向ベストヒット（Reciprocal best hit）」の基準を用いて対応づけを行う* 。
  - この配列相同性検索手段としては、Blastの使用がデファクトスタンダード。

ファイルの準備

①のファイルは、②「生物配列解析基礎」では、③で用意されていたものを使った。

1. 生物配列解析基礎 ②

授業の目標・概要

生命科学のためのデータベースの利用と基本的な解析手法について講義します。 配列データベースや機能データベースの使用法を紹介するとともに、ホモロジー検索、モチーフ解析、Perlプログラミング、系統解析などの基本的な手法について、実習形式で解説します。
バイオインフォマティクス関連の各種データベースにアクセスしたことのない人は、ぜひ本講義を受講して下さい。

担当教員

清水謙多郎 （東大・農・応用生命工学専攻 / 教授）
大島研郎 （法政大学生命科学部 / 教授）

お知らせ

ご自身のノートPCを利用される場合はこちらを参考にして必要なソフトウェアを予めインストールしておいてください。

講義日程（2019年度）

1. 2019年04月10日（PC使用）

講師：大島研郎
- 20190410.pdf
- kiso1
- Mgenitalium.faa ③
- Mpneumoniae.faa ③
- parse-blast7.pl
- test1.seq
- test2.seq
- test3.seq
- Ureaplasma.faa

する相同な機能を持った遺伝子群のこと。
。
、共通祖先に由来するという意味。
ていることが経験的に分かっている。それを
類似性が高いタンパク質コード領域の配列
づける作業が行われます。 ①
たい2つの生物種のCDS領域を抽出した
用意する。そして、2つのゲノム間でCDSの
ゲノムをクエリー（質問配列）にして他方を
ようなCDSの関係、すなわち「双方向ベスト
を用いて対応づけを行う*。
、Blastの使用がデファクトスタンダード。

問題設定

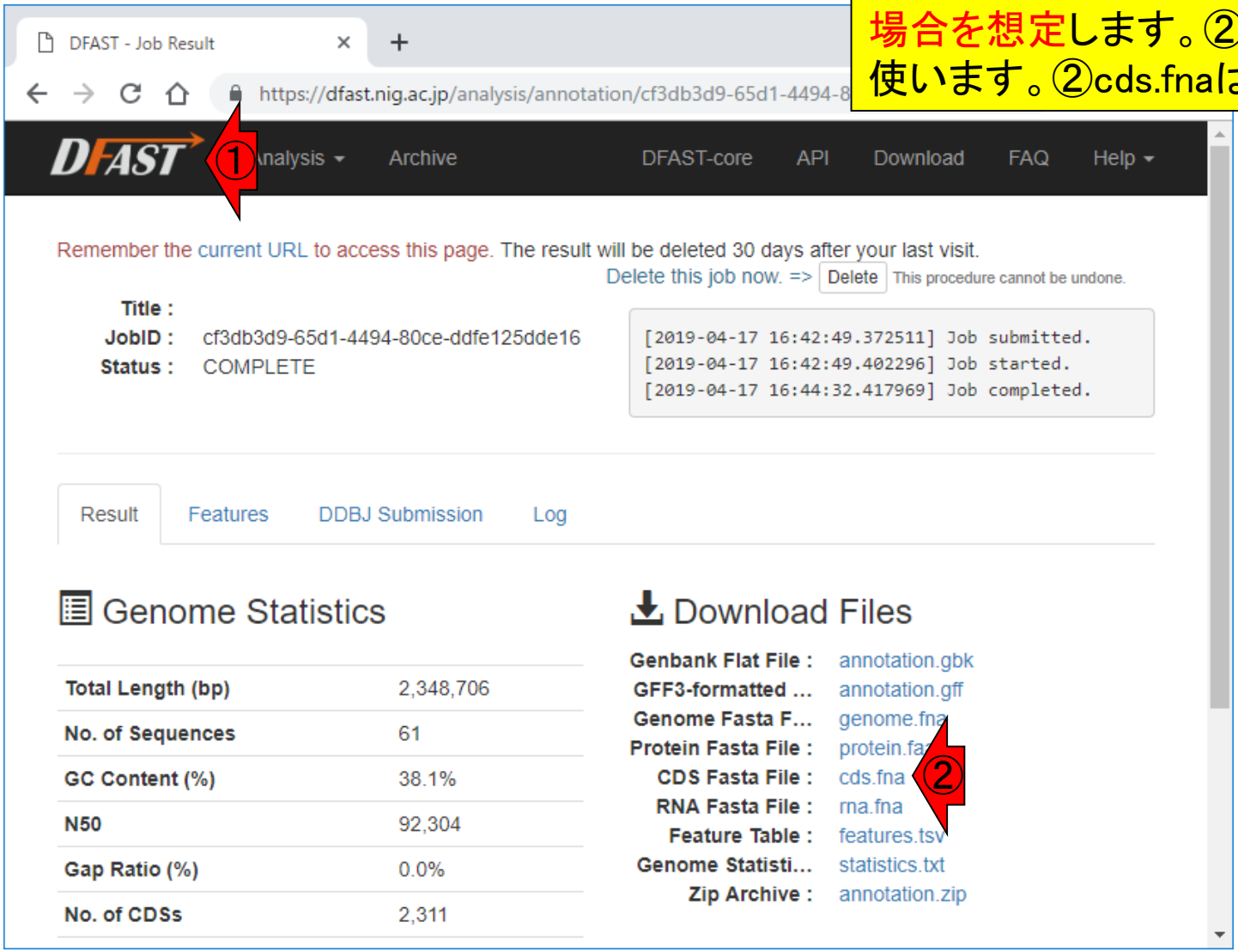
比較対象のほうはともかく、アノテーション結果までしかない手持ちデータのほうは、どのようにして①のファイルを作成すればよいのか？

- オーソログの同定
  - オーソログとは、異なる生物に存在する相同な機能を持った遺伝子群のこと。種分化により派生した相同遺伝子*。
  - 相同（homologous; ホモロガス）とは、共通祖先に由来するという意味。
  - 配列が似たタンパク質は機能も似ていることが経験的に分かっている。それを利用して、2つの生物種間で配列の類似性が高いタンパク質コード領域の配列（coding sequence; CDS）同士を対応づける作業が行われます。
  - 具体的な作業としては、まず比較したい2つの生物種のCDS領域を抽出したFASTAファイルをそれぞれ独立に用意する。そして、2つのゲノム間でCDSの総当たり検索を行った際、どちらのゲノムをクエリー（質問配列）にして他方を検索した場合でもベストヒットになるようなCDSの関係、すなわち「双方向ベストヒット（Reciprocal best hit）」の基準を用いて対応づけを行う*。
  - この配列相同性検索手段としては、Blastの使用がデファクトスタンダード。

①

問題設定

実は①DFASTの場合は、②でCDSのFASTAファイルをダウンロードすることができる。このようなファイルが用意されていない（どこにあるのかわからない）場合を想定します。②cds.fnaは答え合わせ用として使います。②cds.fnaは…



問題設定

実は①DFASTの場合は、②でCDSのFASTAファイルをダウンロードすることができる。このようなファイルが用意されていない（どこにあるのかわからない）場合を想定します。②cds.fnaは答え合わせ用として使います。②cds.fnaは、③からダウンロード可能。

講義日程（2019年度）

1. 2019年04月08日（PC使用）
   講義資料PDF(最終更新：2019.04.09)
   学会(国外)：ISCB
   学会(国内)：JSBi
   QAサイト：Biostar (Parnell et al., PLoS Comput Biol., 2011)
   QAサイト：SEQanswers (Li et al., Bioinformatics, 2012)
   学習教材：バイオインフォマティクス人材育成のための講習会(平成26-29年度)
   学習教材：(Rで)塩基配列解析
   学習教材：(Rで)塩基配列解析のサブ
   RStudio

2. 2019年04月15日（PC使用）
   講義資料PDF(最終更新：2019.04.15)
   (Rで)塩基配列解析
   (Rで)塩基配列解析のサブ

3. 2019年04月22日（PC使用）
   講義資料PDF
   課題：docx版とPDF版
   (Rで)塩基配列解析
   hoge8.fa (課題用)
   Bioconductor
   CRAN

4. 2019年05月13日（PC使用）
   講義資料PDF
   (Rで)塩基配列解析のサブ
   out_gapClosed.fa (約2.3MB)
   (Rで)塩基配列解析
   DFAST：Tanizawa et al., Bioinformatics, 2018
   DFAST実行結果：genome.fna
   DFAST実行結果：annotation.gff
   DFAST実行結果：cds.fna

4. 2019年05月13日（PC使用）
講義資料PDF
(Rで)塩基配列解析のサブ
out_gapClosed.fa (約2.3MB)
(Rで)塩基配列解析
DFAST：Tanizawa et al., Bioinformatics, 2018
DFAST実行結果：genome.fna
DFAST実行結果：annotation.gff
DFAST実行結果：cds.fna
③

問題設定

①このCDS領域を抽出したファイル（cds.fna）は、②アノテーション結果（GFF3ファイル）とゲノム配列（FASTAファイル）があれば作成できるので、それをやります。

講義日程（2019年度）

1. 2019年04月08日（PC使用）
   講義資料PDF(最終更新：2019.04.09)
   学会(国外)：ISCB
   学会(国内)：JSBi
   QAサイト：Biostar (Parnell et al., PLoS Comput Biol., 2011)
   QAサイト：SEQanswers (Li et al., Bioinformatics, 2012)
   学習教材：バイオインフォマティクス人材育成のための講習会(平成26-29年度)
   学習教材：(Rで)塩基配列解析
   学習教材：(Rで)塩基配列解析のサブ
   RStudio

2. 2019年04月15日（PC使用）
   講義資料PDF(最終更新：2019.04.15)
   (Rで)塩基配列解析
   (Rで)塩基配列解析のサブ

3. 2019年04月22日（PC使用）
   講義資料PDF
   課題：docx版とPDF版
   (Rで)塩基配列解析
   hoge8.fa (課題用)
   Bioconductor
   CRAN

4. 2019年05月13日（PC使用）
   講義資料PDF
   (Rで)塩基配列解析のサブ
   out_gapClosed.fa (約2.3MB)
   (Rで)塩基配列解析
   DFAST：Tanizawa et al., Bioinformatics, 2018
   DFAST実行結果：genome.fna
   DFAST実行結果：annotation.gff
   DFAST実行結果：cds.fna

4. 2019年05月13日（PC使用）
講義資料PDF
(Rで)塩基配列解析のサブ
out_gapClosed.fa (約2.3MB)
(Rで)塩基配列解析
DFAST：Tanizawa et al., Bioinformatics, 2018
DFAST実行結果：genome.fna
DFAST実行結果：annotation.gff
DFAST実行結果：cds.fna

②

①

想定外の問題に遭遇

①このCDS領域を抽出したファイル（cds.fna）は、②アノテーション結果（GFF3ファイル）とゲノム配列（FASTAファイル）があれば作成できるので、それをやります。当初、「②を用いて簡単に配列取得できますよ、はいできましたね。では次のトピック（ドットプロット）へ。」の予定でしたが、なぜかR上でこのGFFファイルをうまく読み込めない問題に遭遇しました。

講義日程（2019年度）

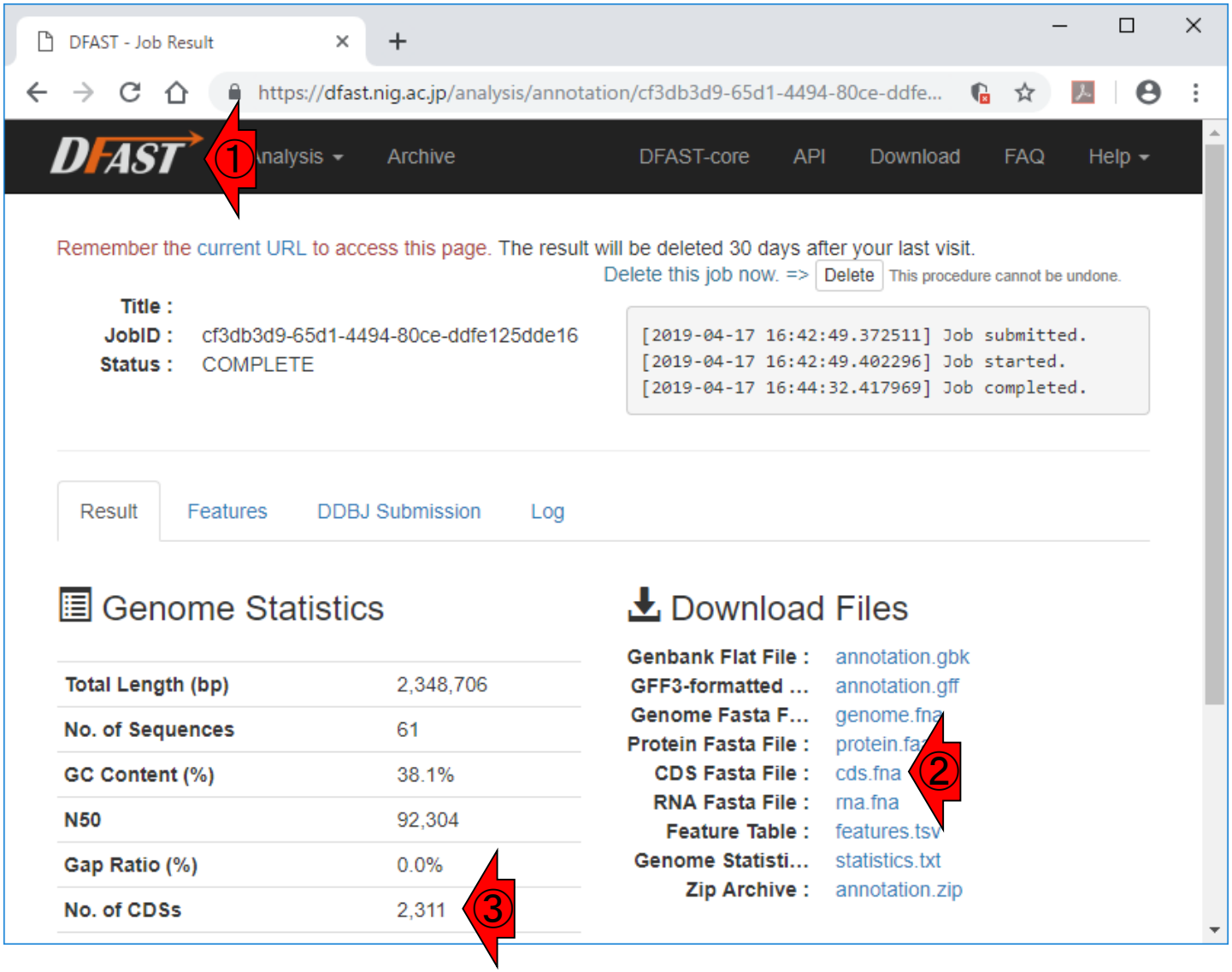
1. 2019年04月08日（PC使用）
   - 講義資料PDF(最終更新：2019.04.09)
   - 学会(国外)：ISCB
   - 学会(国内)：JSBi
   - QAサイト：Biostar (Parnell et al., PLoS Comput Biol., 2011)
   - QAサイト：SEQanswers (Li et al., Bioinformatics, 2012)
   - 学習教材：バイオインフォマティクス人材育成のための講習会(平成26-29年度)
   - 学習教材：(Rで)塩基配列解析
   - 学習教材：(Rで)塩基配列解析のサブ
   - RStudio
2. 2019年04月15日（PC使用）
   - 講義資料PDF(最終更新：2019.04.15)
   - (Rで)塩基配列解析
   - (Rで)塩基配列解析のサブ
3. 2019年04月22日（PC使用）
   - 講義資料PDF
   - 課題：docx版とPDF版
   - (Rで)塩基配列解析
   - hoge8.fa (課題用)
   - Bioconductor
   - CRAN
4. 2019年05月13日（PC使用）
   - 講義資料PDF
   - (Rで)塩基配列解析のサブ
   - out_gapClosed.fa (約2.3MB)
   - (Rで)塩基配列解析
   - DFAST：Tanizawa et al., Bioinformatics, 2018
   - DFAST実行結果：genome.fna
   - DFAST実行結果：annotation.gff
   - DFAST実行結果：cds.fna

Contents

- 配列長でフィルタリング（ゲノムアセンブリ結果の後処理）
  - イントロ、本番
- ゲノムアノテーション
  - イントロ、参考文献
  - DFAST実行、課題1
- CDS（タンパク質コード領域の配列；coding sequenceの略）の取得
  - 比較ゲノム解析のイントロ、問題設定
  - 答え（cds.fna）を眺めておく、sequence logos
  - CDS配列取得の成功例、TxDbオブジェクト、CDS座標情報取得
  - CDS配列取得の失敗例、GFF3をExcelで概観
  - 基本テクを駆使して解決する基本戦略、CDSを含む行の抽出、補足説明
  - Excelで確認、リストファイルの作成、基本テクでCDS配列取得を実行
  - ストランド情報つきのリストファイルの作成、例題6のコード解説
  - ストランド情報反映戦略を練り、コードに修正を加えて実行（例題7）

答えを眺めておく

①DFASTのアノテーション結果画面。②cds.fnaの配列数は2,311個である。



答えを眺めておく

①DFASTのアノテーション結果画面。②cds.fnaの配列数は2,311個である。本科目「ゲノム情報解析基礎」のページ上にある③cds.fnaをデスクトップ上にダウンロード。

講義日程（2019年度）

1. 2019年04月08日（PC使用）
   講義資料PDF(最終更新：2019.04.09)
   学会(国外)：ISCB
   学会(国内)：JSBi
   QAサイト：Biostar (Parnell et al., PLoS Comput Biol., 2011)
   QAサイト：SEQanswers (Li et al., Bioinformatics, 2012)
   学習教材：バイオインフォマティクス人材育成のための講習会(平成26-29年度)
   学習教材：(Rで)塩基配列解析
   学習教材：(Rで)塩基配列解析のサブ
   RStudio

2. 2019年04月15日（PC使用）
   講義資料PDF(最終更新：2019.04.15)
   (Rで)塩基配列解析
   (Rで)塩基配列解析のサブ

3. 2019年04月22日（PC使用）
   講義資料PDF
   課題：docx版とPDF版
   (Rで)塩基配列解析
   hoge8.fa (課題用)
   Bioconductor
   CRAN

4. 2019年05月13日（PC使用）
   講義資料PDF
   (Rで)塩基配列解析のサブ
   out_gapClosed.fa (約2.3MB)
   (Rで)塩基配列解析
   DFAST：Tanizawa et al., Bioinformatics, 2018
   DFAST実行結果：genome.fna
   DFAST実行結果：annotation.gff
   DFAST実行結果：cds.fna

答えを眺めておく

①DFASTのアノテーション結果画面。②cds.fnaの配列数は2,311個である。本科目「ゲノム情報解析基礎」のページ上にある③cds.fnaをデスクトップ上にダウンロード。ここでやりたいのは、デスクトップ上にダウンロードしたcds.fnaをR上で読み込んで、2,311個になっていることや、多くの配列がATG からスタートしていることを確認すること。mRNAの5‘末端から最初に現れる AUG が開始コドンである場合が多い*からです。

講義日程（2019年度）

1. 2019年04月08日（PC使用）
   講義資料PDF(最終更新：2019.04.09)
   学会(国外)：ISCB
   学会(国内)：JSBi
   QAサイト：Biostar (Parnell et al., PLoS Comput Biol., 2011)
   QAサイト：SEQanswers (Li et al., Bioinformatics, 2012)
   学習教材：バイオインフォマティクス人材育成のための講習会(平成26-29年度)
   学習教材：(Rで)塩基配列解析
   学習教材：(Rで)塩基配列解析のサブ
   RStudio

2. 2019年04月15日（PC使用）
   講義資料PDF(最終更新：2019.04.15)
   (Rで)塩基配列解析
   (Rで)塩基配列解析のサブ

3. 2019年04月22日（PC使用）
   講義資料PDF
   課題：docx版とPDF版
   (Rで)塩基配列解析
   hoge8.fa (課題用)
   Bioconductor
   CRAN

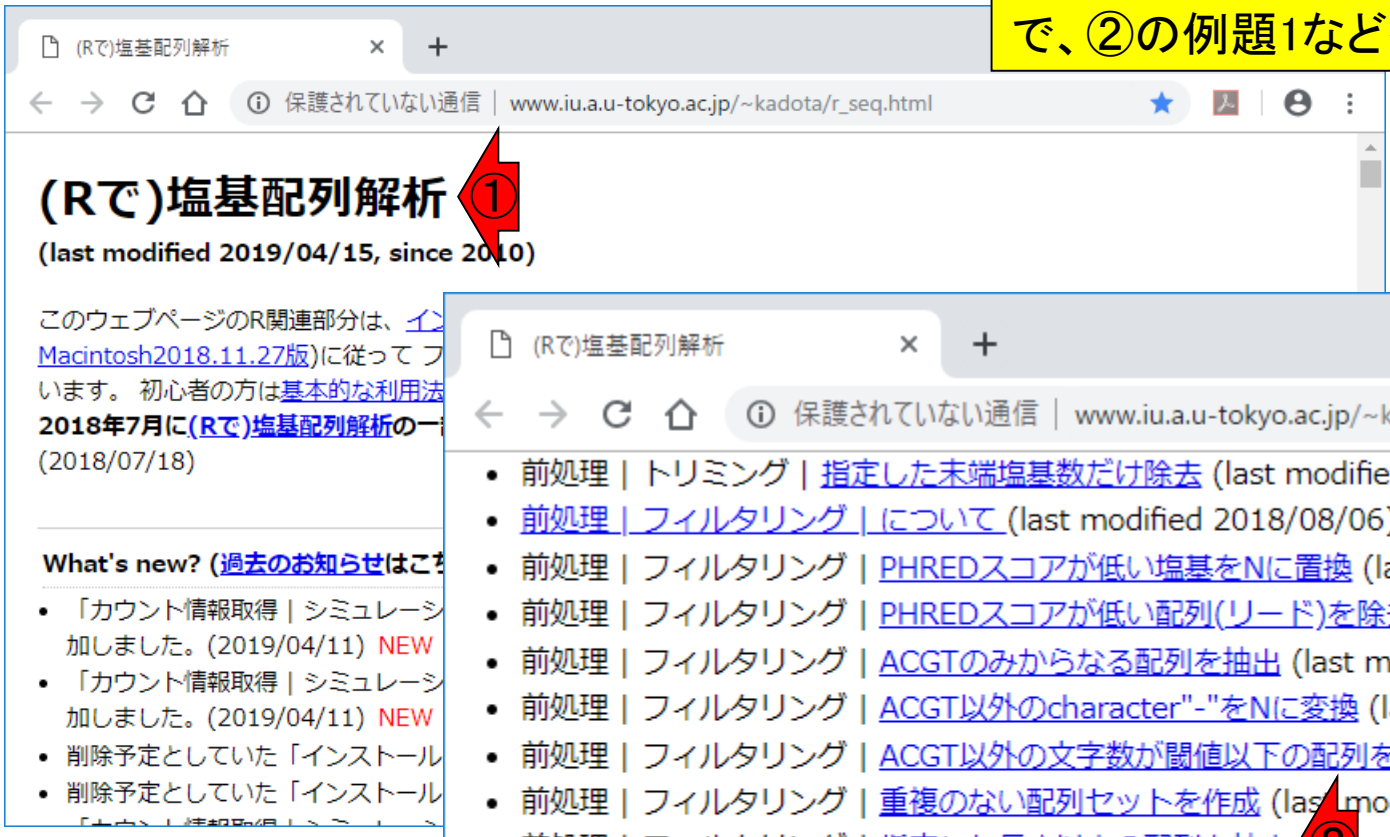
4. 2019年05月13日（PC使用）
   講義資料PDF
   (Rで)塩基配列解析のサブ
   out_gapClosed.fa (約2.3MB)
   (Rで)塩基配列解析
   DFAST：Tanizawa et al., Bioinformatics, 2018
   DFAST実行結果：genome.fna
   DFAST実行結果：annotation.gff
   DFAST実行結果：cds.fna

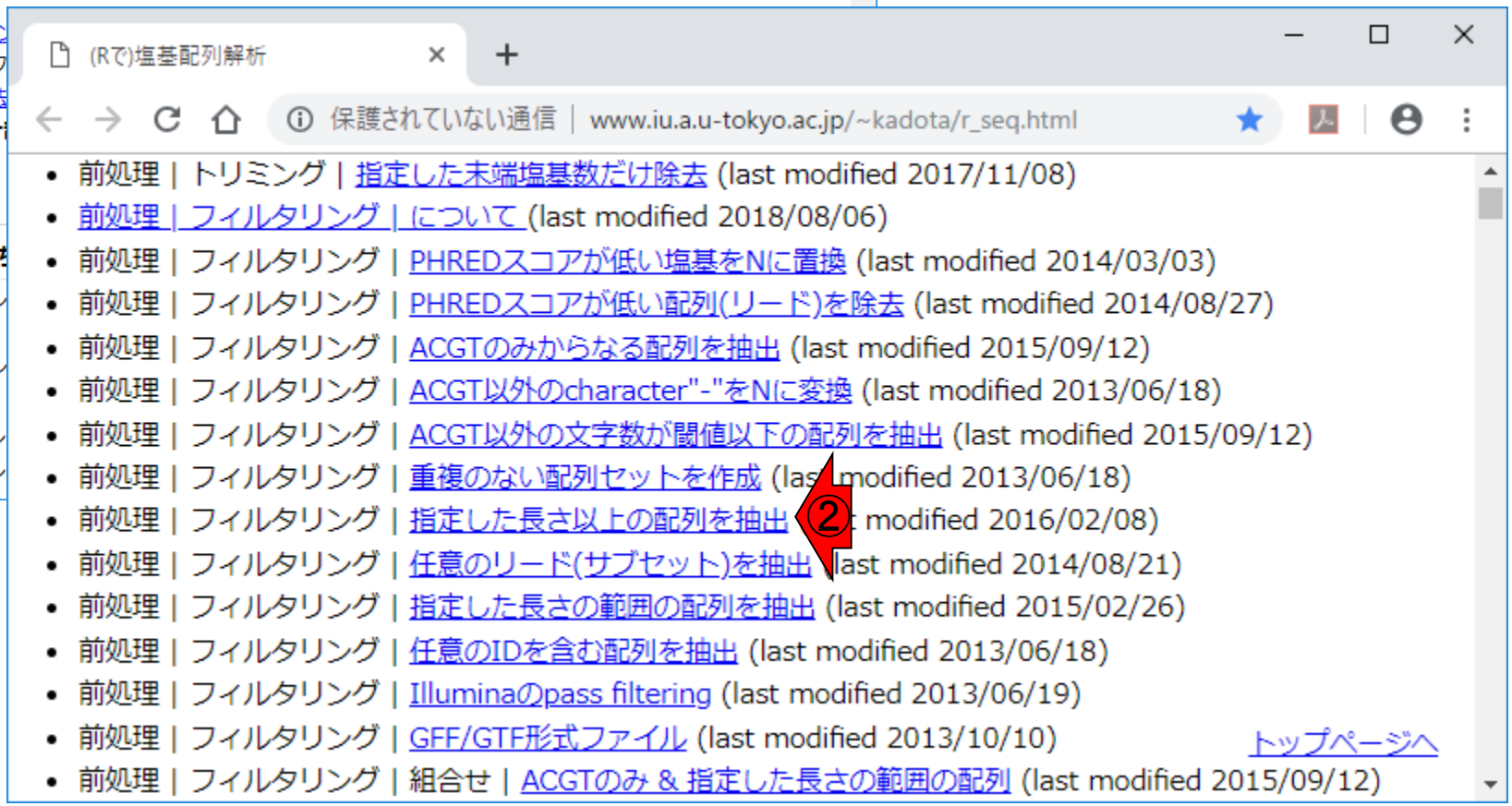
4. 2019年05月13日（PC使用）
講義資料PDF
(Rで)塩基配列解析のサブ
out_gapClosed.fa (約2.3MB)
(Rで)塩基配列解析
DFAST：Tanizawa et al., Bioinformatics, 2018
DFAST実行結果：genome.fna
DFAST実行結果：annotation.gff
DFAST実行結果：cds.fna
③

*https://ja.wikipedia.org/wiki/開始コドン

答えを眺めておく

デスクトップ上にダウンロードしたcds.fnaをR上で読み込んで、2,311個になっていることや多くの配列がATG からスタートしていることを確認するだけなので、②の例題1などをテンプレートとして使ってもよい。





答えを眺めておく

デスクトップ上にダウンロードしたcds.fnaをR上で読み込んで、2,311個になっていることや多くの配列がATG からスタートしていることを確認するだけなので、②の例題1などをテンプレートとして使ってもよい。
例えば、③の赤枠部分のみをテンプレートとして、④をcds.fnaに変更してコピペ実行し、⑤を眺めればよい。

(Rで)塩基配列解析

保護されていない通信 | www.iu.a.u-t

前処理 | フィルタリング | 指定した長さ以上の配列を抽出

FASTA形式やFASTQ形式ファイルを入力として、指定した配列長以上の配列を抽出するやり方を示します。
「ファイル」－「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピペ。

1. multi-FASTAファイル(hoge4.fa)の場合:

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ファイルです。

④

```
in_f <- "hoge4.fa"                          #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_length <- 50                          #配列長の閾値を指定

#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
fasta                                       #確認してるだけです
```

③

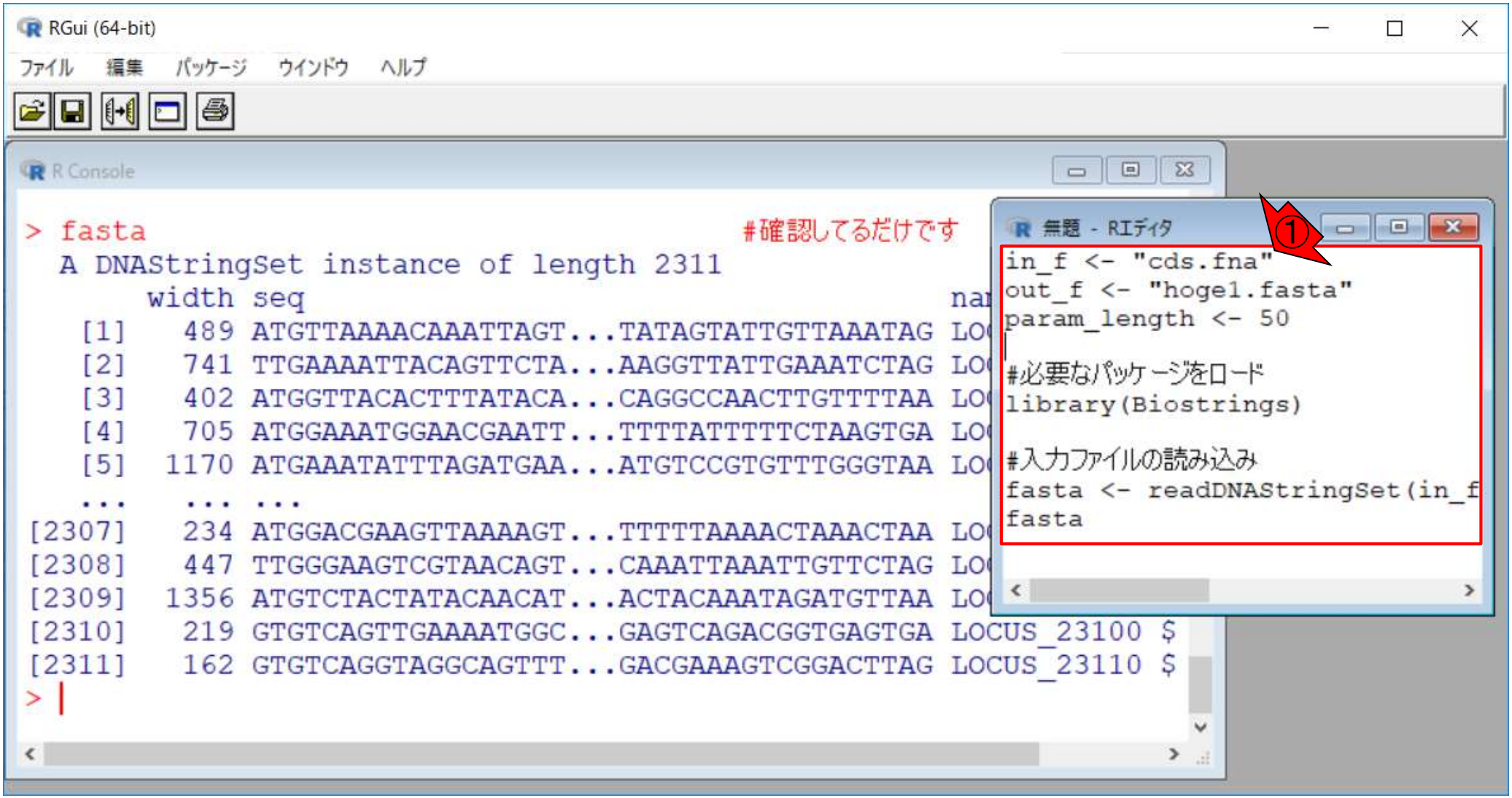
⑤

トップページへ

```
#本番
obj <- as.logical(width(fasta) >= param_length)#条件を満たすかどうかを判定した結果をobjに格納
```

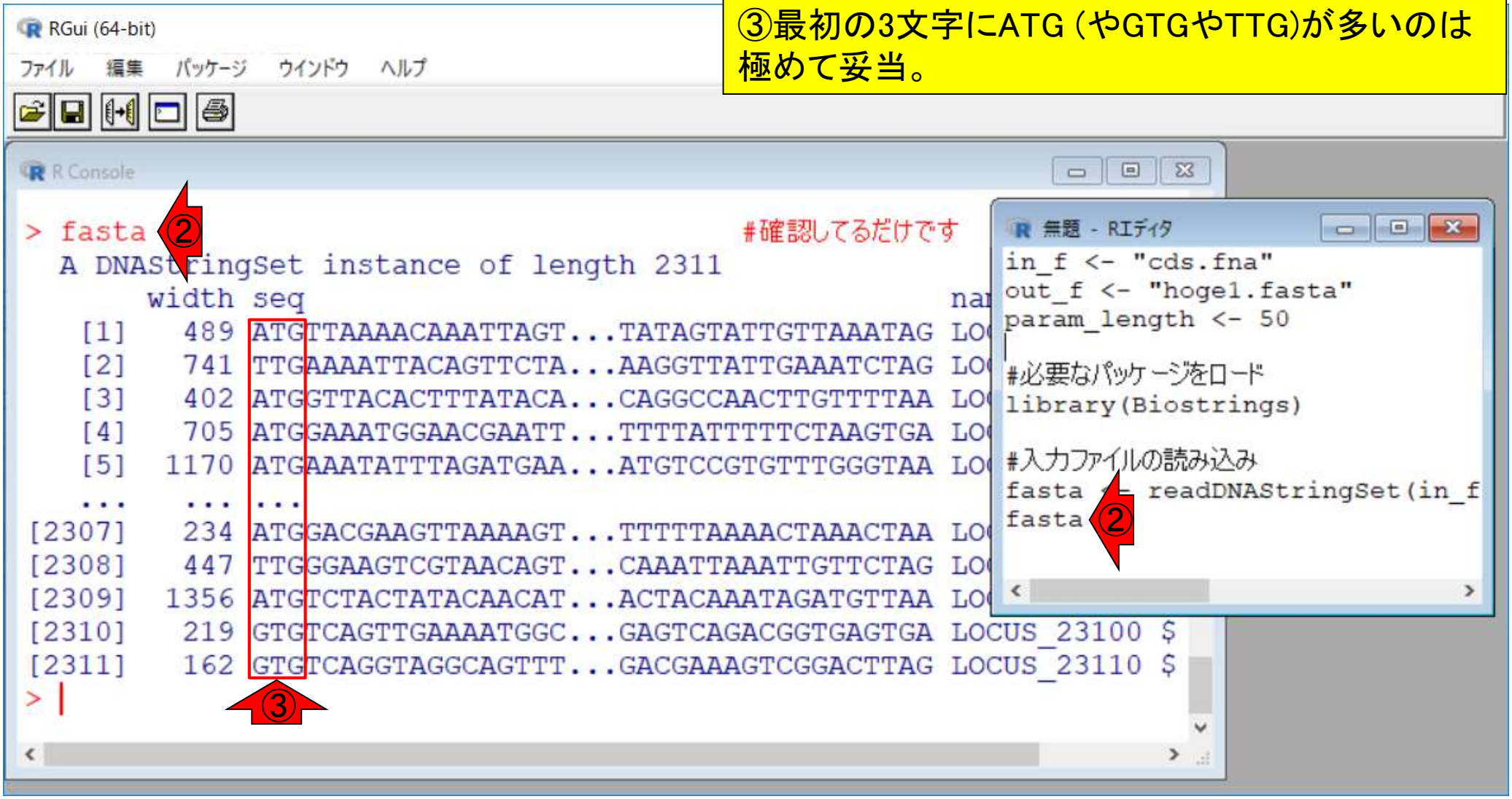
赤枠内で、①cds.fnaに変更してコピペ実行した結果。

答えを眺めておく



答えを眺めておく

赤枠内で、①cds.fnaに変更してコピペ実行した結果。
②fastaオブジェクトが表示されています。今見ているのは、CDS(タンパク質コード領域の配列)なので、
③最初の3文字にATG（やGTGやTTG)が多いのは極めて妥当。



Sequence logos: Schneider and Stephens, *Nucleic Acids Res.*, **18**: 6097–6100, 1990

答えを眺めておく

赤枠内で、①cds.fnaに変更してコピペ実行した結果。②fastaオブジェクトが表示されています。今見ているのは、CDS(タンパク質コード領域の配列)なので、③最初の3文字にATG（やGTGやTTG)が多いのは極めて妥当。ポジションごとにどのような塩基が多いかを表示する手段としてよく用いられるのが④sequence logos。

RGui (64-bit)

ファイル　編集　パッケージ　ウインドウ　ヘルプ

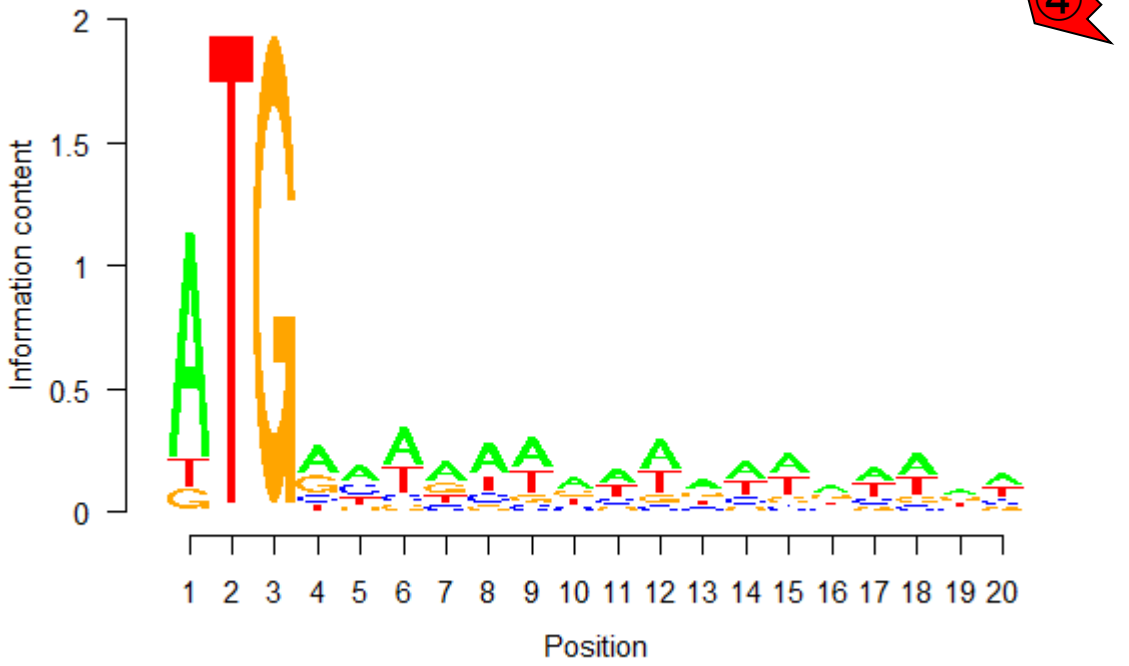
R Console

```
> fasta
A DNAStringSet instance of
      width seq
   [1]   489 ATGTTAAAACAAATTA
   [2]   741 TTGAAAATTACAGTTC
   [3]   402 ATGGTTACACTTTATA
   [4]   705 ATGGAAATGGAACGAA
   [5]  1170 ATGAAATATTTAGATG
   ...   ... ...
[2307]   234 ATGGACGAAGTTAAAA
[2308]   447 TTGGGAAGTCGTAACA
[2309]  1356 ATGTCTACTATACAAC
[2310]   219 GTGTCAGTTGAAAATG
[2311]   162 GTGTCAGGTAGGCAGT
> |
```

②

③

④



Sequence logos: Schneider and Stephens, *Nucleic Acids Res.*, **18**: 6097–6100, 1990

Sequence logos

赤枠内で、①cds.fnaに変更してコピペ実行した結果。②fastaオブジェクトが表示されています。今見ているのは、CDS(タンパク質コード領域の配列)なので、③最初の3文字にATG（やGTGやTTG)が多いのは極めて妥当。ポジションごとにどのような塩基が多いかを表示する手段としてよく用いられるのが④sequence logos。⑤ATGが多いことが分かりますね。

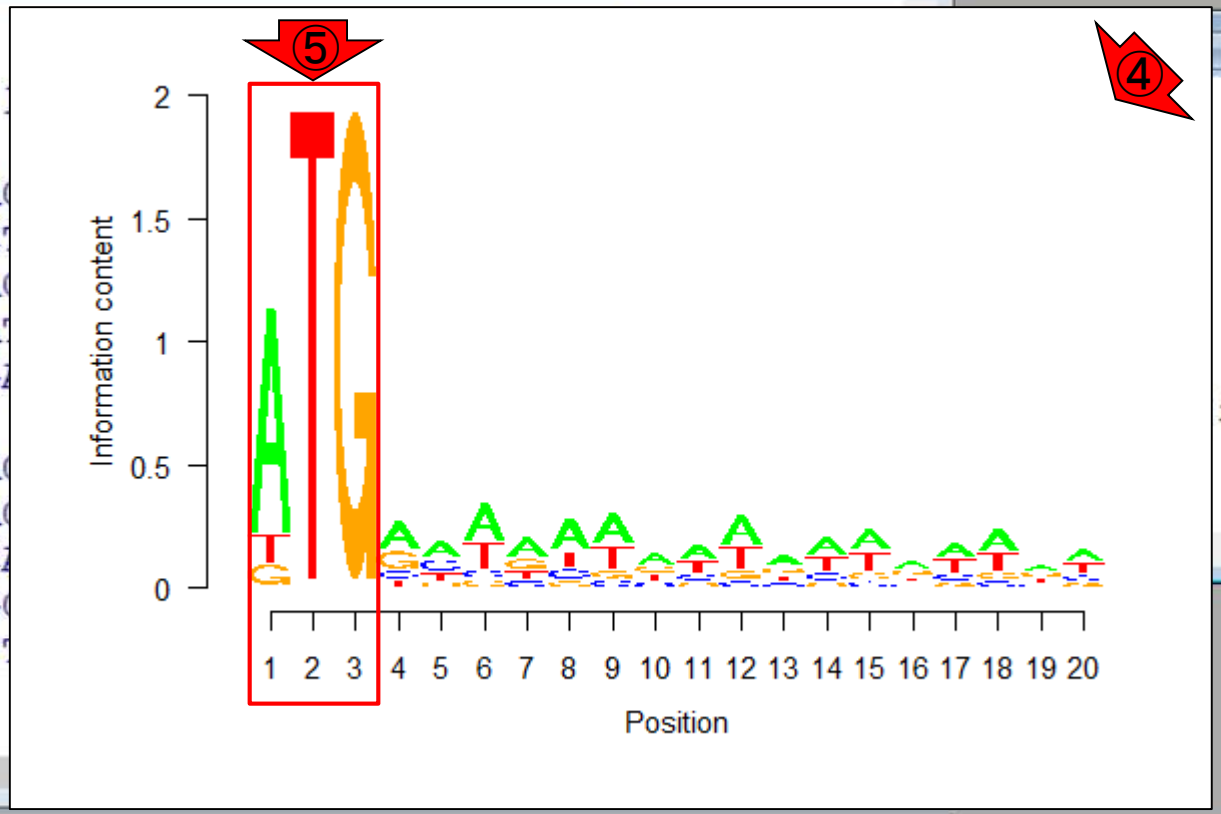
RGui (64-bit)

ファイル　編集　パッケージ　ウインドウ　ヘルプ

R Console

```
> fasta
  A DNAStringSet instance of
      width seq
   [1]   489 ATGTTAAAACAAATTA
   [2]   741 TTGAAAATTACAGTTC
   [3]   402 ATGGTTACACTTTATA
   [4]   705 ATGGAAATGGAACGAA
   [5]  1170 ATGAAATATTTAGATG
   ...   ... ...
[2307]   234 ATGGACGAAGTTAAAA
[2308]   447 TTGGGAAGTCGTAACA
[2309]  1356 ATGTCTACTATACAAC
[2310]   219 GTGTCAGTTGAAAATG
[2311]   162 GTGTCAGGTAGGCAGT
> |
```

③



Sequence logos: Schneider and Stephens, *Nucleic Acids Res.*, **18**: 6097–6100, 1990

Contents

- 配列長でフィルタリング（ゲノムアセンブリ結果の後処理）
  - イントロ、本番
- ゲノムアノテーション
  - イントロ、参考文献
  - DFAST実行、課題1
- CDS（タンパク質コード領域の配列；coding sequenceの略）の取得
  - 比較ゲノム解析のイントロ、問題設定
  - 答え（cds.fna）を眺めておく、sequence logos
  - CDS配列取得の成功例、TxDbオブジェクト、CDS座標情報取得
  - CDS配列取得の失敗例、GFF3をExcelで概観
  - 基本テクを駆使して解決する基本戦略、CDSを含む行の抽出、補足説明
  - Excelで確認、リストファイルの作成、基本テクでCDS配列取得を実行
  - ストランド情報つきのリストファイルの作成、例題6のコード解説
  - ストランド情報反映戦略を練り、コードに修正を加えて実行（例題7）

参考

Sequence logos

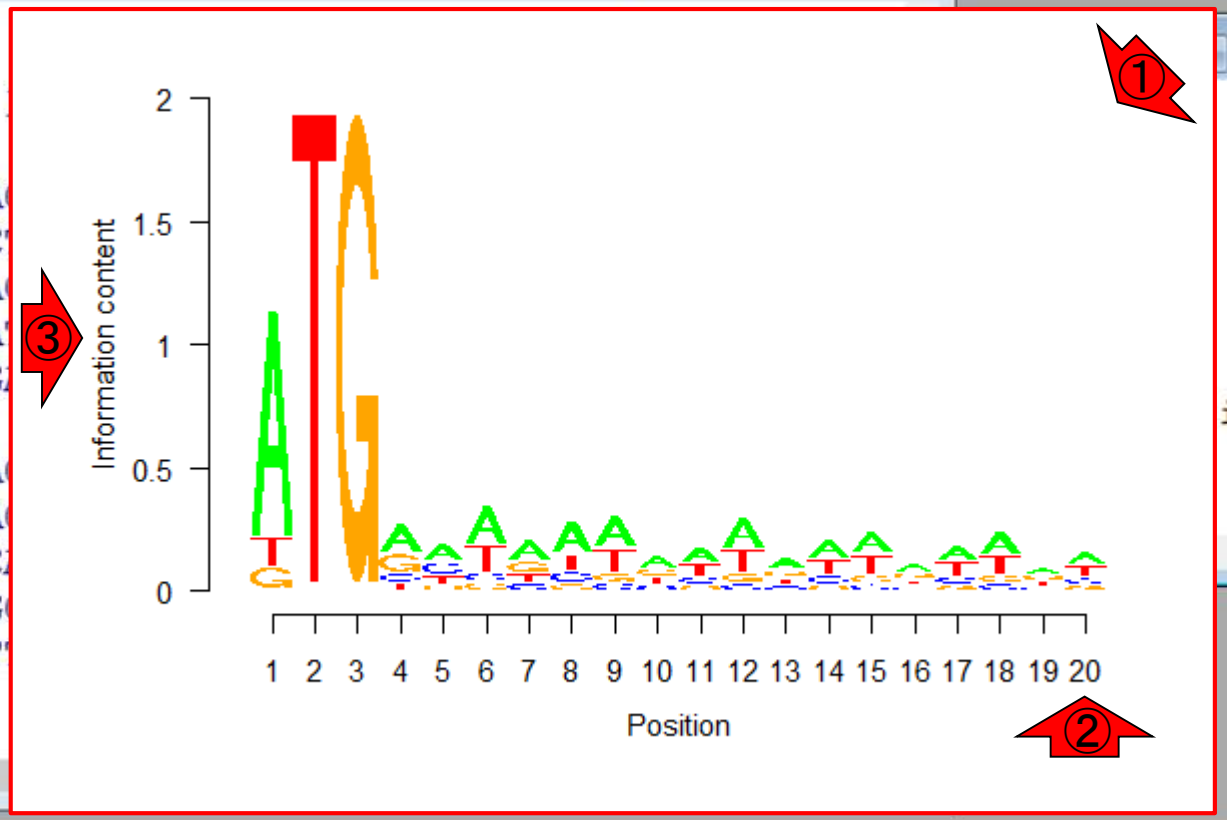
①はcds.fnaを入力として、②最初の20塩基に限定してSequence logosを作成したもの。③縦軸は情報量。

RGui (64-bit)

ファイル　編集　パッケージ　ウインドウ　ヘルプ

R Console

```
> fasta
  A DNAStringSet instance of
      width seq
   [1]   489 ATGTTAAAACAAATTA
   [2]   741 TTGAAAATTACAGTTC
   [3]   402 ATGGTTACACTTTATA
   [4]   705 ATGGAAATGGAACGAA
   [5]  1170 ATGAAATATTTAGATG
   ...   ... ...
[2307]   234 ATGGACGAAGTTAAAA
[2308]   447 TTGGGAAGTCGTAACA
[2309]  1356 ATGTCTACTATACAAC
[2310]   219 GTGTCAGTTGAAAATG
[2311]   162 GTGTCAGGTAGGCAGT
> |
```



参考

Sequence logos

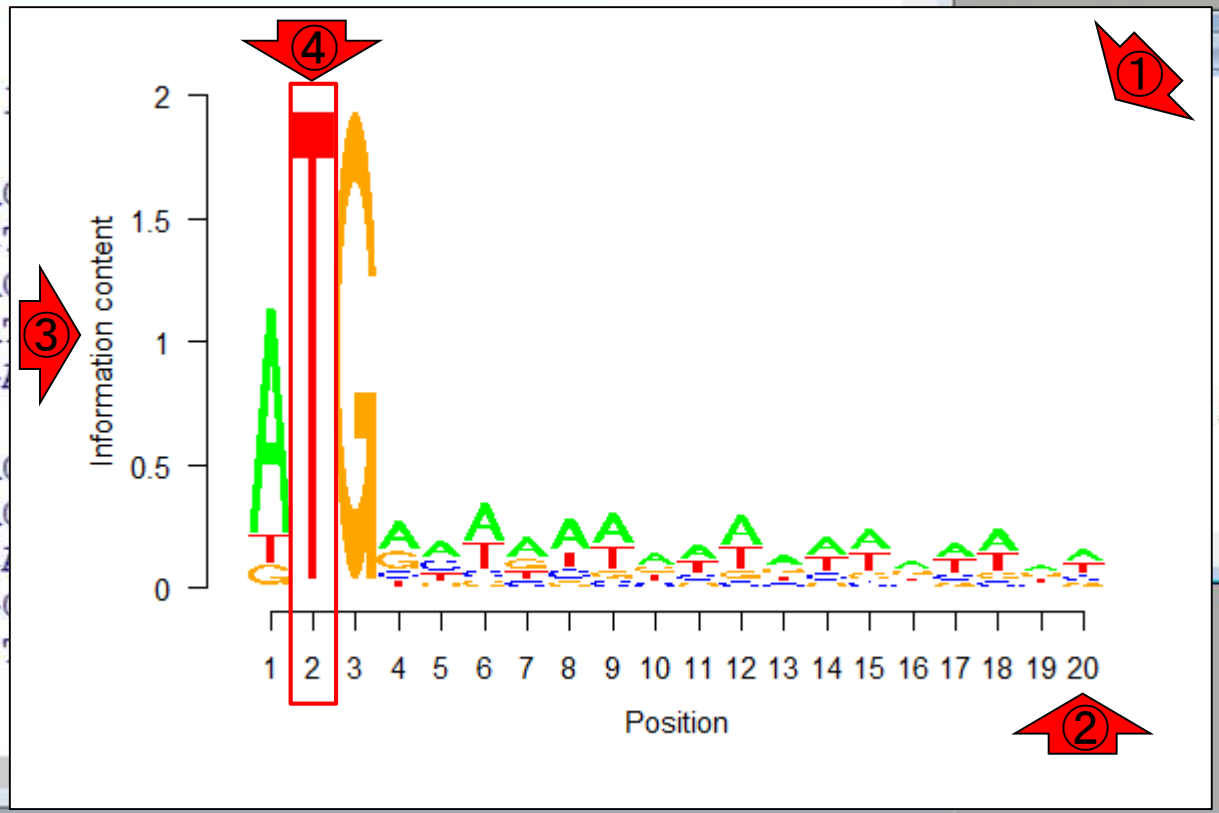
①はcds.fnaを入力として、②最初の20塩基に限定してSequence logosを作成したもの。③縦軸は情報量。例えば④2番目の位置はTのみからなっていることがわかる。

RGui (64-bit)

ファイル　編集　パッケージ　ウインドウ　ヘルプ

R Console

```
> fasta
  A DNAStringSet instance of
      width
   [1]    489 ATGTTAAAACAAATTA
   [2]    741 TTGAAAATTACAGTTC
   [3]    402 ATGGTTACACTTTATA
   [4]    705 ATGGAAATGGAACGAA
   [5]   1170 ATGAAATATTTAGATG
   ...    ... ...
[2307]    234 ATGGACGAAGTTAAAA
[2308]    447 TTGGGAAGTCGTAACA
[2309]   1356 ATGTCTACTATACAAC
[2310]    219 GTGTCAGTTGAAAATG
[2311]    162 GTGTCAGGTAGGCAGT
> |
```



参考

Sequence logos

①はcds.fnaを入力として、②最初の20塩基に限定してSequence logosを作成したもの。③縦軸は情報量。例えば④2番目の位置はTのみからなっていることがわかる。⑤Tの一番上が、⑥2の少し下になっていますが、④のポジションは実際には全てTなので、最大値の2です。文字を積み上げた高さが実際の値よりも若干低くなっているが気にしない。

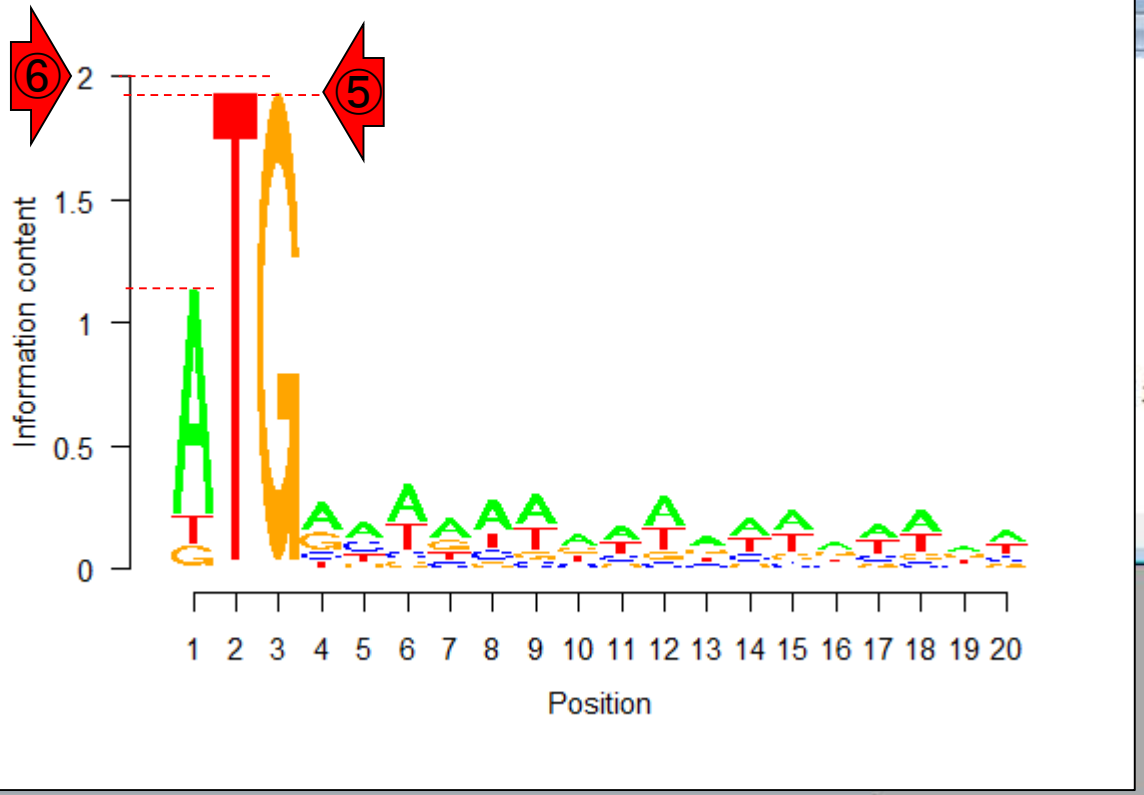
RGui (64-bit)

ファイル 編集 パッケージ ウインドウ ヘルプ

R Console

```
> fasta
  A DNAStringSet instance of
      width
   [1]    489 ATGTTAAAACAAATTA
   [2]    741 TTGAAAATTACAGTTC
   [3]    402 ATGGTTACACTTTATA
   [4]    705 ATGGAAATGGAACGAA
   [5]   1170 ATGAAATATTTAGATG
   ...    ... ...
[2307]    234 ATGGACGAAGTTAAAA
[2308]    447 TTGGGAAGTCGTAACA
[2309]   1356 ATGTCTACTATACAAC
[2310]    219 GTGTCAGTTGAAAATG
[2311]    162 GTGTCAGGTAGGCAGT
> |
```

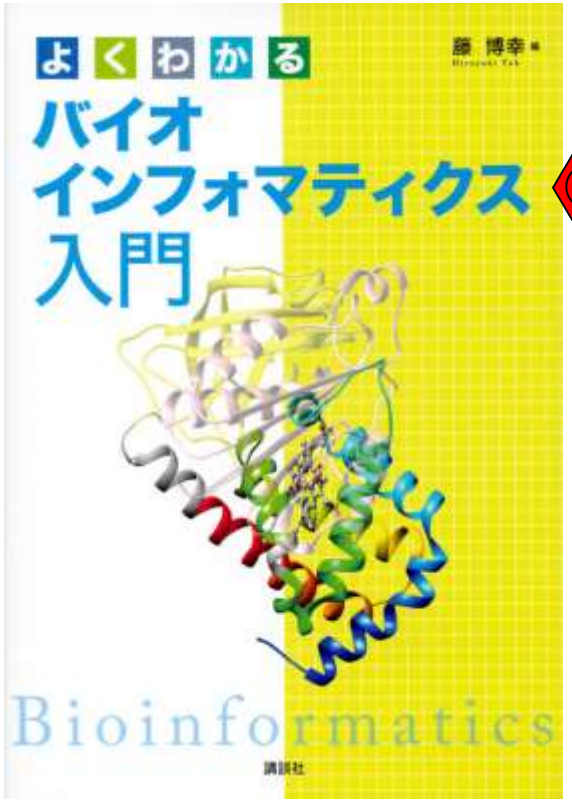
④



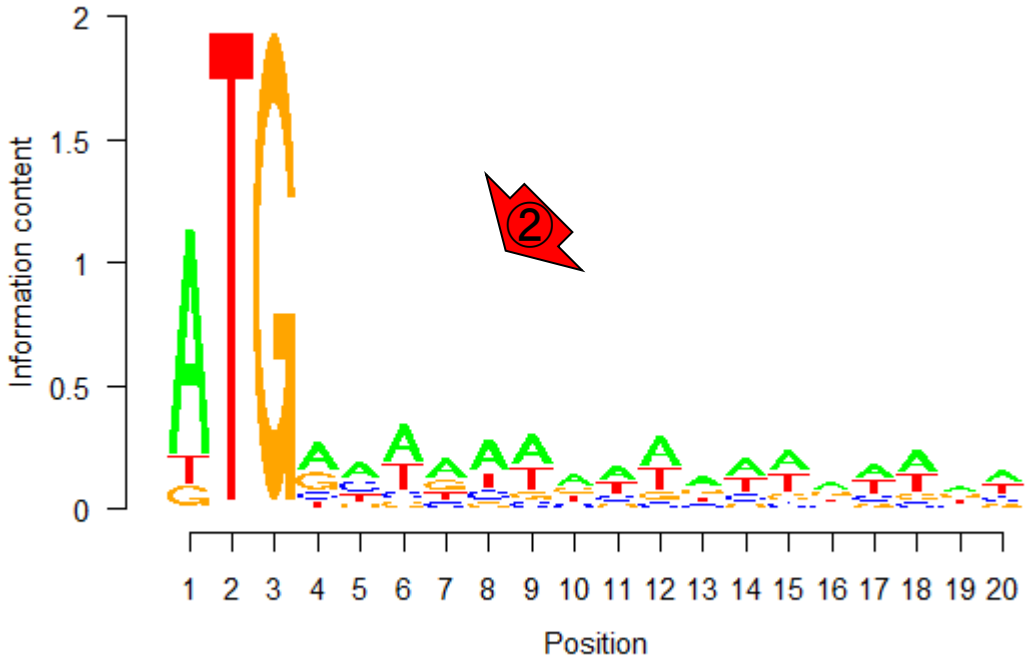
参考

Sequence logos

Sequence logosについては、①参考図書の第6章ゲノム解析（p91-92）でも述べられています。ちなみに②この図は…



①

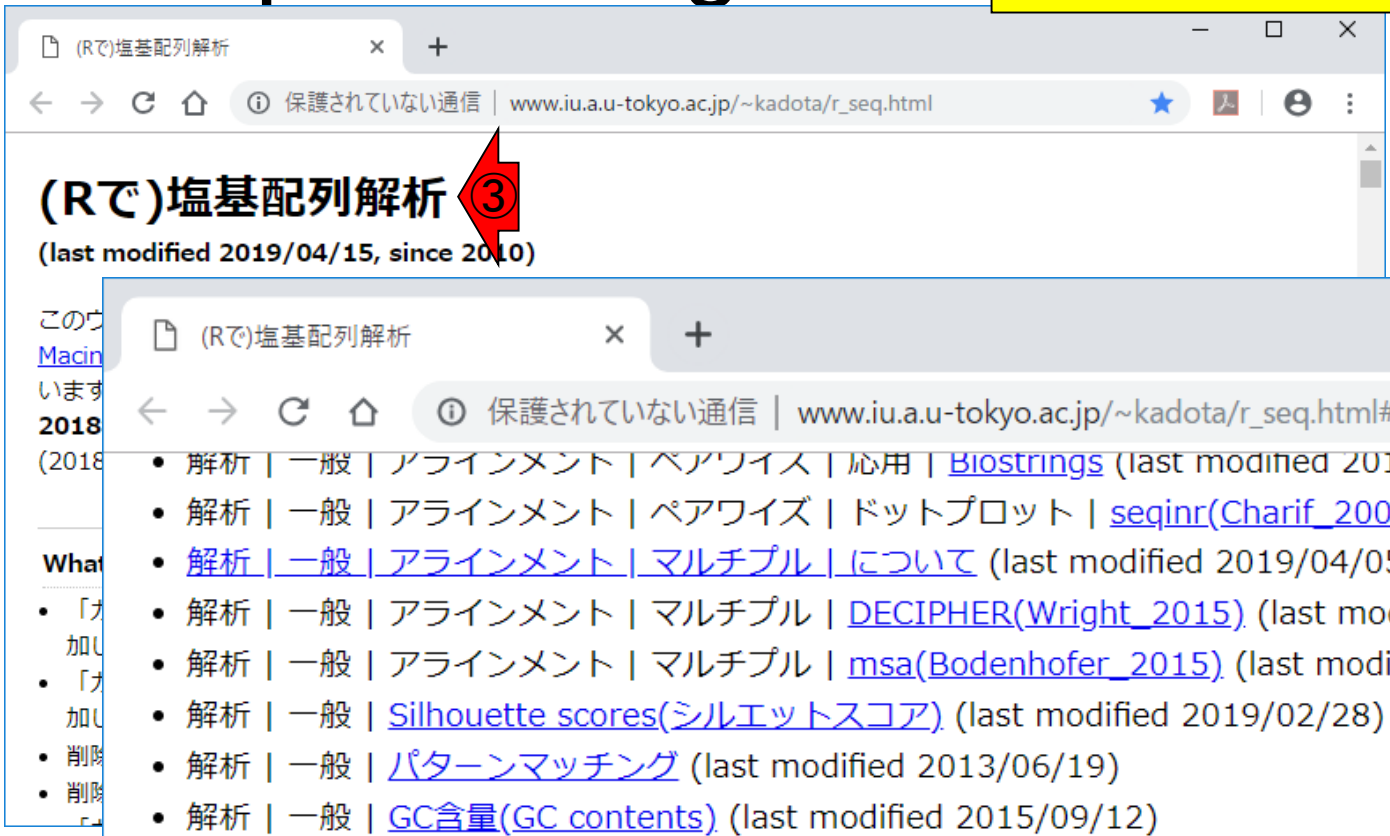


②

参考

Sequence logos

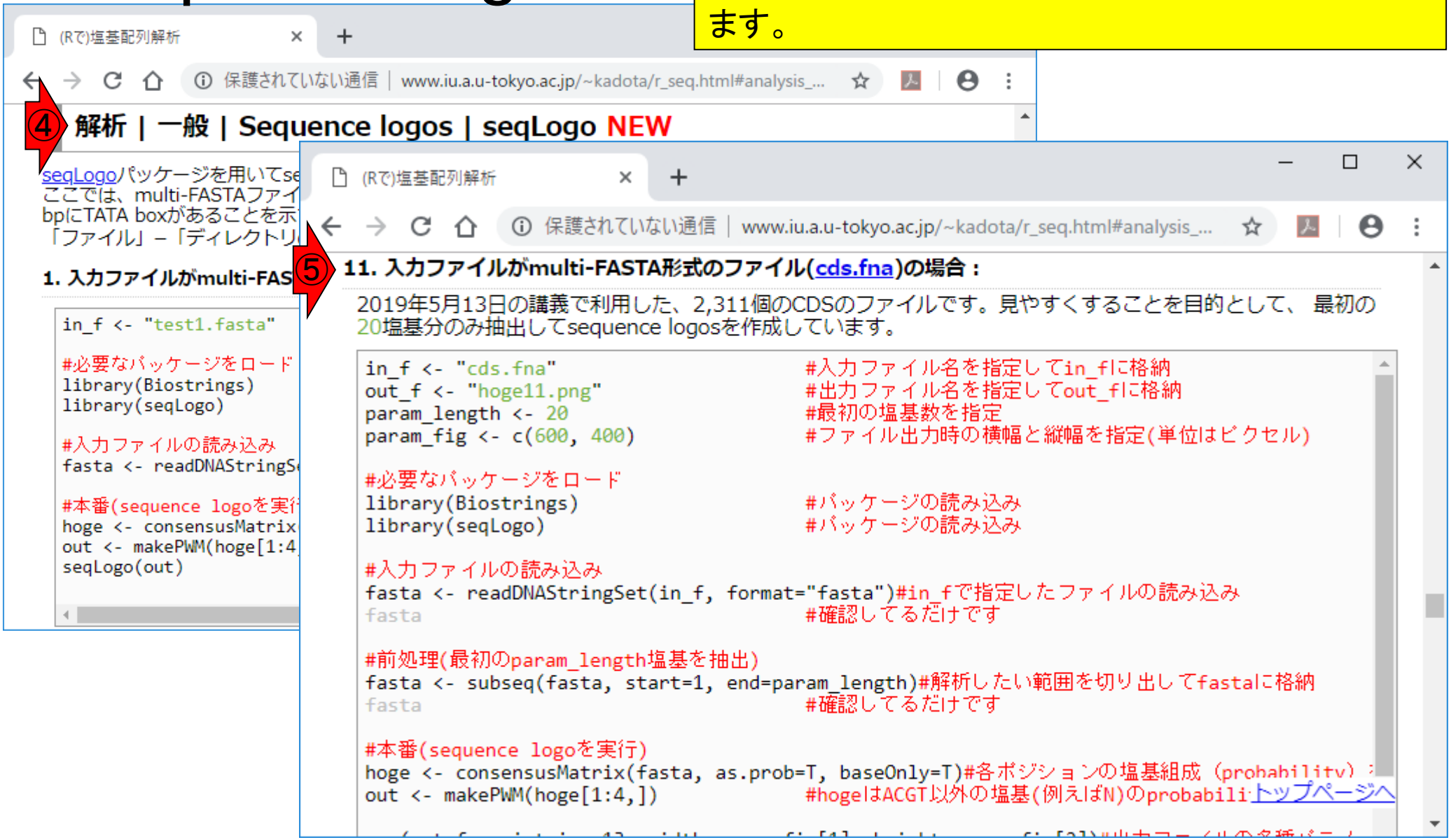
Sequence logosについては、①参考図書の第6章ゲノム解析（p91-92）でも述べられています。ちなみに②この図は、④の…



参考

Sequence logos

Sequence logosについては、①参考図書の第6章ゲノム解析（p91-92）でも述べられています。ちなみに②この図は、④の、⑤例題11のコピペ実行で得られます。



参考

Sequence logos

Sequence logosについては、①参考図書の第6章ゲノム解析（p91-92）でも述べられています。ちなみに②この図は、④の、⑤例題11のコピペ実行で得られます。⑥が部分配列の取得を、⑦subseq関数で行っているところ。⑧1塩基目から、⑨20塩基目までの部分配列を抽出した結果を、再びfastaオブジェクトに格納していることがわかります。

(Rで)塩基配列解析

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

④ 解析 | 一般 | Sequence logos | seqLogo NEW

seqLogoパッケージを用いてse
ここでは、multi-FASTAファイ
bpにTATA boxがあることを示
「ファイル」－「ディレクトリ

1. 入力ファイルがmulti-FAS

```
in_f <- "test1.fasta"

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

#入力ファイルの読み込み
fasta <- readDNAStringS

#本番(sequence logoを実
hoge <- consensusMatrix
out <- makePWM(hoge[1:4
seqLogo(out)
```

(Rで)塩基配列解析

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

⑤ 11. 入力ファイルがmulti-FASTA形式のファイル(cds.fna)の場合：

2019年5月13日の講義で利用した、2,311個のCDSのファイルです。見やすくすることを目的として、 最初の20塩基分のみ抽出してsequence logosを作成しています。

```
in_f <- "cds.fna"                          #入力ファイル名を指定してin_fに格納
out_f <- "hoge11.png"                      #出力ファイル名を指定してout_fに格納
param_length <- 20                         #最初の塩基数を指定
param_fig <- c(600, 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(seqLogo)                           #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

#前処理(最初のparam_length塩基を抽出)
fasta <- subseq(fasta, start=1, end=param_length)#解析したい範囲を切り出してfastaに格納
fasta                                      #確認してるだけです

#本番(sequence logoを実行)
hoge <- consensusMatrix(fasta, as.prob=T, baseOnly=T)#各ポジションの塩基組成（probability）
out <- makePWM(hoge[1:4,])                 #hogeはACGT以外の塩基(例えばN)のprobabili
```

⑥ ⑦ ⑧ ⑨

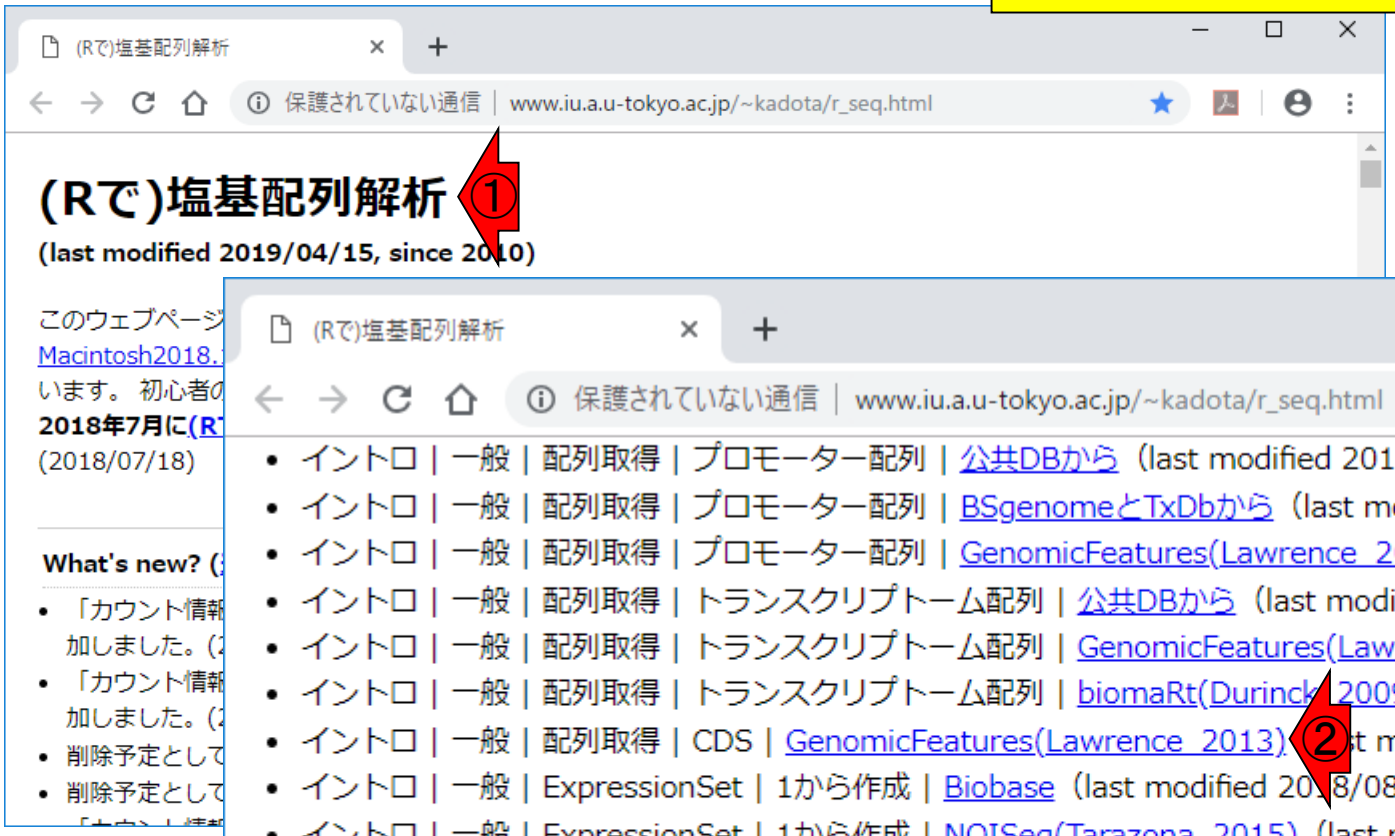
トップページへ

Contents

- 配列長でフィルタリング（ゲノムアセンブリ結果の後処理）
  - イントロ、本番
- ゲノムアノテーション
  - イントロ、参考文献
  - DFAST実行、課題1
- CDS（タンパク質コード領域の配列；coding sequenceの略）の取得
  - 比較ゲノム解析のイントロ、問題設定
  - 答え（cds.fna）を眺めておく、sequence logos
  - CDS配列取得の成功例、TxDbオブジェクト、CDS座標情報取得
  - CDS配列取得の失敗例、GFF3をExcelで概観
  - 基本テクを駆使して解決する基本戦略、CDSを含む行の抽出、補足説明
  - Excelで確認、リストファイルの作成、基本テクでCDS配列取得を実行
  - ストランド情報つきのリストファイルの作成、例題6のコード解説
  - ストランド情報反映戦略を練り、コードに修正を加えて実行（例題7）

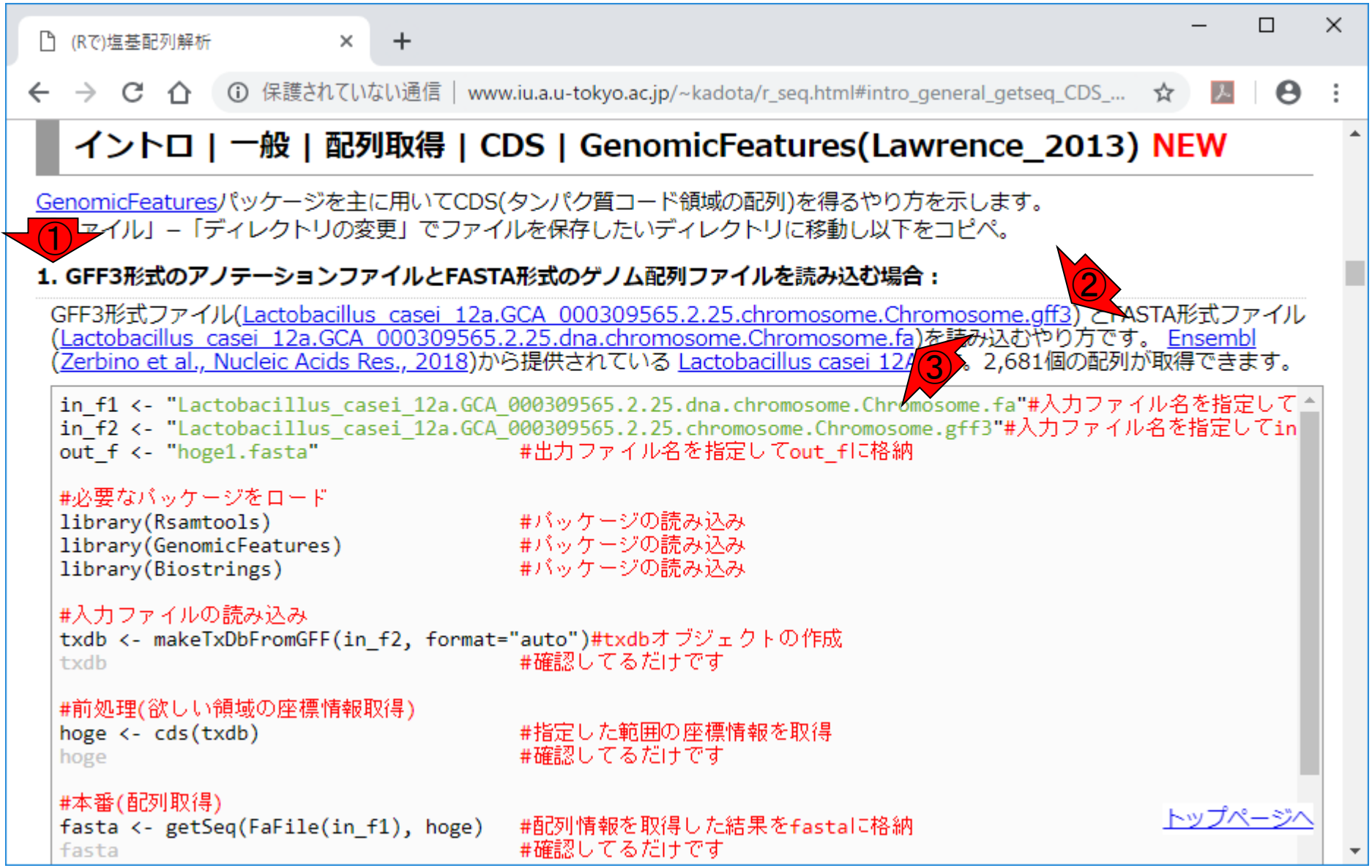
CDS配列取得

②では、FASTAファイルとGFF3ファイルを入力として、CDS配列取得を行うやり方を示しています。例題1が成功例、例題2が失敗例です。



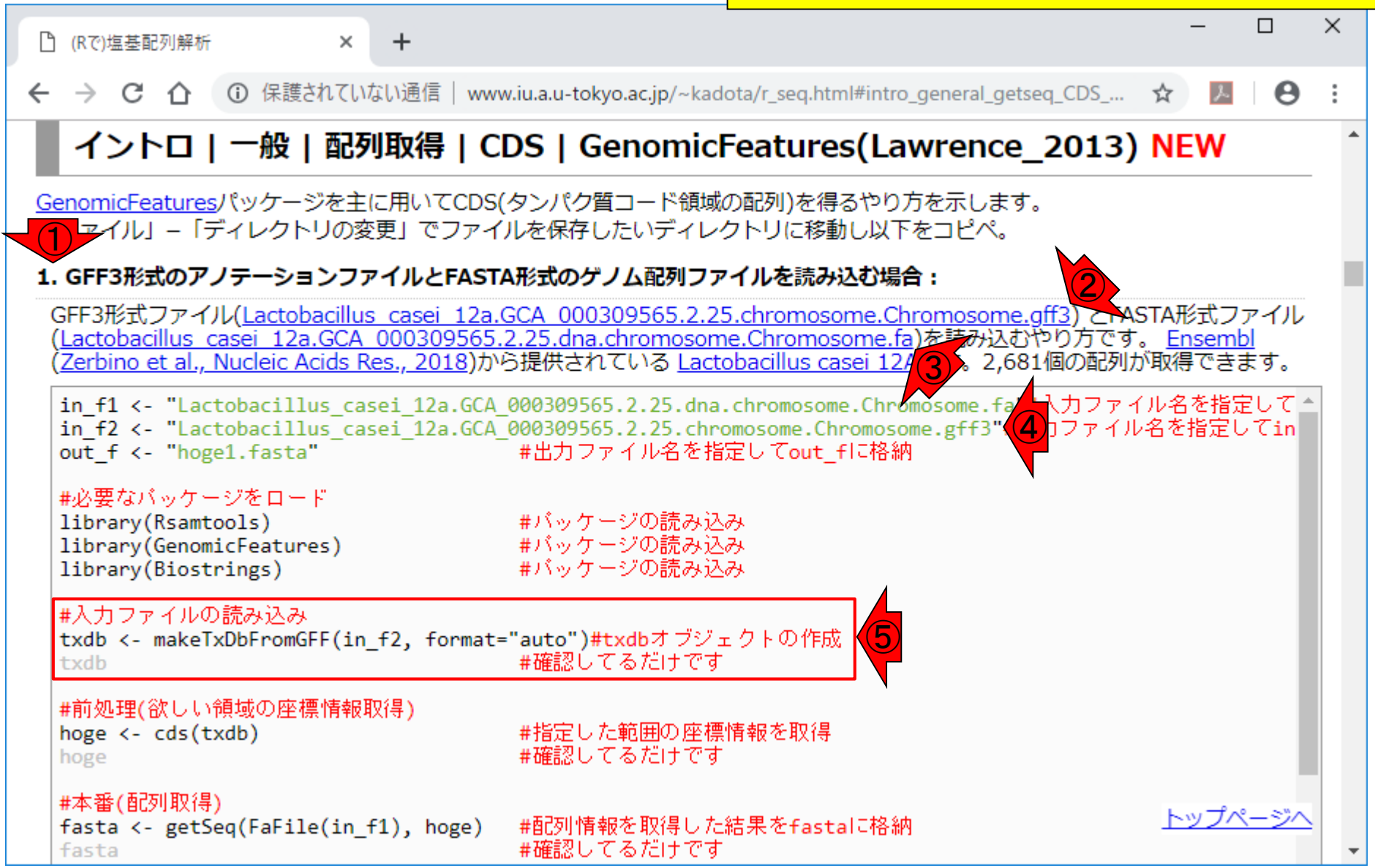
成功例

①例題1が成功例です。入力ファイルは、②と③です。デスクトップにダウンロード。



成功例

①例題1が成功例です。入力ファイルは、②と③です。デスクトップにダウンロード。④GFF3ファイルの読込みは、⑤で行っています。



成功例

①例題1が成功例です。入力ファイルは、②と③です。デスクトップにダウンロード。④GFF3ファイルの読込みは、⑤で行っています。赤枠内をコピペ実行して、⑥GFF3ファイルの情報であるtxdbを眺める。

(Rで)塩基配列解析

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

イントロ | 一般 | 配列取得 | CDS | GenomicFeatures(Lawrence_2013) NEW

GenomicFeaturesパッケージを主に用いてCDS(タンパク質コード領域の配列)を得るやり方を示します。

①「ファイル」－「ディレクトリの変更」でファイルを保存したいディレクトリに移動し以下をコピペ。

1. GFF3形式のアノテーションファイルとFASTA形式のゲノム配列ファイルを読み込む場合：

GFF3形式ファイル(Lactobacillus_casei_12a.GCA_000309565.2.25.chromosome.Chromosome.gff3)とFASTA形式ファイル(Lactobacillus_casei_12a.GCA_000309565.2.25.dna.chromosome.Chromosome.fa)を読み込むやり方です。Ensembl(Zerbino et al., Nucleic Acids Res., 2018)から提供されている Lactobacillus casei 12A。2,681個の配列が取得できます。

②

③

④

```
in_f1 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.dna.chromosome.Chromosome.fa"#入力ファイル名を指定して
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.chromosome.Chromosome.gff3"#入力ファイル名を指定してin
out_f <- "hoge1.fasta"                    #出力ファイル名を指定してout_fに格納

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

#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="auto")#txdbオブジェクトの作成
txdb                                      #確認してるだけです

#前準備(欲しい領域の座標情報取得)
hoge <- cds(txdb)                         #指定した範囲の座標情報を取得
hoge                                      #確認してるだけです

#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)      #配列情報を取得した結果をfastaに格納
fasta                                     #確認してるだけです
```

⑤

⑥

トップページへ

成功例

①例題1が成功例です。入力ファイルは、②と③です。デスクトップにダウンロード。④GFF3ファイルの読込みは、⑤で行っています。赤枠内をコピペ実行して、⑥GFF3ファイルの情報であるtxdbを眺める。
うまく読み込めたら⑦のような感じになります。

(Rで)塩基配列解析

保護されていない通信 | www.iu.a.u

イントロ | 一般 | 配列取得 | CDS

GenomicFeaturesパッケージを主に用いてCDS(タン
「ファイル」－「ディレクトリの変更」でファイルを

1. GFF3形式のアノテーションファイルとFASTA形

GFF3形式ファイル(Lactobacillus_casei_12a.GCA
(Lactobacillus_casei_12a.GCA_000309565.2.25.
(Zerbino et al., Nucleic Acids Res., 2018)から提供

```
in_f1 <- "Lactobacillus_casei_12a.GCA_0003
in_f2 <- "Lactobacillus_casei_12a.GCA_0003
out_f <- "hoge1.fasta"                    #出

#必要なパッケージをロード
library(Rsamtools)                        #パ
library(GenomicFeatures)                  #パ
library(Biostrings)                       #パ

#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="aut
txdb                                      #確

#前準備(欲しい領域の座標情報取得)
hoge <- cds(txdb)                         #指
hoge                                      #確

#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)      #配
fasta                                     #確
```

⑥

⑦

RGui (64-bit)

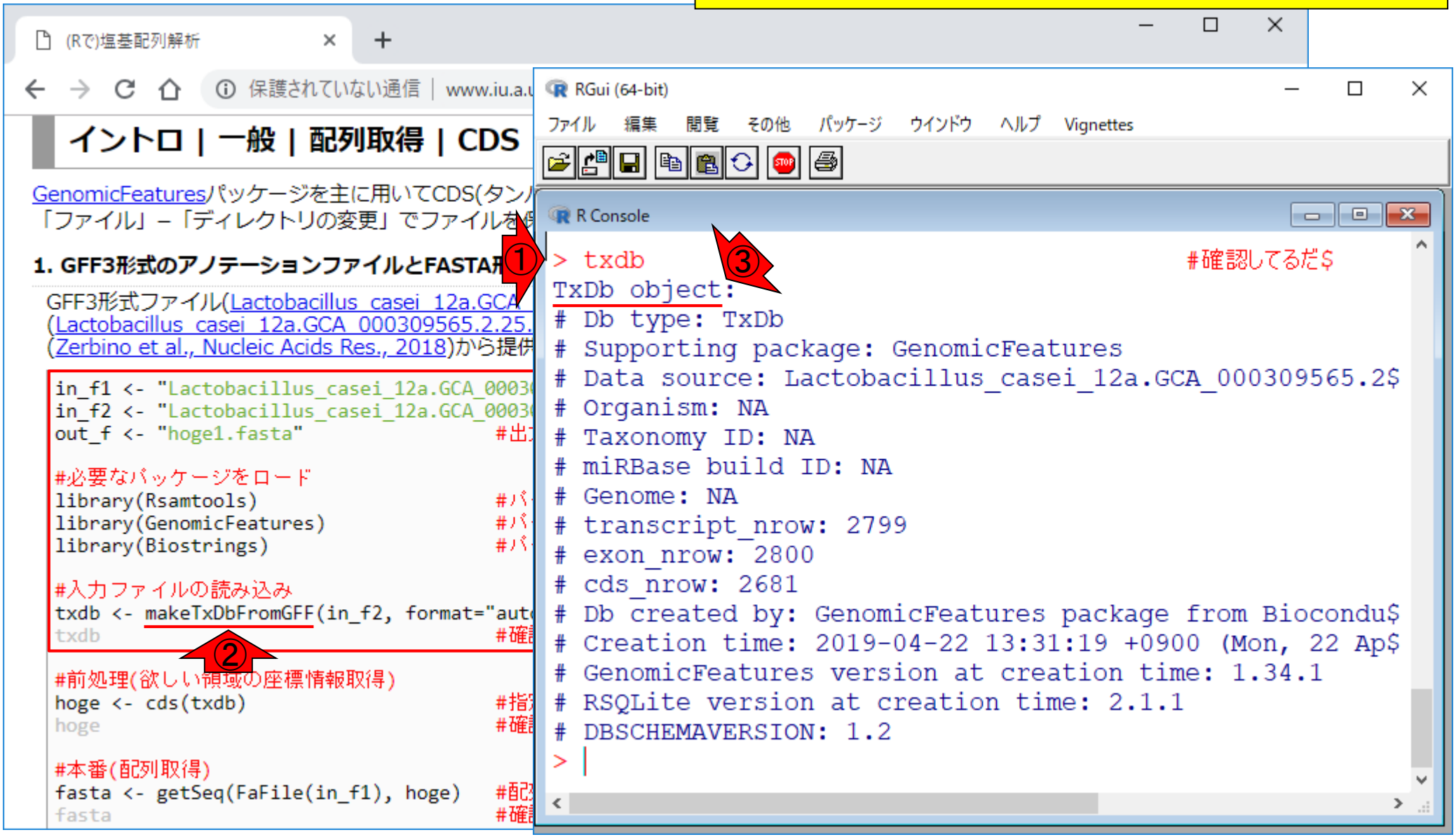
ファイル 編集 閲覧 その他 パッケージ ウインドウ ヘルプ Vignettes

R Console

```
> txdb                                    #確認してるだ$
TxDb object:
# Db type: TxDb
# Supporting package: GenomicFeatures
# Data source: Lactobacillus_casei_12a.GCA_000309565.2$
# Organism: NA
# Taxonomy ID: NA
# miRBase build ID: NA
# Genome: NA
# transcript_nrow: 2799
# exon_nrow: 2800
# cds_nrow: 2681
# Db created by: GenomicFeatures package from Biocondu$
# Creation time: 2019-04-22 13:31:19 +0900 (Mon, 22 Ap$
# GenomicFeatures version at creation time: 1.34.1
# RSQLite version at creation time: 2.1.1
# DBSCHEMAVERSION: 1.2
>
```

TxDbオブジェクト

①は、②makeTxDbFromGFF関数を用いて、GFF3ファイルを読み込んで作成した、③TxDbという形式のオブジェクトです。



TxDbオブジェクト

①は、②makeTxDbFromGFF関数を用いて、GFF3ファイルを読み込んで作成した、③TxDbという形式のオブジェクトです。④CDSは2,681個であることがわかる。このような情報は、当然入力ファイルからも得られます。⑤transcriptが2,799個というのは入力ファイルから読み取れないが、そういうこともある。

(Rで)塩基配列解析

保護されていない通信 | www.iu.a.u

イントロ | 一般 | 配列取得 | CDS

GenomicFeaturesパッケージを主に用いてCDS(タン
「ファイル」－「ディレクトリの変更」でファイルを

1. GFF3形式のアノテーションファイルとFASTA形

GFF3形式ファイル(Lactobacillus_casei_12a.GCA
(Lactobacillus_casei_12a.GCA_000309565.2.25.
(Zerbino et al., Nucleic Acids Res., 2018)から提供

```
in_f1 <- "Lactobacillus_casei_12a.GCA_0003
in_f2 <- "Lactobacillus_casei_12a.GCA_0003
out_f <- "hoge1.fasta"                    #出

#必要なパッケージをロード
library(Rsamtools)                        #パ
library(GenomicFeatures)                  #パ
library(Biostrings)                       #パ

#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="aut
txdb                                      #確

#前処理(欲しい領域の座標情報取得)
hoge <- cds(txdb)                         #指
hoge                                      #確

#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)      #配
fasta                                     #確
```

RGui (64-bit)

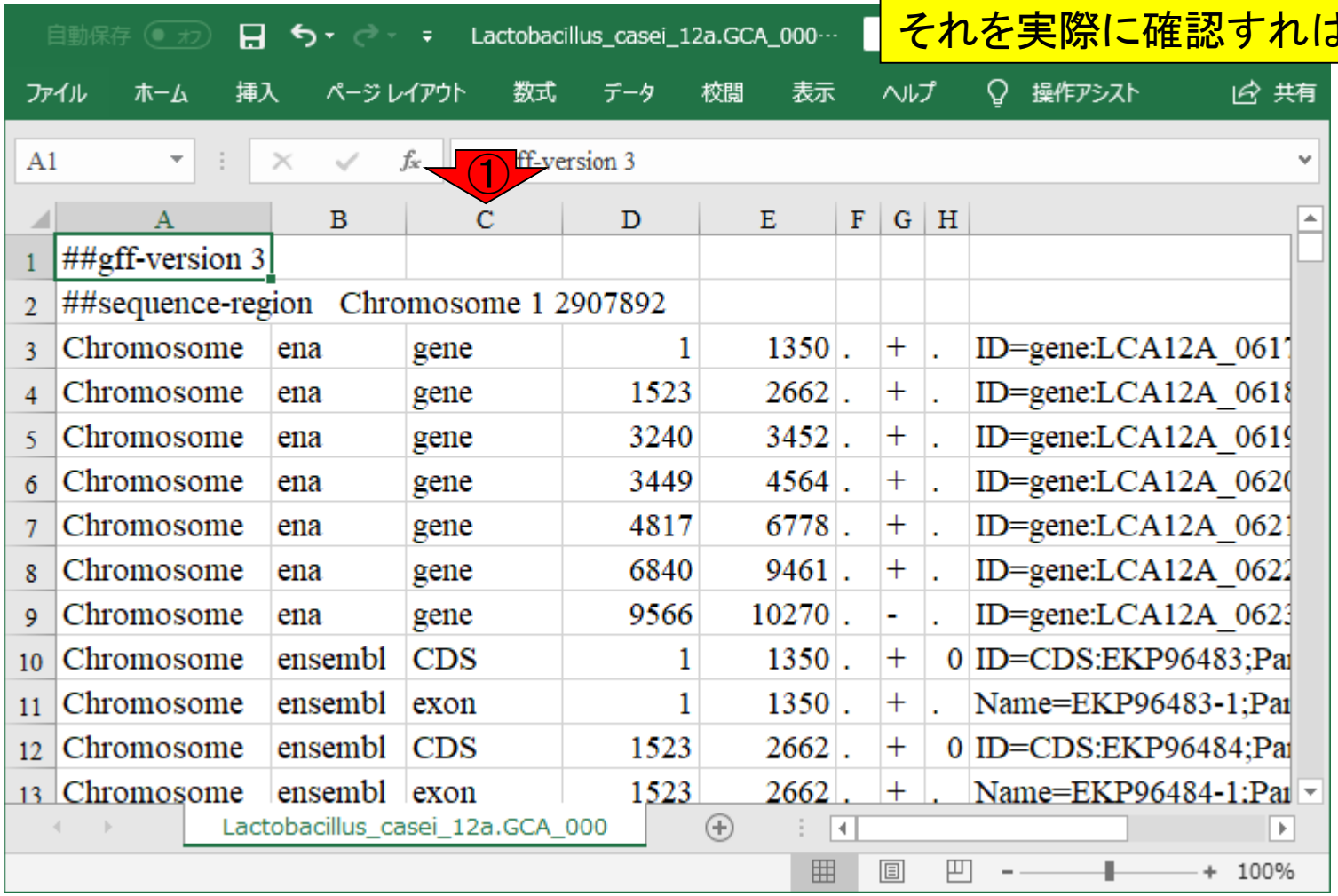
ファイル 編集

R Console

```
> txdb                                    #確認してるだ$
TxDb object:
# Db type: TxDb
# Supporting package: GenomicFeatures
# Data source: Lactobacillus_casei_12a.GCA_000309565.2$
# Organism: NA
# Taxonomy ID: NA
# miRBase build ID: NA
# Genome: NA
# transcript_nrow: 2799
# exon_nrow: 2800
# cds_nrow: 2681
# Db created by: GenomicFeatures package from Biocondu$
# Creation time: 2019-04-22 13:31:19 +0900 (Mon, 22 Ap$
# GenomicFeatures version at creation time: 1.34.1
# RSQLite version at creation time: 2.1.1
# DBSCHEMAVERSION: 1.2
>
```

GFF3をExcelで概観

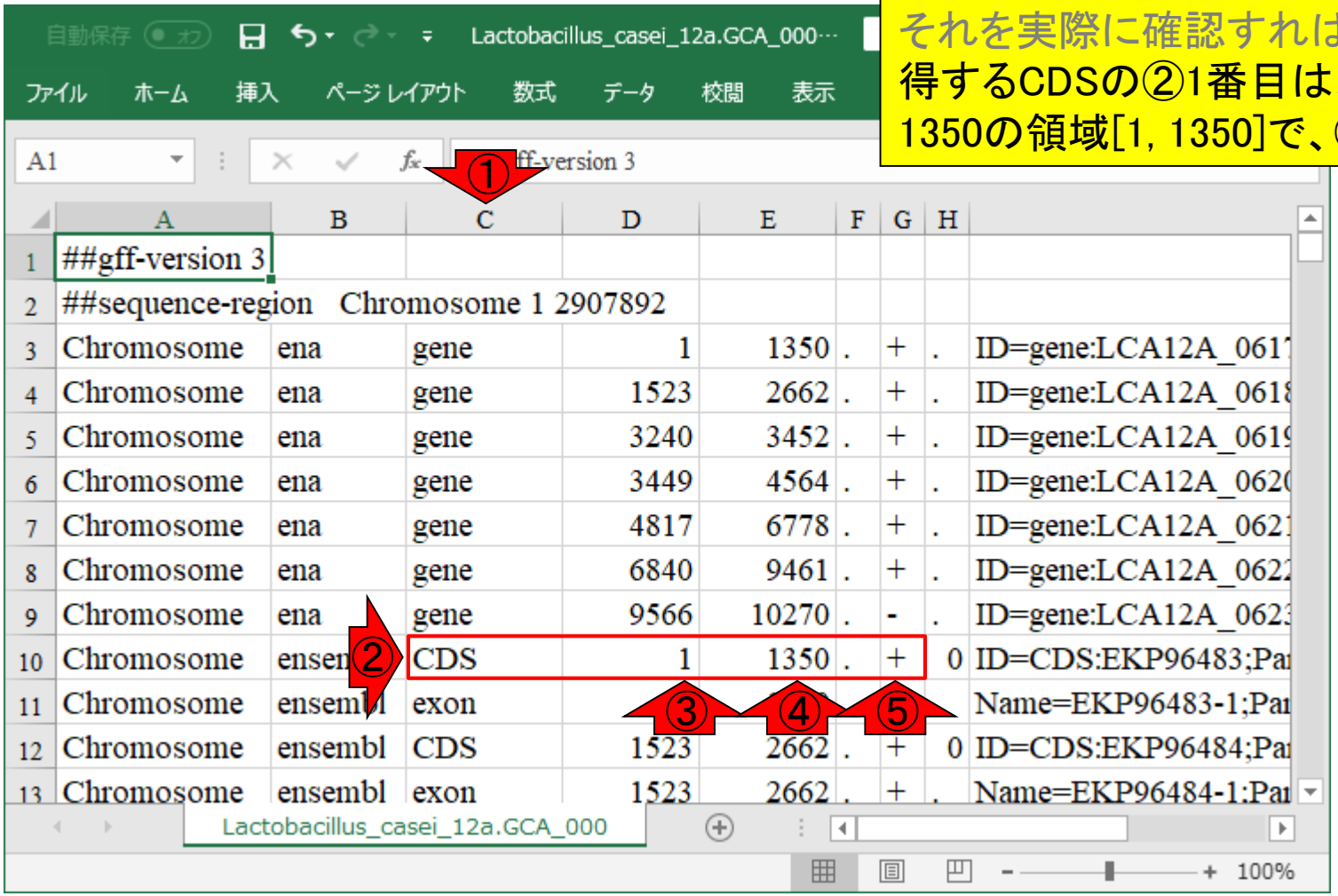
GFF3ファイルをExcelで読み込んだ結果のスクリーンショット。例えば今着目している2,681個のCDSは、①3列目がCDSの行数が2,681なのだろうと予想して、それを実際に確認すれば理解も深まる。



	A	B	C	D	E	F	G	H	
1	##gff-version 3								
2	##sequence-region	Chromosome 1 2907892							
3	Chromosome	ena	gene	1	1350	.	+	.	ID=gene:LCA12A_061
4	Chromosome	ena	gene	1523	2662	.	+	.	ID=gene:LCA12A_061
5	Chromosome	ena	gene	3240	3452	.	+	.	ID=gene:LCA12A_061
6	Chromosome	ena	gene	3449	4564	.	+	.	ID=gene:LCA12A_062
7	Chromosome	ena	gene	4817	6778	.	+	.	ID=gene:LCA12A_062
8	Chromosome	ena	gene	6840	9461	.	+	.	ID=gene:LCA12A_062
9	Chromosome	ena	gene	9566	10270	.	-	.	ID=gene:LCA12A_062
10	Chromosome	ensembl	CDS	1	1350	.	+	0	ID=CDS:EKP96483;Pa
11	Chromosome	ensembl	exon	1	1350	.	+	.	Name=EKP96483-1;Pa
12	Chromosome	ensembl	CDS	1523	2662	.	+	0	ID=CDS:EKP96484;Pa
13	Chromosome	ensembl	exon	1523	2662	.	+	.	Name=EKP96484-1;Pa

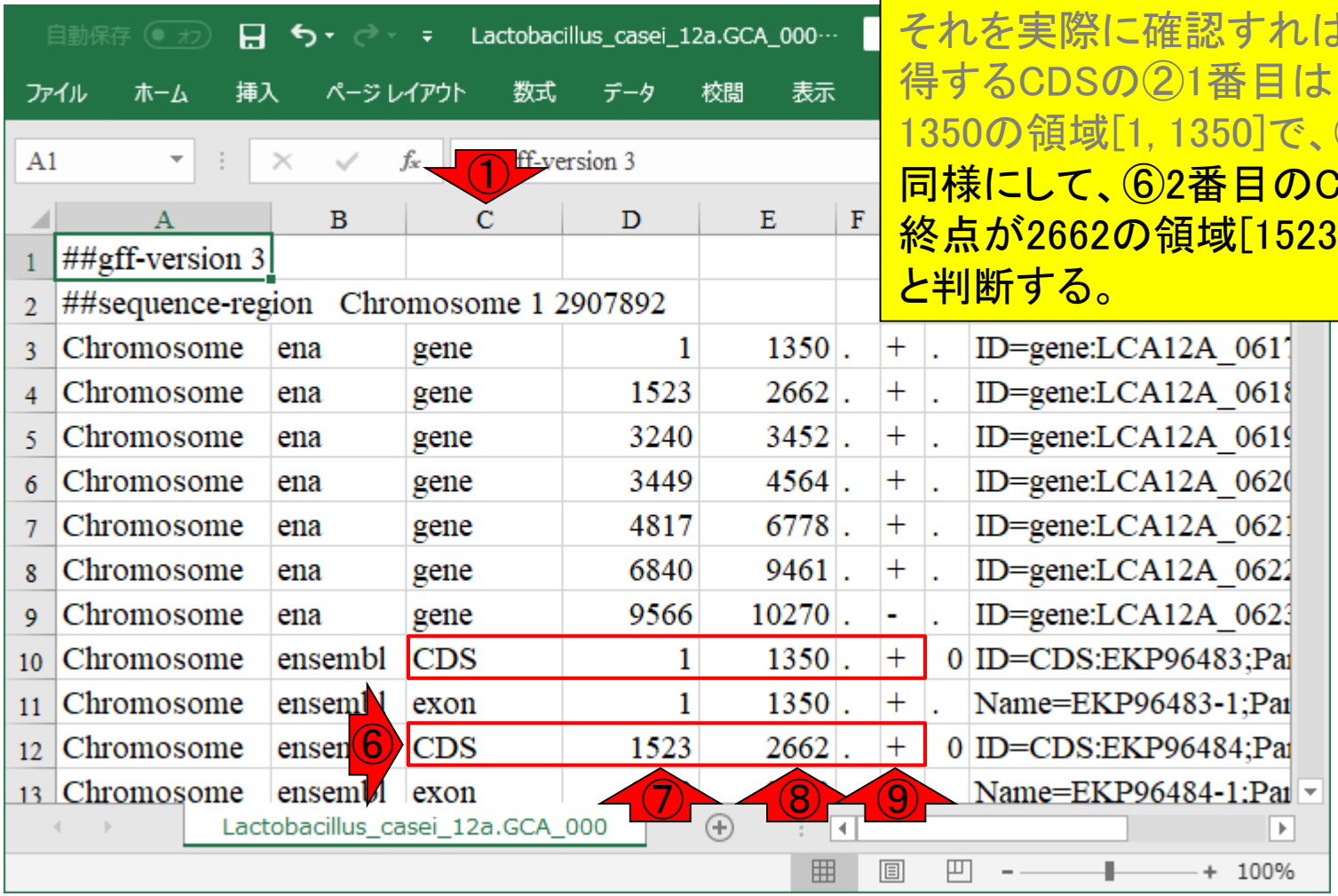
GFF3をExcelで概観

GFF3ファイルをExcelで読み込んだ結果のスクリーンショット。例えば今着目している2,681個のCDSは、①3列目がCDSの行数が2,681なのだろうと予想して、それを実際に確認すれば理解も深まる。そして、取得するCDSの②1番目は、③始点が1、④終点が1350の領域[1, 1350]で、⑤+鎖上のものと判断する。



	A	B	C	D	E	F	G	H	
1	##gff-version 3								
2	##sequence-region	Chromosome 1 2907892							
3	Chromosome	ena	gene	1	1350	.	+	.	ID=gene:LCA12A_061
4	Chromosome	ena	gene	1523	2662	.	+	.	ID=gene:LCA12A_061
5	Chromosome	ena	gene	3240	3452	.	+	.	ID=gene:LCA12A_061
6	Chromosome	ena	gene	3449	4564	.	+	.	ID=gene:LCA12A_062
7	Chromosome	ena	gene	4817	6778	.	+	.	ID=gene:LCA12A_062
8	Chromosome	ena	gene	6840	9461	.	+	.	ID=gene:LCA12A_062
9	Chromosome	ena	gene	9566	10270	.	-	.	ID=gene:LCA12A_062
10	Chromosome	ensembl	CDS	1	1350	.	+	0	ID=CDS:EKP96483;Par
11	Chromosome	ensembl	exon						Name=EKP96483-1;Par
12	Chromosome	ensembl	CDS	1523	2662	.	+	0	ID=CDS:EKP96484;Par
13	Chromosome	ensembl	exon	1523	2662	.	+	.	Name=EKP96484-1;Par

GFF3をExcelで概観

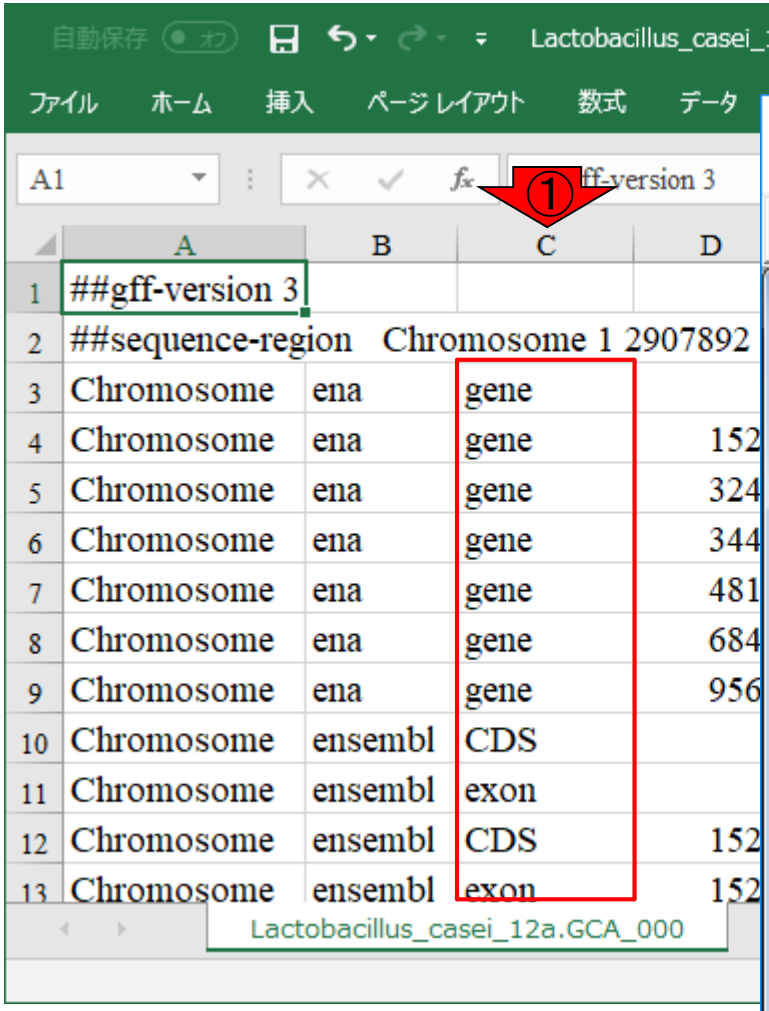


	A	B	C	D	E	F			
1	##gff-version 3								
2	##sequence-region	Chromosome 1 2907892							
3	Chromosome	ena	gene	1	1350	.	+	.	ID=gene:LCA12A_061
4	Chromosome	ena	gene	1523	2662	.	+	.	ID=gene:LCA12A_061
5	Chromosome	ena	gene	3240	3452	.	+	.	ID=gene:LCA12A_061
6	Chromosome	ena	gene	3449	4564	.	+	.	ID=gene:LCA12A_062
7	Chromosome	ena	gene	4817	6778	.	+	.	ID=gene:LCA12A_062
8	Chromosome	ena	gene	6840	9461	.	+	.	ID=gene:LCA12A_062
9	Chromosome	ena	gene	9566	10270	.	-	.	ID=gene:LCA12A_062
10	Chromosome	ensembl	CDS	1	1350	.	+	0	ID=CDS:EKP96483;Pa
11	Chromosome	ensembl	exon	1	1350	.	+	.	Name=EKP96483-1;Pa
12	Chromosome	ensembl	CDS	1523	2662	.	+	0	ID=CDS:EKP96484;Pa
13	Chromosome	ensembl	exon						Name=EKP96484-1;Pa

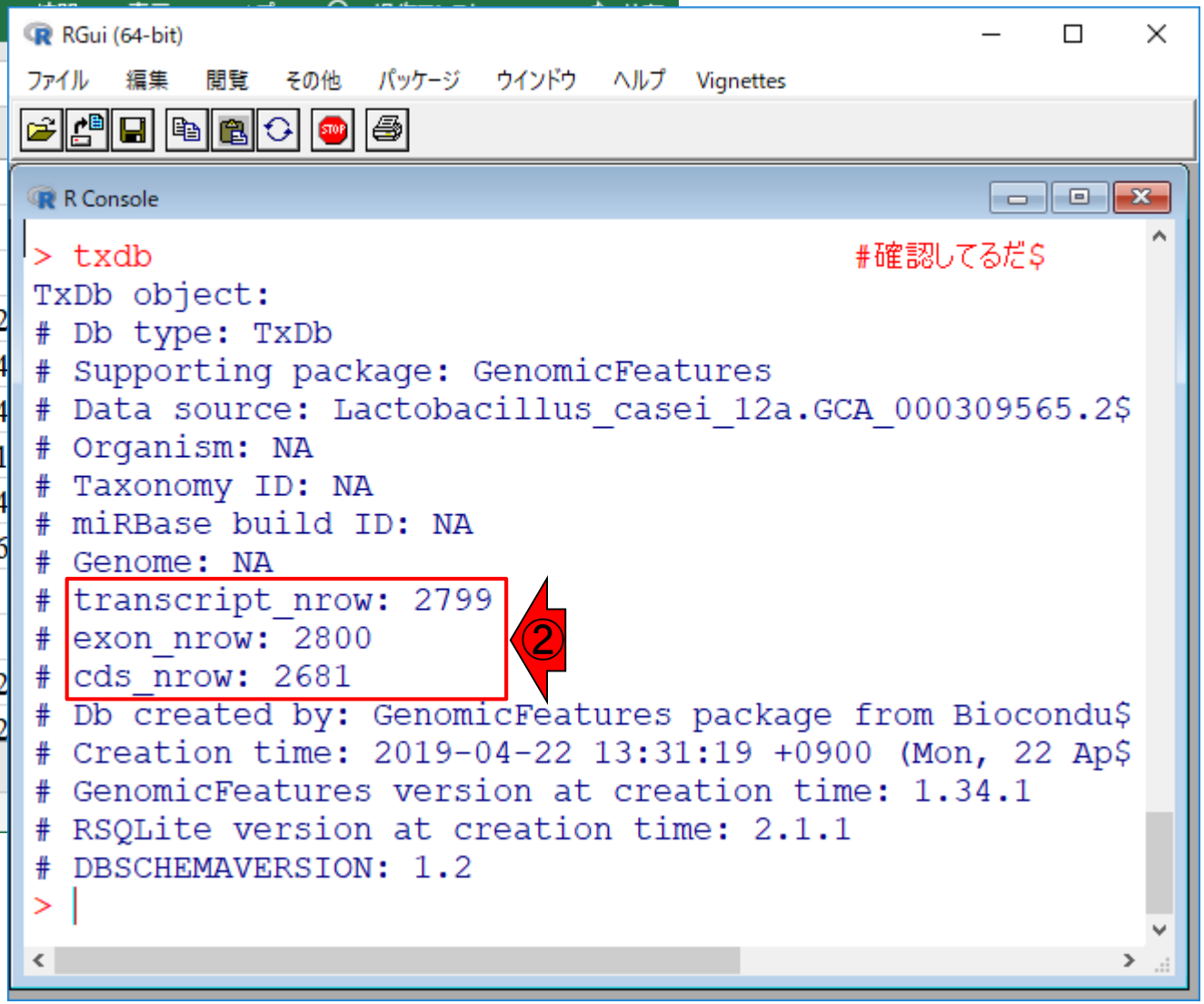
GFF3ファイルをExcelで読み込んだ結果のスクリーンショット。例えば今着目している2,681個のCDSは、①3列目がCDSの行数が2,681なのだろうと予想して、それを実際に確認すれば理解も深まる。そして、取得するCDSの②1番目は、③始点が1、④終点が1350の領域[1, 1350]で、⑤+鎖上のものと判断する。同様にして、⑥2番目のCDSは、⑦始点が1523、⑧終点が2662の領域[1523, 2662]で、⑨+鎖上のものと判断する。

GFF3をExcelで概観

GFFファイルは、①3列目を見てもわかるように、様々な情報を含んでいる。それゆえ、②のような感じで、featureごとに情報を格納していると考えればよい。



	A	B	C	D
1	##gff-version 3			
2	##sequence-region	Chromosome 1 2907892		
3	Chromosome	ena	gene	
4	Chromosome	ena	gene	152
5	Chromosome	ena	gene	324
6	Chromosome	ena	gene	344
7	Chromosome	ena	gene	481
8	Chromosome	ena	gene	684
9	Chromosome	ena	gene	956
10	Chromosome	ensembl	CDS	
11	Chromosome	ensembl	exon	
12	Chromosome	ensembl	CDS	152
13	Chromosome	ensembl	exon	152

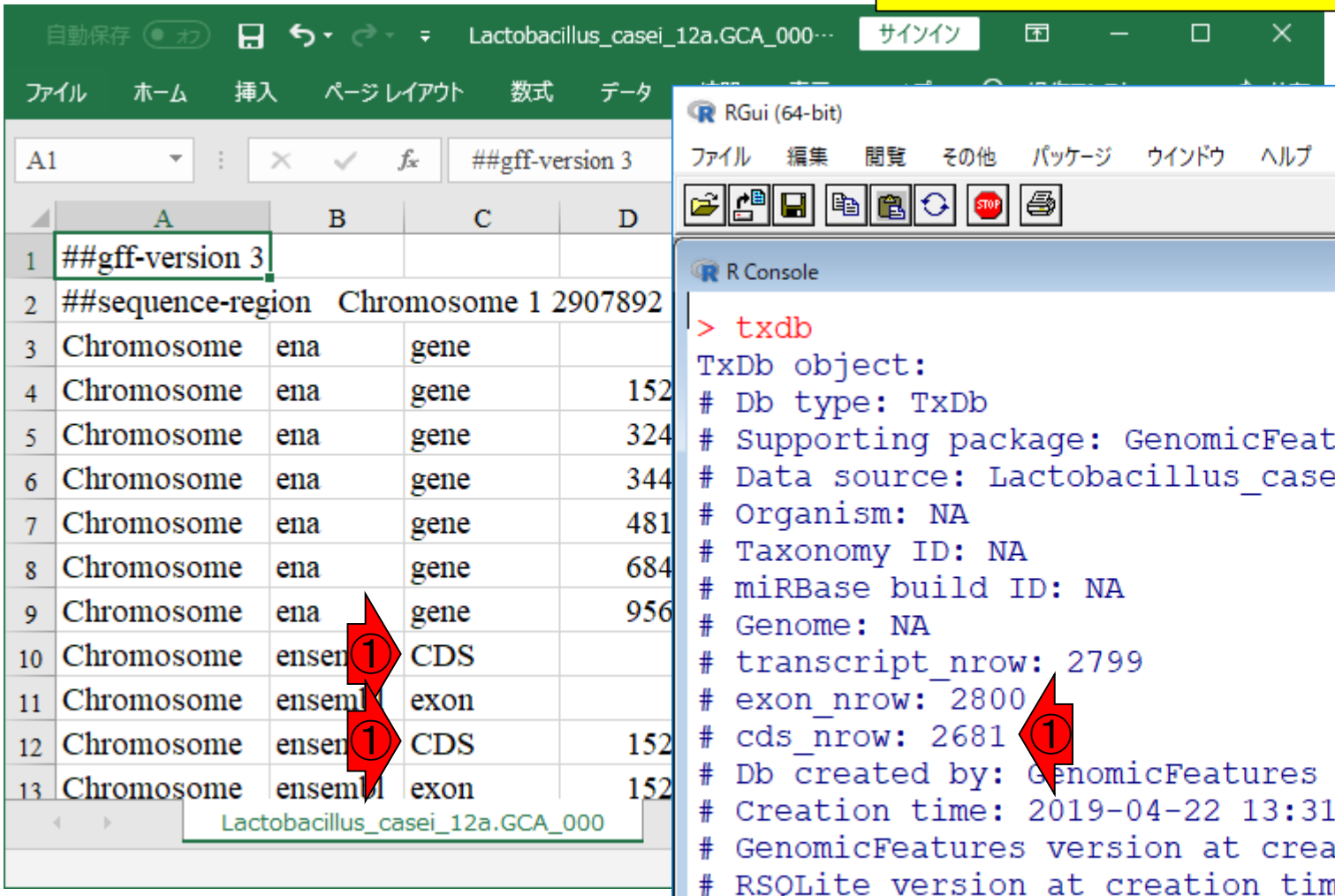


Contents

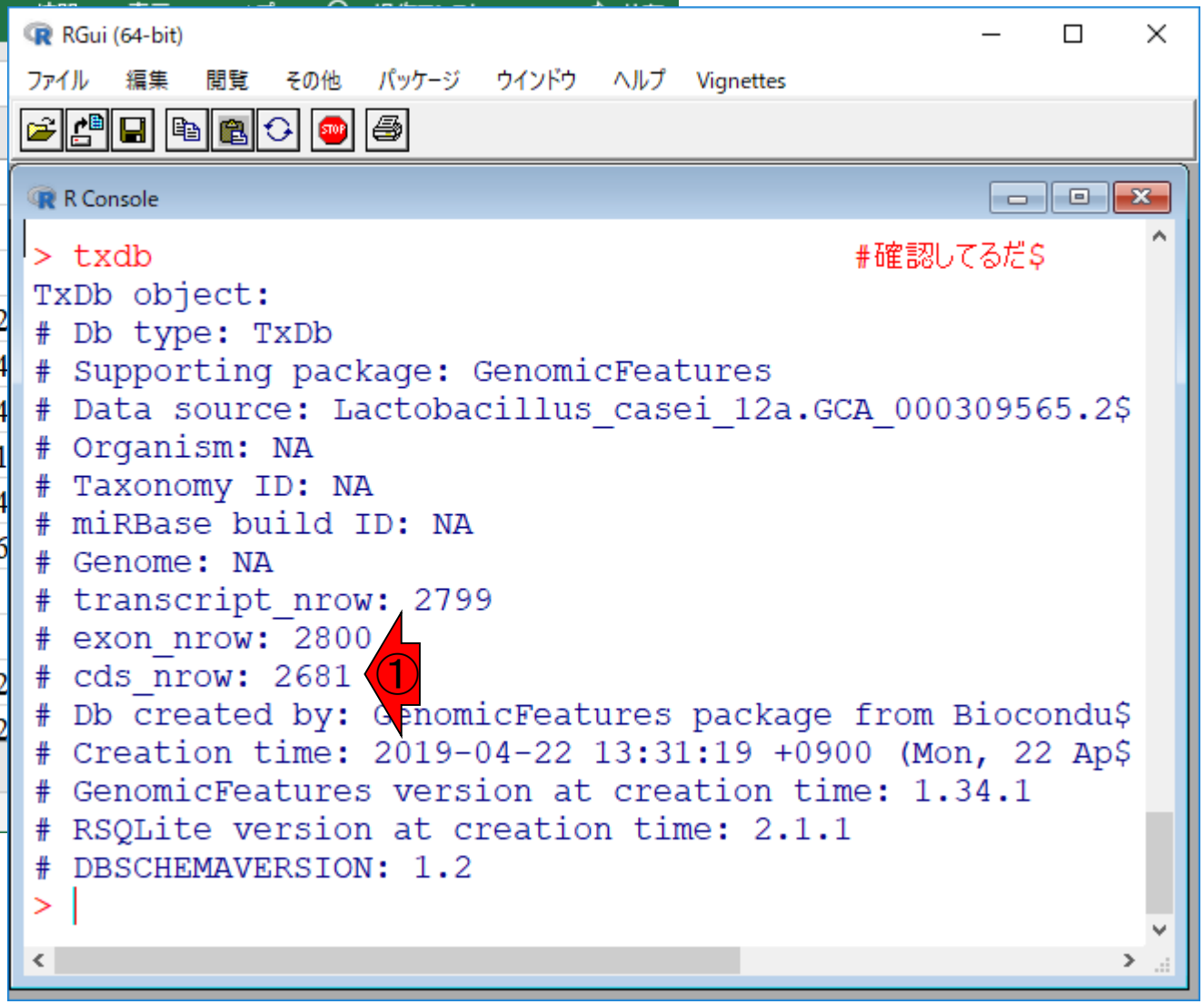
- 配列長でフィルタリング（ゲノムアセンブリ結果の後処理）
  - イントロ、本番
- ゲノムアノテーション
  - イントロ、参考文献
  - DFAST実行、課題1
- CDS（タンパク質コード領域の配列；coding sequenceの略）の取得
  - 比較ゲノム解析のイントロ、問題設定
  - 答え（cds.fna）を眺めておく、sequence logos
  - CDS配列取得の成功例、TxDbオブジェクト、CDS座標情報取得
  - CDS配列取得の失敗例、GFF3をExcelで概観
  - 基本テクを駆使して解決する基本戦略、CDSを含む行の抽出、補足説明
  - Excelで確認、リストファイルの作成、基本テクでCDS配列取得を実行
  - ストランド情報つきのリストファイルの作成、例題6のコード解説
  - ストランド情報反映戦略を練り、コードに修正を加えて実行（例題7）

CDS座標情報取得

今はCDSの配列情報取得が目的なので、①CDSの座標情報のみ取り出して利用したい。それを行っているのが…



	A	B	C	D
1	##gff-version 3			
2	##sequence-region	Chromosome 1 2907892		
3	Chromosome	ena	gene	
4	Chromosome	ena	gene	152
5	Chromosome	ena	gene	324
6	Chromosome	ena	gene	344
7	Chromosome	ena	gene	481
8	Chromosome	ena	gene	684
9	Chromosome	ena	gene	956
10	Chromosome	ensem①	CDS	
11	Chromosome	ensem	exon	
12	Chromosome	ensem①	CDS	152
13	Chromosome	ensem	exon	152



CDS座標情報取得

今はCDSの配列情報取得が目的なので、①CDSの座標情報のみ取り出して利用したい。それを行っているのが、②の部分。ここをコピペ実行。

(Rで)塩基配列解析

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

イントロ | 一般 | 配列取得 | CDS | GenomicFeatures(Lawrence_2013) NEW

GenomicFeaturesパッケージを主に用いてCDS(タンパク質コード領域の配列)を得るやり方を示します。
「ファイル」－「ディレクトリの変更」でファイルを保存したいディレクトリに移動し以下をコピペ。

1. GFF3形式のアノテーションファイルとFASTA形式のゲノム配列ファイルを読み込む場合：

GFF3形式ファイル(Lactobacillus_casei_12a.GCA_000309565.2.25.chromosome.Chromosome.gff3) とFASTA形式ファイル(Lactobacillus_casei_12a.GCA_000309565.2.25.dna.chromosome.Chromosome.fa)を読み込むやり方です。 Ensembl (Zerbino et al., Nucleic Acids Res., 2018)から提供されている Lactobacillus casei 12Aです。2,681個の配列が取得できます。

```
in_f1 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.dna.chromosome.Chromosome.fa"#入力ファイル名を指定して
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.chromosome.Chromosome.gff3"#入力ファイル名を指定してin
out_f <- "hoge1.fasta"                    #出力ファイル名を指定してout_fに格納

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

#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="auto")#txdbオブジェクトの作成
txdb                                      #確認してるだけです

#前処理(欲しい領域の座標情報取得)
hoge <- cds(txdb)                         #指定した範囲の座標情報を取得
hoge                                      #確認してるだけです

#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)      #配列情報を取得した結果をfastaに格納
fasta                                     #確認してるだけです
```

②

トップページへ

CDS座標情報取得

今はCDSの配列情報取得が目的なので、①CDSの座標情報のみ取り出して利用したい。それを行っているのが、②の部分。ここをコピペ実行。③座標情報を含んだhogeオブジェクトの中身はこんな感じ。

(Rで)塩基配列解析

保護されていない通信 | www.iu.a.u

イントロ | 一般 | 配列取得 | CDS

GenomicFeaturesパッケージを主に用いてCDS(タン
「ファイル」－「ディレクトリの変更」でファイルを

1. GFF3形式のアノテーションファイルとFASTA形

GFF3形式ファイル(Lactobacillus_casei_12a.GCA
(Lactobacillus_casei_12a.GCA_000309565.2.25.
(Zerbino et al., Nucleic Acids Res., 2018)から提供

```
in_f1 <- "Lactobacillus_casei_12a.GCA_0003
in_f2 <- "Lactobacillus_casei_12a.GCA_0003
out_f <- "hoge1.fasta"                    #出

#必要なパッケージをロード
library(Rsamtools)                        #パ
library(GenomicFeatures)                  #パ
library(Biostrings)                       #パ

#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="aut
txdb                                      #確

#前処理(欲しい領域の座標情報取得)
hoge <- cds(txdb)                         #指
hoge                                      #確

#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)      #配
fasta                                     #確
```

③

③

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウインドウ ヘルプ Vignettes

R Console

```
> hoge                                    #確認してるだ$
GRanges object with 2681 ranges and 1 metadata column:
             seqnames              ranges strand |     cds_id
                <Rle>           <IRanges>  <Rle> | <integer>
      [1] Chromosome               1-1350      + |           1
      [2] Chromosome            1523-2662      + |           2
      [3] Chromosome            3240-3452      + |           3
      [4] Chromosome            3449-4564      + |           4
      [5] Chromosome            4817-6778      + |           5
      ...        ...                  ...    ... .         ...
   [2677] Chromosome 2902483-2903871      - |        2677
   [2678] Chromosome 2904384-2905148      - |        2678
   [2679] Chromosome 2905166-2906002      - |        2679
   [2680] Chromosome 2906147-2906698      - |        2680
   [2681] Chromosome 2906832-2906972      - |        2681
   -------
   seqinfo: 1 sequence from an unspecified genome; no s$
>
```

GRanges

hogeは、①GRangesという形式のオブジェクトです。GrangesというのはGenomic Rangesの略であるが、重要な事柄のみ（座標情報を取得したいだけ）に注力して確認すればよい。②配列名（この場合はChromosomeしかないので全部同じ）、③範囲情報、④ストランド情報（+鎖か−鎖かということ）。⑤全部で2,681配列。

(Rで)塩基配列解析

保護されていない通信 | www.iu.a.u

イントロ | 一般 | 配列取得 | CDS

GenomicFeaturesパッケージを主に用いてCDS(タン
「ファイル」－「ディレクトリの変更」でファイルを

1. GFF3形式のアノテーションファイルとFASTA形

GFF3形式ファイル(Lactobacillus_casei_12a.GC
(Lactobacillus_casei_12a.GCA_000309565.2.25
(Zerbino et al., Nucleic Acids Res., 2018)から提供

```
in_f1 <- "Lactobacillus_casei_12a.GCA_0003
in_f2 <- "Lactobacillus_casei_12a.GCA_0003
out_f <- "hoge1.fasta"                    #出

#必要なパッケージをロード
library(Rsamtools)                        #パ
library(GenomicFeatures)                  #パ
library(Biostrings)                       #パ

#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="aut
txdb                                      #確

#前処理(欲しい領域の座標情報取得)
hoge <- cds(txdb)                         #指
hoge                                      #確

#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)      #配
fasta                                     #確
```

RGui (64-bit)

ファイル 編集

R Console

```
> hoge                                          #確認してるだ$
GRanges object with 2681 ranges and 1 metadata column:
             seqnames              ranges strand |     cds_id
                <Rle>           <IRanges>  <Rle> | <integer>
     [1] Chromosome             1-1350      + |           1
     [2] Chromosome          1523-2662      + |           2
     [3] Chromosome          3240-3452      + |           3
     [4] Chromosome          3449-4564      + |           4
     [5] Chromosome          4817-6778      + |           5
     ...          ...                ...    ... .         ...
  [2677] Chromosome 2902483-2903871      - |        2677
  [2678] Chromosome 2904384-2905148      - |        2678
  [2679] Chromosome 2905166-2906002      - |        2679
  [2680] Chromosome 2906147-2906698      - |        2680
  [2681] Chromosome 2906832-2906972      - |        2681
  -------
  seqinfo: 1 sequence from an unspecified genome; no s$
> |
```

GRanges

hogeは、①GRangesという形式のオブジェクトです。Grangesというのは Genomic Rangesの略であるが、重要な事柄のみ（座標情報を取得したいだけ）に注力して確認すればよい。②配列名（この場合はChromosomeしかないので全部同じ）、③範囲情報、④ストランド情報（+鎖か−鎖かということ）。⑤全部で2,681配列。⑥最初の2つの座標情報は、見覚えのある数値ですね。

(Rで)塩基配列解析

保護されていない通信 | www.iu.a.u

イントロ | 一般 | 配列取得 | CDS

GenomicFeaturesパッケージを主に用いてCDS(タン
「ファイル」－「ディレクトリの変更」でファイルを

1. GFF3形式のアノテーションファイルとFASTA形式

GFF3形式ファイル(Lactobacillus_casei_12a.GCA
(Lactobacillus_casei_12a.GCA_000309565.2.25.
(Zerbino et al., Nucleic Acids Res., 2018)から提

```
in_f1 <- "Lactobacillus_casei_12a.GCA_0003
in_f2 <- "Lactobacillus_casei_12a.GCA_0003
out_f <- "hoge1.fasta"                    #出

#必要なパッケージをロード
library(Rsamtools)                        #パ
library(GenomicFeatures)                  #パ
library(Biostrings)                       #パ

#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="aut
txdb                                      #確

#前処理(欲しい領域の座標情報取得)
hoge <- cds(txdb)                         #指
hoge                                      #確

#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)      #配
fasta                                     #確
```

RGui (64-bit)

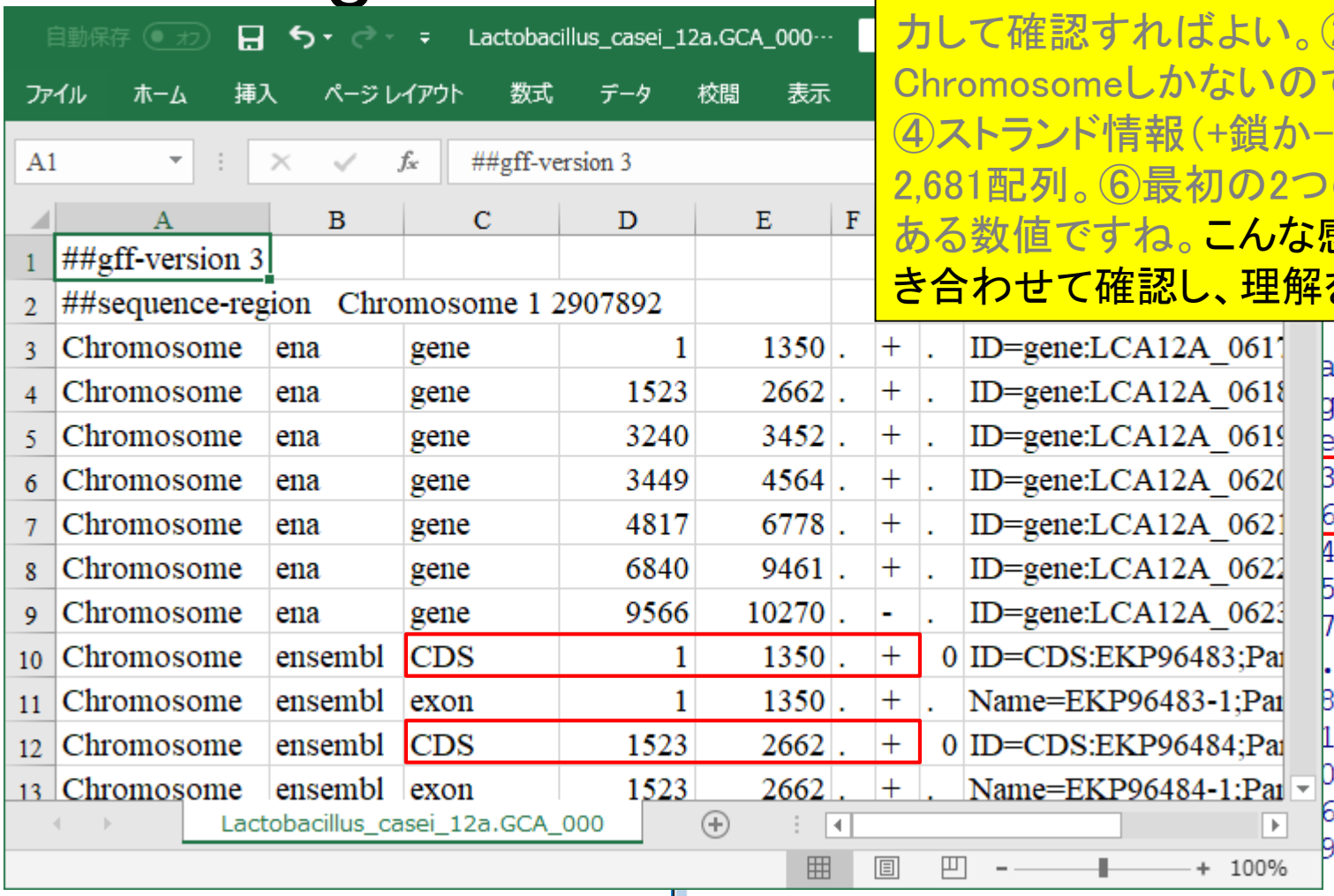
ファイル 編集

R Console

```
> hoge                                    #確認してるだ$
GRanges object with 2681 ranges and 1 metadata column:
          seqnames          ranges strand |     cds_id
             <Rle>       <IRanges>  <Rle> | <integer>
      [1] Chromosome         1-1350      + |         1
      [2] Chromosome      1523-2662      + |         2
      [3] Chromosome      3240-3452      + |         3
      [4] Chromosome      3449-4564      + |         4
      [5] Chromosome      4817-6778      + |         5
      ...        ...            ...    ... .       ...
   [2677] Chromosome 2902483-2903871      - |      2677
   [2678] Chromosome 2904384-2905148      - |      2678
   [2679] Chromosome 2905166-2906002      - |      2679
   [2680] Chromosome 2906147-2906698      - |      2680
   [2681] Chromosome 2906832-2906972      - |      2681
   -------
   seqinfo: 1 sequence from an unspecified genome; no s$
>
```

GRanges

hogeは、①GRangesという形式のオブジェクトです。Grangesというのは Genomic Rangesの略であるが、重要な事柄のみ（座標情報を取得したいだけ）に注力して確認すればよい。②配列名（この場合はChromosomeしかないので全部同じ）、③範囲情報、④ストランド情報（+鎖か−鎖かということ）。⑤全部で2,681配列。⑥最初の2つの座標情報は、見覚えのある数値ですね。こんな感じで、手持ちの情報と突き合わせて確認し、理解を深めてゆくとよいです。



	A	B	C	D	E	F			
1	##gff-version 3								
2	##sequence-region	Chromosome	1	2907892					
3	Chromosome	ena	gene	1	1350	.	+	.	ID=gene:LCA12A_061
4	Chromosome	ena	gene	1523	2662	.	+	.	ID=gene:LCA12A_061
5	Chromosome	ena	gene	3240	3452	.	+	.	ID=gene:LCA12A_061
6	Chromosome	ena	gene	3449	4564	.	+	.	ID=gene:LCA12A_062
7	Chromosome	ena	gene	4817	6778	.	+	.	ID=gene:LCA12A_062
8	Chromosome	ena	gene	6840	9461	.	+	.	ID=gene:LCA12A_062
9	Chromosome	ena	gene	9566	10270	.	-	.	ID=gene:LCA12A_062
10	Chromosome	ensembl	CDS	1	1350	.	+	0	ID=CDS:EKP96483;Par
11	Chromosome	ensembl	exon	1	1350	.	+	.	Name=EKP96483-1;Par
12	Chromosome	ensembl	CDS	1523	2662	.	+	0	ID=CDS:EKP96484;Par
13	Chromosome	ensembl	exon	1523	2662	.	+	.	Name=EKP96484-1;Par

```
#確認してるだら
and 1 metadata column:
ges strand |     cds_id
es>  <Rle> | <integer>
350      + |          1
662      + |          2
452      + |          3
564      + |          4
778      + |          5
...    ... .        ...
871      - |       2677
148      - |       2678
002      - |       2679
698      - |       2680
972      - |       2681

  seqinfo: 1 sequence from an unspecified genome; no s$
> |
```

CDS座標情報取得

コード全体を最後までコピペ実行すると、こんな感じになります。

(Rで)塩基配列解析

保護されていない通信 | www.iu.a.u

イントロ | 一般 | 配列取得 | CDS

GenomicFeaturesパッケージを主に用いてCDS(タン
「ファイル」－「ディレクトリの変更」でファイルを

1. GFF3形式のアノテーションファイルとFASTA形式

GFF3形式ファイル(Lactobacillus_casei_12a.GCA_
(Lactobacillus_casei_12a.GCA_000309565.2.25.
(Zerbino et al., Nucleic Acids Res., 2018)から提

```
in_f1 <- "Lactobacillus_casei_12a.GCA_0003
in_f2 <- "Lactobacillus_casei_12a.GCA_0003
out_f <- "hoge1.fasta"                    #出

#必要なパッケージをロード
library(Rsamtools)                        #パ
library(GenomicFeatures)                  #パ
library(Biostrings)                       #パ

#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="aut
txdb                                      #確

#前処理(欲しい領域の座標情報取得)
hoge <- cds(txdb)                         #指
hoge                                      #確

#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)      #配
fasta                                     #確
```

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウインドウ ヘルプ Vignettes

R Console

```
> fasta                                   #確認してるだ$
  A DNAStringSet instance of length 2681
        width seq                         names
    [1]  1350 ATGCCCAAT...CGCGGGTAA Chromosome_1_1350
    [2]  1140 ATGAAATTT...ACGTTCTAA Chromosome_1523_2662
    [3]   213 ATGACAACA...AGCGCATGA Chromosome_3240_3452
    [4]  1116 ATGAAACTG...AGAGCGTAA Chromosome_3449_4564
    [5]  1962 GTGACGGAC...GATGCTTGA Chromosome_4817_6778
     ...   ... ...
[2677]  1389 ATGTCTGCA...GGCAAGTAA Chromosome_290248...
[2678]   765 ATGCCAACC...AGACCGTAA Chromosome_290438...
[2679]   837 GTGAAAAAT...CGTAAATAA Chromosome_290516...
[2680]   552 TTGTTTGCG...GTCATCTGA Chromosome_290614...
[2681]   141 ATGACAACC...TCTGCCTAA Chromosome_290683...
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", w$
>
```

CDS座標情報取得

コード全体を最後までコピペ実行すると、こんな感じになります。①ATGが多く、妥当ですね。この②例題1が成功例です。

(Rで)塩基配列解析

保護されていない通信 | www.iu.a.u

イントロ | 一般 | 配列取得 | CDS

GenomicFeaturesパッケージを主に用いてCDS(タン
ァイル」－「ディレクトリの変更」でファイルを保

②

1. GFF3形式のアノテーションファイルとFASTA形式

GFF3形式ファイル(Lactobacillus_casei_12a.GCA_
(Lactobacillus_casei_12a.GCA_000309565.2.25.
(Zerbino et al., Nucleic Acids Res., 2018)から提供

```
in_f1 <- "Lactobacillus_casei_12a.GCA_0003
in_f2 <- "Lactobacillus_casei_12a.GCA_0003
out_f <- "hoge1.fasta"                    #出

#必要なパッケージをロード
library(Rsamtools)                        #パ
library(GenomicFeatures)                  #パ
library(Biostrings)                       #パ

#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="aut
txdb                                      #確

#前処理(欲しい領域の座標情報取得)
hoge <- cds(txdb)                         #指
hoge                                      #確

#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)      #配
fasta                                     #確
```

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウインドウ ヘルプ Vignettes

R Console

①

```
> fasta                                   #確認してるだ$
  A DNAStringSet instance of length 2681
        width seq                         names
    [1]  1350 ATGCCCAAT...CGCGGGTAA Chromosome_1_1350
    [2]  1140 ATGAAATTT...ACGTTCTAA Chromosome_1523_2662
    [3]   213 ATGACAACA...AGCGCATGA Chromosome_3240_3452
    [4]  1116 ATGAAACTG...AGAGCGTAA Chromosome_3449_4564
    [5]  1962 GTGACGGAC...GATGCTTGA Chromosome_4817_6778
     ...   ... ...
[2677]  1389 ATGTCTGCA...GGCAAGTAA Chromosome_290248...
[2678]   765 ATGCCAACC...AGACCGTAA Chromosome_290438...
[2679]   837 GTGAAAAAT...CGTAAATAA Chromosome_290516...
[2680]   552 TTGTTTGCG...GTCATCTGA Chromosome_290614...
[2681]   141 ATGACAACC...TCTGCCTAA Chromosome_290683...
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", w$
>
```

Contents

- 配列長でフィルタリング（ゲノムアセンブリ結果の後処理）
  - イントロ、本番
- ゲノムアノテーション
  - イントロ、参考文献
  - DFAST実行、課題1
- CDS（タンパク質コード領域の配列；coding sequenceの略）の取得
  - 比較ゲノム解析のイントロ、問題設定
  - 答え（cds.fna）を眺めておく、sequence logos
  - CDS配列取得の成功例、TxDbオブジェクト、CDS座標情報取得
  - CDS配列取得の失敗例、GFF3をExcelで概観
  - 基本テクを駆使して解決する基本戦略、CDSを含む行の抽出、補足説明
  - Excelで確認、リストファイルの作成、基本テクでCDS配列取得を実行
  - ストランド情報つきのリストファイルの作成、例題6のコード解説
  - ストランド情報反映戦略を練り、コードに修正を加えて実行（例題7）

失敗例は例題2

失敗例は例題2です。①を押していって、例題2までページ下部に移動。

(Rで)塩基配列解析

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

イントロ | 一般 | 配列取得 | CDS | GenomicFeatures(Lawrence_2013) NEW

GenomicFeaturesパッケージを主に用いてCDS(タンパク質コード領域の配列)を得るやり方を示します。
「ファイル」－「ディレクトリの変更」でファイルを保存したいディレクトリに移動し以下をコピペ。

1. GFF3形式のアノテーションファイルとFASTA形式のゲノム配列ファイルを読み込む場合：

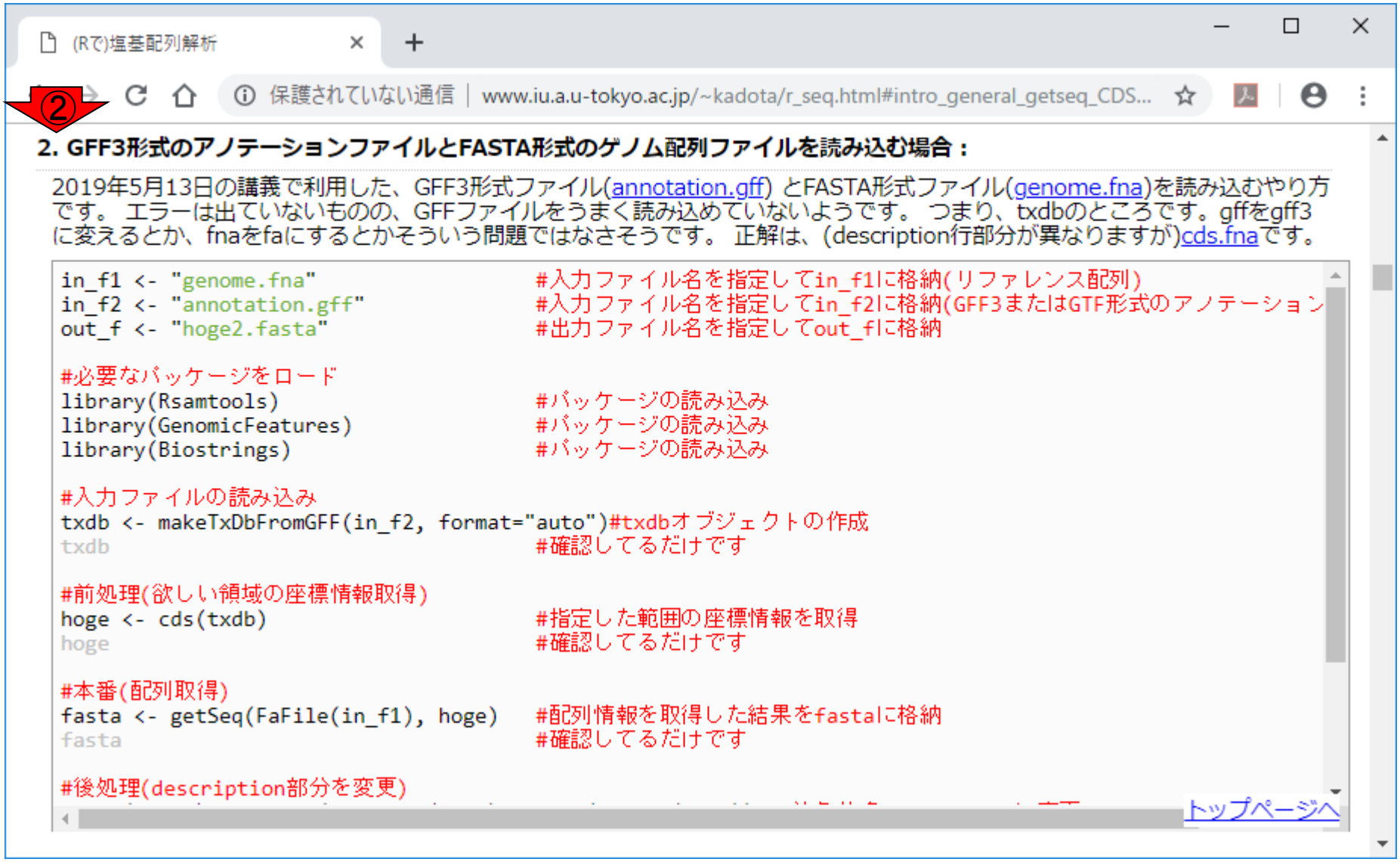
GFF3形式ファイル(Lactobacillus_casei_12a.GCA_000309565.2.25
(Lactobacillus_casei_12a.GCA_000309565.2.25.dna.chromosome
(Zerbino et al., Nucleic Acids Res., 2018)から提供されている Lacto

```
in_f1 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.dna.
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.chro
out_f <- "hoge1.fasta"                    #出力ファイル名を指定

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

#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="auto")#txdbオブジェ
txdb                                      #確認してるだけです

#前処理(欲しい領域の座標情報取得)
hoge <- cds(txdb)                         #指定した範囲の座標情報を取得
hoge                                      #確認してるだけです

#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)      #配列情報を取得した結果をfastaに格納
fasta                                     #確認してるだけです
```

トップページへ

失敗例は例題2

失敗例は例題2です。①を押していって、例題2までページ下部に移動。②例題2です。



2. GFF3形式のアノテーションファイルとFASTA形式のゲノム配列ファイルを読み込む場合：

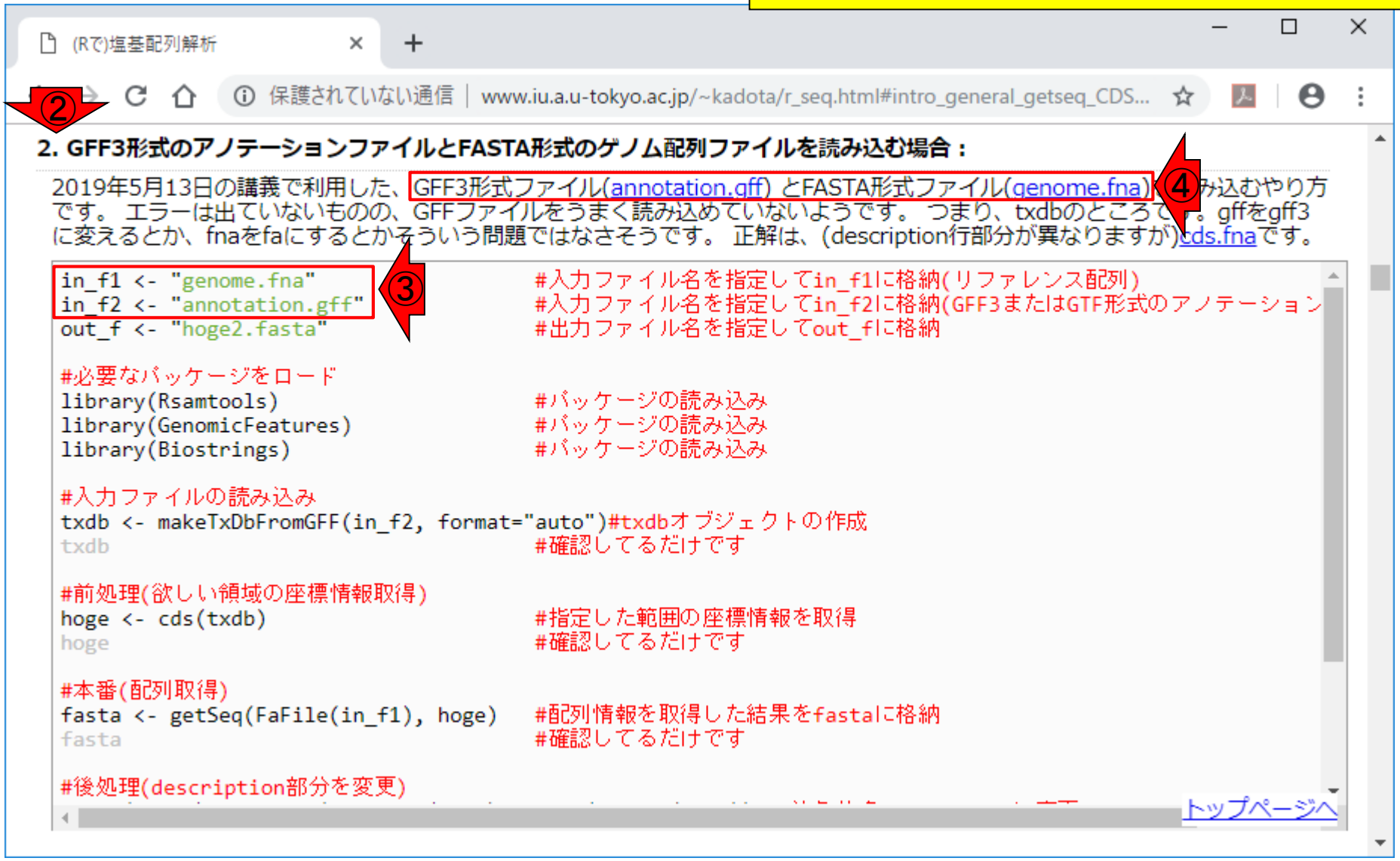
2019年5月13日の講義で利用した、GFF3形式ファイル(annotation.gff) とFASTA形式ファイル(genome.fna)を読み込むやり方です。 エラーは出ていないものの、GFFファイルをうまく読み込めていないようです。 つまり、txdbのところです。gffをgff3に変えるとか、fnaをfaにするとかそういう問題ではなさそうです。 正解は、(description行部分が異なりますが)cds.fnaです。

```
in_f1 <- "genome.fna"                    #入力ファイル名を指定してin_f1に格納(リファレンス配列)
in_f2 <- "annotation.gff"                #入力ファイル名を指定してin_f2に格納(GFF3またはGTF形式のアノテーション
out_f <- "hoge2.fasta"                   #出力ファイル名を指定してout_fに格納

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

#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="auto")#txdbオブジェクトの作成
txdb                                     #確認してるだけです

#前処理(欲しい領域の座標情報取得)
hoge <- cds(txdb)                        #指定した範囲の座標情報を取得
hoge                                     #確認してるだけです

#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)     #配列情報を取得した結果をfastaに格納
fasta                                    #確認してるだけです

#後処理(description部分を変更)
```

トップページへ

失敗例は例題2

失敗例は例題2です。①を押していって、例題2までページ下部に移動。②例題2です。③入力ファイルは、④ですが…



失敗例は例題2

失敗例は例題2です。①を押していって、例題2までページ下部に移動。②例題2です。③入力ファイルは、④ですが、⑤と同じものです。

講義日程（2019年度）

1. 2019年04月08日（PC使用）
   講義資料PDF(最終更新：2019.04.09)
   学会(国外)：ISCB
   学会(国内)：JSBi
   QAサイト：Biostar (Parnell et al., PLoS Comput Biol., 2011)
   QAサイト：SEQanswers (Li et al., Bioinformatics, 2012)
   学習教材：バイオインフォマティクス人材育成のための講習会(平成26-29年度)
   学習教材：(Rで)塩基配列解析
   学習教材：(Rで)塩基配列解析のサブ
   RStudio

2. 2019年04月15日（PC使用）
   講義資料PDF(最終更新：2019.04.15)
   (Rで)塩基配列解析
   (Rで)塩基配列解析のサブ

3. 2019年04月22日（PC使用）
   講義資料PDF
   課題：docx版とPDF版
   (Rで)塩基配列解析
   hoge8.fa (課題用)
   Bioconductor
   CRAN

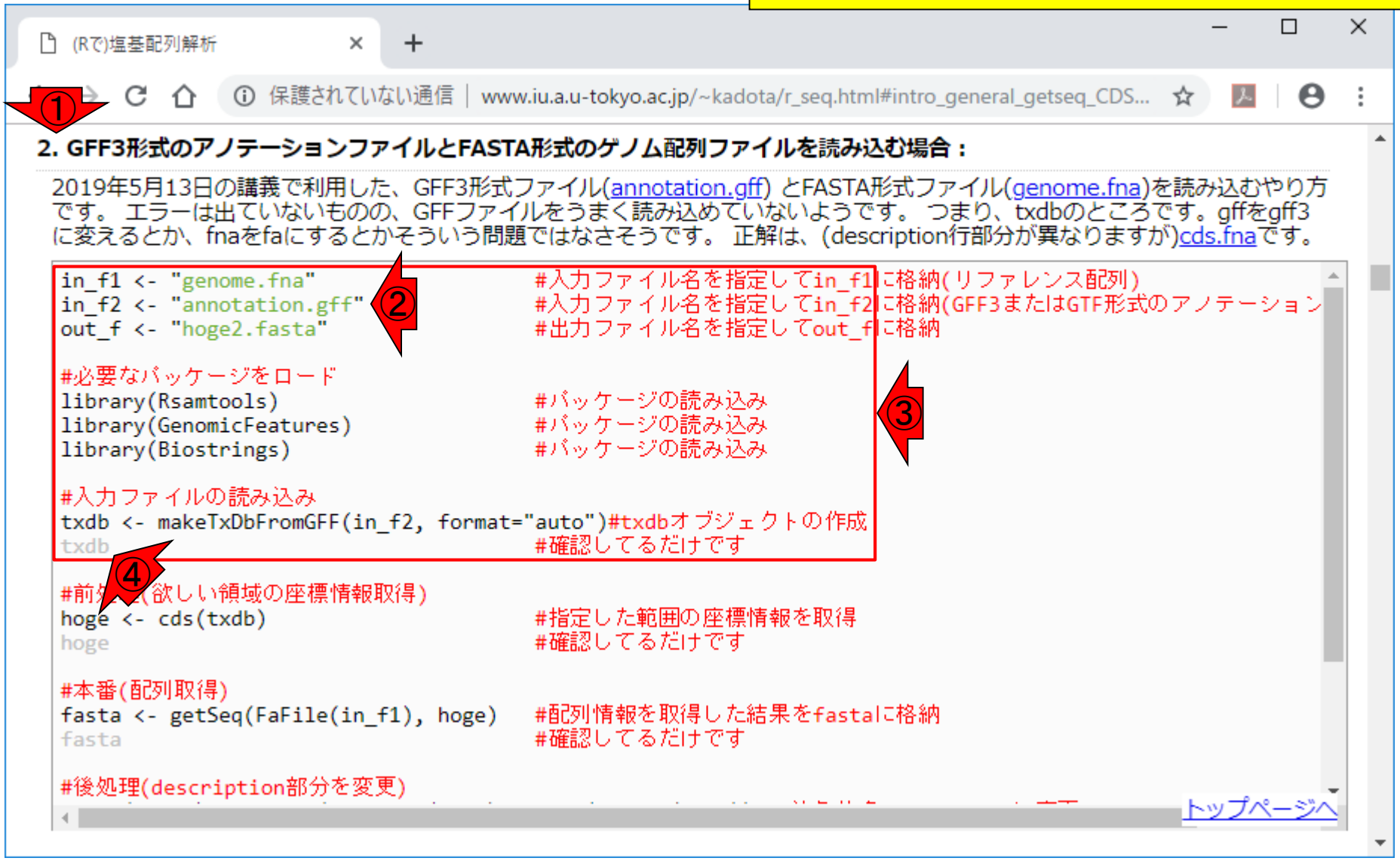
4. 2019年05月13日（PC使用）
   講義資料PDF
   (Rで)塩基配列解析のサブ
   out_gapClosed.fa (約2.3MB)
   (Rで)塩基配列解析
   DFAST：Tanizawa et al., Bioinformatics, 2018
   DFAST実行結果：genome.fna
   DFAST実行結果：annotation.gff
   DFAST実行結果：cds.fna

4. 2019年05月13日（PC使用）
講義資料PDF
(Rで)塩基配列解析のサブ
out_gapClosed.fa (約2.3MB)
(Rで)塩基配列解析
DFAST：Tanizawa et al., Bioinformatics, 2018
DFAST実行結果：genome.fna
DFAST実行結果：annotation.gff
DFAST実行結果：cds.fna

⑤

txdbの作成まで

①例題2は、②GFFファイルの読込みのところがうまくいってないことが分かっています。③赤枠内をコピペ実行して、④を眺めます。




```
in_f1 <- "genome.fna"                    #入力ファイル名を指定してin_f1に格納(リファレンス配列)
in_f2 <- "annotation.gff"                #入力ファイル名を指定してin_f2に格納(GFF3またはGTF形式のアノテーション
out_f <- "hoge2.fasta"                   #出力ファイル名を指定してout_fに格納

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

#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="auto")#txdbオブジェクトの作成
txdb                                     #確認してるだけです

#前処理(欲しい領域の座標情報取得)
hoge <- cds(txdb)                        #指定した範囲の座標情報を取得
hoge                                     #確認してるだけです

#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)     #配列情報を取得した結果をfastaに格納
fasta                                    #確認してるだけです

#後処理(description部分を変更)
```

txdbの作成まで

①例題2は、②GFFファイルの読込みのところがうまくいってないことが分かっています。③赤枠内をコピペ実行して、④を眺めます。④txdbオブジェクトの中身。

(Rで)塩基配列解析

保護されていない通信 | www.iu.a

2. GFF3形式のアノテーションファイルとFASTA形式

2019年5月13日の講義で利用した、GFF3形式ファイ
です。 エラーは出ていないものの、GFFファイルを
に変えるとか、fnaをfaにするとかそういう問題で

```
in_f1 <- "genome.fna"                    #
in_f2 <- "annotation.gff"                #
out_f <- "hoge2.fasta"                   #出

#必要なパッケージをロード
library(Rsamtools)                       #パ
library(GenomicFeatures)                 #パ
library(Biostrings)                      #パ

#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="aut
txdb                                     #確

#前処理(欲しい領域の座標情報取得)
hoge <- cds(txdb)                        #指
hoge                                     #確

#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)     #配
fasta                                    #確

#後処理(description部分を変更)
```

④

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウインドウ ヘルプ Vignettes

R Console

```
> txdb                                   #確認してる$
TxDb object:
# Db type: TxDb
# Supporting package: GenomicFeatures
# Data source: annotation.gff
# Organism: NA
# Taxonomy ID: NA
# miRBase build ID: NA
# Genome: NA
# transcript_nrow: 60
# exon_nrow: 60
# cds_nrow: 0
# Db created by: GenomicFeatures package from Biocond$
# Creation time: 2019-04-23 15:08:43 +0900 (Tue, 23 A$
# GenomicFeatures version at creation time: 1.34.1
# RSQLite version at creation time: 2.1.1
# DBSCHEMAVERSION: 1.2
>
```

txdbの作成まで

①例題2は、②GFFファイルの読込みのところがうまくいってないことが分かっています。③赤枠内をコピペ実行して、④を眺めます。④txdbオブジェクトの中身。⑤CDSが0となっており、オカシイことがわかる。

(Rで)塩基配列解析

保護されていない通信 | www.iu.a.u

2. GFF3形式のアノテーションファイルとFASTA形式

2019年5月13日の講義で利用した、GFF3形式ファイ
です。 エラーは出ていないものの、GFFファイルを
に変えるとか、fnaをfaにするとかそういう問題で

```
in_f1 <- "genome.fna"                    #
in_f2 <- "annotation.gff"                #
out_f <- "hoge2.fasta"                   #出

#必要なパッケージをロード
library(Rsamtools)                       #パ
library(GenomicFeatures)                 #パ
library(Biostrings)                      #パ

#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="aut
txdb                                     #確

#前処理(欲しい領域の座標情報取得)
hoge <- cds(txdb)                        #指
hoge                                     #確

#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)     #配
fasta                                    #確

#後処理(description部分を変更)
```

④

④

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウインドウ ヘルプ Vignettes

R Console

```
> txdb                                   #確認してる$
TxDb object:
# Db type: TxDb
# Supporting package: GenomicFeatures
# Data source: annotation.gff
# Organism: NA
# Taxonomy ID: NA
# miRBase build ID: NA
# Genome: NA
# transcript_nrow: 60
# exon_nrow: 60
# cds_nrow: 0
# Db created by: GenomicFeatures package from Biocond$
# Creation time: 2019-04-23 15:08:43 +0900 (Tue, 23 A$
# GenomicFeatures version at creation time: 1.34.1
# RSQLite version at creation time: 2.1.1
# DBSCHEMAVERSION: 1.2
>
```

⑤

txdbの作成まで

①例題2は、②GFFファイルの読込みのところがうまくいってないことが分かっています。③赤枠内をコピペ実行して、④を眺めます。④txdbオブジェクトの中身。⑤CDSが0となっており、オカシイことがわかる。赤枠のGFF読み込み時のメッセージを見ても、⑥OKとなっている。警告すら出ていないが、⑤CDSが0個となってしまっているのが厄介なところ。

(Rで)塩基配列解析

保護されていない通信 | www.iu.a.u

2. GFF3形式のアノテーションファイルとFASTA形式

2019年5月13日の講義で利用した、GFF3形式ファイ
です。 エラーは出ていないものの、GFFファイルを
に変えるとか、fnaをfaにするとかそういう問題では

```
in_f1 <- "genome.fna"                    #入
in_f2 <- "annotation.gff"                #入
out_f <- "hoge2.fasta"                   #出

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

#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="aut
txdb                                     #確

#前処理(欲しい領域の座標情報取得)
hoge <- cds(txdb)                        #指
hoge                                     #確

#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)     #配
fasta                                    #確

#後処理(description部分を変更)
```

④

④

RGui (64-bit)

ファイル 編集

R Console

```
> #入力ファイルの読み込み
> txdb <- makeTxDbFromGFF(in_f2, format="auto")#txdb$
Import genomic features from the file as a GRanges ob$
Prepare the 'metadata' data frame ... OK
Make the TxDb object ... OK
> txdb                                   #確認してる$
TxDb object:
# Db type: TxDb
# Supporting package: GenomicFeatures
# Data source: annotation.gff
# Organism: NA
# Taxonomy ID: NA
# miRBase build ID: NA
# Genome: NA
# transcript_nrow: 60
# exon_nrow: 60
# cds_nrow: 0
# Db created by: GenomicFeatures package from Biocond$
```

⑥

⑥

⑤

CDS座標情報取得

④txdbオブジェクトの段階で⑤CDSが0なので、当然ながら赤枠で示したCDS領域の座標情報もうまく取得できていないことがわかる。

(Rで)塩基配列解析

保護されていない通信 | www.iu.a

2. GFF3形式のアノテーションファイルとFASTA形式

2019年5月13日の講義で利用した、GFF3形式ファイ
です。 エラーは出ていないものの、GFFファイルを
に変えるとか、fnaをfaにするとかそういう問題では

```
in_f1 <- "genome.fna"                    #入
in_f2 <- "annotation.gff"                #入
out_f <- "hoge2.fasta"                   #出

#必要なパッケージをロード
library(Rsamtools)                       #パ
library(GenomicFeatures)                 #パ
library(Biostrings)                      #パ

#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="aut
txdb                                     #確

#前処理(欲しい領域の座標情報取得)
hoge <- cds(txdb)                        #指
hoge                                     #確

#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)     #配
fasta                                    #確

#後処理(description部分を変更)
```

④

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウインドウ ヘルプ Vignettes

R Console

```
# Genome: NA
# transcript_nrow: 60
# exon_nrow: 60
# cds_nrow: 0
# Db created by GenomicFeatures package from Biocond$
# Creation time: 2019-04-23 15:08:43 +0900 (Tue, 23 A$
# GenomicFeatures version at creation time: 1.34.1
# RSQLite version at creation time: 2.1.1
# DBSCHEMAVERSION: 1.2
> #前処理(欲しい領域の座標情報取得)
> hoge <- cds(txdb)                      #指定した範$
> hoge                                   #確認してる$
GRanges object with 0 ranges and 1 metadata column:
   seqnames    ranges strand |     cds_id
      <Rle> <IRanges>  <Rle> | <integer>
  -------
  seqinfo: 45 sequences from an unspecified genome; n$
>
```

⑤

CDS座標情報取得

④txdbオブジェクトの段階で⑤CDSが0なので、当然ながら赤枠で示したCDS領域の座標情報もうまく取得できていないことがわかる。⑥cds関数実行時にエラーが出ないのは、⑤CDSが0個なので、⑦CDSの領域数が0だという情報がhogeオブジェクトに格納されているだけだからと解釈すればよい。

(Rで)塩基配列解析

保護されていない通信 | www.iu.a.u

2. GFF3形式のアノテーションファイルとFASTA形式

2019年5月13日の講義で利用した、GFF3形式ファイ
です。 エラーは出ていないものの、GFFファイルを
に変えるとか、fnaをfaにするとかそういう問題では

```
in_f1 <- "genome.fna"                    #入
in_f2 <- "annotation.gff"                #入
out_f <- "hoge2.fasta"                   #出

#必要なパッケージをロード
library(Rsamtools)                       #パ
library(GenomicFeatures)                 #パ
library(Biostrings)                      #パ

#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="aut
txdb                                     #確

#前処理(欲しい領域の座標情報取得)
hoge <- cds(txdb)                        #
hoge                                     #確

#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)     #配
fasta                                    #確

#後処理(description部分を変更)
```

④

⑥

RGui (64-bit)

ファイル 編集

R Console

```
# Genome: NA
# transcript_nrow: 60
# exon_nrow: 60
# cds_nrow: 0
# Db created by: GenomicFeatures package from Biocond$
# Creation time: 2019-04-23 15:08:43 +0900 (Tue, 23 A$
# GenomicFeatures version at creation time: 1.34.1
# RSQLite version at creation time: 2.1.1
# DBSCHEMAVERSION: 1.2
> #前処理(欲しい領域の座標情報取得)
> hoge <- cds(txdb)                      #指定した範$
> hoge                                   #確認してる$
GRanges object with 0 ranges and 1 metadata column:
   seqnames    ranges strand |     cds_id
      <Rle> <IRanges>  <Rle> | <integer>
   -------
   seqinfo: 45 sequences from an unspecified genome; n$
>
```

⑤

⑦

CDS座標情報取得

当然最後までコピペ実行しても、①や②の表示結果からも予想できるように、③出力ファイル（hoge2.fasta）中には何も書きだされません。

(Rで)塩基配列解析

保護されていない通信 | www.iu.a.u

2. GFF3形式のアノテーションファイルとFASTA形式

2019年5月13日の講義で利用した、GFF3形式ファイ
です。 エラーは出ていないものの、GFFファイルを
に変えるとか、fnaをfaにするとかそういう問題では

```
in_f1 <- "genome.fna"                    #入
in_f2 <- "annotation.gf                  #入
out_f <- "hoge2.fasta"                   #出

#必要なパッケージをロード
library(Rsamtools)                       #パ
library(GenomicFeatures)                 #パ
library(Biostrings)                      #

#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="aut
txdb                                     #確

#前処理(欲しい領域の座標情報取得)
hoge <- cds(txdb)                        #指
hoge                                     #

#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)     #配
fasta                                    #確

#後処理(description部分を変更)
```

③

①

②

RGui (64-bit)

ファイル　編集　閲覧　その他　パッケージ　ウインドウ　ヘルプ　Vignettes

R Console

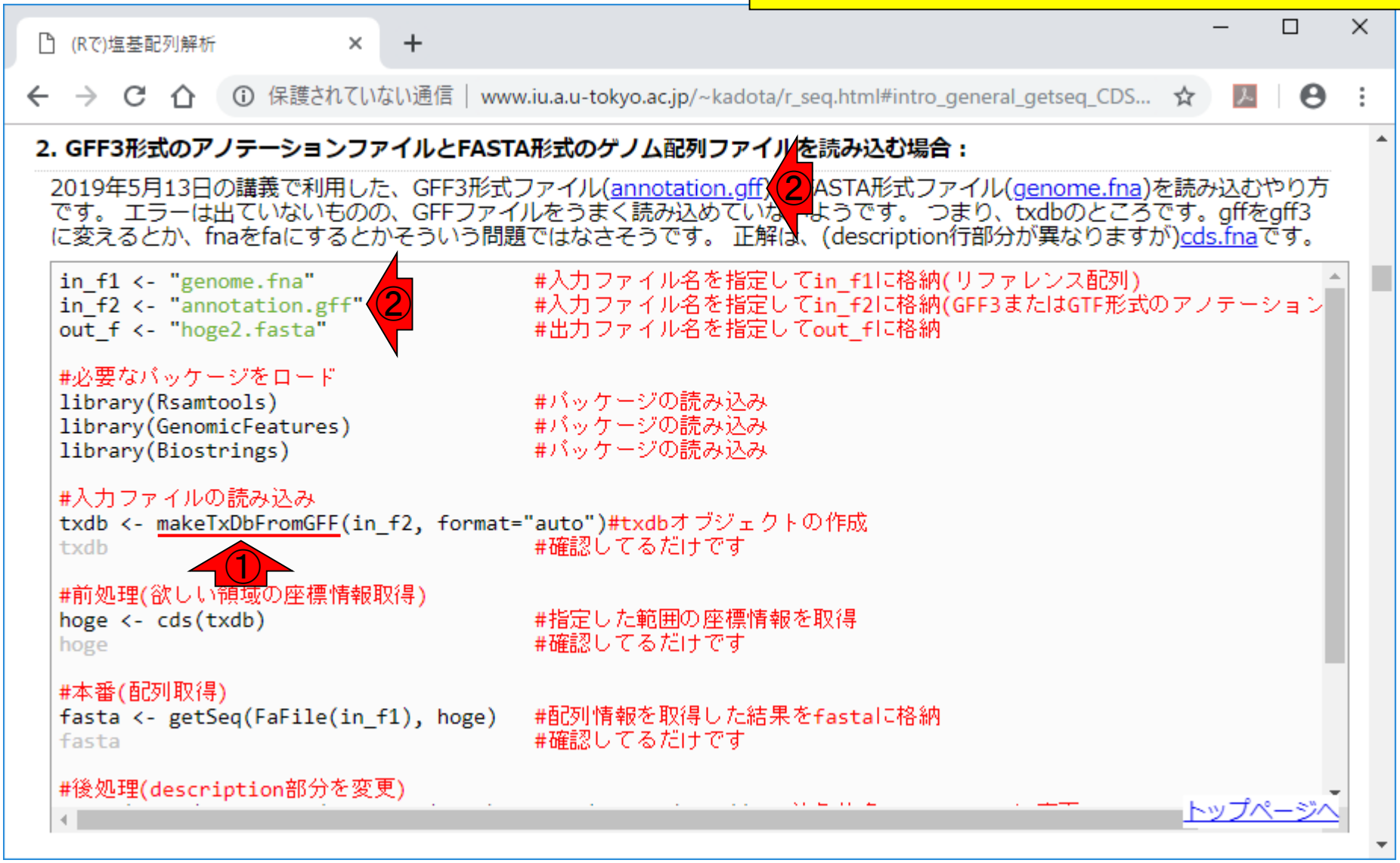
```
   seqnames    ranges strand |    cds_id
      <Rle> <IRanges>  <Rle> | <integer>
  -------
  seqinfo: 45 sequences from an unspecified genome; n$
> #本番(配列取得)
> fasta <- getSeq(FaFile(in_f1), hoge)    #配列情報を$
> fasta                                   #確認してる$
  A DNAStringSet instance of length 0
>
> #後処理(description部分を変更)
> names(fasta) <- paste(seqnames(hoge), start(ranges($
+                       end(ranges(hoge)), sep="_")#"$
> fasta                                   #確認してる$
  A DNAStringSet instance of length 0
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", $
>
```

Contents

- 配列長でフィルタリング（ゲノムアセンブリ結果の後処理）
  - イントロ、本番
- ゲノムアノテーション
  - イントロ、参考文献
  - DFAST実行、課題1
- CDS（タンパク質コード領域の配列；coding sequenceの略）の取得
  - 比較ゲノム解析のイントロ、問題設定
  - 答え（cds.fna）を眺めておく、sequence logos
  - CDS配列取得の成功例、TxDbオブジェクト、CDS座標情報取得
  - CDS配列取得の失敗例、GFF3をExcelで概観
  - 基本テクを駆使して解決する基本戦略、CDSを含む行の抽出、補足説明
  - Excelで確認、リストファイルの作成、基本テクでCDS配列取得を実行
  - ストランド情報つきのリストファイルの作成、例題6のコード解説
  - ストランド情報反映戦略を練り、コードに修正を加えて実行（例題7）

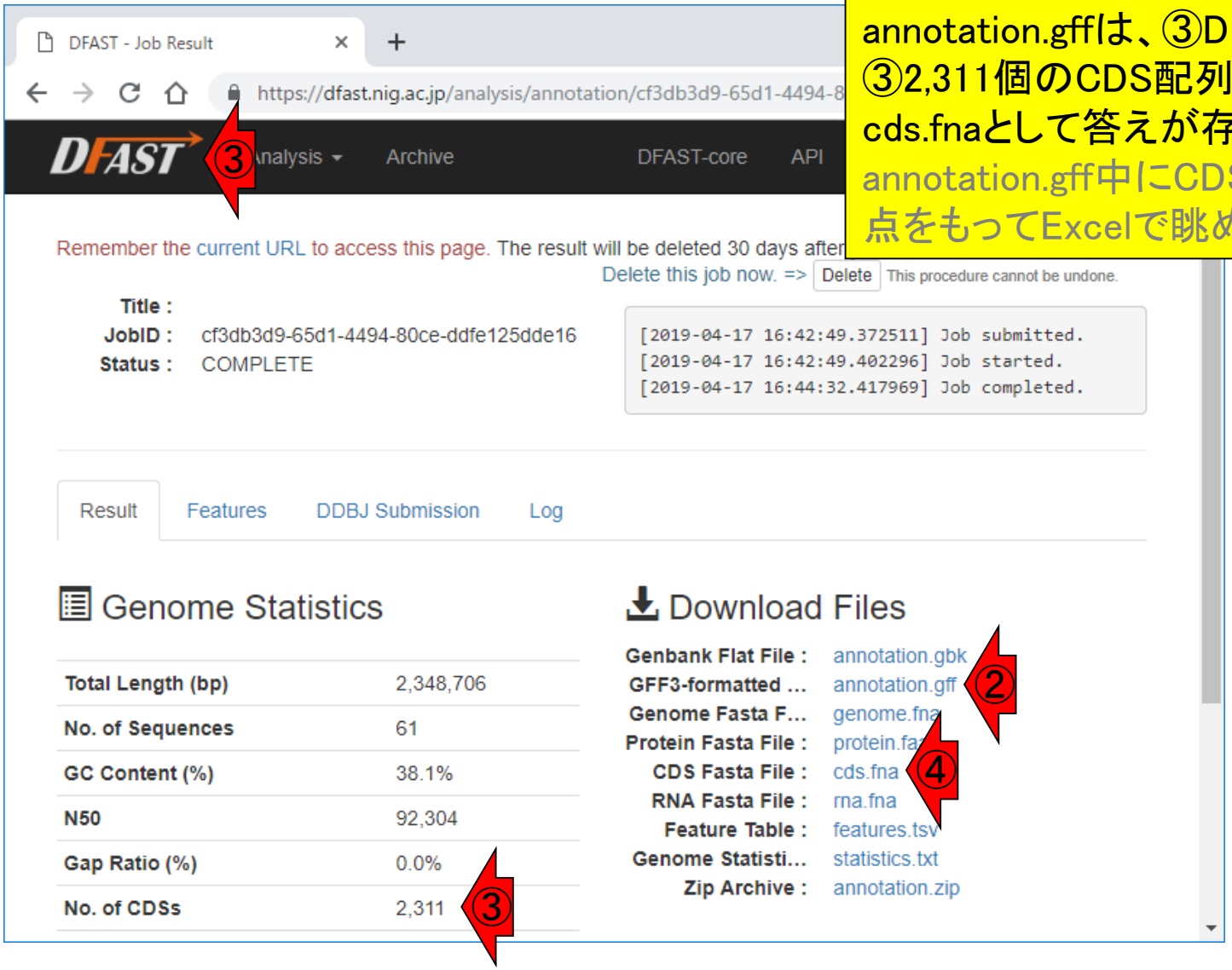
GFF3をExcelで概観

①makeTxDbFromGFF関数で正しく読み込めなかった原因を探る。Excelで②GFFファイルを読み込んで、CDSの情報が本当にあるかを調べる。



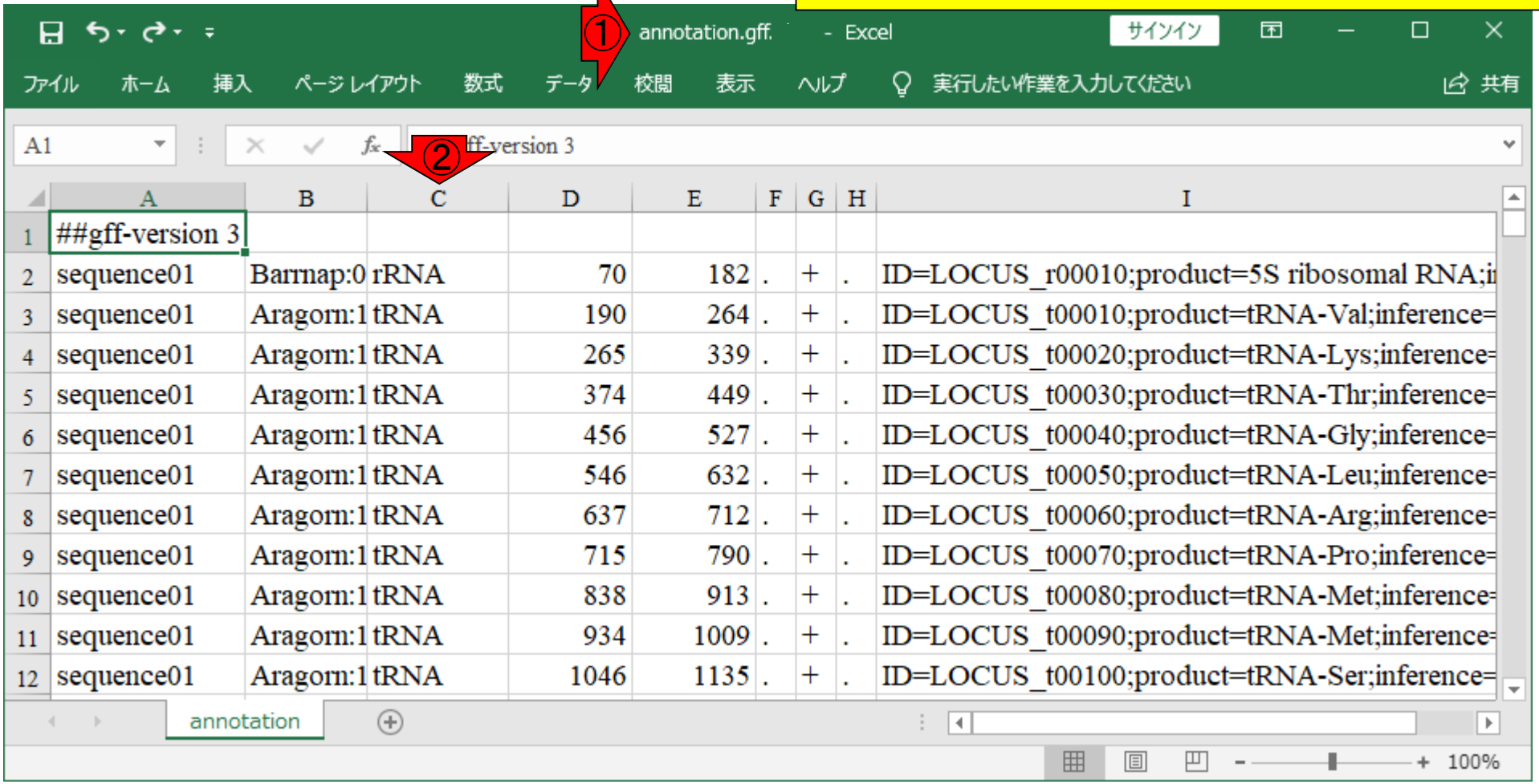
CDSはあるはず!

①makeTxDbFromGFF関数で正しく読み込めなかった原因を探る。Excelで②GFFファイルを読み込んで、CDSの情報が本当にあるかを調べる。とはいえ、②annotation.gffは、③DFAST実行結果ファイルであり、③2,311個のCDS配列があるはずだし、実際に④cds.fnaとして答えが存在する。それゆえ、②annotation.gff中にCDSの記載があるはず!という視点をもってExcelで眺める。



GFF3をExcelで概観

①annotation.gffをExcelで読み込んだ結果のスクリーンショット。②3列目にCDSという文字列を含む行が2,311個存在するはずという視点で眺めると…



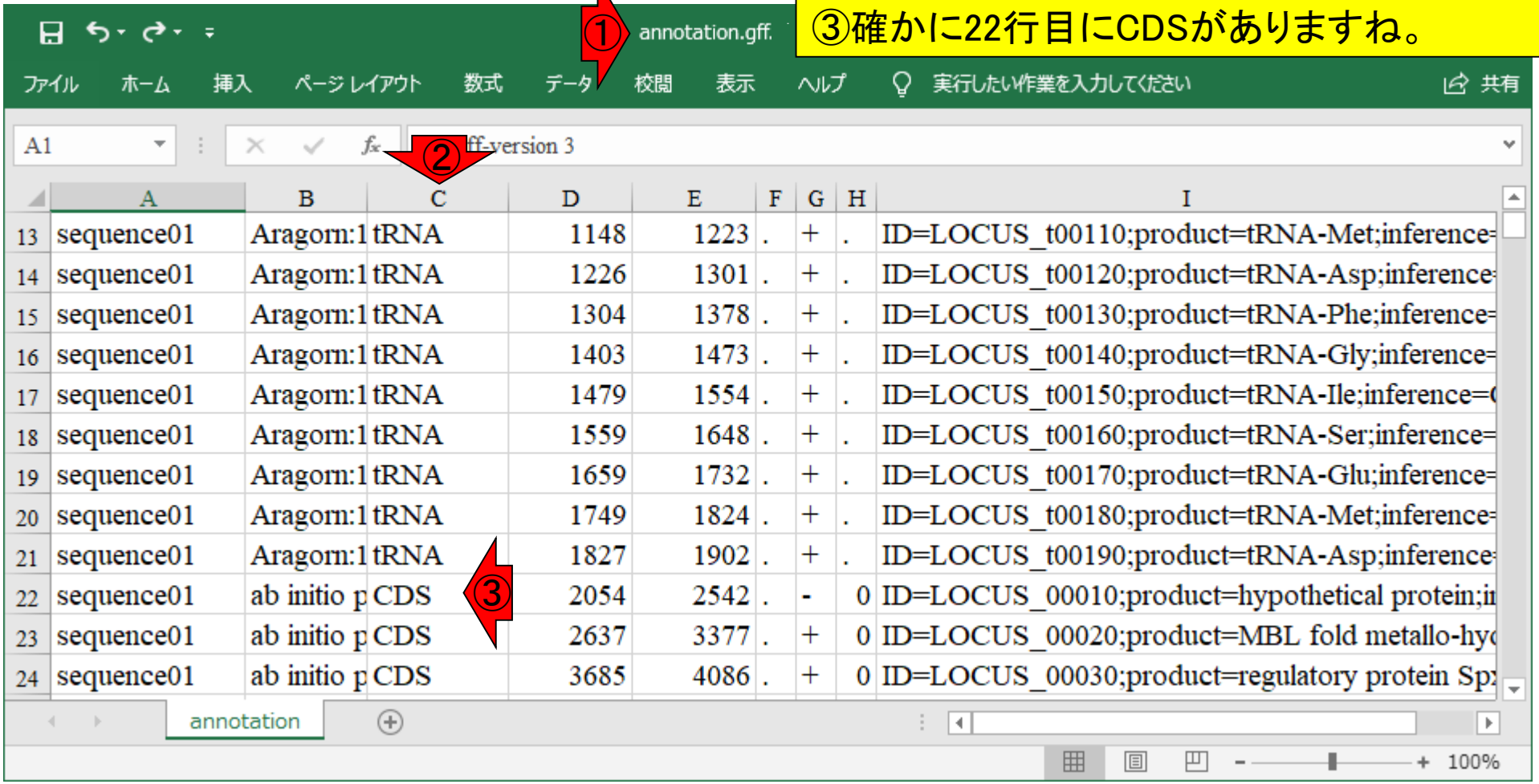
	A	B	C	D	E	F	G	H	I
1	##gff-version 3								
2	sequence01	Barrnap:0	rRNA	70	182	.	+	.	ID=LOCUS_r00010;product=5S ribosomal RNA;i
3	sequence01	Aragorn:1	tRNA	190	264	.	+	.	ID=LOCUS_t00010;product=tRNA-Val;inference=
4	sequence01	Aragorn:1	tRNA	265	339	.	+	.	ID=LOCUS_t00020;product=tRNA-Lys;inference=
5	sequence01	Aragorn:1	tRNA	374	449	.	+	.	ID=LOCUS_t00030;product=tRNA-Thr;inference=
6	sequence01	Aragorn:1	tRNA	456	527	.	+	.	ID=LOCUS_t00040;product=tRNA-Gly;inference=
7	sequence01	Aragorn:1	tRNA	546	632	.	+	.	ID=LOCUS_t00050;product=tRNA-Leu;inference=
8	sequence01	Aragorn:1	tRNA	637	712	.	+	.	ID=LOCUS_t00060;product=tRNA-Arg;inference=
9	sequence01	Aragorn:1	tRNA	715	790	.	+	.	ID=LOCUS_t00070;product=tRNA-Pro;inference=
10	sequence01	Aragorn:1	tRNA	838	913	.	+	.	ID=LOCUS_t00080;product=tRNA-Met;inference=
11	sequence01	Aragorn:1	tRNA	934	1009	.	+	.	ID=LOCUS_t00090;product=tRNA-Met;inference=
12	sequence01	Aragorn:1	tRNA	1046	1135	.	+	.	ID=LOCUS_t00100;product=tRNA-Ser;inference=

annotation

100%

GFF3をExcelで概観

①annotation.gffをExcelで読み込んだ結果のスクリーンショット。②3列目にCDSという文字列を含む行が2,311個存在するはずという視点で眺めると、③確かに22行目にCDSがありますね。



	A	B	C	D	E	F	G	H	I
13	sequence01	Aragorn:1	tRNA	1148	1223	.	+	.	ID=LOCUS_t00110;product=tRNA-Met;inference=
14	sequence01	Aragorn:1	tRNA	1226	1301	.	+	.	ID=LOCUS_t00120;product=tRNA-Asp;inference=
15	sequence01	Aragorn:1	tRNA	1304	1378	.	+	.	ID=LOCUS_t00130;product=tRNA-Phe;inference=
16	sequence01	Aragorn:1	tRNA	1403	1473	.	+	.	ID=LOCUS_t00140;product=tRNA-Gly;inference=
17	sequence01	Aragorn:1	tRNA	1479	1554	.	+	.	ID=LOCUS_t00150;product=tRNA-Ile;inference=(
18	sequence01	Aragorn:1	tRNA	1559	1648	.	+	.	ID=LOCUS_t00160;product=tRNA-Ser;inference=
19	sequence01	Aragorn:1	tRNA	1659	1732	.	+	.	ID=LOCUS_t00170;product=tRNA-Glu;inference=
20	sequence01	Aragorn:1	tRNA	1749	1824	.	+	.	ID=LOCUS_t00180;product=tRNA-Met;inference=
21	sequence01	Aragorn:1	tRNA	1827	1902	.	+	.	ID=LOCUS_t00190;product=tRNA-Asp;inference=
22	sequence01	ab initio p	CDS	2054	2542	.	-	0	ID=LOCUS_00010;product=hypothetical protein;i
23	sequence01	ab initio p	CDS	2637	3377	.	+	0	ID=LOCUS_00020;product=MBL fold metallo-hyd
24	sequence01	ab initio p	CDS	3685	4086	.	+	0	ID=LOCUS_00030;product=regulatory protein Sp

なぜ読み込めない?!

CDSをうまく読み込めない理由は、今のところ不明です。ぜひ課題2で挑戦してください!講義はここまで

(Rで)塩基配列解析

保護されていない通信 | www.iu.a.u

2. GFF3形式のアノテーションファイルとFASTA形式

2019年5月13日の講義で利用した、GFF3形式ファイ
です。 エラーは出ていないものの、GFFファイルを
に変えるとか、fnaをfaにするとかそういう問題で

```
in_f1 <- "genome.fna"                    #④
in_f2 <- "annotation.gff"                #
out_f <- "hoge2.fasta"                   #出

#必要なパッケージをロード
library(Rsamtools)                       #パ
library(GenomicFeatures)                 #パ
library(Biostrings)                      #パ

#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="aut
txdb                                     #確

#前処理(欲しい領域の座標情報取得)
hoge <- cds(txdb)                        #指
hoge                                     #確

#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)     #配
fasta                                    #確

#後処理(description部分を変更)
```

④

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウインドウ ヘルプ Vignettes

R Console

```
> txdb                                   #確認してる$
TxDb object:
# Db type: TxDb
# Supporting package: GenomicFeatures
# Data source: annotation.gff
# Organism: NA
# Taxonomy ID: NA
# miRBase build ID: NA
# Genome: NA
# transcript_nrow: 60
# exon_nrow: 60
# cds_nrow: 0
# Db created by: GenomicFeatures package from Biocond$
# Creation time: 2019-04-23 15:08:43 +0900 (Tue, 23 A$
# GenomicFeatures version at creation time: 1.34.1
# RSQLite version at creation time: 2.1.1
# DBSCHEMAVERSION: 1.2
>
```

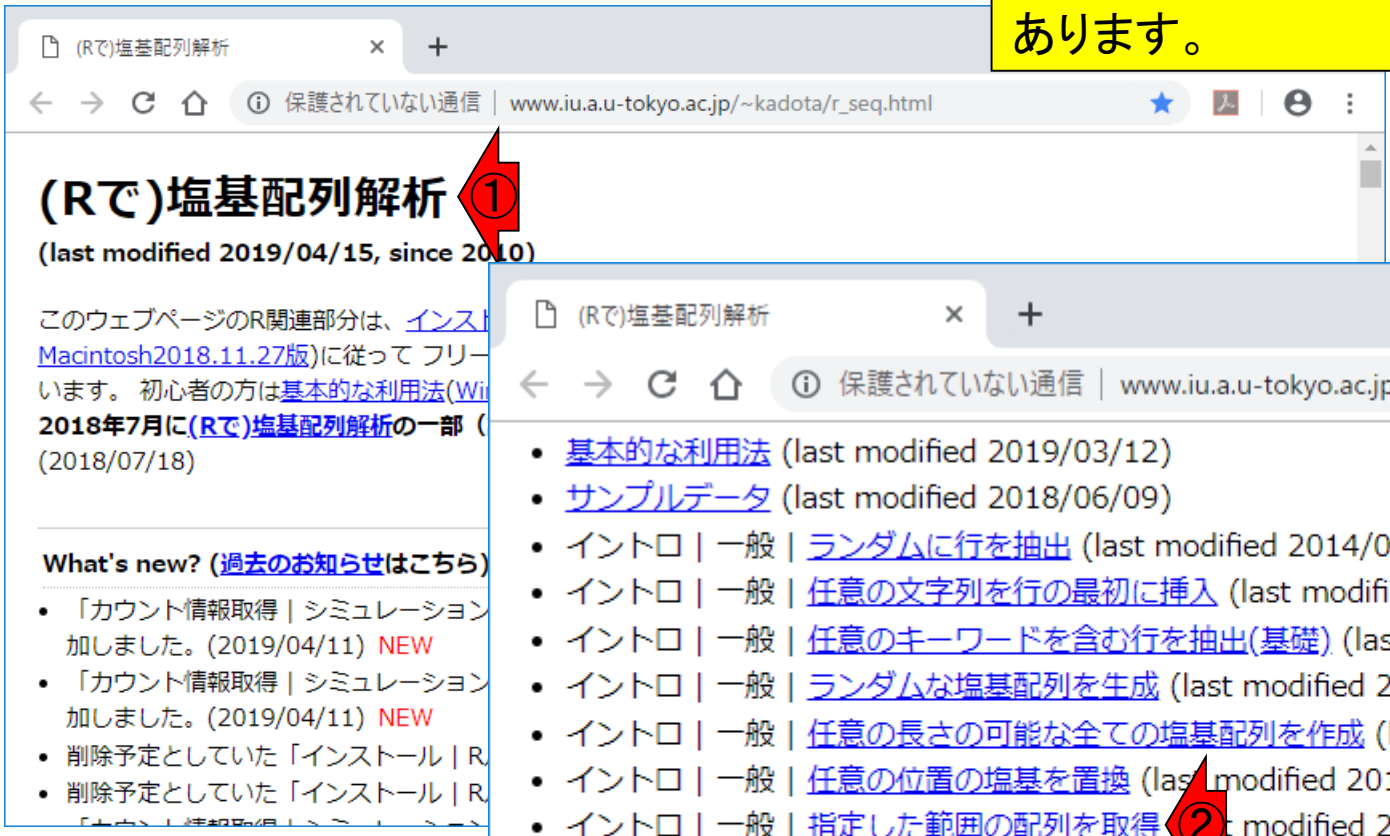
⑤

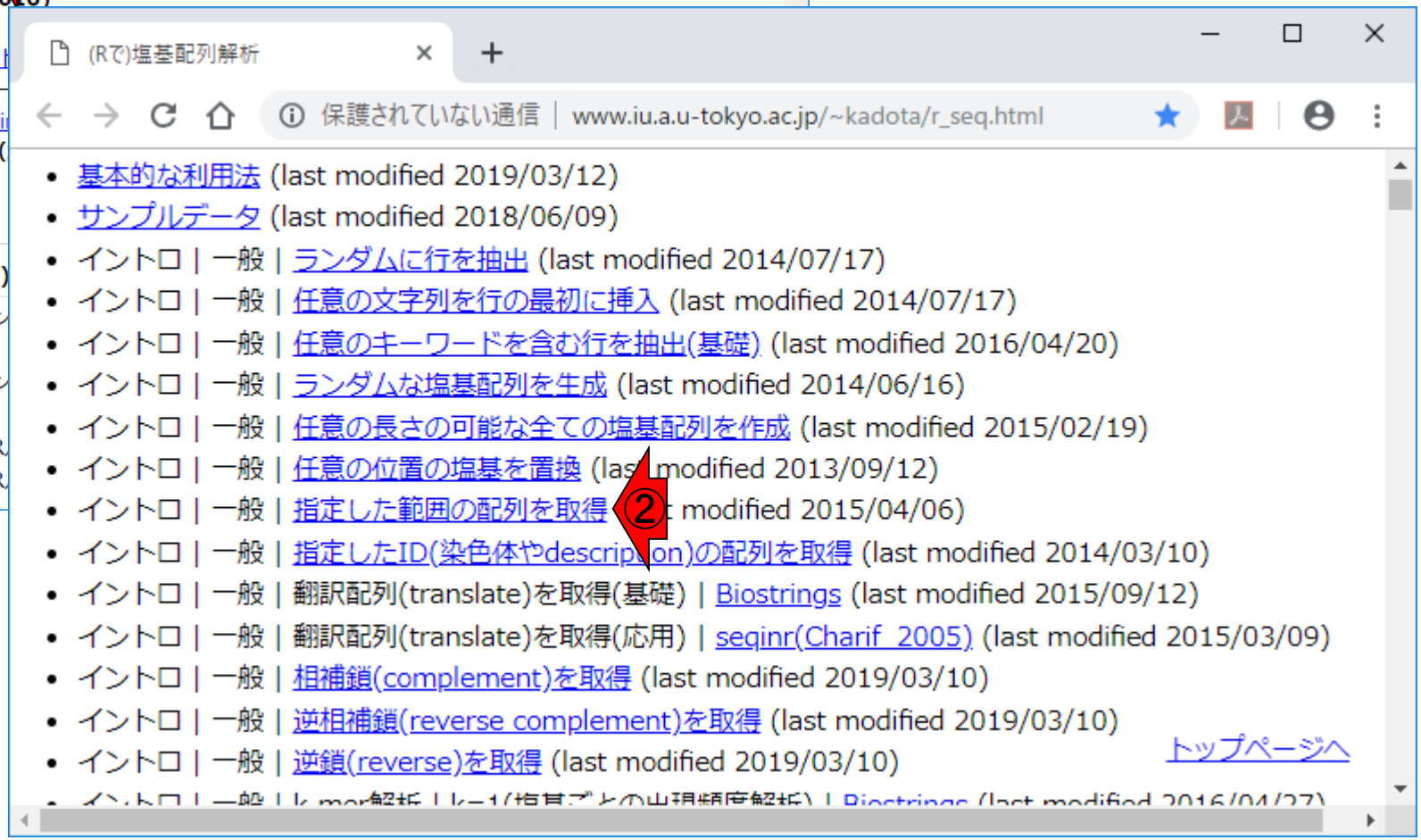
Contents

- 配列長でフィルタリング（ゲノムアセンブリ結果の後処理）
  - イントロ、本番
- ゲノムアノテーション
  - イントロ、参考文献
  - DFAST実行、課題1
- CDS（タンパク質コード領域の配列；coding sequenceの略）の取得
  - 比較ゲノム解析のイントロ、問題設定
  - 答え（cds.fna）を眺めておく、sequence logos
  - CDS配列取得の成功例、TxDbオブジェクト、CDS座標情報取得
  - CDS配列取得の失敗例、GFF3をExcelで概観
  - 基本テクを駆使して解決する基本戦略、CDSを含む行の抽出、補足説明
  - Excelで確認、リストファイルの作成、基本テクでCDS配列取得を実行
  - ストランド情報つきのリストファイルの作成、例題6のコード解説
  - ストランド情報反映戦略を練り、コードに修正を加えて実行（例題7）

基本戦略

私は①の作者なので、②を使うことで、「この配列名の、ここから、ここまで」という指定の仕方で任意の範囲の部分配列を取得できることが記憶の片隅にあります。





基本戦略

私は①の作者なので、②を使うことで、「この配列名の、ここから、ここまで」という指定の仕方で任意の範囲の部分配列を取得できることが記憶の片隅にあります。③例題5がよさそうと判断。

(Rで)塩基配列解析

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

イントロ | 一般 | 指定した範囲の配列を取得 ②

single-FASTA形式やmulti-FASTA
は、「この染色体の、ここから、
だった場合に、chr3の20000から
chr4とchr8の配列のみ抽出といっ
ロード時に、*.fastaという拡張子
「ファイル」－「ディレクトリの
へ。

1. (single-)FASTA形式ファイル

任意の範囲 (始点が3, 終点が9)

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

#必要なパッケージをロード
library(Biostrings)

#入力ファイルの読み込み
```

(Rで)塩基配列解析

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

③

5. multi-FASTA形式のファイル (ref_genome.fa)ファイルの場合:

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_sub3.txt) を読み込ませて、目的の部分配列のmulti-FASTAファイルをゲットするやり方です。

```
in_f1 <- "ref_genome.fa"                #入力ファイル名を指定してin_f1に格納(multi-FASTAフ
in_f2 <- "list_sub3.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 <- "hoge5.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
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
```

トップページへ

基本戦略

私は①の作者なので、②を使うことで、「この配列名の、ここから、ここまで」という指定の仕方で任意の範囲の部分配列を取得できることが記憶の片隅にあります。③例題5がよさそうと判断。④が抽出したい領域情報をリスト化したファイル。⑤が中身。

(R で)塩基配列解析

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

イントロ | 一般 | 指定した範囲の配列を取得 ②

single-FASTA形式やmulti-FASTA
は、「この染色体の、ここから、
だった場合に、chr3の20000から
chr4とchr8の配列のみ抽出といっ
ロード時に、*.fastaという拡張子
「ファイル」－「ディレクトリの
ペ。

1. (single-)FASTA形式ファイル

任意の範囲 (始点が3, 終点が9)(

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

#必要なパッケージをロード
library(Biostrings)

#入力ファイルの読み込み
```

(R で)塩基配列解析

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

③

5. multi-FASTA形式のファイル (ref_genome.fa)ファイルの場合:

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_sub3.txt) を読み込ませて、目的の部分配列のmulti-FASTAファイルをゲットするやり方です。

④

```
in_f1 <- "ref_genome.fa"                #入力ファイル名を指定してin_f1に格納(m
in_f2 <- "list_sub3.txt"                #入力ファイル名を指定してin_f2に格納(リ
out_f <- "hoge5.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
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 ブレー
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
```

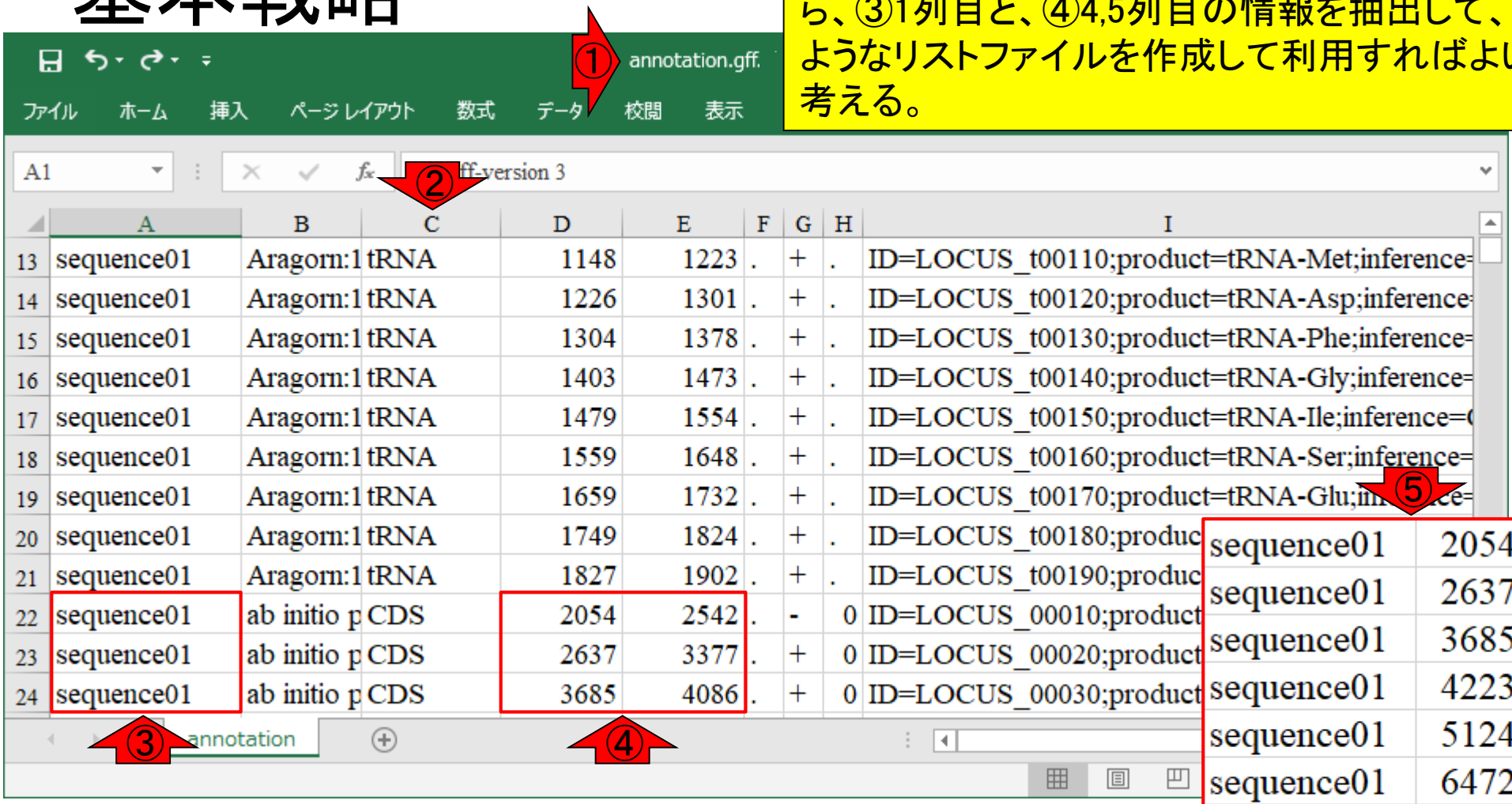
④

⑤

chr1	11	45
chr2	16	50
chr2	1	35
chr3	11	45
chr3	15	49
chr3	3	37
chr3	1	35
chr5	1	35

基本戦略

①annotation.gffの場合は、②3列目にCDSという文字列を含む行をまず抽出しておく。そしてその中から、③1列目と、④4,5列目の情報を抽出して、⑤のようなリストファイルを作成して利用すればよい、と考える。



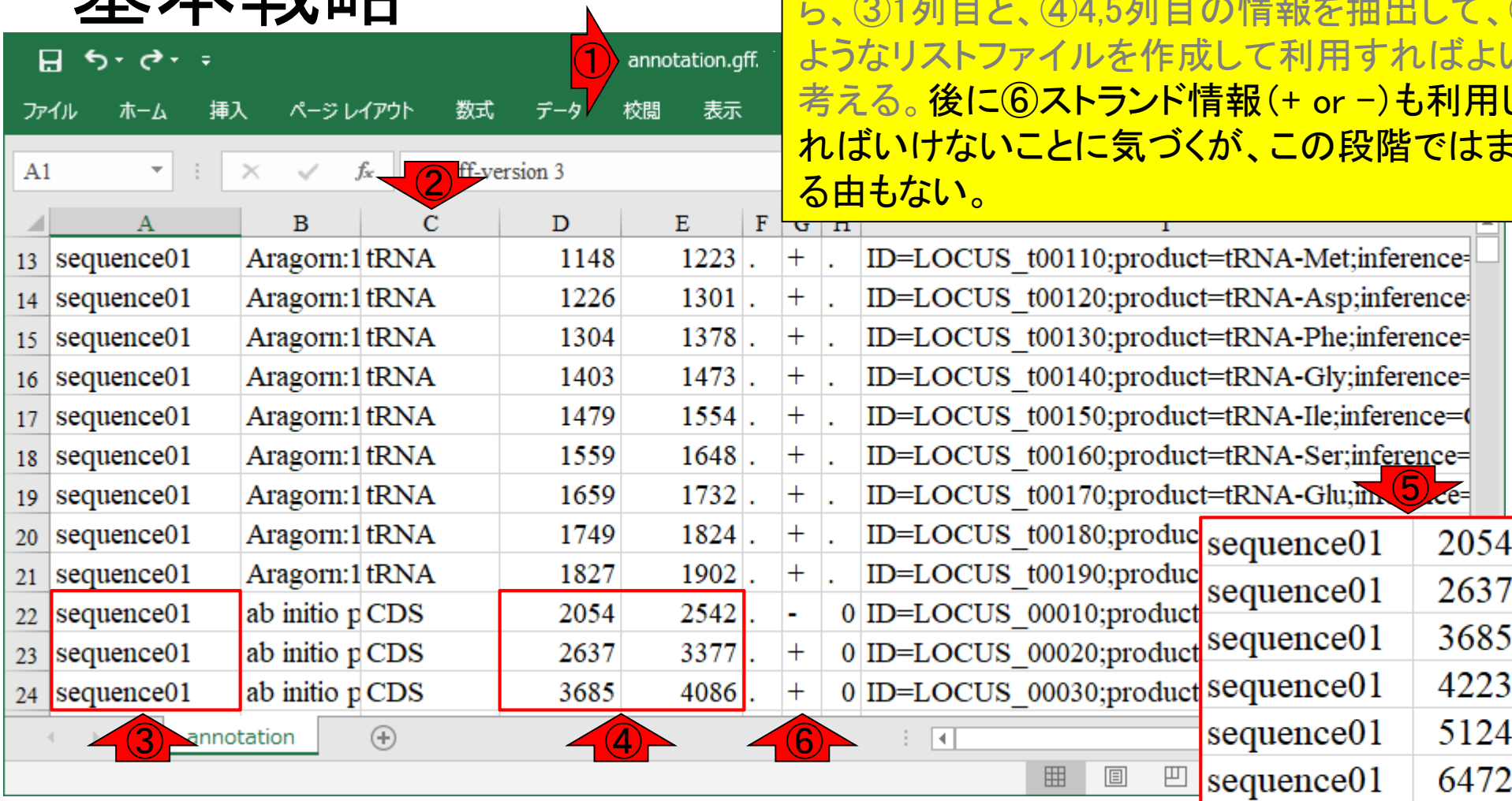
	A	B	C	D	E	F	G	H	I
13	sequence01	Aragorn:1	tRNA	1148	1223	.	+	.	ID=LOCUS_t00110;product=tRNA-Met;inference=
14	sequence01	Aragorn:1	tRNA	1226	1301	.	+	.	ID=LOCUS_t00120;product=tRNA-Asp;inference=
15	sequence01	Aragorn:1	tRNA	1304	1378	.	+	.	ID=LOCUS_t00130;product=tRNA-Phe;inference=
16	sequence01	Aragorn:1	tRNA	1403	1473	.	+	.	ID=LOCUS_t00140;product=tRNA-Gly;inference=
17	sequence01	Aragorn:1	tRNA	1479	1554	.	+	.	ID=LOCUS_t00150;product=tRNA-Ile;inference=(
18	sequence01	Aragorn:1	tRNA	1559	1648	.	+	.	ID=LOCUS_t00160;product=tRNA-Ser;inference=
19	sequence01	Aragorn:1	tRNA	1659	1732	.	+	.	ID=LOCUS_t00170;product=tRNA-Glu;in ce=
20	sequence01	Aragorn:1	tRNA	1749	1824	.	+	.	ID=LOCUS_t00180;produc
21	sequence01	Aragorn:1	tRNA	1827	1902	.	+	.	ID=LOCUS_t00190;produc
22	sequence01	ab initio p	CDS	2054	2542	.	-	0	ID=LOCUS_00010;product
23	sequence01	ab initio p	CDS	2637	3377	.	+	0	ID=LOCUS_00020;product
24	sequence01	ab initio p	CDS	3685	4086	.	+	0	ID=LOCUS_00030;product

⑤

sequence01	2054	2542
sequence01	2637	3377
sequence01	3685	4086
sequence01	4223	4927
sequence01	5124	6293
sequence01	6472	7248
sequence01	7366	7878
sequence01	7976	8380
…		

基本戦略

①annotation.gffの場合は、②3列目にCDSという文字列を含む行をまず抽出しておく。そしてその中から、③1列目と、④4,5列目の情報を抽出して、⑤のようなリストファイルを作成して利用すればよい、と考える。後に⑥ストランド情報（+ or −）も利用しなければいけないことに気づくが、この段階ではまだ知る由もない。



	A	B	C	D	E	F	G	H	I
13	sequence01	Aragorn:1	tRNA	1148	1223	.	+	.	ID=LOCUS_t00110;product=tRNA-Met;inference=
14	sequence01	Aragorn:1	tRNA	1226	1301	.	+	.	ID=LOCUS_t00120;product=tRNA-Asp;inference=
15	sequence01	Aragorn:1	tRNA	1304	1378	.	+	.	ID=LOCUS_t00130;product=tRNA-Phe;inference=
16	sequence01	Aragorn:1	tRNA	1403	1473	.	+	.	ID=LOCUS_t00140;product=tRNA-Gly;inference=
17	sequence01	Aragorn:1	tRNA	1479	1554	.	+	.	ID=LOCUS_t00150;product=tRNA-Ile;inference=(
18	sequence01	Aragorn:1	tRNA	1559	1648	.	+	.	ID=LOCUS_t00160;product=tRNA-Ser;inference=
19	sequence01	Aragorn:1	tRNA	1659	1732	.	+	.	ID=LOCUS_t00170;product=tRNA-Glu;in…ce=
20	sequence01	Aragorn:1	tRNA	1749	1824	.	+	.	ID=LOCUS_t00180;produc
21	sequence01	Aragorn:1	tRNA	1827	1902	.	+	.	ID=LOCUS_t00190;produc
22	sequence01	ab initio p	CDS	2054	2542	.	-	0	ID=LOCUS_00010;product
23	sequence01	ab initio p	CDS	2637	3377	.	+	0	ID=LOCUS_00020;product
24	sequence01	ab initio p	CDS	3685	4086	.	+	0	ID=LOCUS_00030;product

⑤

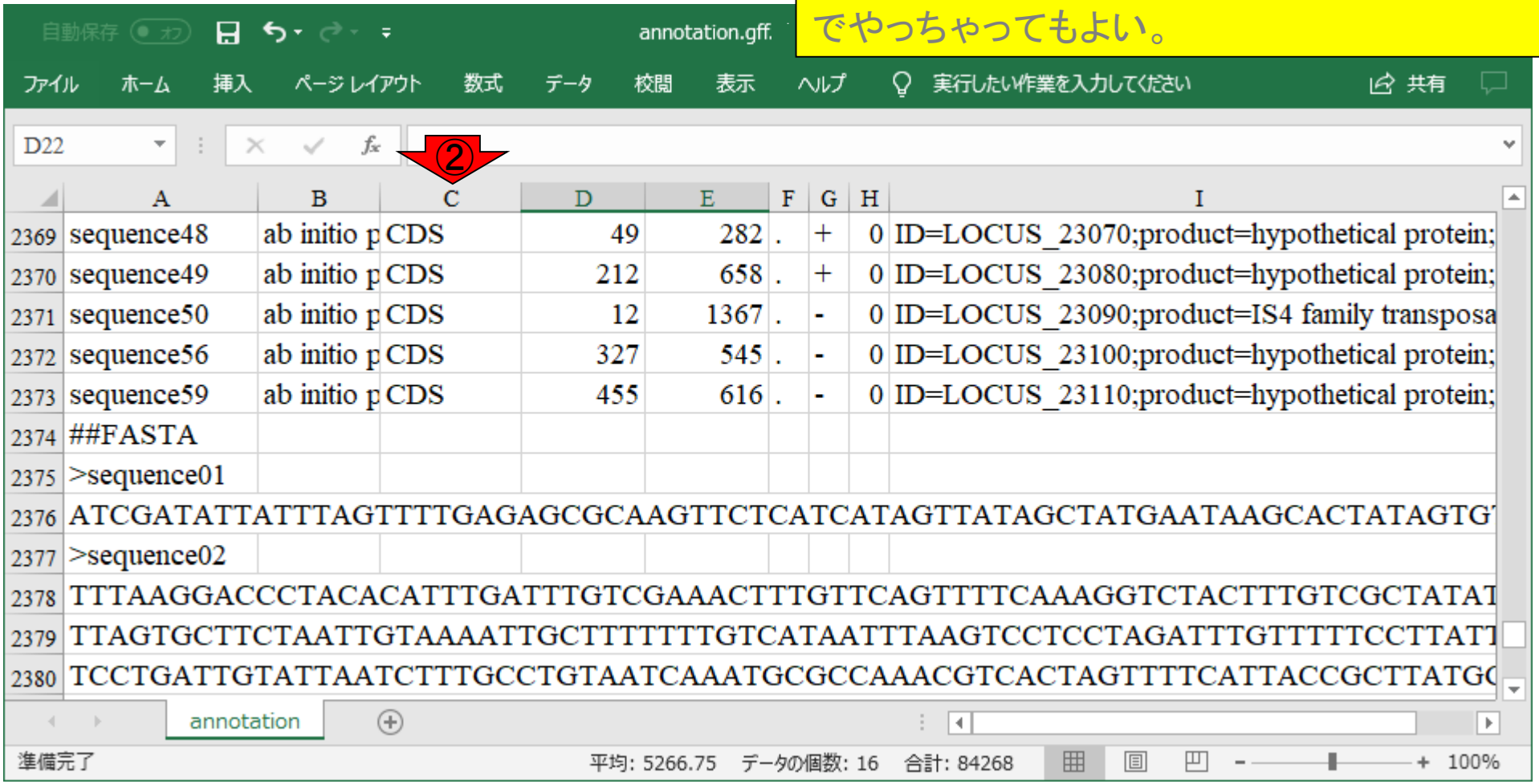
sequence01	2054	2542
sequence01	2637	3377
sequence01	3685	4086
sequence01	4223	4927
sequence01	5124	6293
sequence01	6472	7248
sequence01	7366	7878
sequence01	7976	8380
…		

Contents

- 配列長でフィルタリング（ゲノムアセンブリ結果の後処理）
  - イントロ、本番
- ゲノムアノテーション
  - イントロ、参考文献
  - DFAST実行、課題1
- CDS（タンパク質コード領域の配列；coding sequenceの略）の取得
  - 比較ゲノム解析のイントロ、問題設定
  - 答え（cds.fna）を眺めておく、sequence logos
  - CDS配列取得の成功例、TxDbオブジェクト、CDS座標情報取得
  - CDS配列取得の失敗例、GFF3をExcelで概観
  - 基本テクを駆使して解決する基本戦略、CDSを含む行の抽出、補足説明
  - Excelで確認、リストファイルの作成、基本テクでCDS配列取得を実行
  - ストランド情報つきのリストファイルの作成、例題6のコード解説
  - ストランド情報反映戦略を練り、コードに修正を加えて実行（例題7）

CDSを含む行の抽出

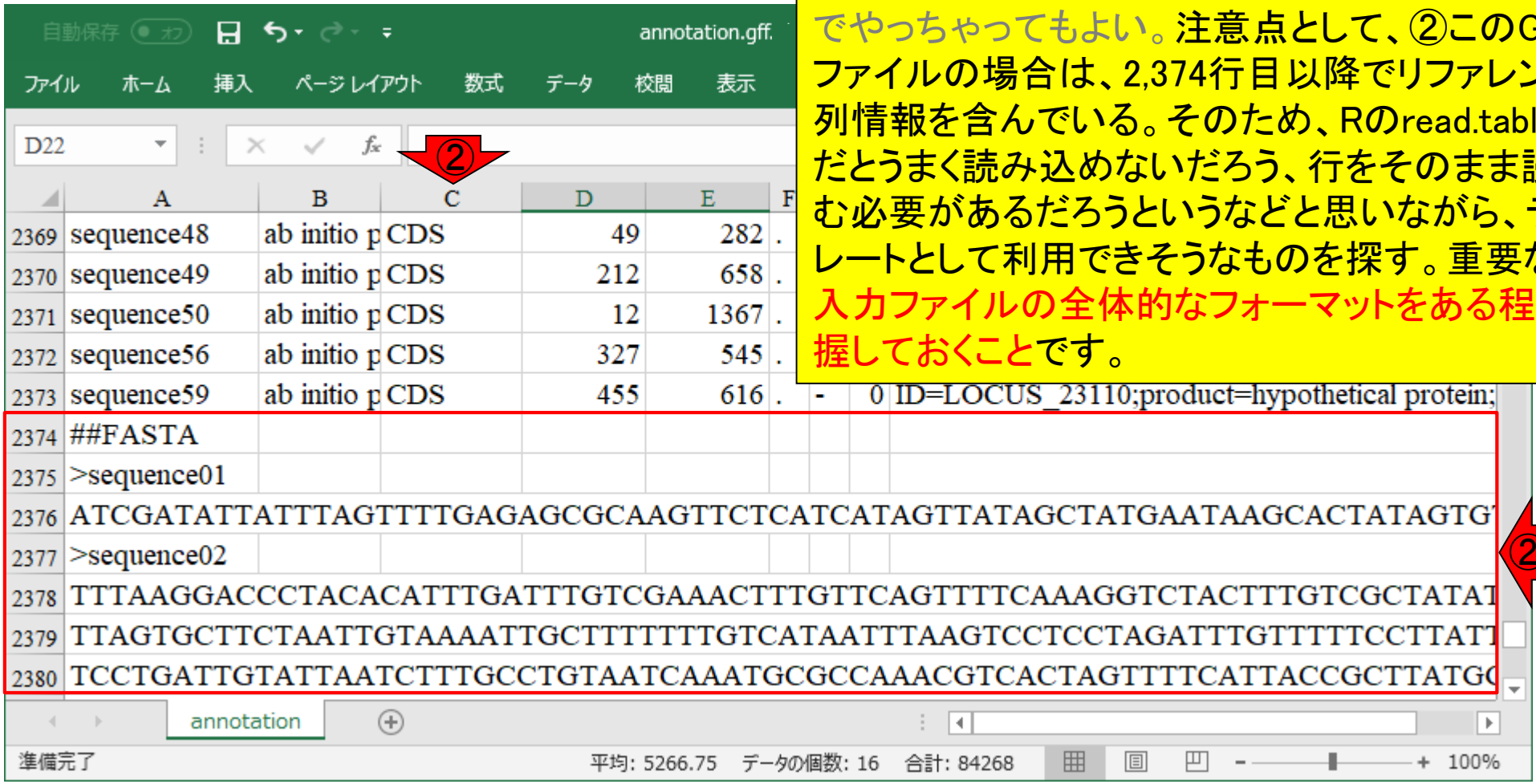
まずは、①3列目にCDSという文字列を含む行を抽出する。CDS配列は2,311個なので、2,311行のファイルになればよい、という感覚をもってやる。Excelでやっちゃってもよい。



	A	B	C	D	E	F	G	H	I
2369	sequence48	ab initio p	CDS	49	282	.	+	0	ID=LOCUS_23070;product=hypothetical protein;
2370	sequence49	ab initio p	CDS	212	658	.	+	0	ID=LOCUS_23080;product=hypothetical protein;
2371	sequence50	ab initio p	CDS	12	1367	.	-	0	ID=LOCUS_23090;product=IS4 family transposa
2372	sequence56	ab initio p	CDS	327	545	.	-	0	ID=LOCUS_23100;product=hypothetical protein;
2373	sequence59	ab initio p	CDS	455	616	.	-	0	ID=LOCUS_23110;product=hypothetical protein;
2374	##FASTA								
2375	>sequence01								
2376	ATCGATATTATTTAGTTTTGAGAGCGCAAGTTCTCATCATAGTTATAGCTATGAATAAGCACTATAGTG								
2377	>sequence02								
2378	TTTAAGGACCCTACACATTTGATTTGTCGAAACTTTGTTCAGTTTTCAAAGGTCTACTTTGTCGCTATAT								
2379	TTAGTGCTTCTAATTGTAAAATTGCTTTTTTTGTCATAATTTAAGTCCTCCTAGATTTGTTTTTCCTTATT								
2380	TCCTGATTGTATTAATCTTTGCCTGTAATCAAATGCGCCAAACGTCACTAGTTTTCATTACCGCTTATGC								

CDSを含む行の抽出

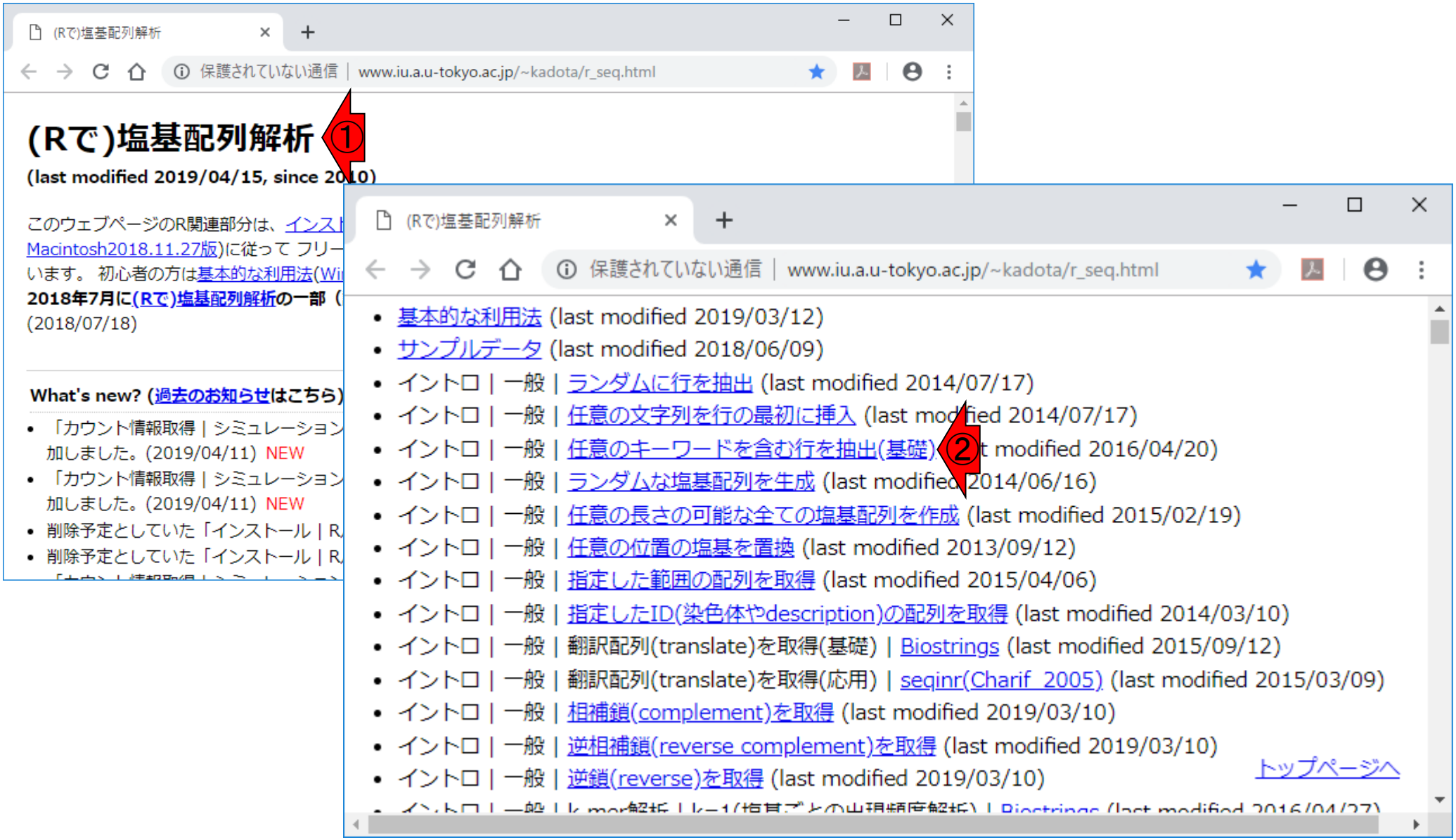
まずは、①3列目にCDSという文字列を含む行を抽出する。CDS配列は2,311個なので、2,311行のファイルになればよい、という感覚をもってやる。Excelでやっちゃってもよい。注意点として、②このGFFファイルの場合は、2,374行目以降でリファレンス配列情報を含んでいる。そのため、Rのread.table関数だとうまく読み込めないだろう、行をそのまま読み込む必要があるだろうというなどと思いながら、テンプレートとして利用できそうなものを探す。重要なのは、入力ファイルの全体的なフォーマットをある程度把握しておくことです。



	A	B	C	D	E	F			
2369	sequence48	ab initio p	CDS	49	282	.			
2370	sequence49	ab initio p	CDS	212	658	.			
2371	sequence50	ab initio p	CDS	12	1367	.			
2372	sequence56	ab initio p	CDS	327	545	.			
2373	sequence59	ab initio p	CDS	455	616	.	-	0	ID=LOCUS_23110;product=hypothetical protein;
2374	##FASTA								
2375	>sequence01								
2376	ATCGATATTATTTAGTTTTGAGAGCGCAAGTTCTCATCATAGTTATAGCTATGAATAAGCACTATAGTG								
2377	>sequence02								
2378	TTTAAGGACCCTACACATTTGATTTGTCGAAACTTTGTTCAGTTTTCAAAGGTCTACTTTGTCGCTATAT								
2379	TTAGTGCTTCTAATTGTAAAATTGCTTTTTTTGTCATAATTTAAGTCCTCCTAGATTTGTTTTTCCTTATT								
2380	TCCTGATTGTATTAATCTTTGCCTGTAATCAAATGCGCCAAACGTCACTAGTTTTCATTACCGCTTATGC								

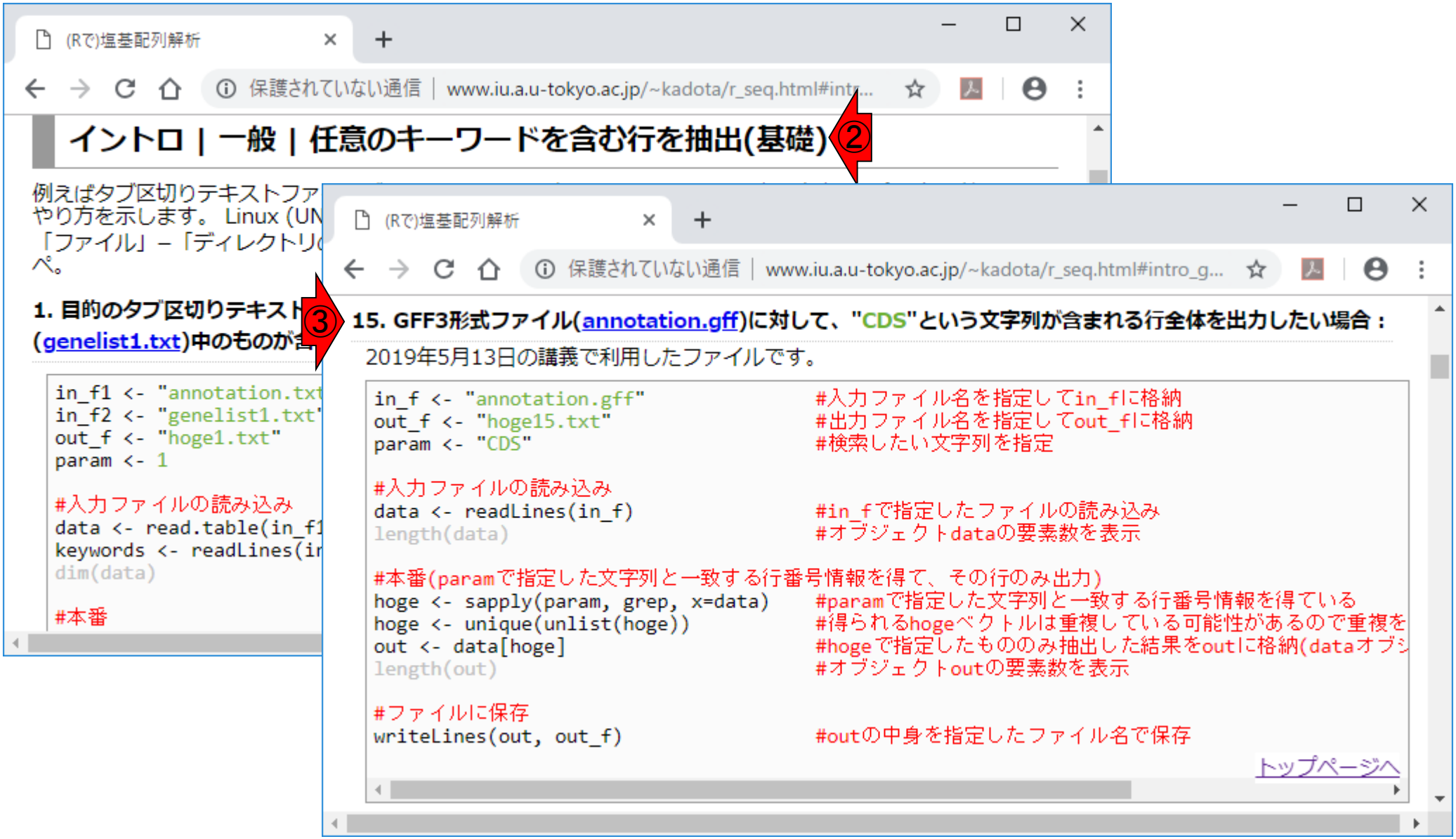
②で目的を達成できそう。

CDSを含む行の抽出



②で目的を達成できそう。③例題15。

CDSを含む行の抽出



CDSを含む行の抽出

②で目的を達成できそう。③例題15。コピペ実行後の状態。

(Rで)塩基配列解析

保護されていない通信 | www.iu.a.

③

15. GFF3形式ファイル(annotation.gff)に対して、

2019年5月13日の講義で利用したファイルです。

```
in_f <- "annotation.gff"                #入
out_f <- "hoge15.txt"                   #出
param <- "CDS"                          #検

#入力ファイルの読み込み
data <- readLines(in_f)                 #in
length(data)                            #オ

#本番(paramで指定した文字列と一致する行番号情
hoge <- sapply(param, grep, x=data)     #pa
hoge <- unique(unlist(hoge))            #得
out <- data[hoge]                       #ho
length(out)                             #オ

#ファイルに保存
writeLines(out, out_f)                  #ou
```

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウインドウ ヘルプ

R Console

```
> param <- "CDS"                        #検索したい$
>
> #入力ファイルの読み込み
> data <- readLines(in_f)               #in_fで指定$
> length(data)                          #オブジェク$
[1] 2496
>
> #本番(paramで指定した文字列と一致する行番号情報を得$
> hoge <- sapply(param, grep, x=data)   #paramで指定$
> hoge <- unique(unlist(hoge))          #得られるhog$
> out <- data[hoge]                     #hogeで指定$
> length(out)                           #オブジェク$
[1] 2311
>
> #ファイルに保存
> writeLines(out, out_f)                #outの中身を$
>
```

CDSを含む行の抽出

①GFFファイルの、②元の行数は2496。そのうち、
③CDSを含む行は、④2311。

(Rで)塩基配列解析

保護されていない通信 | www.iu.a...

15. GFF3形式ファイル(annotation.gff)に対して、

2019年5月13日の講義で利用したファイルです。

```
in_f <- "annotation.gff"                    #入
out_f <- "hoge15.txt"                       #出
param <- "CDS"                              #検

#入力ファイルの読み込み
data <- readLines(in_f)                     #in
length(data)                                #オ

#本番(paramで指定した文字列と一致する行番号情
hoge <- sapply(param, grep, x=data)         #
hoge <- unique(unlist(hoge))                #
out <- data[hoge]                           #ho
length(out)                                 #オ

#ファイルに保存
writeLines(out, out_f)                      #ou
```

① ② ③ ④

RGui (64-bit)

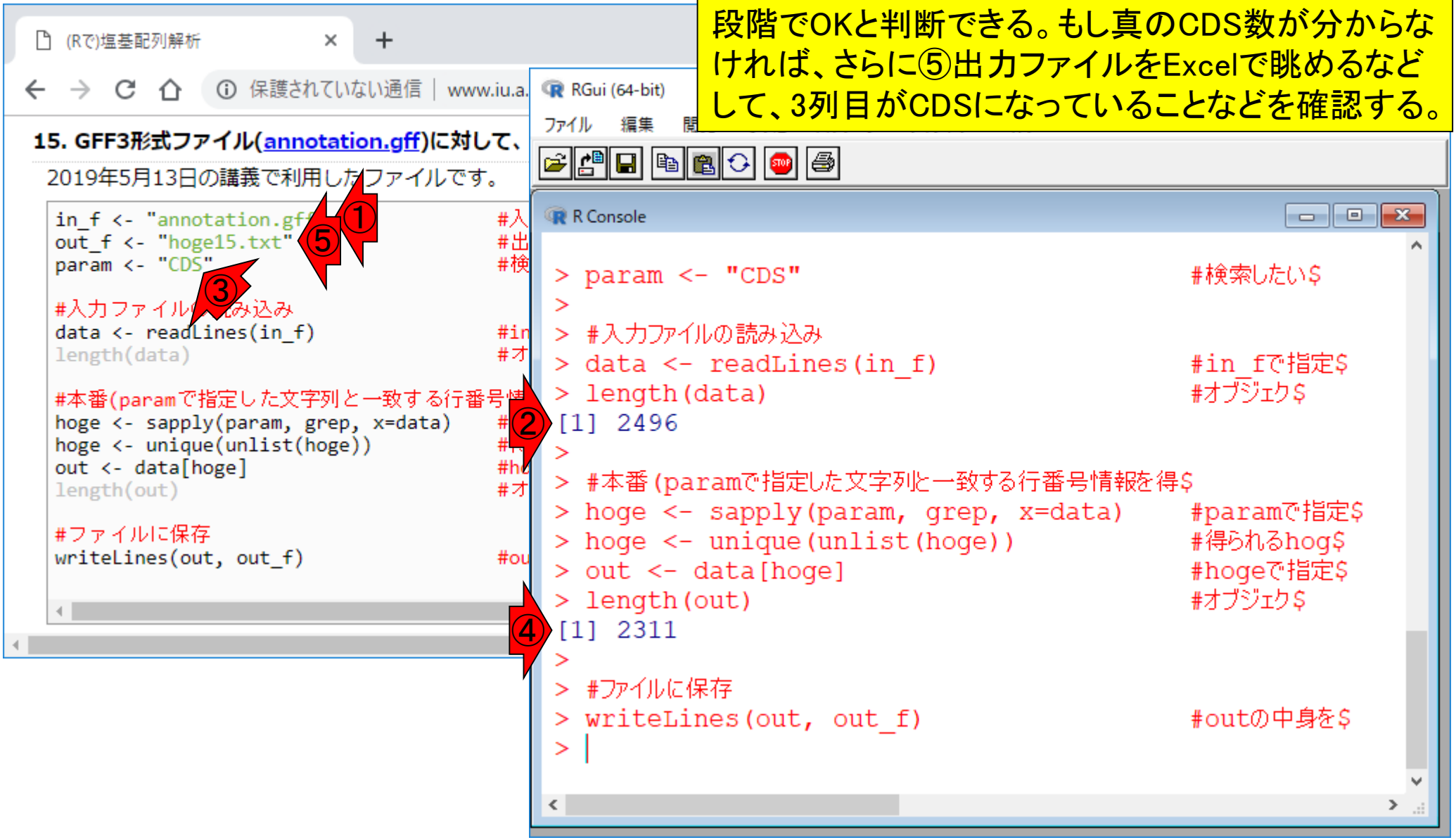
ファイル 編集 閲覧 その他 パッケージ ウインドウ ヘルプ

R Console

```
> param <- "CDS"                            #検索したい$
>
> #入力ファイルの読み込み
> data <- readLines(in_f)                   #in_fで指定$
> length(data)                              #オブジェク$
[1] 2496
>
> #本番(paramで指定した文字列と一致する行番号情報を得$
> hoge <- sapply(param, grep, x=data)       #paramで指定$
> hoge <- unique(unlist(hoge))              #得られるhog$
> out <- data[hoge]                         #hogeで指定$
> length(out)                               #オブジェク$
[1] 2311
>
> #ファイルに保存
> writeLines(out, out_f)                    #outの中身を$
>
```

CDSを含む行の抽出

①GFFファイルの、②元の行数は2496。そのうち、③CDSを含む行は、④2311。今回は真のCDS数が2311個だと分かっているので、④で数値が一致した段階でOKと判断できる。もし真のCDS数が分からなければ、さらに⑤出力ファイルをExcelで眺めるなどして、3列目がCDSになっていることなどを確認する。



Contents

- 配列長でフィルタリング（ゲノムアセンブリ結果の後処理）
  - イントロ、本番
- ゲノムアノテーション
  - イントロ、参考文献
  - DFAST実行、課題1
- CDS（タンパク質コード領域の配列；coding sequenceの略）の取得
  - 比較ゲノム解析のイントロ、問題設定
  - 答え（cds.fna）を眺めておく、sequence logos
  - CDS配列取得の成功例、TxDbオブジェクト、CDS座標情報取得
  - CDS配列取得の失敗例、GFF3をExcelで概観
  - 基本テクを駆使して解決する基本戦略、CDSを含む行の抽出、補足説明
  - Excelで確認、リストファイルの作成、基本テクでCDS配列取得を実行
  - ストランド情報つきのリストファイルの作成、例題6のコード解説
  - ストランド情報反映戦略を練り、コードに修正を加えて実行（例題7）

補足説明

R上では①元の行数は2496となっているが、Excelではそうは見えません。

(Rで)塩基配列解析

保護されていない通信 | www.iu.a.

15. GFF3形式ファイル(annotation.gff)に対して、

2019年5月13日の講義で利用したファイルです。

```
in_f <- "annotation.gff"                #入
out_f <- "hoge15.txt"                   #出
param <- "CDS"                          #検

#入力ファイルの読み込み
data <- readLines(in_f)                 #in
length(data)                            #オ

#本番(paramで指定した文字列と一致する行番号情
hoge <- sapply(param, grep, x=data)     #
hoge <- unique(unlist(hoge))            #
out <- data[hoge]                       #ho
length(out)                             #オ

#ファイルに保存
writeLines(out, out_f)                  #ou
```

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウインドウ ヘルプ

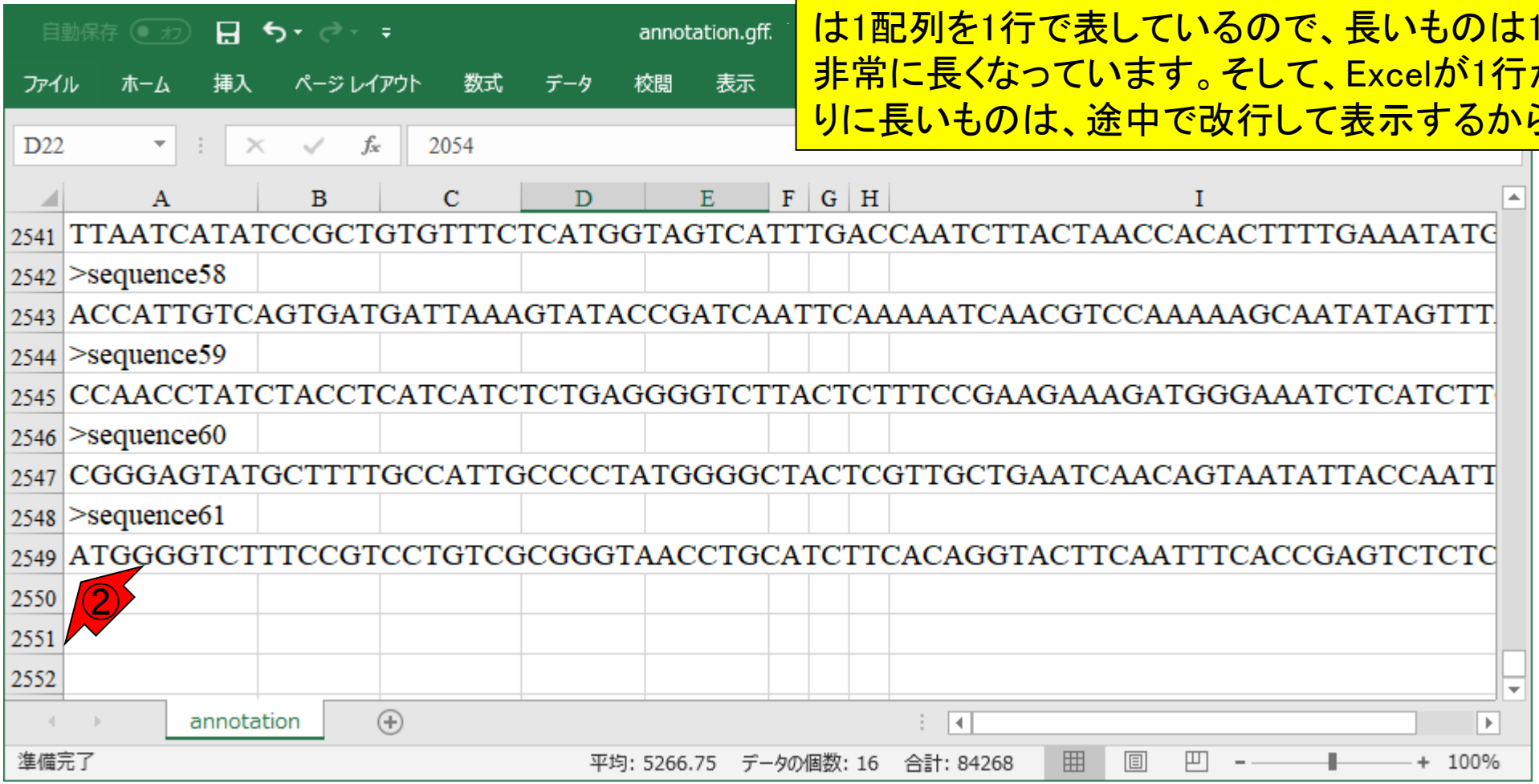
R Console

```
> param <- "CDS"                        #検索したい$
>
> #入力ファイルの読み込み
> data <- readLines(in_f)               #in_fで指定$
> length(data)                          #オブジェク$
[1] 2496
>
> #本番(paramで指定した文字列と一致する行番号情報を得$
> hoge <- sapply(param, grep, x=data)   #paramで指定$
> hoge <- unique(unlist(hoge))          #得られるhog$
> out <- data[hoge]                     #hogeで指定$
> length(out)                           #オブジェク$
[1] 2311
>
> #ファイルに保存
> writeLines(out, out_f)                #outの中身を$
>
```

①

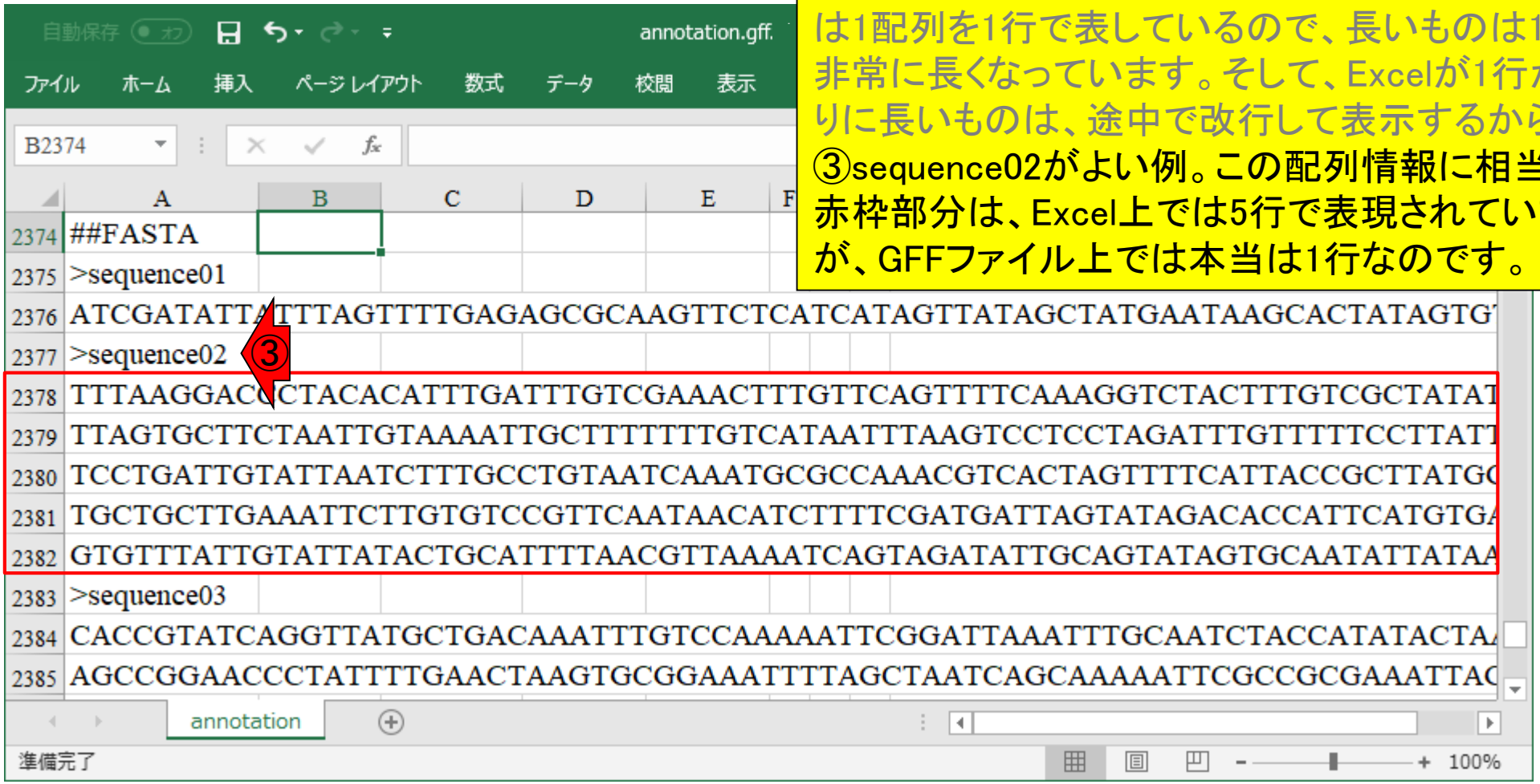
補足説明

R上では①元の行数は2496となっているが、Excelではそうは見えません。こんな感じになって、②最終行は2549となります。この理由は、このGFFファイルは1配列を1行で表しているので、長いものは1行が非常に長くなっています。そして、Excelが1行があまりに長いものは、途中で改行して表示するからです。



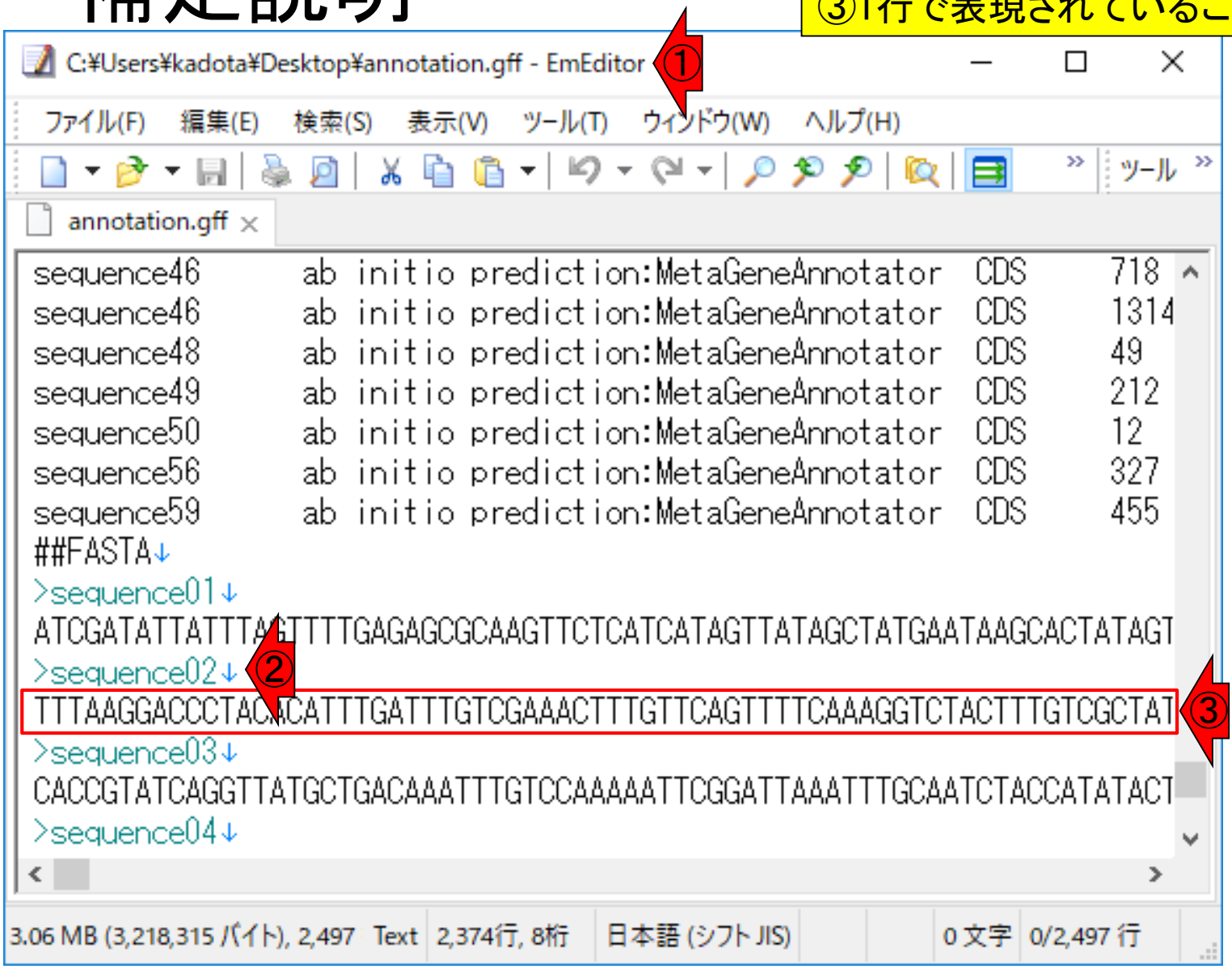
補足説明

R上では①元の行数は2496となっているが、Excelではそうは見えません。こんな感じになって、②最終行は2549となります。この理由は、このGFFファイルは1配列を1行で表しているので、長いものは1行が非常に長くなっています。そして、Excelが1行があまりに長いものは、途中で改行して表示するからです。
③sequence02がよい例。この配列情報に相当する赤枠部分は、Excel上では5行で表現されていますが、GFFファイル上では本当は1行なのです。



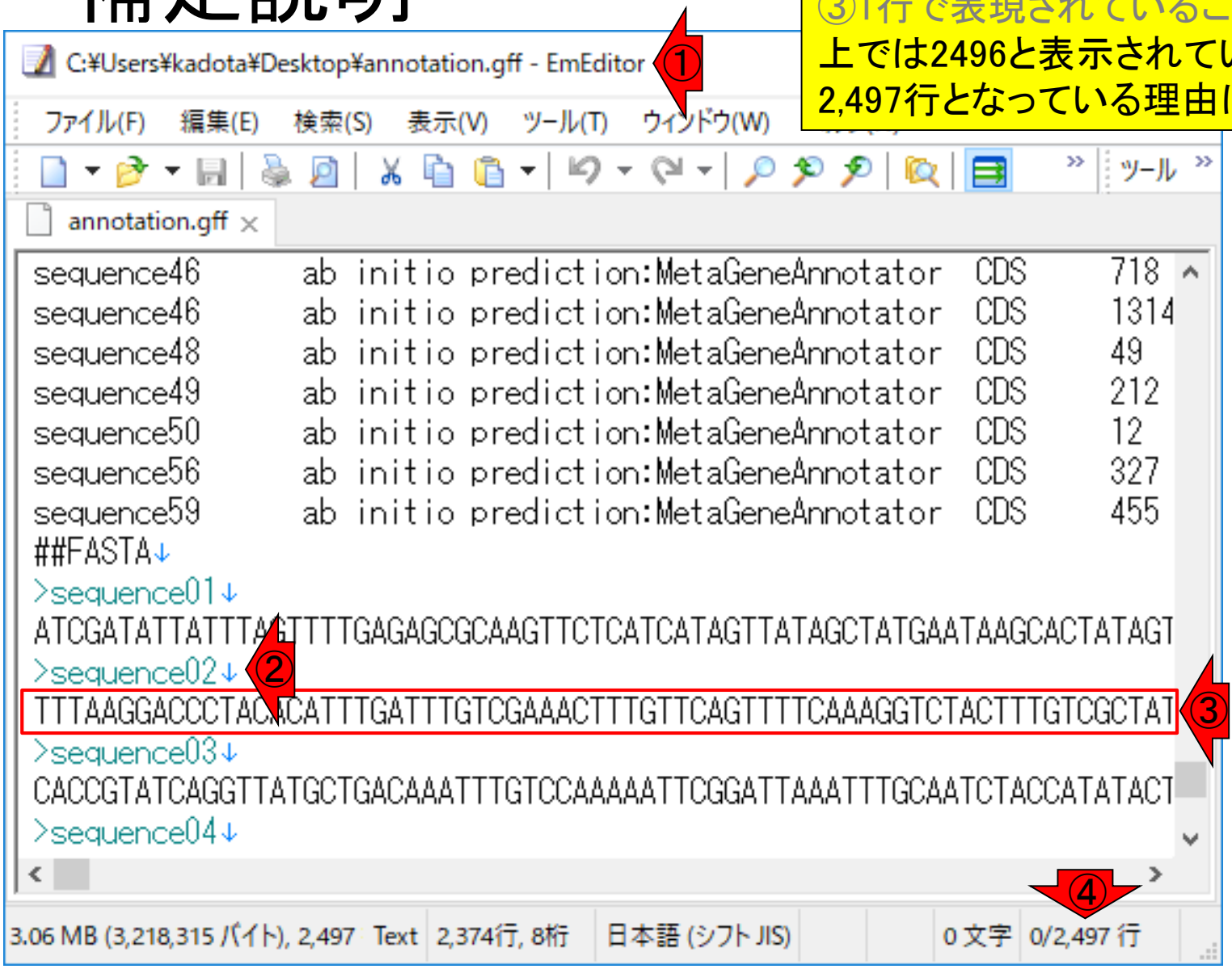
補足説明

Windows用の高機能エディタの1つである、①EmEditorで表示させると、確かに②sequence02は、③1行で表現されていることが確認できます。



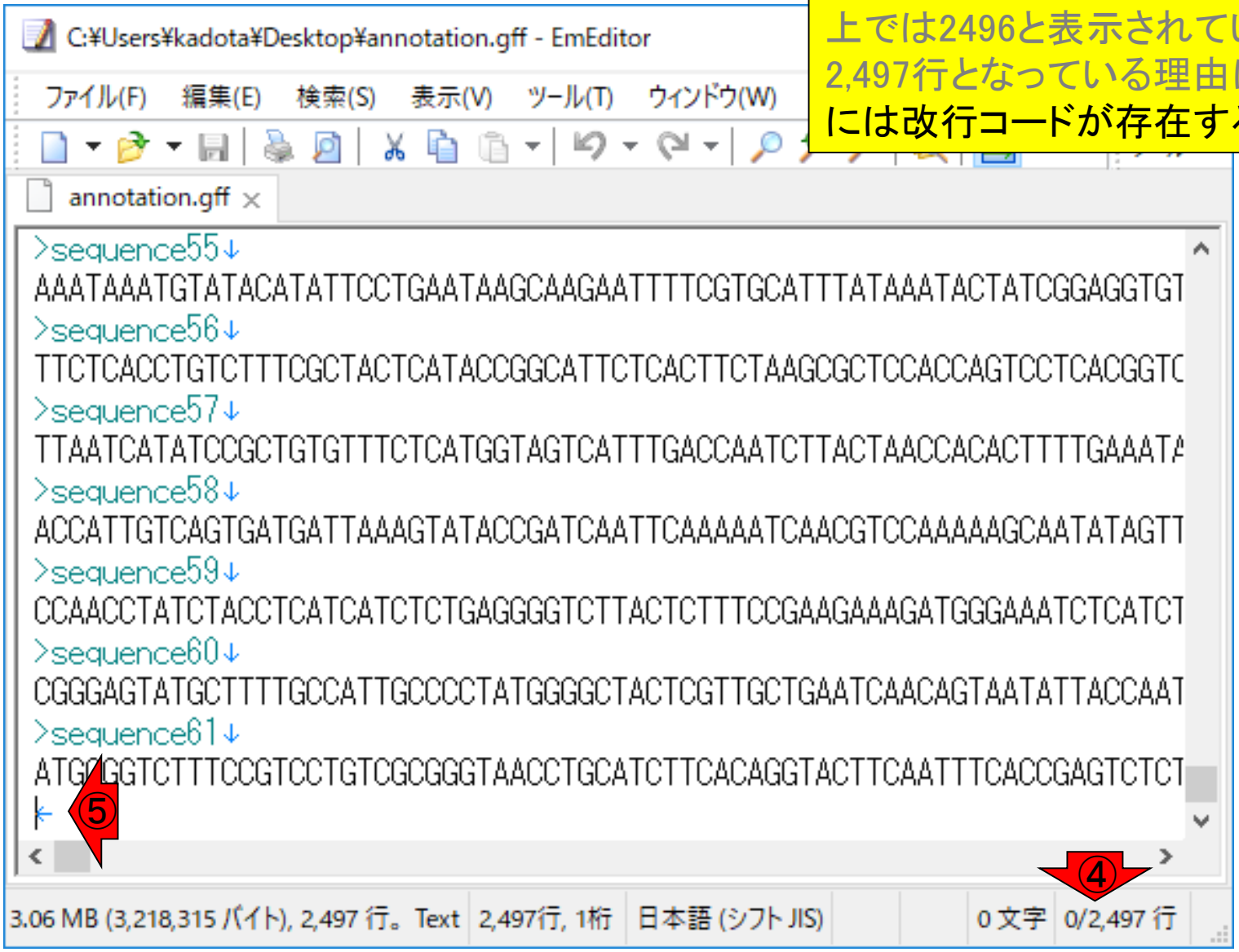
補足説明

Windows用の高機能エディタの1つである、①EmEditorで表示させると、確かに②sequence02は、③1行で表現されていることが確認できます。尚、R上では2496と表示されていたにも関わらず、④が2,497行となっている理由は…



補足説明

Windows用の高機能エディタの1つである、①EmEditorで表示させると、確かに②sequence02は、③1行で表現されていることが確認できます。尚、R上では2496と表示されていたにも関わらず、④が2,497行となっている理由は、⑤この何もない（実際には改行コードが存在する）行が2,497行目だから。

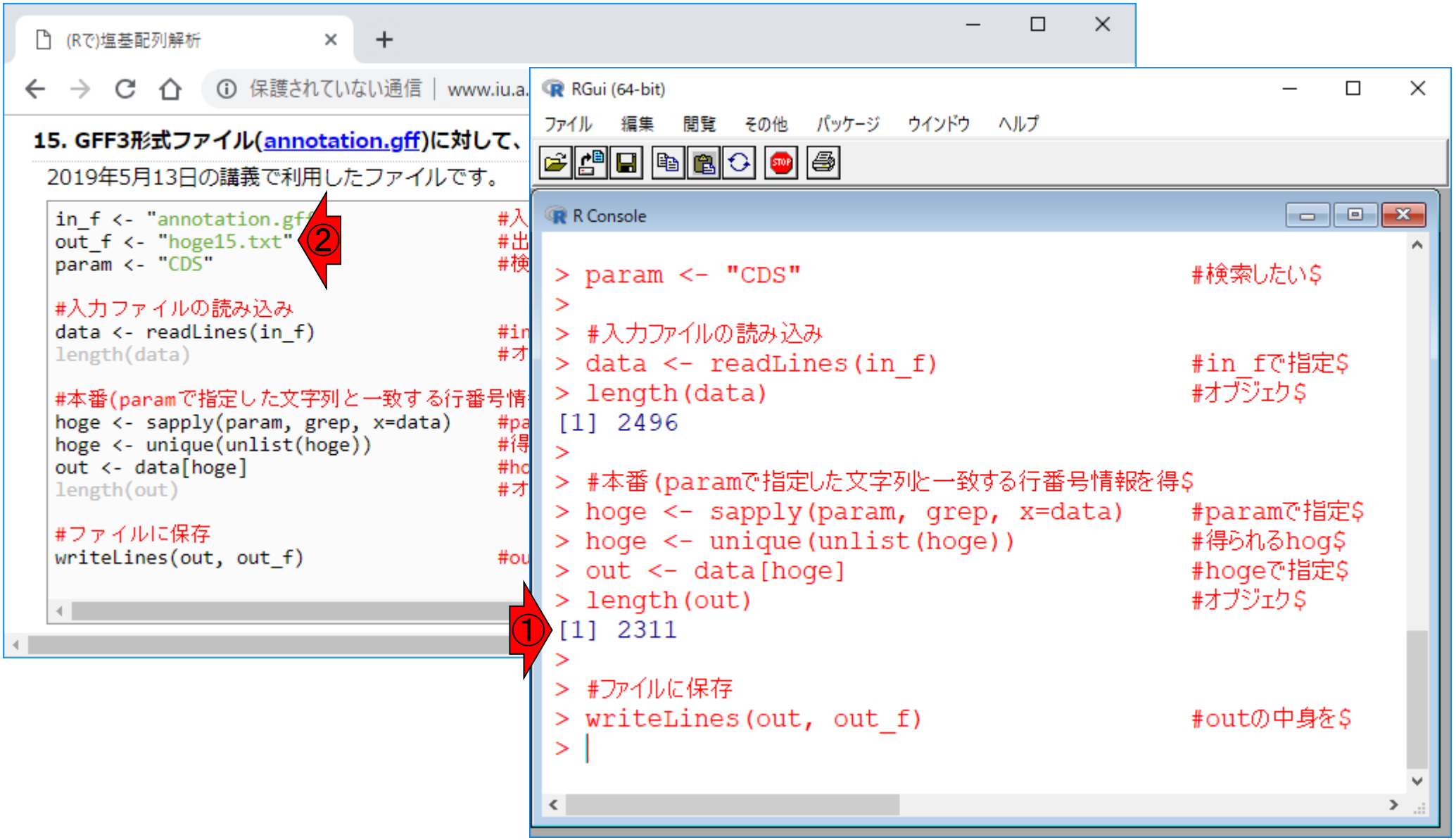


Contents

- 配列長でフィルタリング（ゲノムアセンブリ結果の後処理）
  - イントロ、本番
- ゲノムアノテーション
  - イントロ、参考文献
  - DFAST実行、課題1
- CDS（タンパク質コード領域の配列；coding sequenceの略）の取得
  - 比較ゲノム解析のイントロ、問題設定
  - 答え（cds.fna）を眺めておく、sequence logos
  - CDS配列取得の成功例、TxDbオブジェクト、CDS座標情報取得
  - CDS配列取得の失敗例、GFF3をExcelで概観
  - 基本テクを駆使して解決する基本戦略、CDSを含む行の抽出、補足説明
  - Excelで確認、リストファイルの作成、基本テクでCDS配列取得を実行
  - ストランド情報つきのリストファイルの作成、例題6のコード解説
  - ストランド情報反映戦略を練り、コードに修正を加えて実行（例題7）

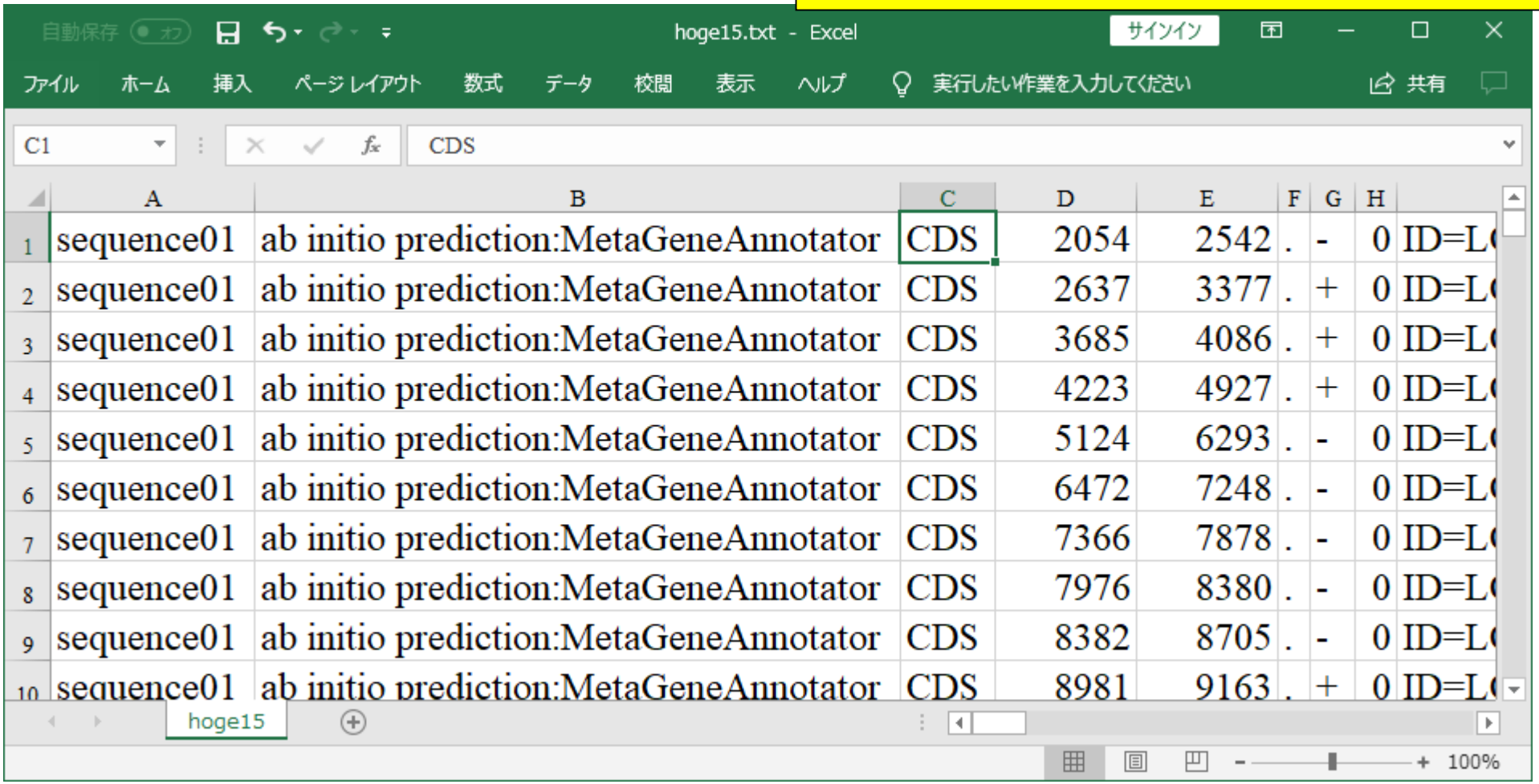
①2,311行からなる、②hoge15.txtを、Excelで確認。

Excelで確認



Excelで確認

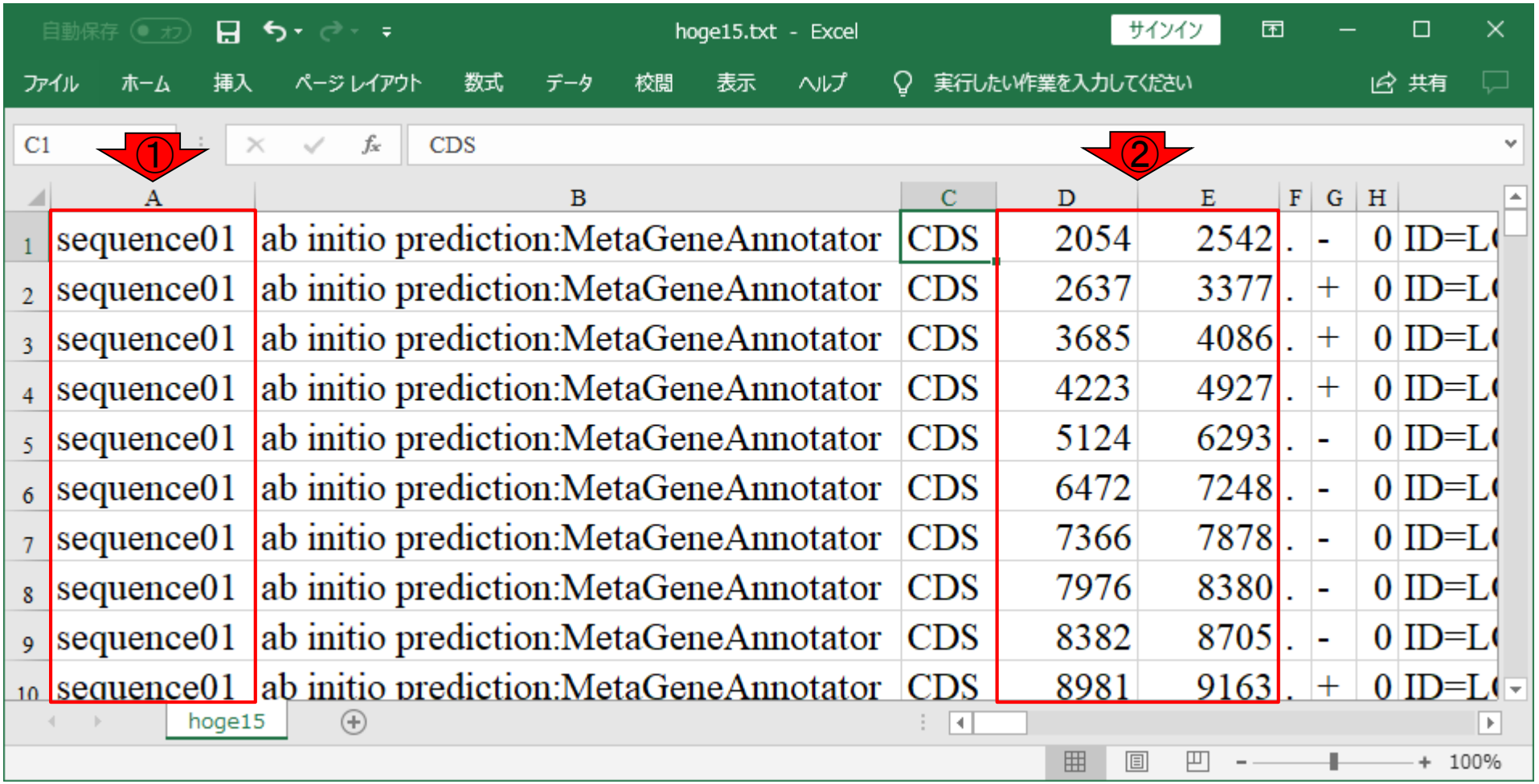
①2,311行からなる、②hoge15.txtを、Excelで確認。
ここでは、3列目が全部CDSになっていることが確認できたので、大丈夫だと判断しました。



	A	B	C	D	E	F	G	H	
1	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	2054	2542	.	-	0	ID=L
2	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	2637	3377	.	+	0	ID=L
3	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	3685	4086	.	+	0	ID=L
4	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	4223	4927	.	+	0	ID=L
5	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	5124	6293	.	-	0	ID=L
6	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	6472	7248	.	-	0	ID=L
7	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	7366	7878	.	-	0	ID=L
8	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	7976	8380	.	-	0	ID=L
9	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	8382	8705	.	-	0	ID=L
10	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	8981	9163	.	+	0	ID=L

①1列目と、②4,5列目の情報のみ抽出して…

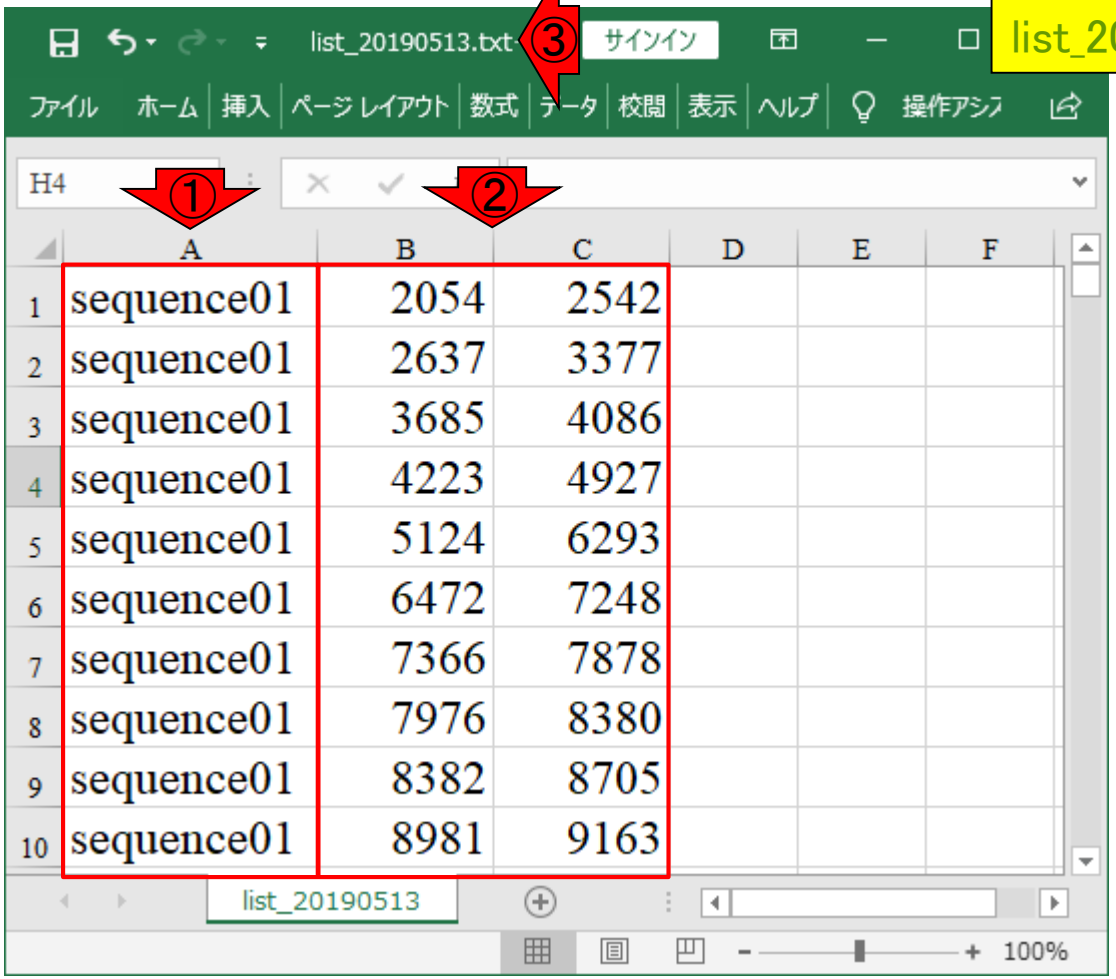
リストファイル作成



	A	B	C	D	E	F	G	H	
1	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	2054	2542	.	-	0	ID=L(
2	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	2637	3377	.	+	0	ID=L(
3	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	3685	4086	.	+	0	ID=L(
4	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	4223	4927	.	+	0	ID=L(
5	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	5124	6293	.	-	0	ID=L(
6	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	6472	7248	.	-	0	ID=L(
7	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	7366	7878	.	-	0	ID=L(
8	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	7976	8380	.	-	0	ID=L(
9	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	8382	8705	.	-	0	ID=L(
10	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	8981	9163	.	+	0	ID=L(

リストファイル作成

①1列目と、②4,5列目の情報のみ抽出して、こんな感じにして「この配列名の、ここから、ここまで」というリストファイルを作成します。ここでは③list_20190513.txtとしました。

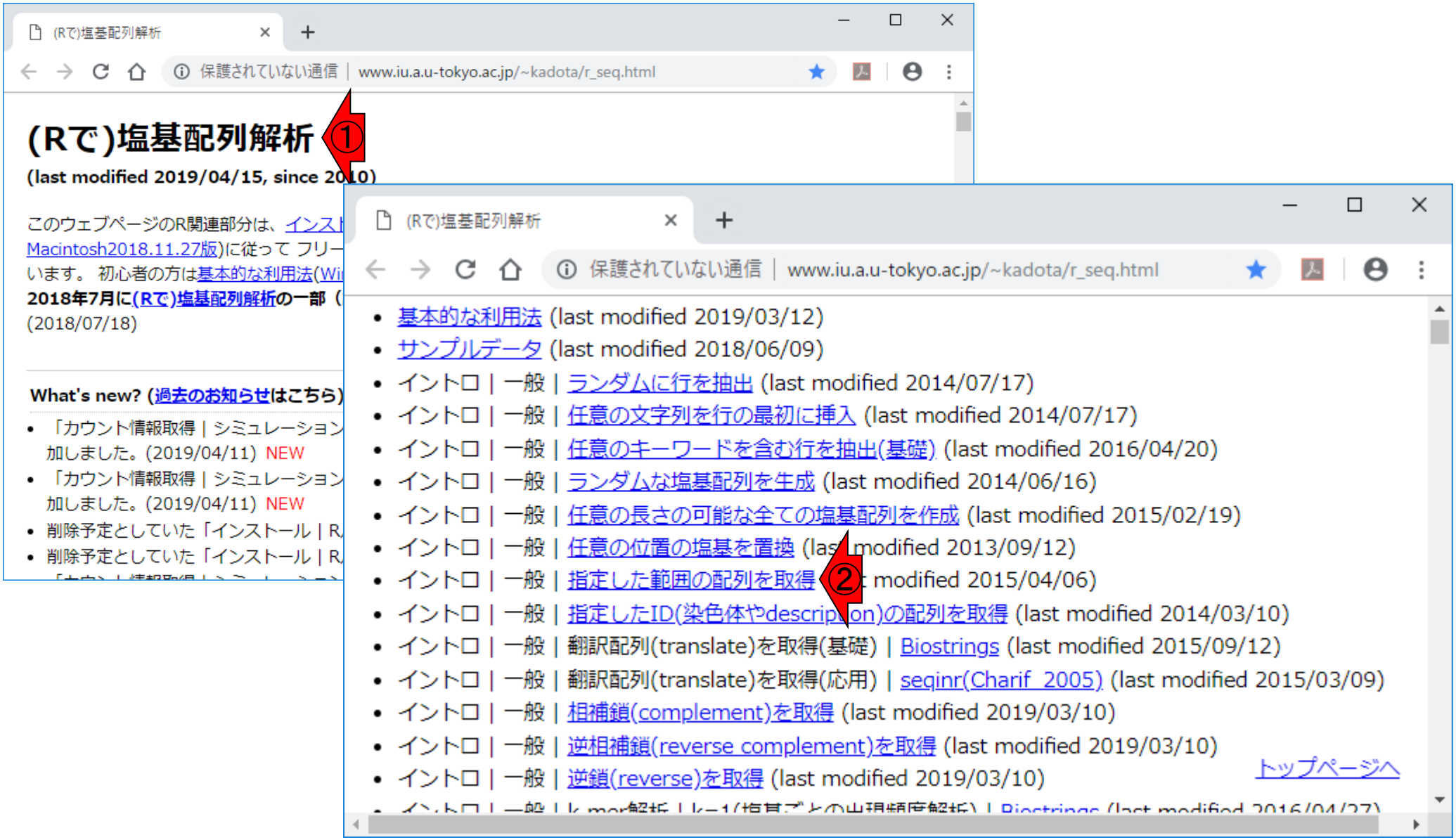


Contents

- 配列長でフィルタリング（ゲノムアセンブリ結果の後処理）
  - イントロ、本番
- ゲノムアノテーション
  - イントロ、参考文献
  - DFAST実行、課題1
- CDS（タンパク質コード領域の配列；coding sequenceの略）の取得
  - 比較ゲノム解析のイントロ、問題設定
  - 答え（cds.fna）を眺めておく、sequence logos
  - CDS配列取得の成功例、TxDbオブジェクト、CDS座標情報取得
  - CDS配列取得の失敗例、GFF3をExcelで概観
  - 基本テクを駆使して解決する基本戦略、CDSを含む行の抽出、補足説明
  - Excelで確認、リストファイルの作成、基本テクでCDS配列取得を実行
  - ストランド情報つきのリストファイルの作成、例題6のコード解説
  - ストランド情報反映戦略を練り、コードに修正を加えて実行（例題7）

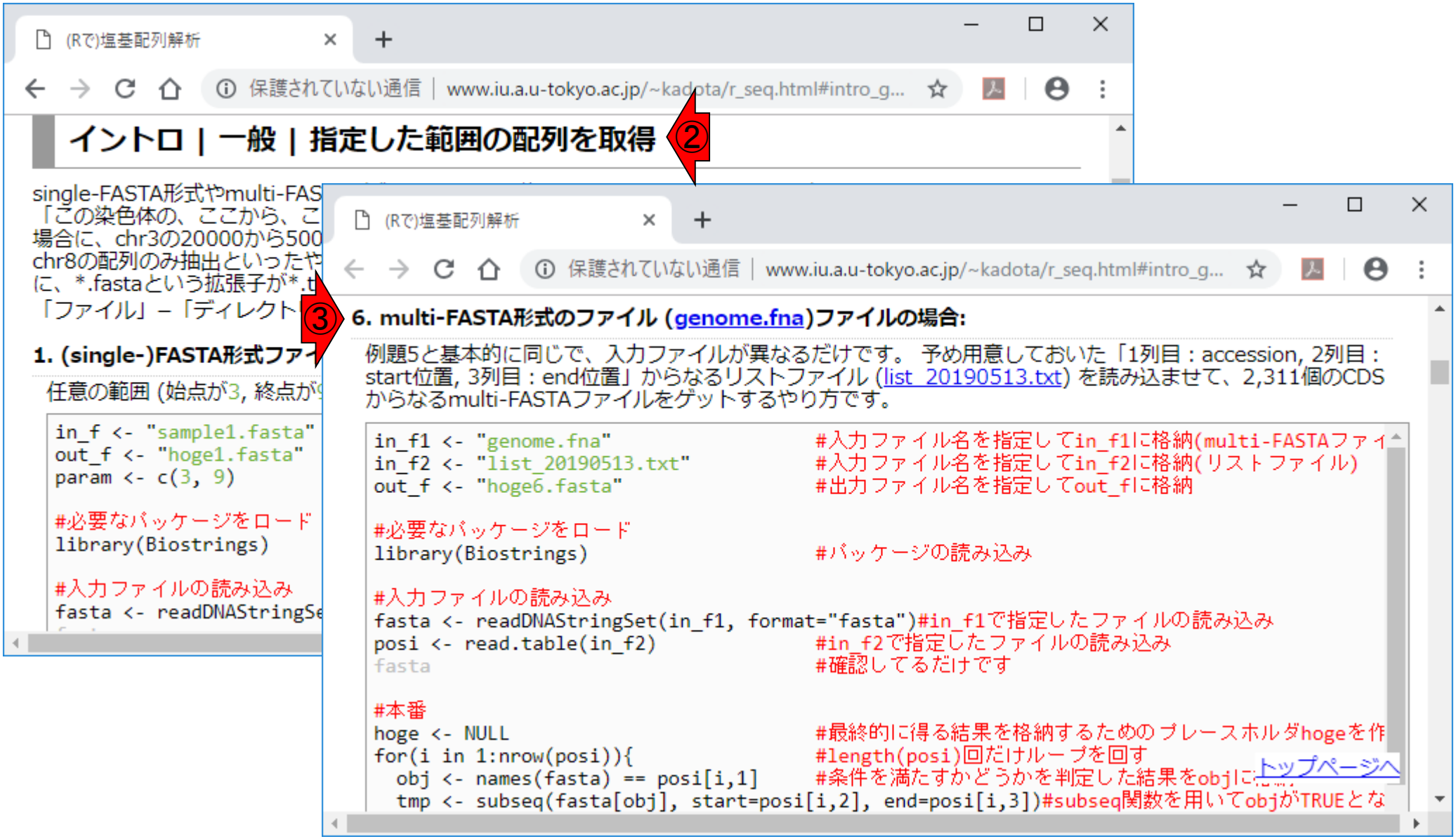
②「指定した範囲の配列を取得」の例題6をやります。

基本テクで実行



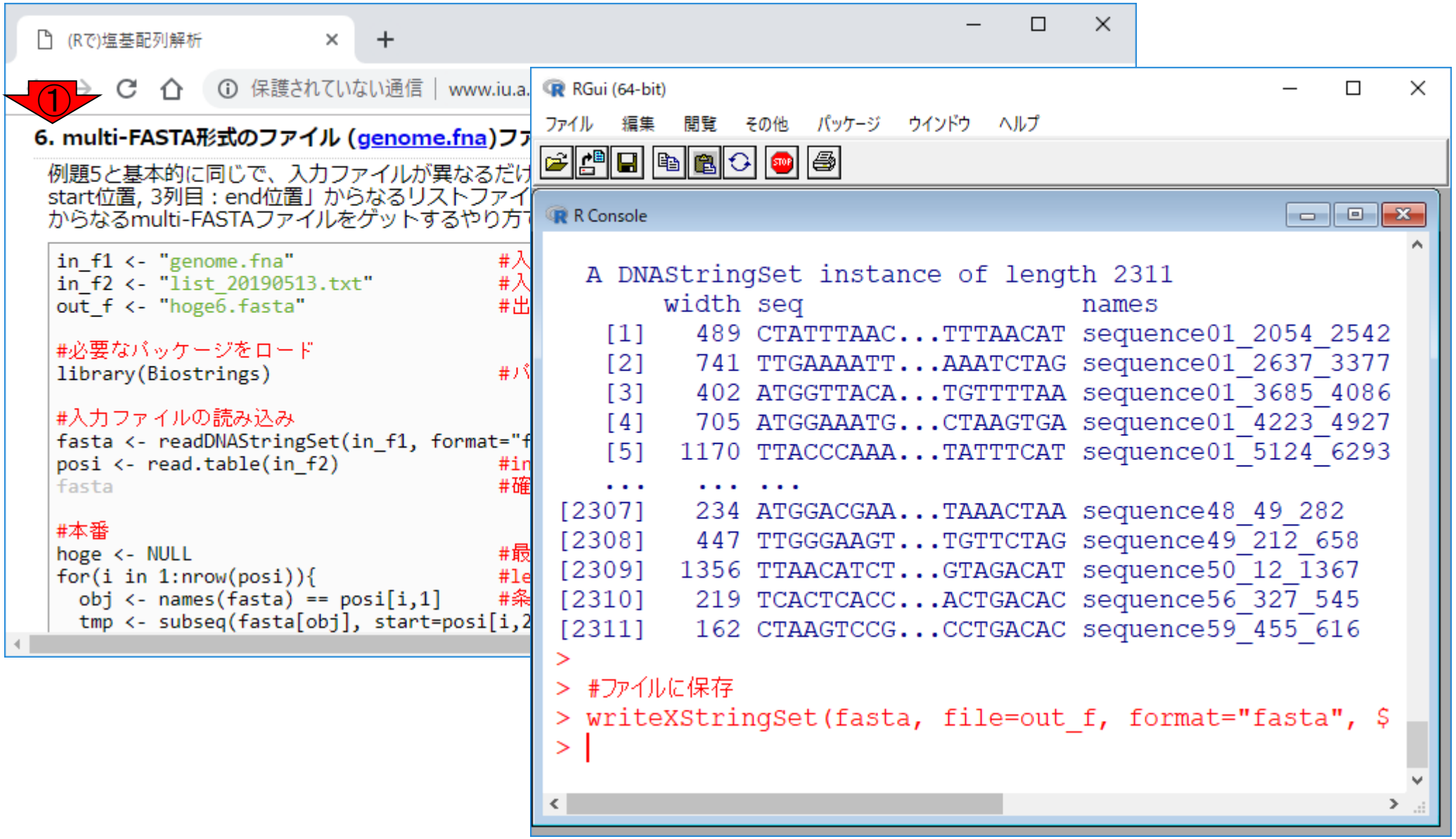
②「指定した範囲の配列を取得」の、③例題6です。

基本テクで実行



①例題6のコピペ実行結果。

基本テクで実行



基本テクで実行

①例題6のコピペ実行結果。今抽出している部分配列は、CDS（タンパク質コード領域の配列；coding sequence）。したがって、メチオニンをコードするATG（やGTGやTTG）が多くみられるはずだが、②2番目と3番目のポジションを眺めた段階で、「何かミスっている」ことに気づく。

(Rで)塩基配列解析

①

6. multi-FASTA形式のファイル (genome.fna)ファ

例題5と基本的に同じで、入力ファイルが異なるだけ
start位置, 3列目：end位置」からなるリストファイ
からなるmulti-FASTAファイルをゲットするやり方

```
in_f1 <- "genome.fna"                    #入
in_f2 <- "list_20190513.txt"             #入
out_f <- "hoge6.fasta"                   #出

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

#入力ファイルの読み込み
fasta <- readDNAStringSet(in_f1, format="f
posi <- read.table(in_f2)                #in
fasta                                    #確

#本番
hoge <- NULL                             #最
for(i in 1:nrow(posi)){                  #le
  obj <- names(fasta) == posi[i,1]       #条
  tmp <- subseq(fasta[obj], start=posi[i,2
```

RGui (64-bit)

ファイル 編集

R Console

②

```
  A DNAStringSet instance of length 2311
      width seq                                names
   [1]   489 CTATTTAAC...TTTAACAT sequence01_2054_2542
   [2]   741 TTGAAAATT...AAATCTAG sequence01_2637_3377
   [3]   402 ATGGTTACA...TGTTTTAA sequence01_3685_4086
   [4]   705 ATGGAAATG...CTAAGTGA sequence01_4223_4927
   [5]  1170 TTACCCAAA...TATTTCAT sequence01_5124_6293
   ...   ... ...
[2307]   234 ATGGACGAA...TAAACTAA sequence48_49_282
[2308]   447 TTGGGAAGT...TGTTCTAG sequence49_212_658
[2309]  1356 TTAACATCT...GTAGACAT sequence50_12_1367
[2310]   219 TCACTCACC...ACTGACAC sequence56_327_545
[2311]   162 CTAAGTCCG...CCTGACAC sequence59_455_616
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", $
>
```

Sequence logos

①例題6のコピペ実行結果。今抽出している部分配列は、CDS(タンパク質コード領域の配列；coding sequence)。したがって、メチオニンをコードするATG(やGTGやTTG)が多くみられるはずだが、②2番目と3番目のポジションを眺めた段階で、「何かミスっている」ことに気づく。Sequence logosで確認してもよい。③こんな感じの結果が得られるはずです。

(Rで)塩基配列解析

保護されていない通信 | www.iu.a...

6. multi-FASTA形式のファイル (genome.fna)ファ...

例題5と基本的に同じで、入力ファイルが異なるだけ... start位置, 3列目：end位置」からなるリストファイ... からなるmulti-FASTAファイルをゲットするやり方...

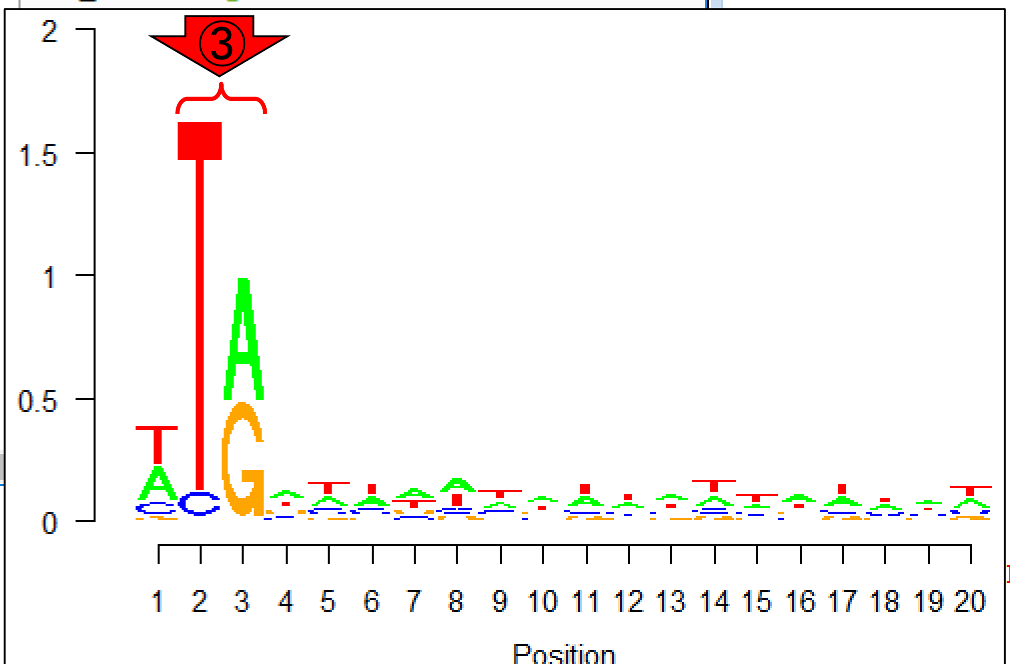
```
in_f1 <- "genome.fna"                #入
in_f2 <- "list_20190513.txt"         #入
out_f <- "hoge6.fasta"               #出
```

RGui (64-bit)

ファイル　編集　閲

R Console

```
A DNAStringSet instance of length 2311
      width seq                          names
          CTATTTAAC...TTTAACAT sequence01_2054_2542
          TTGAAAATT...AAATCTAG sequence01_2637_3377
          ATGGTTACA...TGTTTTAA sequence01_3685_4086
          ATGGAAATG...CTAAGTGA sequence01_4223_4927
          TTACCCAAA...TATTTCAT sequence01_5124_6293
          ...
          ATGGACGAA...TAAACTAA sequence48_49_282
          TTGGGAAGT...TGTTCTAG sequence49_212_658
          TTAACATCT...GTAGACAT sequence50_12_1367
          TCACTCACC...ACTGACAC sequence56_327_545
          CTAAGTCCG...CCTGACAC sequence59_455_616

ngSet(fasta, file=out_f, format="fasta", $
```

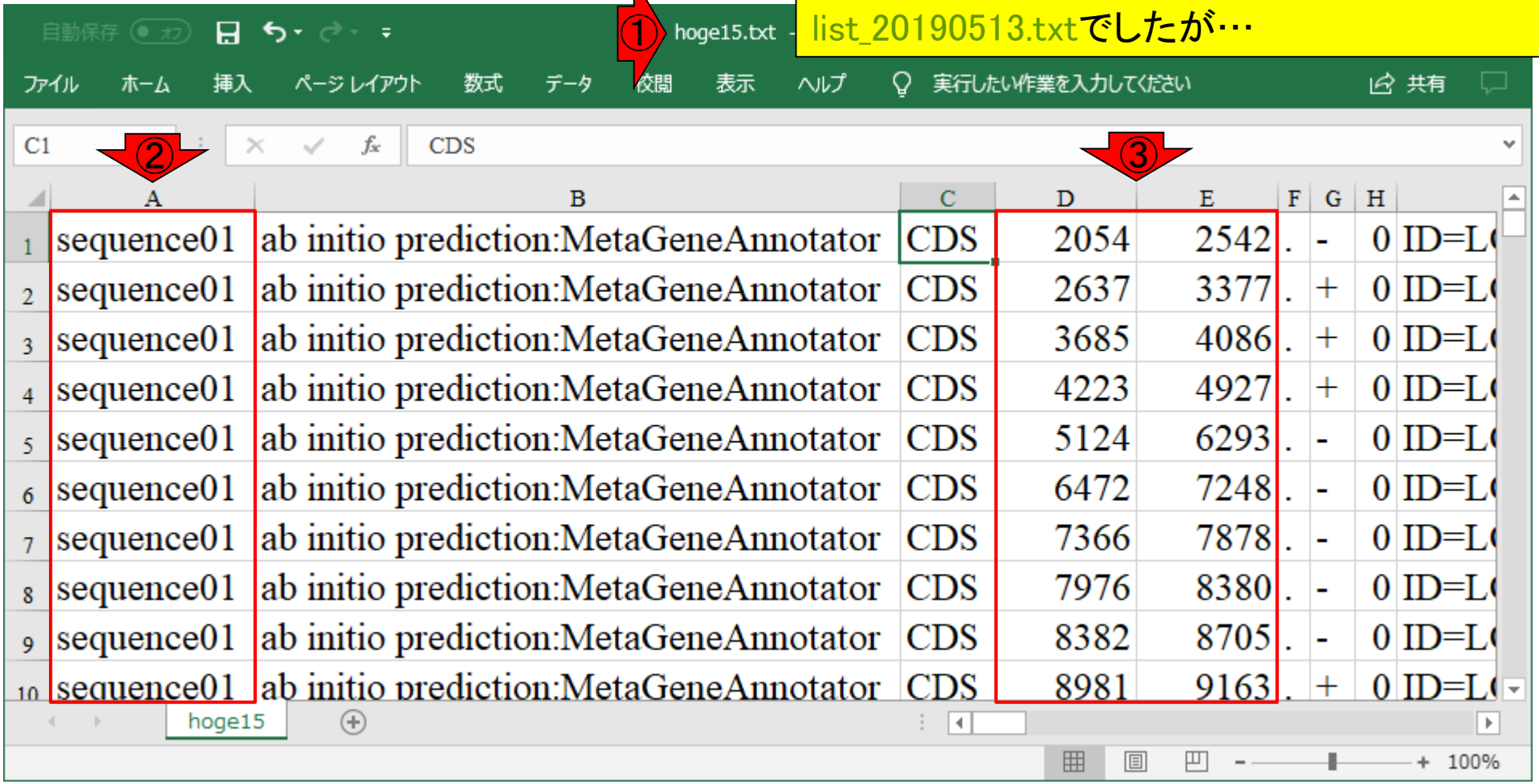


Contents

- 配列長でフィルタリング（ゲノムアセンブリ結果の後処理）
  - イントロ、本番
- ゲノムアノテーション
  - イントロ、参考文献
  - DFAST実行、課題1
- CDS（タンパク質コード領域の配列；coding sequenceの略）の取得
  - 比較ゲノム解析のイントロ、問題設定
  - 答え（cds.fna）を眺めておく、sequence logos
  - CDS配列取得の成功例、TxDbオブジェクト、CDS座標情報取得
  - CDS配列取得の失敗例、GFF3をExcelで概観
  - 基本テクを駆使して解決する基本戦略、CDSを含む行の抽出、補足説明
  - Excelで確認、リストファイルの作成、基本テクでCDS配列取得を実行
  - ストランド情報つきのリストファイルの作成、例題6のコード解説
  - ストランド情報反映戦略を練り、コードに修正を加えて実行（例題7）

ストランド情報追加

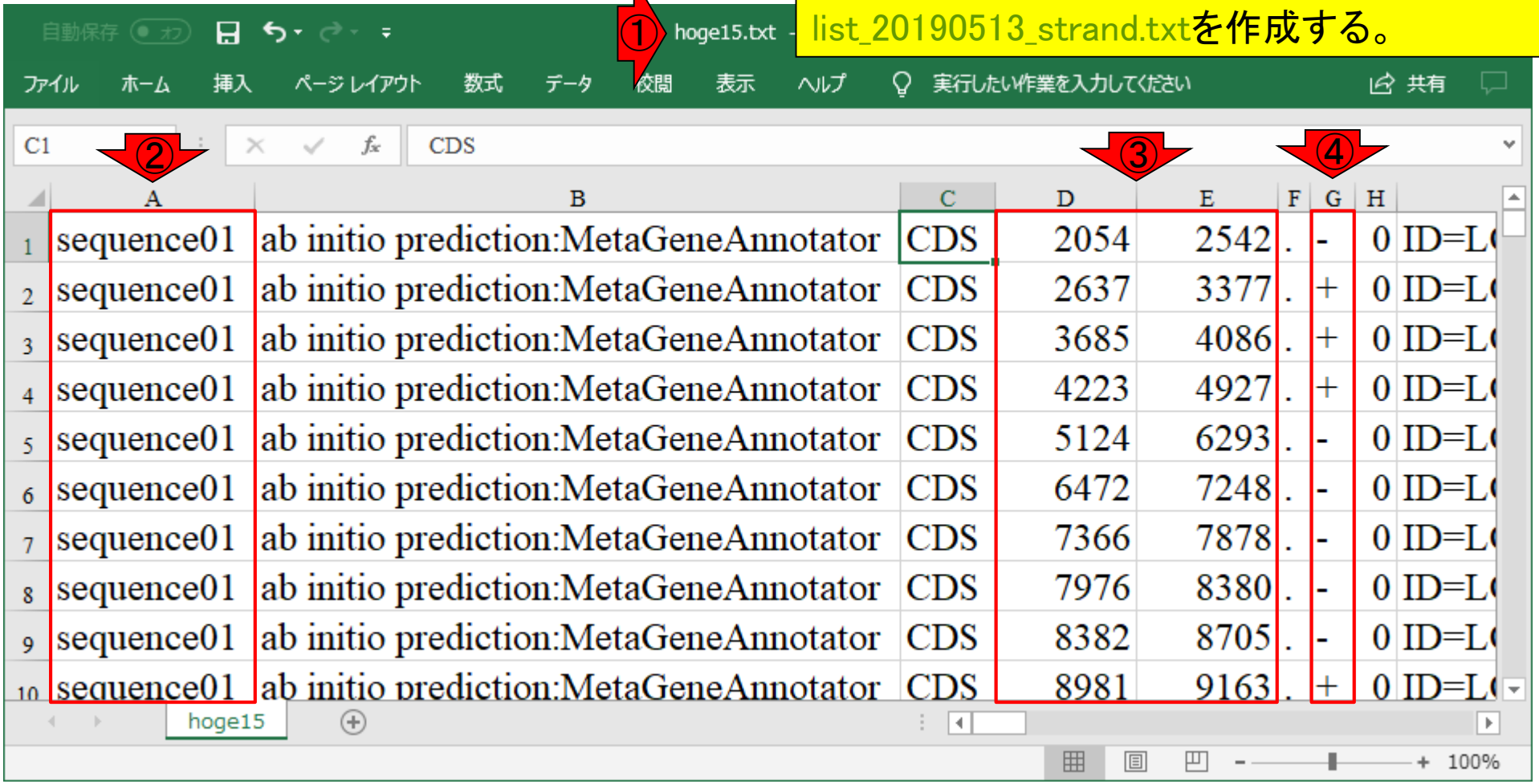
①hoge15.txt（annotation.gffからCDSという文字列を含む2,311行を抽出したファイル）から、②1列目と、③4,5列目の情報のみ抽出したのが list_20190513.txtでしたが…



	A	B	C	D	E	F	G	H	
1	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	2054	2542	.	-	0	ID=L
2	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	2637	3377	.	+	0	ID=L
3	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	3685	4086	.	+	0	ID=L
4	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	4223	4927	.	+	0	ID=L
5	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	5124	6293	.	-	0	ID=L
6	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	6472	7248	.	-	0	ID=L
7	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	7366	7878	.	-	0	ID=L
8	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	7976	8380	.	-	0	ID=L
9	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	8382	8705	.	-	0	ID=L
10	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	8981	9163	.	+	0	ID=L

ストランド情報追加

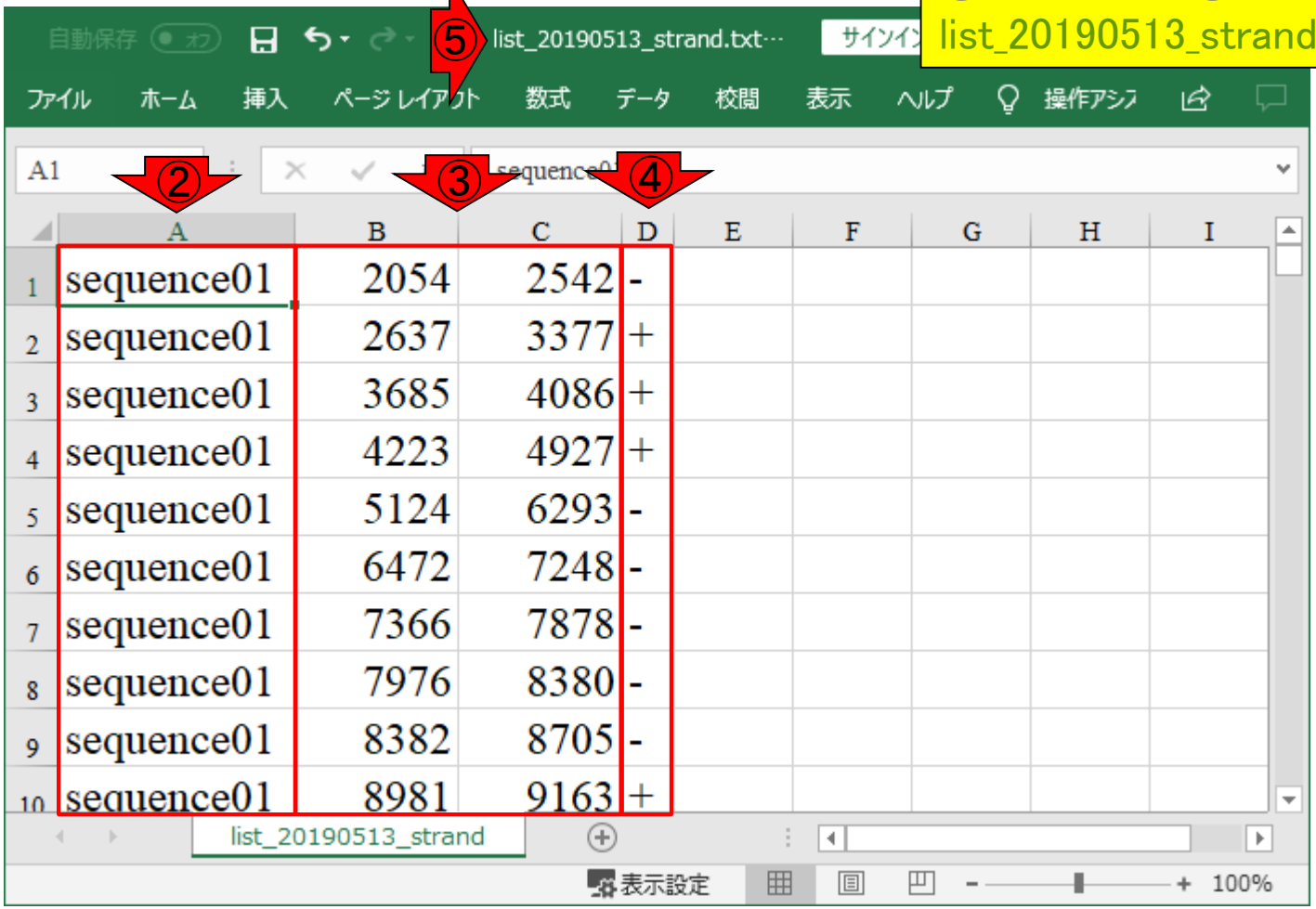
①hoge15.txt（annotation.gffからCDSという文字列を含む2,311行を抽出したファイル）から、②1列目と、③4,5列目と、④7列目の情報も加えたlist_20190513_strand.txtを作成する。



	A	B	C	D	E	F	G	H	
1	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	2054	2542	.	-	0	ID=L
2	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	2637	3377	.	+	0	ID=L
3	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	3685	4086	.	+	0	ID=L
4	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	4223	4927	.	+	0	ID=L
5	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	5124	6293	.	-	0	ID=L
6	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	6472	7248	.	-	0	ID=L
7	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	7366	7878	.	-	0	ID=L
8	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	7976	8380	.	-	0	ID=L
9	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	8382	8705	.	-	0	ID=L
10	sequence01	ab initio prediction:MetaGeneAnnotator	CDS	8981	9163	.	+	0	ID=L

ストランド情報追加

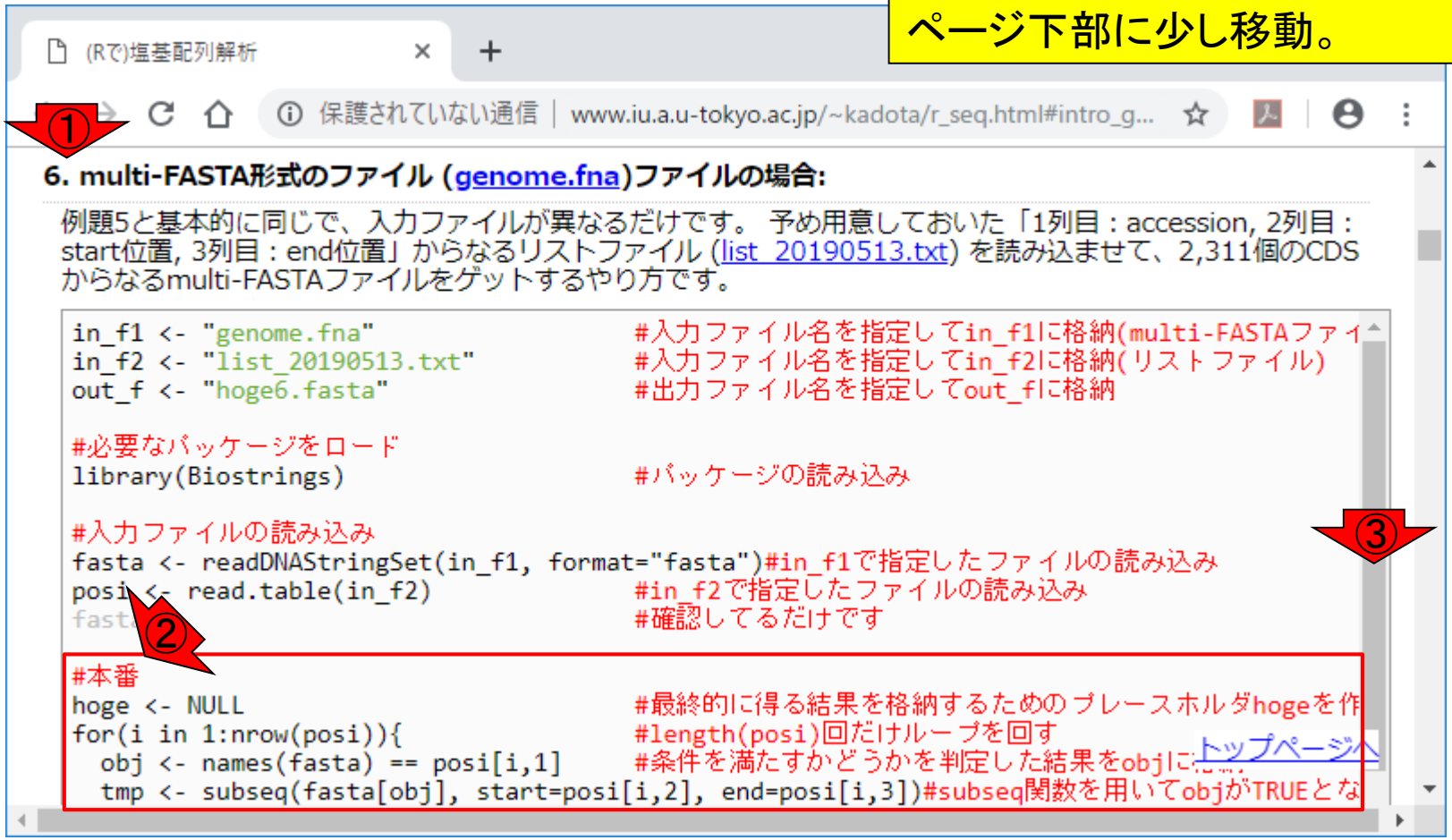
①hoge15.txt（annotation.gffからCDSという文字列を含む2,311行を抽出したファイル）から、②1列目と、③4,5列目と、④7列目の情報も加えた⑤list_20190513_strand.txtを作成した。



	A	B	C	D
1	sequence01	2054	2542	-
2	sequence01	2637	3377	+
3	sequence01	3685	4086	+
4	sequence01	4223	4927	+
5	sequence01	5124	6293	-
6	sequence01	6472	7248	-
7	sequence01	7366	7878	-
8	sequence01	7976	8380	-
9	sequence01	8382	8705	-
10	sequence01	8981	9163	+

例題6を眺める

「指定した範囲の配列を取得」の①例題6をもう一度眺める。CDSの部分配列を抽出しているのは②のあたりなので、これがページ上部に来るように、③ページ下部に少し移動。



例題6を眺める

「指定した範囲の配列を取得」の①例題6をもう一度眺める。CDSの部分配列を抽出しているのは②のあたりなので、これがページ上部に来るように、③ページ下部に少し移動。こんな感じ。

コードの解説

①例題6の、②リストファイル（list_20190513.txt）は、③のところで読み込まれ、posiというオブジェクト名で格納されている。従って、①の1列目の情報はposi[,1]、2列目の情報はposi[,2]、そして3列目の情報はposi[,3]で表現される。

(Rで)塩基配列解析

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

①

6. multi-FASTA形式のファイル (genome.fna)ファイルの場合:

例題5と基本的に同じで、入力ファイルが異なるだけです。 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_20190513.txt) を読み込ませて、2,311個のCDSからなるmulti-FASTAファイルをゲットするやり方です。

```
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hog
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
  tmp <- subseq(fasta[obj], start=posi[i,2], end=posi[i,3])#subseq関数を用いてobjがTRU
  hoge <- append(hoge, tmp)              #tmpの情報をhogeに追加で格納
}
fasta <- hoge                            #hogeの中身をfastaに格納
fasta                                    #確認してるだけです

#後処理(description部分の作成)
description <- paste(posi[,1], posi[,2], posi[,3], sep="_")#行列posiの各列を"_"で結合し
names(fasta) <- description              #description行に相当する記述を追加している
fasta                                    #確認してるだけです

#ファイルに保存                                                    トップページへ
writeXStringSet(fasta, file=out_f, format="fasta", width=50)#fastaの中身を指定したファイル名
```

③

②

sequence01	2054	2542
sequence01	2637	3377
sequence01	3685	4086
sequence01	4223	4927
sequence01	5124	6293
sequence01	6472	7248
sequence01	7366	7878
sequence01	7976	8380
sequence01	8382	8705
sequence01	8981	9163

コードの解説

①nrow(posi)は、②行列posiの行数に相当するので、
③このforループで、リストファイルの行数分だけ
ループを回していることになる。

(Rで)塩基配列解析

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

6. multi-FASTA形式のファイル (genome.fna)ファイルの場合:

例題5と基本的に同じで、入力ファイルが異なるだけです。 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_20190513.txt) を読み込ませて、2,311個のCDSからなるmulti-FASTAファイルをゲットするやり方です。

```
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
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
  tmp <- subseq(fasta[obj], start=posi[i,2], end=posi[i,3])#subseq関数を用いてobjがTRUE
  hoge <- append(hoge, tmp)              #tmpの情報をhogeに追加で格納
}
fasta <- hoge                            #hogeの中身をfastaに格納
fasta                                    #確認してるだけです

#後処理(description部分の作成)
description <- paste(posi[,1], posi[,2], posi[,3], sep="_")#行列posiの各列を"_"で結合し
names(fasta) <- description              #description行に相当する記述を追加している
fasta                                    #確認してるだけです

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

トップページへ

sequence01	2054	2542
sequence01	2637	3377
sequence01	3685	4086
sequence01	4223	4927
sequence01	5124	6293
sequence01	6472	7248
sequence01	7366	7878
sequence01	7976	8380
sequence01	8382	8705
sequence01	8981	9163

コードの解説

①nrow(posi)は、②行列posiの行数に相当するので、③このforループで、リストファイルの行数分だけループを回していることになる。④ここでiという添え字を使っているので、i行目という表現方法になっている。

(Rで)塩基配列解析

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

6. multi-FASTA形式のファイル (genome.fna)ファイルの場合:

例題5と基本的に同じで、入力ファイルが異なるだけです。 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_20190513.txt) を読み込ませて、2,311個のCDSからなるmulti-FASTAファイルをゲットするやり方です。

```
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

#ホ
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
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
  tmp <- subseq(fasta[obj], start=posi[i,2], end=posi[i,3])#subseq関数を用いてobjがTRUE
  hoge <- append(hoge, tmp)              #tmpの情報をhogeに追加で格納
}
fasta <- hoge                            #hogeの中身をfastaに格納
fasta                                    #確認してるだけです

#後処理(description部分の作成)
description <- paste(posi[,1], posi[,2], posi[,3], sep="_")#行列posiの各列を"_"で結合し
names(fasta) <- description              #description行に相当する記述を追加している
fasta                                    #確認してるだけです

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

トップページへ

sequence01	2054	2542
sequence01	2637	3377
sequence01	3685	4086
sequence01	4223	4927
sequence01	5124	6293
sequence01	6472	7248
sequence01	7366	7878
sequence01	7976	8380
sequence01	8382	8705
sequence01	8981	9163

コードの解説

①では、i行1列の要素（例えばi=3として、3行1列なら②が該当する）に相当する配列名posi[i,1]を抽出し、「リファレンス配列の配列名names(fasta)と一致する、配列名の要素の位置のみがTRUEとなる論理値ベクトルobjを作成している。この作業は、リファレンス配列genome.fnaの配列数が61個あるので、どの配列上の部分配列を取得したいかを特定するために重要です。

(Rで)塩基配列解析

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

6. multi-FASTA形式のファイル (genome.fna)ファイルの場合:

例題5と基本的に同じで、入力ファイルが異なるだけです。 予め用意し
start位置, 3列目：end位置」からなるリストファイル (list_2019051
からなるmulti-FASTAファイルをゲットするやり方です。

```
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
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
  tmp <- subseq(fasta[obj], start=posi[i,2], end=posi[i,3])#subseq関数を用いてobjがTRUE
  hoge <- append(hoge, tmp)              #tmpの情報をhogeに追加で格納
}
fasta <- hoge                            #hogeの中身をfastaに格納
fasta                                    #確認してるだけです

#後処理(description部分の作成)
description <- paste(posi[,1], posi[,2], posi[,3], sep="_")#行列posiの各列を"_"で結合し
names(fasta) <- description              #description行に相当する記述を追加している
fasta                                    #確認してるだけです

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

①

②

sequence01	2054	2542
sequence01	2637	3377
sequence01	3685	4086
sequence01	4223	4927
sequence01	5124	6293
sequence01	6472	7248
sequence01	7366	7878
sequence01	7976	8380
sequence01	8382	8705
sequence01	8981	9163

トップページへ

コードの解説

①では、fasta[obj]として目的の塩基配列（②の場合だとsequence01の配列）に限定して…

(Rで)塩基配列解析

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

6. multi-FASTA形式のファイル (genome.fna)ファイルの場合:

例題5と基本的に同じで、入力ファイルが異なるだけです。 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_20190513.txt) を読み込ませて、2,311個のCDSからなるmulti-FASTAファイルをゲットするやり方です。

```
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
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
  tmp <- subseq(fasta[obj], start=posi[i,2], end=posi[i,3])#subseq関数を用いてobjがTRUE
  hoge <- append(hoge, tmp)              #tmpの情報をhogeに追加して格納
}
fasta <- hoge                            #hogeの中身をfastaに格納
fasta                                    #確認してるだけです

#後処理(description部分の作成)
description <- paste(posi[,1], posi[,2], posi[,3], sep="_")#行列posiの各列を"_"で結合し
names(fasta) <- description              #description行に相当する記述を追加している
fasta                                    #確認してるだけです

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

トップページへ

sequence01	2054	2542
sequence01	2637	3377
sequence01	3685	4086
sequence01	4223	4927
sequence01	5124	6293
sequence01	6472	7248
sequence01	7366	7878
sequence01	7976	8380
sequence01	8382	8705
sequence01	8981	9163

コードの解説

①では、fasta[obj]として目的の塩基配列（②の場合だとsequence01の配列）に限定して、③start位置の要素に相当するi行2列の要素③posi[i,2]を指定している。i = 3の場合は、④の要素に相当する。こんな感じで、⑤subseq関数を用いて部分配列を抽出している。

(Rで)塩基配列解析

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

6. multi-FASTA形式のファイル (genome.fna)ファイルの場合:

例題5と基本的に同じで、入力ファイルが異なるだけです。 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_20190513.txt) を読み込ませて、2,311個のCDSからなるmulti-FASTAファイルをゲットするやり方です。

```
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
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
  tmp <- subseq(fasta[obj], start=posi[i,2], end=posi[i,3])#subseq関数を用いてobjがTRUE
  hoge <- append(hoge, tmp)              #tmpの情報をhogeに追加して格納
}
fasta <- hoge                            #hogeの中身をfastaに格納
fasta                                    #確認してるだけです

#後処理(description部分の作成)
description <- paste(posi[,1], posi[,2], posi[,3], sep="_")#行列posiの各列を"_"で結合し
names(fasta) <- description              #description行に相当する記述を追加している
fasta                                    #確認してるだけです

#ファイルに保存                                              トップページへ
writeXStringSet(fasta, file=out_f, format="fasta", width=50)#fastaの中身を指定したファイル名
```

sequence01	2054	2542
sequence01	3057	3377
sequence01	3685	4086
sequence01	4223	4927
sequence01	5124	6293
sequence01	6472	7248
sequence01	7366	7878
sequence01	7976	8380
sequence01	8382	8705
sequence01	8981	9163

コードの解説

①でhogeという何もない（NULL）オブジェクトを作成し、②append関数のところで、③利用している。④部分配列情報に相当するtmpの情報を、hogeにどんどん追加で格納しているだけです。なので、⑤がhogeなのです。例えば、i=1のときは、③のhogeには何も入っていない状態で、④tmpの情報を入れた結果が⑤hogeに格納される。i=2のときは、③のhogeにはi=1のときに抽出した部分配列のtmpが含まれていて、④でi=2のときの部分配列tmpを追加で入れた結果が⑤のhogeに格納される。

(Rで)塩基配列解析

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

6. multi-FASTA形式のファイル (genome.fna)ファイルの場合:

例題5と基本的に同じで、入力ファイルが異なるだけです。 予め用意し
start位置, 3列目：end位置」からなるリストファイル (list_2019051
からなるmulti-FASTAファイルをゲットするやり方です。

```
fasta <- readDNAStringSet(in_f1, format="fasta")#in_f1で指
posi <- read.table(in_f2)                  #in_f2で指定したフ
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
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
  tmp <- subseq(fasta[obj], start=posi[i,2], end=posi[i,3])#subseq関数を用いてobjがTRUE
  hoge <- append(hoge, tmp)                #tmpの情報をhogeに追加で格納
}
fasta <- hoge                              #hogeの中身をfastaに格納
fasta                                      #確認してるだけです

#後処理(description部分の作成)
description <- paste(posi[,1], posi[,2], posi[,3], sep="_")#行列posiの各列を"_"で結合し
names(fasta) <- description                #description行に相当する記述を追加している
fasta                                      #確認してるだけです

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

①②③④⑤

トップページへ

sequence01	2637	3377
sequence01	3685	4086
sequence01	4223	4927
sequence01	5124	6293
sequence01	6472	7248
sequence01	7366	7878
sequence01	7976	8380
sequence01	8382	8705
sequence01	8981	9163

Contents

- 配列長でフィルタリング（ゲノムアセンブリ結果の後処理）
  - イントロ、本番
- ゲノムアノテーション
  - イントロ、参考文献
  - DFAST実行、課題1
- CDS（タンパク質コード領域の配列；coding sequenceの略）の取得
  - 比較ゲノム解析のイントロ、問題設定
  - 答え（cds.fna）を眺めておく、sequence logos
  - CDS配列取得の成功例、TxDbオブジェクト、CDS座標情報取得
  - CDS配列取得の失敗例、GFF3をExcelで概観
  - 基本テクを駆使して解決する基本戦略、CDSを含む行の抽出、補足説明
  - Excelで確認、リストファイルの作成、基本テクでCDS配列取得を実行
  - ストランド情報つきのリストファイルの作成、例題6のコード解説
  - ストランド情報反映戦略を練り、コードに修正を加えて実行（例題7）

ストランド反映戦略

ストランドが"+"のときは、①で得られたtmpがそのまま②の部分で使われてよい。しかしストランドが"-"のときは、抽出した部分配列tmpを逆相補鎖（reverse complement）にしてから②に供するようにせねばならないと判断する。

(Rで)塩基配列解析

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

6. multi-FASTA形式のファイル (genome.fna)ファイルの場合:

例題5と基本的に同じで、入力ファイルが異なるだけです。 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_20190513.txt) を読み込ませて、2,311個のCDSからなるmulti-FASTAファイルをゲットするやり方です。

```
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
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
  tmp <- subseq(fasta[obj], start=posi[i,2], end=posi[i,3])#subseq関数を用いてobjがTRUE
  hoge <- append(hoge, tmp)              #tmpの情報をhogeに追加して格納
}
fasta <- hoge                            #hogeの中身をfastaに格納
fasta                                    #確認してるだけです

#後処理(description部分の作成)
description <- paste(posi[,1], posi[,2], posi[,3], sep="_")#行列posiの各列を"_"で結合し
names(fasta) <- description              #description行に相当する記述を追加している
fasta                                    #確認してるだけです

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

①

②

①

トップページへ

sequence01	2054	2542
sequence01	2637	3377
sequence01	3685	4086
sequence01	4223	4927
sequence01	5124	6293
sequence01	6472	7248
sequence01	7366	7878
sequence01	7976	8380
sequence01	8382	8705
sequence01	8981	9163

例題7

①例題7が、「ストランドが"-"のときに、抽出した部分配列tmpを逆相補鎖（reverse complement）にする」変更を追加したコード。②4列目にストランド情報を含むリストファイル。

(Rで)塩基配列解析

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

①

7. multi-FASTA形式のファイル (genome.fna)ファイルの場合:

例題6と基本的に同じですが、4列目にストランド情報を含むリストファイル (list_20190513_strand.txt) を読み込ませて、ストランドを適切に反映させた2,311個のCDSからなるmulti-FASTAファイルをゲットするやり方です。

```
in_f1 <- "genome.fna"                          #入力ファイル名を指定してin_f1に格納(multi-
in_f2 <- "list_20190513_strand.txt"            ②#入力ファイル名を指定してin_f2に格納(リスト
out_f <- "hoge7.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
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
  tmp <- subseq(fasta[obj], start=posi[i,2], end=posi[i,3])#subseq関数を用いてob
  if(posi[i,4] == "-"){tmp <- reverseComplement(tmp)}#ストランドが"-"の場合は逆相
  hoge <- append(hoge, tmp)                    #tmpの情報をhogeに追加で格納
}                                                                    トップページへ
fasta <- hoge                                  #hogeの中身をfastaに格納
fasta                                          #確認してるだけです
```

②

sequence01	2054	2542	-
sequence01	2637	3377	+
sequence01	3685	4086	+
sequence01	4223	4927	+
sequence01	5124	6293	-
sequence01	6472	7248	-
sequence01	7366	7878	-
sequence01	7976	8380	-
sequence01	8382	8705	-
sequence01	8981	9163	+

例題7

①例題7が、「ストランドが"-"のときに、抽出した部分配列tmpを逆相補鎖（reverse complement）にする」変更を追加したコード。②4列目にストランド情報を含むリストファイル。③posi[i,4]が-なら、tmpの中身を逆相補鎖に変更するコードを追加しただけです。④がposi[i,4]に相当する部分です。

(Rで)塩基配列解析

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

①

7. multi-FASTA形式のファイル (genome.fna)ファイルの場合:

例題6と基本的に同じですが、4列目にストランド情報を含むリストファイル (list_20190513_strand.txt) を読み込ませて、ストランドを適切に反映させた2,311個のCDSからなるmulti-FASTAファイルをゲットするやり方です。

```
in_f1 <- "genome.fna"                    #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_20190513_strand.txt"      #入力ファイル名を指定してin_f2に格納(リスト
out_f <- "hoge7.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
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
  tmp <- subseq(fasta[obj], start=posi[i,2], end=posi[i,3])#subseq関数を用いてob
  if(posi[i,4] == "-"){tmp <- reverseComplement(tmp)}#ストランドが"-"の場合は逆相
  hoge <- append(hoge, tmp)              #tmpの情報をhoge に追加で格納
}
fasta <- hoge                            #hogeの中身をfastaに格納
fasta                                    #確認してるだけです
```

②

③

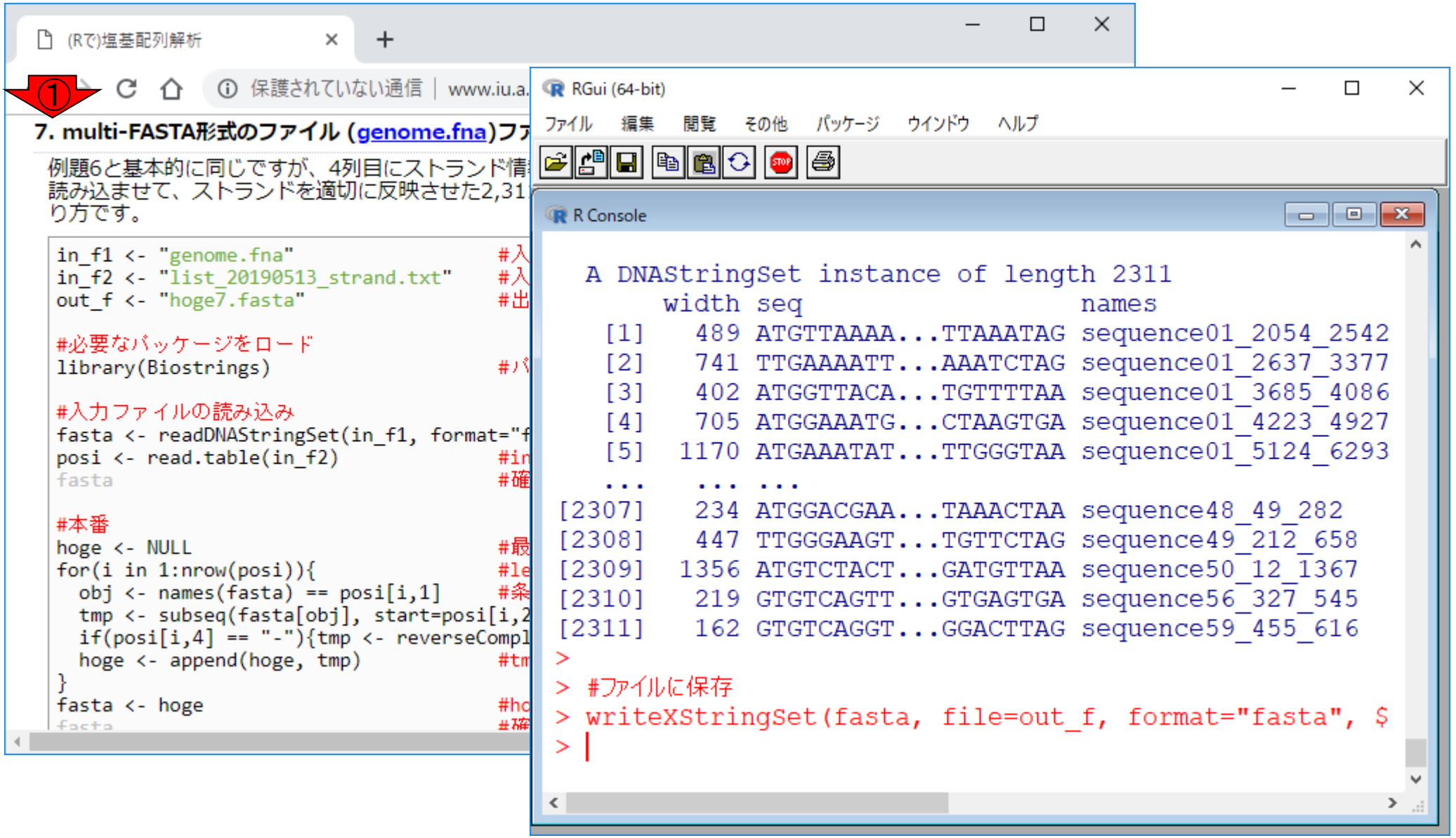
④

sequence01	2054	2542	-
sequence01	2637	3377	+
sequence01	3685	4086	+
sequence01	4223	4927	+
sequence01	5124	6293	-
sequence01	6472	7248	-
sequence01	7366	7878	-
sequence01	7976	8380	-
sequence01	8382	8705	-
sequence01	8981	9163	+

トップページへ

①例題7のコピペ実行結果。

例題7



例題7

①例題7のコピペ実行結果。②をざっとみただけですが、うまくいってることがわかります。こんな感じで必要に応じて③if文をつけるなどして必要最小限の変更を施して目的を達成してゆくのです

(Rで)塩基配列解析

① 保護されていない通信 | www.iu.a

7. multi-FASTA形式のファイル (genome.fna)ファ

例題6と基本的に同じですが、4列目にストランド情
読み込ませて、ストランドを適切に反映させた2,31
り方です。

```
in_f1 <- "genome.fna"                          #入
in_f2 <- "list_20190513_strand.txt"            #入
out_f <- "hoge7.fasta"                         #出

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

#入力ファイルの読み込み
fasta <- readDNAStringSet(in_f1, format="f
posi <- read.table(in_f2)                      #in
fasta                                          #確

#本番
hoge <- NULL                                   #最
for(i in 1:nrow(posi)){                        #le
  obj <- names(fasta) == posi[i,1]             #条
  tmp <- subseq(fasta[obj], start=posi[i,2
③ if(posi[i,4] == "-"){tmp <- reverseCompl
  hoge <- append(hoge, tmp)                    #tm
}
fasta <- hoge                                  #ho
fasta                                          #確
```

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウインドウ ヘルプ

R Console

```
  A DNAStringSet instance of length 2311
       width seq                                names
   [1]   489 ATGTTAAAA...TTAAATAG sequence01_2054_2542
   [2]   741 TTGAAAATT...AAATCTAG sequence01_2637_3377
   [3]   402 ATGGTTACA...TGTTTTAA sequence01_3685_4086
   [4]   705 ATGGAAATG...CTAAGTGA sequence01_4223_4927
   [5]  1170 ATGAAATAT...TTGGGTAA sequence01_5124_6293
   ...   ... ...
[2307]   234 ATGGACGAA...TAAACTAA sequence48_49_282
[2308]   447 TTGGGAAGT...TGTTCTAG sequence49_212_658
[2309]  1356 ATGTCTACT...GATGTTAA sequence50_12_1367
[2310]   219 GTGTCAGTT...GTGAGTGA sequence56_327_545
[2311]   162 GTGTCAGGT...GGACTTAG sequence59_455_616
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", $
>
```

② ATGTTAAAA...TTAAATAG