

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。



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

ようこそ!!
アグリバイオ
教育研究センター

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

東京大学
THE UNIVERSITY OF TOKYO

教育プログラム

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

▼ スケジュール

プログラム概要

本プログラムで開講する講義科目は()に分けられます。カテゴリと

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

お知らせ

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

各講義のページ

(科目名をクリックすると各講義のページに移動します)

先端トピックス セミナー・討論形式 研究指導	農学生命情報科学特別演習			
	農学生命情報科学特論 I	農学生命情報科学特論 II	農学生命情報科学特論 III	農学生命情報科学特論 IV
方法論 講義・実習を一体化	生物配列統計学 システム生物学概論 知識情報処理論			
	オーム情報解析 機能ゲノム学 分子モデリングと分子シミュレーション			
基礎 講義・実習を一体化	フィールドインフォマティクス			
	ゲノム情報解析基礎	構造バイオインフォマティクス基礎	生物配列解析基礎	バイオスタティスティクス基礎論

③

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trand情報付きのリストファイルの作成、例題6のコード解説
 - スtrand情報反映戦略を練り、コードに修正を加えて実行(例題7)

本科目のページ

講義日程 (2019年度)

- 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
- 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
- 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
- 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

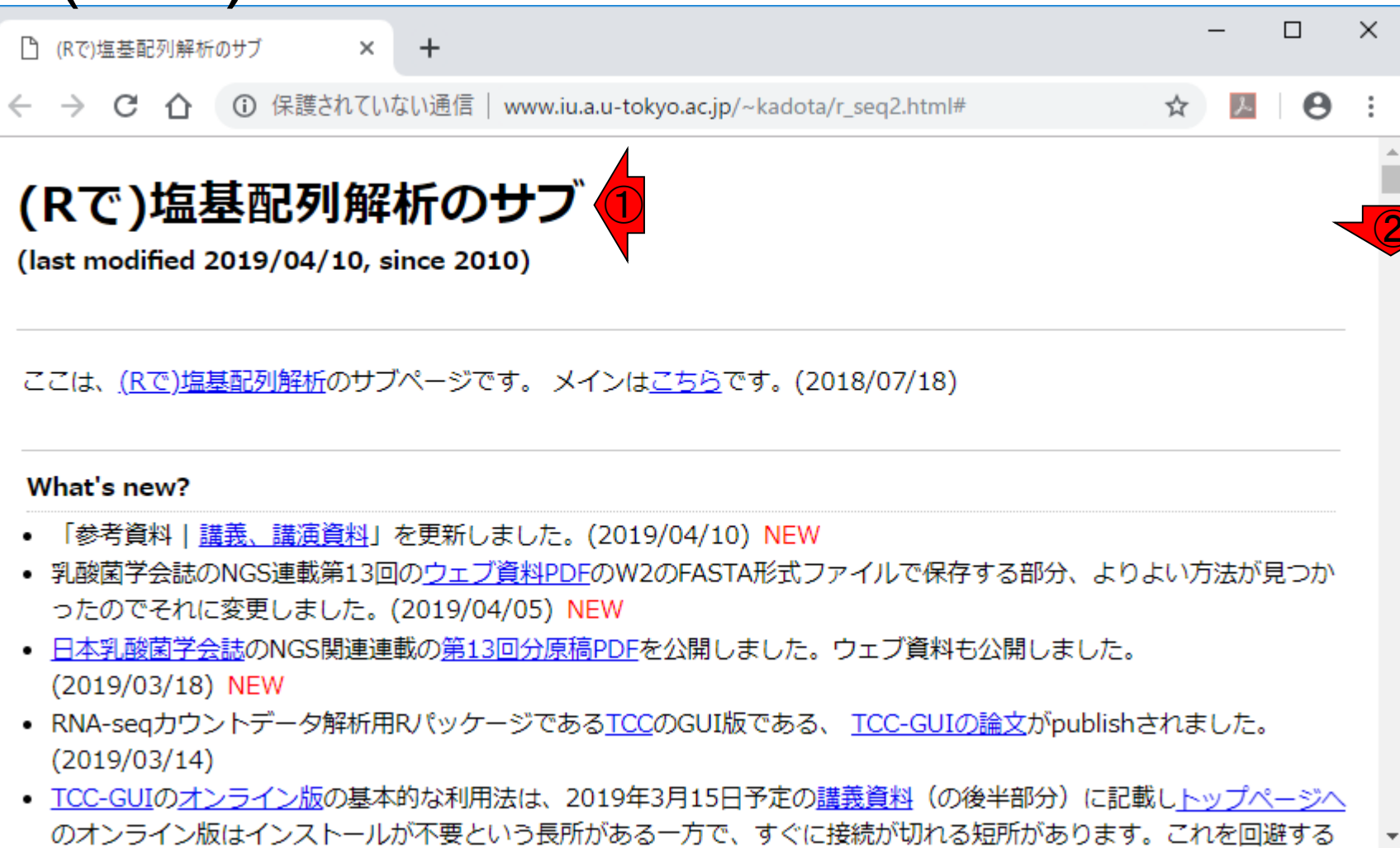
DFAST実行結果: genome.fna

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。



The screenshot shows a web browser window with the address bar displaying 'www.iu.a.u-tokyo.ac.jp/~kadota/r_seq2.html#'. The page title is '(Rで)塩基配列解析のサブ' and it indicates it was last modified on 2019/04/10. A red arrow labeled '1' points to the title. Another red arrow labeled '2' points to the vertical scrollbar on the right side of the page. The main content area contains a paragraph and a 'What's new?' section with several bullet points.

(Rで)塩基配列解析のサブ
(last modified 2019/04/10, since 2010)

ここは、[\(Rで\)塩基配列解析](#)のサブページです。メインは[こちら](#)です。(2018/07/18)

What's new?

- 「参考資料 | [講義、講演資料](#)」を更新しました。(2019/04/10) **NEW**
- 乳酸菌学会誌のNGS連載第13回の[ウェブ資料PDF](#)のW2のFASTA形式ファイルで保存する部分、よりよい方法が見つかったのでそれに変更しました。(2019/04/05) **NEW**
- [日本乳酸菌学会誌](#)のNGS関連連載の[第13回分原稿PDF](#)を公開しました。ウェブ資料も公開しました。(2019/03/18) **NEW**
- RNA-seqカウントデータ解析用Rパッケージである[ICC](#)のGUI版である、[ICC-GUIの論文](#)がpublishされました。(2019/03/14)
- [ICC-GUIのオンライン版](#)の基本的な利用法は、2019年3月15日予定の[講義資料](#)（の後半部分）に記載し[トップページ](#)へのオンライン版はインストールが不要という長所がある一方で、すぐに接続が切れる短所があります。これを回避する

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

(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#

- 書籍 | トランスクリプトーム解析 | [4.3.3 2群間比較](#) (last modified 2014/04/28)
- 書籍 | トランスクリプトーム解析 | [4.3.4 他の実験デザイン\(3群間\)](#) (last modified 2014/04/28)
- 書籍 | 日本乳酸菌学会誌 | [について](#) (last modified 2019/04/05) **NEW**
- 書籍 | 日本乳酸菌学会誌 | [第1回イントロダクション](#) (last modified 2018/09/03)
- 書籍 | 日本乳酸菌学会誌 | [第2回GUI環境からコマンドライン環境へ](#) (last modified 2018/09/03)
- 書籍 | 日本乳酸菌学会誌 | [第3回Linux環境構築からNGSデータ取得まで](#) (last modified 2017/07/02)
- 書籍 | 日本乳酸菌学会誌 | [第4回クオリティコントロールとプログラムのインストール](#) (last modified 2018/05/10)
- 書籍 | 日本乳酸菌学会誌 | [第5回アセンブル、マッピング、そしてQC](#) (last modified 2017/06/25)
- 書籍 | 日本乳酸菌学会誌 | [第6回ゲノムアセンブリ](#) (last modified 2017/06/21)
- 書籍 | 日本乳酸菌学会誌 | [第7回ロングリードアセンブリ](#) (last modified 2017/06/28)
- 書籍 | 日本乳酸菌学会誌 | [第8回アセンブリ後の解析](#) (last modified 2017/06/28)
- 書籍 | 日本乳酸菌学会誌 | [第9回ゲノムアノテーションとその可視化、DDBJへの登録](#) (last modified 2017/03/13)
- 書籍 | 日本乳酸菌学会誌 | [第10回DDBJへの塩基配列の登録\(後編\)](#) (last modified 2017/06/28)
- 書籍 | 日本乳酸菌学会誌 | [第11回統合データ解析環境Galaxy](#) (last modified 2017/11/13)
- 書籍 | 日本乳酸菌学会誌 | [第12回Galaxy: ヒストリーとワークフロー](#) (last modified 2018/07/04)
- 書籍 | 日本乳酸菌学会誌 | [第13回RNA-seq解析\(その1\)](#) (last modified 2019/04/05) **NEW**

[トップページへ](#)



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。

(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#

- 書籍 | トランスクリプトーム解析 | [4.3.3 2群間比較](#) (last modified 2014/04/28)
- 書籍 | トランスクリプトーム解析 | [4.3.4 他の実験デザイン\(3群間\)](#) (last modified 2014/04/28)
- 書籍 | [日本乳酸菌学会誌](#) | [について](#) (last modified 2019/04/05) NEW
- 書籍 | [日本乳酸菌学会誌](#) | [第1回イントロダクション](#) (last modified 2018/09/03) ①
- 書籍 | [日本乳酸菌学会誌](#) | [第2回GUI環境からコマンドライン環境へ](#) (last modified 2018/09/03)
- 書籍 | [日本乳酸菌学会誌](#) | [第3回Linux環境構築からNGSデータ取得まで](#) (last modified 2017/07/02)
- 書籍 | [日本乳酸菌学会誌](#) | [第4回クオリティコントロールとプログラムのインストール](#) (last modified 2018/05/10)
- 書籍 | [日本乳酸菌学会誌](#) | [第5回アセンブル、マッピング、そしてQC](#) (last modified 2017/06/25)
- 書籍 | [日本乳酸菌学会誌](#) | [第6回ゲノムアセンブリ](#) (last modified 2017/06/21) ②
- 書籍 | [日本乳酸菌学会誌](#) | [第7回ロングリードアセンブリ](#) (last modified 2017/06/28)
- 書籍 | [日本乳酸菌学会誌](#) | [第8回アセンブリ後の解析](#) (last modified 2017/06/28)
- 書籍 | [日本乳酸菌学会誌](#) | [第9回ゲノムアノテーションとその可視化、DDBJへの登録](#) (last modified 2017/03/13)
- 書籍 | [日本乳酸菌学会誌](#) | [第10回DDBJへの塩基配列の登録\(後編\)](#) (last modified 2017/06/28)
- 書籍 | [日本乳酸菌学会誌](#) | [第11回統合データ解析環境Galaxy](#) (last modified 2017/11/13)
- 書籍 | [日本乳酸菌学会誌](#) | [第12回Galaxy: ヒストリーとワークフロー](#) (last modified 2018/07/04) ③
- 書籍 | [日本乳酸菌学会誌](#) | [第13回RNA-seq解析\(その1\)](#) (last modified 2019/04/05) NEW

[トップページへ](#)

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

(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 | トランスクリプトーム解析 | [4.3.3 2群間比較](#) (last modified 2019/04/05)
- 書籍 | トランスクリプトーム解析 | [4.3.4 他の実験デザイン\(3群間\)](#)
- 書籍 | [日本乳酸菌学会誌](#) | [について](#) (last modified 2019/04/05)
- 書籍 | [日本乳酸菌学会誌](#) | [第1回イントロダクション](#) (last modified 2018/09/03)
- 書籍 | [日本乳酸菌学会誌](#) | [第2回GUI環境からコマンドライン環境へ](#) (last modified 2018/09/03)
- 書籍 | [日本乳酸菌学会誌](#) | [第3回Linux環境構築からNGSデータ取得まで](#) (last modified 2017/07/02)
- 書籍 | [日本乳酸菌学会誌](#) | [第4回クオリティコントロールとプログラムのインストール](#) (last modified 2018/05/10)
- 書籍 | [日本乳酸菌学会誌](#) | [第5回アセンブル、マッピング、そしてQC](#) (last modified 2017/06/25)
- 書籍 | [日本乳酸菌学会誌](#) | [第6回ゲノムアセンブリ](#) (last modified 2017/06/21)
- 書籍 | [日本乳酸菌学会誌](#) | [第7回ロングリードアセンブリ](#) (last modified 2017/06/28)
- 書籍 | [日本乳酸菌学会誌](#) | [第8回アセンブリ後の解析](#) (last modified 2017/06/28)
- 書籍 | [日本乳酸菌学会誌](#) | [第9回ゲノムアノテーションとその可視化、DDBJへの登録](#) (last modified 2017/03/13)
- 書籍 | [日本乳酸菌学会誌](#) | [第10回DDBJへの塩基配列の登録\(後編\)](#) (last modified 2017/06/28)
- 書籍 | [日本乳酸菌学会誌](#) | [第11回統合データ解析環境Galaxy](#) (last modified 2017/11/13)
- 書籍 | [日本乳酸菌学会誌](#) | [第12回Galaxy: ヒストリーとワークフロー](#) (last modified 2018/07/04)
- 書籍 | [日本乳酸菌学会誌](#) | [第13回RNA-seq解析\(その1\)](#) (last modified 2019/04/05) **NEW**

[トップページへ](#)

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。

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からです（2016年4月23日追加）。

- [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回終了時点以降のovaファイルからのインストール手順：
 - [Windows用](#)(2015.12.28版; 約2MB)
 - [Macintosh用](#)(2015.12.28版; 約2MB)

Linuxコマンド

[トップページへ](#)

- [bzip2](#) (bzip2圧縮、および解凍)

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。

(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

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です (2016年4月23日追加)。

- ① [原稿PDF](#)
- ② [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回終了時点以降のovaファイルからのインストール手順：
 - [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)

[トップページへ](#)

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)。

(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

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
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
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
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
まれていない)です(2016年4月23日追加)。

- [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回終了時点以降のovaファイルからのインストール手順:
 - [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)

[トップページへ](#)

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

(Rで塩基配列解析のサブ × +

← → ↻ 🏠 ⓘ 保護されていない通信 | www.jp.a.u-tokyo.ac.jp/~kadota/r_seq2.html#book_JSLAB_6 ☆ 🧑 | 👤 ⋮

DDBJ Pipeline (基礎処理部Platanusの実行) ①

- [Platanus : Kajitani et al., Genome Res., 2014](#)
- W20-4 : Platanus ver. 1.2.2を実行した結果ファイル([platanusResult.zip](#); 約2.2MB)
- W20-5 : Bio-Linuxで解凍

ホストOSの共有フォルダ(~/Desktop/share)に保存し、ゲストOSの共有フォルダ(~/Desktop/mac_share)から眺めて、zipファイルを解凍します。解凍後に見られるファイルのうち、[out_gapClosed.fa](#) (約2.3MB) がPlatanusの主なアセンブル結果ファイルになります。それ以外には、k-merの出現頻度分布([out_32merFrq.tsv](#))や、インサート長の頻度分布([out_lib1_insFreq.tsv](#))などのtsv形式ファイルも含まれます。

```
cd ~/Desktop/mac_share

pwd
ls -l pla*
unzip -q platanusResult.zip
ls -ld pla*
cd platanusResult
pwd
ls
```

- W20-6 : 配列数確認

[トップページへ](#)

```
pwd
```

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。

(Rで塩基配列解析のサブ

保護されていない通信 | www.iij.a.u-tokyo.ac.jp/~kadota/seq.html#book_334b_0

DDBJ Pipeline (基礎処理部Platanusの実行)

- [Platanus : Kajitani et al., Genome Res., 2014](#)
- W20-4 : Platanus ver. 1.2.2を実行した結果ファイル([platanusResult.zip](#); 約2.2MB)
- W20-5 : Bio-Linuxで解凍

ホストOSの共有フォルダ(~/Desktop/share)に保存し、ゲストOSの共有フォルダ(~/Desktop/mac_share)から眺めて、zipファイルを解凍します。解凍後に見られるファイルのうち、[out_gapClosed.fa](#) (約2.3MB) がPlatanusの主なアセンブル結果ファイルになります。それ以外には、k-merの出現頻度分布([_32merFrq.tsv](#))や、インサート長の頻度分布([out_lib1_insFreq.tsv](#))などのtsv形式ファイルも含まれます。

```
cd ~/Desktop/mac_share

pwd
ls -l pla*
unzip -q platanusResult.zip
ls -ld pla*
cd platanusResult
pwd
ls
```

- W20-6 : 配列数確認

[トップページへ](#)

```
pwd
ls -l *fa
```

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
あります。それ以外には、k-merの出現頻度分布 (out_k32merFrq.tsv)
(out_k32merFrq.tsv)などのtsv形式ファイルも含まれます。

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

③

第6回の原稿PDF

(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 ☆ 人 👤 ⋮

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回で詳述では、「第8回以降」で詳述する予定である。」と読み替えてまわっていません（2016年4月23日追加）。

- [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)

Japanese Journal of Lactic Acid Bacteria
Copyright © 2016, Japan Society for Lactic Acid Bacteria

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、③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おうとして、**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回ドキュメント原稿作成時点で、これらの内容は含まれていないからです(2016年

- ① [原稿PDF](#)
- ウェブ資料PDF
 - [Windows用\(2016.0\)](#)
 - [Macintosh用](#)
- (共有フォルダ設定情報を)
 - [Windows用\(2015.3\)](#)
 - [Macintosh用\(2015.3\)](#)

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 24MB」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

③ 配列長によるフィルタリング

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trand情報付きのリストファイルの作成、例題6のコード解説
 - スtrand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

(last modified 2019/04/15, since 2010)

このウェブページのR関連部分は、[インストーラ \(Macintosh2018.11.27版\)](#)に従ってインストールされています。初心者の方は[基本的な利用法](#)をご覧ください。
2018年7月に(Rで)塩基配列解析の更新を行いました。(2018/07/18)

What's new? (過去のお知らせはこちら)

- 「カウント情報取得 | シミュレーション」を追加しました。(2019/04/11) **NEW**
- 「カウント情報取得 | シミュレーション」を追加しました。(2019/04/11) **NEW**
- 削除予定としていた「インストール」を追加しました。
- 削除予定としていた「インストール」を追加しました。

①

②

- 前処理 | トリミング | [指定した末端塩基数だけ除去](#) (last modified 2017/11/08)
- 前処理 | [フィルタリング](#) | [について](#) (last modified 2018/08/06)
- 前処理 | [フィルタリング](#) | [PHREDスコアが低い塩基をNに置換](#) (last modified 2014/03/03)
- 前処理 | [フィルタリング](#) | [PHREDスコアが低い配列\(リード\)を除去](#) (last modified 2014/08/27)
- 前処理 | [フィルタリング](#) | [ACGTのみからなる配列を抽出](#) (last modified 2015/09/12)
- 前処理 | [フィルタリング](#) | [ACGT以外のcharacter "-"をNに変換](#) (last modified 2013/06/18)
- 前処理 | [フィルタリング](#) | [ACGT以外の文字数が閾値以下の配列を抽出](#) (last modified 2015/09/12)
- 前処理 | [フィルタリング](#) | [重複のない配列セットを作成](#) (last modified 2013/06/18)
- 前処理 | [フィルタリング](#) | [指定した長さ以上の配列を抽出](#) (last modified 2016/02/08)
- 前処理 | [フィルタリング](#) | [任意のリード\(サブセット\)を抽出](#) (last modified 2014/08/21)
- 前処理 | [フィルタリング](#) | [指定した長さの範囲の配列を抽出](#) (last modified 2015/02/26)
- 前処理 | [フィルタリング](#) | [任意のIDを含む配列を抽出](#) (last modified 2013/06/18)
- 前処理 | [フィルタリング](#) | [Illuminaのpass filtering](#) (last modified 2013/06/19)
- 前処理 | [フィルタリング](#) | [GFF/GTF形式ファイル](#) (last modified 2013/10/10) [トップページへ](#)
- 前処理 | [フィルタリング](#) | [組合せ | ACGTのみ & 指定した長さの範囲の配列](#) (last modified 2015/09/12)

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。

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ー。

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

#ファイルに保存
```

[トップページへ](#)

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。

事前準備

The screenshot shows the RGui (64-bit) interface with the R Console window open. The R Console displays the output of the `list.files()` command, which lists files in the current working directory. Two red arrows point to the output: arrow ① points to the directory path `"C:/Users/kadota/Desktop"`, and arrow ② points to the file `"out_gapClosed.fa"`. The R Console also shows the execution of `getwd()` and `list.files()` commands. The RGui window shows the menu bar (ファイル, 編集, 閲覧, その他, パッケージ, ウィンドウ, ヘルプ) and the toolbar. The background shows a web browser window with the URL `www.iu.a.u-tokyo.ac.jp/~kadota/r_seq.html#preprocessing fi...`.

```
R Console  
R は多くの貢献者による共同プロジェクトです。  
詳しくは 'contributors()' と入力してください。  
また、R や R のパッケージを出版物で引用する際の形 $  
'citation()' と入力してください。  
  
'demo()' と入力すればデモをみることができます。  
'help()' とすればオンラインヘルプが出ます。  
'help.start()' で HTML ブラウザによるヘルプがみら $  
'q()' と入力すれば R を終了します。  
  
> getwd()  
[1] "C:/Users/kadota/Desktop" ①  
> list.files()  
[1] "desktop.ini" "hoge"  
[3] "out_gapClosed.fa" ②  
> |  
  
#入力ファイル名を指定してin_fに格納  
#出力ファイル名を指定してout_fに格納  
#配列長の閾値を指定  
  
#パッケージの読み込み  
  
format="fasta")#in_fで指定したファイルの記  
#確認してるだけです  
  
param_length)#条件を満たすかどうかを判定  
#objがTRUEとなる要素のみ抽出した結果  
#確認してるだけです  
  
format="fasta", width=50)#fasta
```

Rエディタで編集

①デスクトップ上に保存した、②out_gapClosed.fasta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。

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

```
in_f <- "hoge4.fa"
out_f <- "hogel.fasta"
param_length <- 50

#必要なパッケージをロード
library(Biostrings)

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

#本番
obj <- as.logical(width(fasta) >= param_length) #条件を満たすかどうかを判定
fasta <- fasta[obj]
fasta

#ファイルに保存
writeXStringSet(fasta, file=out_f, format="fasta", width=50) #fasta
```

Red arrows indicate the following actions:

- ③: Points to the first part of the code in the R Console window.
- ④: Points to the second part of the code in the R Console window.

必要最小限の変更

①デスクトップ上に保存した、②out_gapClosed.fasta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…

(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 |

RGui (64-bit)

ファイル 編集 パッケージ ウィンドウ ヘルプ



FASTA形式やFASTA
「ファイル」 - 「マ

1. multi-FASTA

イントロ | 一般

```
in_f <- "hoge"
out_f <- "hogel.fasta"
param_length
```

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

```
#入力ファイルの読み込み
fasta <- readDNASTringSet(in_f, format="fasta")
fasta
```

```
#本番
obj <- as.logical(width(fasta) >= param_length)
fasta <- fasta[obj]
```

```
#ファイルに保存
writeXStringSet(fasta, file=out_f, format="fasta", width=50)
```

R Console

```
R は多くの貢献者による共同プロジェクトです。
詳しくは 'contributors()' を見てください。
また、R や R のパッケージをインストールするに
'citation()' と入力すれば、適切な引用を出力します。
'demo()' と入力すればデモを起動します。
'help()' とすればオンラインヘルプを開きます。
'help.start()' で HTML ヘルプを起動します。
'q()' と入力すれば R を終了します。

> getwd()
[1] "C:/Users/kadota/Desktop"
> list.files()
[1] "desktop.ini"
[3] "out_gapClosed.fasta"
> |
```

無題 - RIデータ

```
in_f <- "out_gapClosed.fasta" #入力ファイル名を指定してin_fに格納
out_f <- "hogel.fasta" #出力ファイル名を指定してout_fに格納
param_length <- 300 #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
fasta <- fasta[obj] #objがTRUEとなる要素のみ抽出した結果
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
```



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。

(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/~kadot

前処理 |

RGui (64-bit)

ファイル **7** パッケージ ウィンドウ ヘルプ



FASTA形式やFASTA
「ファイル」 - 「ラ

1. multi-FASTA

イントロ | 一般

```
in_f <- "hoge"
out_f <- "hogel.fasta"
param_length <- 300
```

#必要なパッケージをロード
library(Biost

#入力ファイルの読み込み
fasta <- readDN

fasta

#本番
obj <- as.log

fasta <- fast
fasta

#ファイルに保存
writeXStringSet(fasta, file=out_f, format="fasta", width=50)

カーソル行または選択中のRコードを実行

R Console

R 無題 - RIデータ

R は多くの貢献者による共同開発です。
詳しくは 'contributors()' を見てください。
また、R や R のパッケージをインストールする際、
'citation()' と入力すれば、そのパッケージの
'demo()' と入力すればデモを実行し、
'help()' とすればオンラインヘルプを開き、
'help.start()' で HTML ヘルプを開き、
'q()' と入力すれば R を終了します。

```
> getwd()
[1] "C:/Users/kadot"
> list.files()
[1] "desktop.ini"
[3] "out_gapClosed.f"
> |
```

```
in_f <- "out_gapClosed.fa" #入力ファイル名を指定して読み込み
out_f <- "hogel.fasta" #出力ファイル名を指定してout_fに格納
param_length <- 300 #配列長の閾値を指定

#必要なパッケージをロード
library(Biostings) #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断
fasta <- fasta[obj] #objがTRUEとなる要素のみ抽出した結果
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をout_fに保存
```

実行結果

The screenshot shows the R GUI interface with the R Console window open. The console displays the following output:

```
[1] 10520 ATCGATAT...ATTGCTG scaffold1_cov55
[2] 136075 TTTAAGGA...GAATTAA scaffold2_cov60
[3] 64531 CACCGTAT...TATCTAG scaffold3_cov43
[4] 35091 CAAAATGC...GATTGAA scaffold4_cov52
[5] 467 GTACCAAG...TTAGAAA scaffold8_cov49
...
[48] 798 TGTTACGA...TCTTCCA scaffold96_cov160
[49] 1439 AAATAAAT...AAGCTTC scaffold100_cov358
[50] 1220 TTCTCACC...CGGAAAT scaffold102_cov196
[51] 652 CCAACCTA...TAGAGTG scaffold109_cov158
[52] 747 CGGGAGTA...TTCACGC scaffold114_cov106
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta$
>
> |
> |
```

The script in the background includes the following code and comments:

```
in_f <- "hoge"
out_f <- "hog"
param_length

#必要なパッケージ
library(Biostat)

#入力ファイルの読み込み
fasta <- readLines(in_f)

#本番
obj <- as.log(fasta)
fasta <- fastq(obj)

#ファイルに保存
writeXStringSet(fasta, file=out_f, format="fasta", width=50) #fas
```

Comments in the script explain the steps: #入力ファイル名を指定して (Specify input filename), #出力ファイル名を指定してout_fに格納 (Specify output filename and store in out_f), #配列長の閾値を指定 (Specify sequence length threshold), #パッケージの読み込み (Load package), #in_fで指定したファイルの読み込み (Load file specified in in_f), #確認してるだけです (Just confirming), #条件を満たすかどうかを判定 (Judge if conditions are met), #objがTRUEとなる要素のみ抽出した結果 (Extract only elements where obj is TRUE), #確認してるだけです (Just confirming).

実行後は52配列

こんな感じになります。①を押していった、②fastaオブジェクトが見える位置にただけです。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。

(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...

前処理 |

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ



FASTA形式やFASTA
「ファイル」 - 「ラ

1. multi-FASTA

イントロ |

```
in_f <- "hoge1.fasta"
out_f <- "hoge1.fasta"
param_length <- 50
```

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

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

```
#本番実行
obj <- as.list(fasta)
fasta <- filterByLength(obj,
                        param_length)
```

```
#ファイルに保存
writeDNAStringSet(fasta, out_f,
                  format="fasta", width=50)
```

R Console

```
> fasta #確認して$
A DNAStringSet instance of length 52
      width seq          names
[1]  10520 ATCGATAT...ATTGCTG scaffold1_cov55
[2] 136075 TTTAAGGA...GAATTAA scaffold2_cov60
[3]  64531 CACCGTAT...TATCTAG scaffold3_cov43
[4]  35091 CAAAATGC...GATTGAA scaffold4_cov52
[5]    467 GTACCAAG...TTAGAAA scaffold8_cov49
...
[48]   798 TGTTACGA...TCTTCCA scaffold96_cov160
[49]  1439 AAATAAAT...AAGCTTC scaffold100_cov358
[50]  1220 TTCTCACC...CGGAAAT scaffold102_cov196
[51]   652 CCAACCTA...TAGAGTG scaffold109_cov158
[52]   747 CGGGAGTA...TTCACGC scaffold114_cov106
> #ファイルに保存
```

```
#入力ファイル名を指定して
#出力ファイル名を指定してout_fに格
#配列長の閾値を指定

#パッケージの読み込み

format="fasta")#in_fで指定したファイルの
#確認してるだけです

param_length)#条件を満たすかどうかを判
#objがTRUEとなる要素のみ抽出した
#確認してるだけです

format="fasta", width=50)#fas
```



実行前は117配列

こんな感じになります。①を押していった、②fastaオブジェクトが見える位置にただけです。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。

(Rで塩基配列解析

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

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ



R Console

```
> fasta <- readDNASTringSet(in_f, format="fasta")#$
> fasta                                     #確認して$
A DNASTringSet instance of length 117
      width seq          names
[1]  10520 ATCGATA...ATTGCTG scaffold1_cov55
[2] 136075 TTTAAGG...GAATTAA scaffold2_cov60
[3]  64531 CACCGTA...TATCTAG scaffold3_cov43
[4]  35091 CAAAATG...GATTGAA scaffold4_cov52
[5]    105 CAAAACG...TGTGCTA scaffold5_cov98
...    ...    ...
[113]  113 CATTAGT...ACGATGA scaffold113_cov168
[114]  747 CGGGAGT...TTCACGC scaffold114_cov106
[115]  159 ACAAACT...CTAAATT scaffold115_cov182
[116]  117 CTTTAAC...AAATTGT scaffold116_cov184
[117]  263 ATGGGGT...TCTTTTCG scaffold117_cov182
>
```

④

```
#入力ファイル名を指定して
#出力ファイル名を指定してout_fに格
#配列長の閾値を指定
```

```
#パッケージの読み込み
```

```
format="fasta")#in_fで指定したファイル
#確認してるだけです
```

```
param_length)#条件を満たすかどうかを判
#objがTRUEとなる要素のみ抽出した
#確認してるだけです
```

```
format="fasta", width=50)#fas
```

実行前は117配列

こんな感じになります。①を押していった、②fastaオブジェクトが見える位置にただけです。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た、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。

(Rで塩基配列解析

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

前処理 |

FASTA形式やFAST

「ファイル」 - 「ラ

1. multi-FASTA

イントロ |

```
in_f <- "hoge  
out_f <- "hog  
param_length
```

```
#必要なパッケー  
library(Biost
```

```
#入力ファイルの  
fasta <- read  
fasta
```

```
#本番
```

```
obj <- as.lo  
fasta <- fa  
fasta
```

```
#ファイルに保存
```

⑥

④

```
R Console  
> fasta <- readDNASTringSet(in_f, format="fasta")#$  
> fasta #確認して$  
A DNASTringSet instance of length 117  
width seq names  
[1] 10520 ATCGATA...ATTGCTG scaffold1_cov55  
[2] 136075 TTTAAGG...GAATTAA scaffold2_cov60  
[3] 64531 CACCGTA...TATCTAG scaffold3_cov43  
[4] 35091 CAAAATG...GATTGAA scaffold4_cov52  
[5] 105 CAAAACG...TGTGCTA scaffold5_cov98  
... ..  
[113] 113 CATTAGT...ACGATGA scaffold113_cov168  
[114] 747 CGGGAGT...TTCACGC scaffold114_cov106  
[115] 159 ACAAACT...CTAAATT scaffold115_cov182  
[116] 117 CTTTAAC...AAATTGT scaffold116_cov184  
[117] 263 ATGGGGT...TCTTTCG scaffold117_cov182  
>
```

```
#入力ファイル名を指定して  
#出力ファイル名を指定してout_fに格  
#配列長の閾値を指定  
  
#パッケージの読み込み  
  
format="fasta")#in_fで指定したファイル  
#確認してるだけです  
  
param_length)#条件を満たすかどうかを判  
#objがTRUEとなる要素のみ抽出した  
#確認してるだけです  
  
format="fasta", width=50)#fas
```

⑤

第6回の原稿PDF

(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/~kado

書籍 | 日本乳酸菌学会誌 | 第6回ゲノムアセンブ

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
 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
 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
 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
 まれていないです (2016年4月23日追加)。

- [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回終了時点以降のovaファイル
 - [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)

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もであろう。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、FaQCs⁷⁾ 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。

(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/~kado

書籍 | 日本乳酸菌学会誌 | 第6回ゲノムアセンブ

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
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
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
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
まれている」です(2016年4月23日追加)。

- [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回終了時点以降のovaファイル
 - [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)

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もであろう。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 1000 万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、FaQCs⁷⁾ 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。

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trand情報付きのリストファイルの作成、例題6のコード解説
 - スtrand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

(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#

(Rで)塩基配列解析のサブ

(last modified 2019/04/10, sin

ここは、(Rで)塩基配列解析のサブ

What's new?

- 「参考資料 | [講義、講演資料](#)」を
- 乳酸菌学会誌のNGS連載第13回の
- ったのでそれに変更しました。(2
- 日本乳酸菌学会誌のNGS関連連載
- (2019/03/18) **NEW**
- RNA-seqカウントデータ解析用R
- (2019/03/14)
- [TCC-GUIのオンライン版](#)の基本的
- のオンライン版はインストールが

(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#

- 書籍 | [トランスクリプトーム解析 | 4.3.3 2群間比較](#) (last modified 2014/04/28)
- 書籍 | [トランスクリプトーム解析 | 4.3.4 他の実験デザイン\(3群間\)](#) (last modified 2014/04/28)
- 書籍 | [日本乳酸菌学会誌 | について](#) (last modified 2019/04/05) **NEW**
- 書籍 | [日本乳酸菌学会誌 | 第1回イントロダクション](#) (last modified 2018/09/03)
- 書籍 | [日本乳酸菌学会誌 | 第2回GUI環境からコマンドライン環境へ](#) (last modified 2018/09/03)
- 書籍 | [日本乳酸菌学会誌 | 第3回Linux環境構築からNGSデータ取得まで](#) (last modified 2017/07/02)
- 書籍 | [日本乳酸菌学会誌 | 第4回クオリティコントロールとプログラムのインストール](#) (last modified 2018/05/10)
- 書籍 | [日本乳酸菌学会誌 | 第5回アセンブル、マッピング、そしてQC](#) (last modified 2017/06/25)
- 書籍 | [日本乳酸菌学会誌 | 第6回ゲノムアセンブリ](#) (last modified 2017/06/21)
- 書籍 | [日本乳酸菌学会誌 | 第7回ロングリードアセンブリ](#) (last modified 2017/06/28)
- 書籍 | [日本乳酸菌学会誌 | 第8回アセンブリ後の解析](#) (last modified 2017/06/28)
- 書籍 | [日本乳酸菌学会誌 | 第9回ゲノムアノテーションとその可視化、DDBJへの登録](#) (last modified 2017/03/13)
- 書籍 | [日本乳酸菌学会誌 | 第10回DDBJへの塩基配列の登録\(後編\)](#) (last modified 2017/06/28)
- 書籍 | [日本乳酸菌学会誌 | 第11回統合データ解析環境Galaxy](#) (last modified 2017/11/13)
- 書籍 | [日本乳酸菌学会誌 | 第12回Galaxy: ヒストリーとワークフロー](#) (last modified 2018/07/04)
- 書籍 | [日本乳酸菌学会誌 | 第13回RNA-seq解析\(その1\)](#) (last modified 2019/04/05) **NEW**

[トップページへ](#)

第9回の原稿PDF

(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/~kado

書籍 | 日本乳酸菌学会誌 | 第9回ゲノムアノテーション登録

日本乳酸菌学会誌の第9回分です。Linuxコマンドのリンク先は主に日経

- [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)

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、アミノ酸に翻訳される領域(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。

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。

(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/~kado

書籍 | 日本乳酸菌学会誌 | 第9回ゲノムアノテーション登録

日本乳酸菌学会誌の第9回分です。Linuxコマンドのリンク先は主に日経

- [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)

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、アミノ酸に翻訳される領域(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。

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/~kado

書籍 | 日本乳酸菌学会誌 | 第9回ゲノムアノテーション登録

日本乳酸菌学会誌の第9回分です。Linuxコマンドのリンク先は主に日経

- [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)

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、アミノ酸に翻訳される領域(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、主に後者だと思えます。

ゲノム配列

アノテーションの実行

遺伝子1

遺伝子2

遺伝子3

遺伝子4

ゲノム配列

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trand情報付きのリストファイルの作成、例題6のコード解説
 - スtrand情報反映戦略を練り、コードに修正を加えて実行(例題7)

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

参考文献(全般)

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 PS¹, Shippy T², Miller S², Benoit JB³, Munoz-Torres M^{4,5}, Flores-Gonzalez M¹, Mueller LA¹, Wiersma-Koch H⁶, D'Elia T⁶, Brown SJ², Saha S¹.

⊕ 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)

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 Y¹, Fujisawa T¹, Nakamura Y¹.



⊕ 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/bty011

DFAST: a flexible prokaryotic genome annotation pipeline

Tanizawa Y¹, Fujisawa T¹, Nakamura Y¹.



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.

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。

(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#

(Rで)塩基配列解析のサブ

(last modified 2019/04/10, sin

ここは、[\(Rで\)塩基配列解析のサブ](#)

What's new?

- 「参考資料 | [講義、講演資料](#)」を
- 乳酸菌学会誌のNGS連載第13回の
ったのでそれに変更しました。(2
- [日本乳酸菌学会誌](#)のNGS関連連載
(2019/03/18) **NEW**
- RNA-seqカウントデータ解析用R
(2019/03/14)
- [TCC-GUI](#)の[オンライン版](#)の基本的
のオンライン版はインストールが

(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#

- 書籍 | [トランスクリプトーム解析 | 4.3.3 2群間比較](#) (last modified 2014/04/28)
- 書籍 | [トランスクリプトーム解析 | 4.3.4 他の実験デザイン\(3群間\)](#) (last modified 2014/04/28)
- 書籍 | [日本乳酸菌学会誌 | について](#) (last modified 2019/04/05) **NEW**
- 書籍 | [日本乳酸菌学会誌 | 第1回イントロダクション](#) (last modified 2018/09/03)
- 書籍 | [日本乳酸菌学会誌 | 第2回GUI環境からコマンドライン環境へ](#) (last modified 2018/09/03)
- 書籍 | [日本乳酸菌学会誌 | 第3回Linux環境構築からNGSデータ取得まで](#) (last modified 2017/07/02)
- 書籍 | [日本乳酸菌学会誌 | 第4回クオリティコントロールとプログラムのインストール](#) (last modified 2018/05/10)
- 書籍 | [日本乳酸菌学会誌 | 第5回アセンブル、マッピング、そしてQC](#) (last modified 2017/06/25)
- 書籍 | [日本乳酸菌学会誌 | 第6回ゲノムアセンブリ](#) (last modified 2017/06/21)
- 書籍 | [日本乳酸菌学会誌 | 第7回ロングリードアセンブリ](#) (last modified 2017/06/28)
- 書籍 | [日本乳酸菌学会誌 | 第8回アセンブリ後の解析](#) (last modified 2017/06/28)
- 書籍 | [日本乳酸菌学会誌 | 第9回ゲノムアノテーションとその可視化、DDBJへの登録](#) (last modified 2017/03/13)
- 書籍 | [日本乳酸菌学会誌 | 第10回DDBJへの塩基配列の登録\(後編\)](#) (last modified 2017/06/28)
- 書籍 | [日本乳酸菌学会誌 | 第11回統合データ解析環境Galaxy](#) (last modified 2017/11/13)
- 書籍 | [日本乳酸菌学会誌 | 第12回Galaxy: ヒストリーとワークフロー](#) (last modified 2018/07/04)
- 書籍 | [日本乳酸菌学会誌 | 第13回RNA-seq解析\(その1\)](#) (last modified 2019/04/05) **NEW**

[トップページへ](#)

参考文献(原核生物)

[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 Y¹](#), [Fujisawa T¹](#), [Nakamura Y¹](#).

⊕ 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.



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trand情報付きのリストファイルの作成、例題6のコード解説
 - スtrand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実行

DFAST & DAGA x +

https://dfast.nig.ac.jp

DFAST Analysis Archive DFAST-core API Download FAQ Help

[2019-Feb-6] Updated. More options became available to customize the workflow.

DFAST DDBJ Fast Annotation and Submission Tool

1

Start your project! [Running 0 / Waiting 0]

Please see [FAQ](#) and [Sample Result](#) if this is your first visit.

DAGA : DFAST Archive of Genome Annotation

DAGA stores genomic data collected from the public nucleotide database and the sequence read archive. All the genomes are consistently annotated using DFAST. Currently, DAGA is available only for genomes of Lactic Acid Bacteria.

1421 annotated genome resources are available, covering 2 genera and 191 species. [ENTER](#)

© 2016-2018 National Institute of Genetics. [✉ dfast\(at\)nig.ac.jp](mailto:dfast(at)nig.ac.jp)

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。

The screenshot shows the DFAST web interface for submitting a new job. The browser address bar shows the URL <https://dfast.nig.ac.jp/dfc/>. The navigation menu includes Analysis, Archive, DFAST-core, API, Download, FAQ, and Help. The main heading is "DFAST Prokaryotic genome annotation pipeline".

The form includes the following fields and options:

- Query File (Fasta format, up to 15Mbyte):** A file selection button labeled "ファイルを選択" with a red arrow and the number 1 pointing to it. A checkbox for "Demo mode (Sample annotation for E.coli O26)" is also present.
- Job Title:** An optional text input field.
- Mail Address:** An optional text input field with a note: "E-mail notification will be sent to this address when the job is completed. (optional)".
- Advanced Options:** A dropdown menu.
- Run:** A blue button with a red arrow and the number 2 pointing to it.

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

The image shows a web browser window displaying the DFAST Prokaryotic genome annotation pipeline interface. The browser address bar shows <https://dfast.nig.ac.jp/dfc/>. The interface includes a navigation menu with 'Analysis', 'Archive', 'DFAST-core', 'API', 'Download', 'FAQ', and 'Help'. The main content area is titled 'DFAST Prokaryotic genome annotation pipeline' and features a 'Query File (Fasta format, up to 15Mbyte)' section. A 'ファイルを選択' button is visible, along with a 'Demo mode (Sample annotation)' checkbox. Below this, there are input fields for 'Job Title' and 'Mail Address'. A 'Run' button is located at the bottom left of the interface.

Overlaid on the browser window is a Windows File Explorer window titled '開く'. The address bar shows the path 'PC > デスクトップ', with a red arrow labeled '①' pointing to it. The left sidebar shows the 'デスクトップ' folder selected. The main pane displays a folder named 'hoge' and a file named 'out_gapClosed.fa', with a red arrow labeled '②' pointing to the file. The bottom of the window shows the file name 'out_gapClosed.fa' in the 'ファイル名(N):' field, and the '開く(O)' button is highlighted with a red arrow labeled '③'.

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 - submit a new job - x +

https://dfast.nig.ac.jp/dfc/

DFAST Analysis Archive DFAST-core API Download FAQ Help

DFAST Prokaryotic genome annotation pipeline

Query File (Fasta format, up to 15Mbyte)

Demo mode (Sample annotation for E.coli O26)

ファイルを選択 out_g...ed.fa ①

Job Title

(optional)

Mail Address

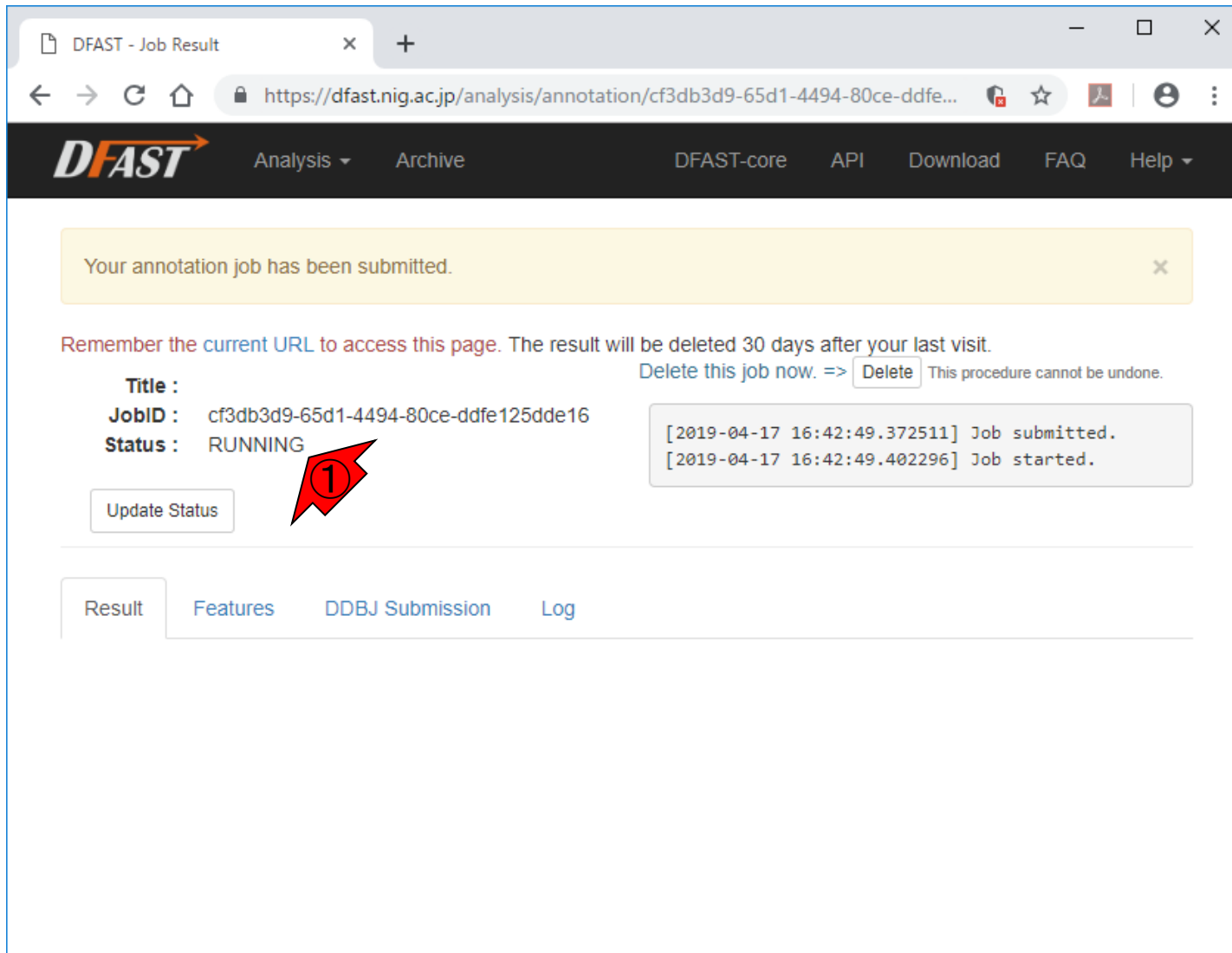
E-mail notification will be sent to this address when the job is completed. (optional)

▼ Advanced Options

② Run

DFAST実行開始

こんな感じになります。①実行中(RUNNING)となっておりますね。



The screenshot shows a web browser window with the URL `https://dfast.nig.ac.jp/analysis/annotation/cf3db3d9-65d1-4494-80ce-ddfe...`. The page header includes the DFAST logo and navigation links: Analysis, Archive, DFAST-core, API, Download, FAQ, and Help. A yellow notification box at the top states: "Your annotation job has been submitted." Below this, a message reads: "Remember the current URL to access this page. The result will be deleted 30 days after your last visit." The job details are as follows:

- Title :
- JobID : cf3db3d9-65d1-4494-80ce-ddfe125dde16
- Status : RUNNING

A red arrow with the number 1 points to the "RUNNING" status. To the right, a log box shows the following entries:

```
[2019-04-17 16:42:49.372511] Job submitted.  
[2019-04-17 16:42:49.402296] Job started.
```

Below the job details is an "Update Status" button. At the bottom, there are tabs for "Result", "Features", "DDBJ Submission", and "Log".

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。

The screenshot shows a web browser window with the URL <https://dfast.nig.ac.jp/analysis/annotation/cf3db3d9-65d1-4494-80ce-ddfe...>. The page title is "DFAST - Job Result". The navigation menu includes "Analysis", "Archive", "DFAST-core", "API", "Download", "FAQ", and "Help".

Remember the [current URL](#) to access this page. The result will be deleted 30 days after your last visit.
Delete this job now. => This procedure cannot be undone.

Title :
JobID : cf3db3d9-65d1-4494-80ce-ddfe125dde16
Status : COMPLETE

The job status log shows the following entries:

```
[2019-04-17 16:42:49.372511] Job submitted.  
[2019-04-17 16:42:49.402296] Job started.  
[2019-04-17 16:44:32.417969] Job completed.
```

The "Status" field and the log entries are highlighted with red arrows and circled numbers 1 and 2, respectively.

Navigation tabs: Result (selected), Features, DDBJ Submission, Log

Genome Statistics

Total Length (bp)	2,348,706
No. of Sequences	61
GC Content (%)	38.1%
N50	92,304
Gap Ratio (%)	0.0%
No. of CDSs	2,311

Download Files

- Genbank Flat File : [annotation.gbk](#)
- GFF3-formatted ... : [annotation.gff](#)
- Genome Fasta F... : [genome.fna](#)
- Protein Fasta File : [protein.faa](#)
- CDS Fasta File : [cds.fna](#)
- RNA Fasta File : [rna.fna](#)
- Feature Table : [features.tsv](#)
- Genome Statisti... : [statistics.txt](#)
- Zip Archive : [annotation.zip](#)

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！

DFAST - Job Result

https://dfast.nig.ac.jp/analysis/annotation/cf3db3d9-65d1-4494-80ce-ddfe...

DFAST Analysis Archive DFAST-core API Download FAQ Help

Remember the [current URL](#) to access this page. The result will be deleted 30 days after your last visit.

Delete this job now. => This procedure cannot be undone.

Title :
JobID : cf3db3d9-65d1-4494-80ce-ddfe125dde16
Status : COMPLETE

```
[2019-04-17 16:42:49.372511] Job submitted.  
[2019-04-17 16:42:49.402296] Job started.  
[2019-04-17 16:44:32.417969] Job completed.
```

Result Features DDBJ Submission Log

Genome Statistics

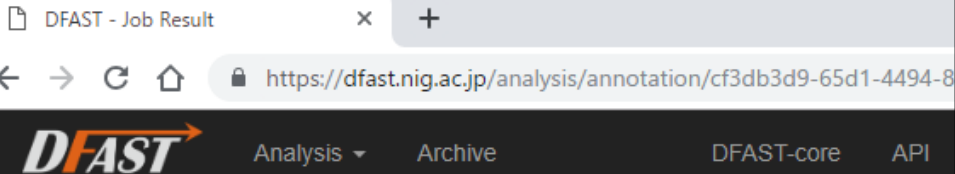
Total Length (bp)	2,348,706
No. of Sequences	61
GC Content (%)	38.1%
N50	92,304
Gap Ratio (%)	0.0%
No. of CDSs	2,311

Download Files

- Genbank Flat File : [annotation.gbk](#)
- GFF3-formatted ... : [annotation.gff](#)
- Genome Fasta F... : [genome.fna](#)
- Protein Fasta File : [protein.faa](#)
- CDS Fasta File : [cds.fna](#)
- RNA Fasta File : [rna.fna](#)
- Feature Table : [features.tsv](#)
- Genome Statisti... : [statistics.txt](#)
- Zip Archive : [annotation.zip](#)

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！



Remember the [current URL](#) to access this page. The result will be deleted 30 days after

Delete this job now. => This procedure cannot be undone.

Title :
JobID : cf3db3d9-65d1-4494-80ce-ddfe125dde16
Status : COMPLETE

[2019-04-17 16:42:49.372511] Job submitted.
[2019-04-17 16:42:49.402296] Job started.
[2019-04-17 16:44:32.417969] Job completed.

Result Features DDBJ Submission Log

Genome Statistics

Total Length (bp)	2,348,706
No. of Sequences	61
GC Content (%)	38.1%
N50	92,304
Gap Ratio (%)	0.0%
No. of CDSs	2,311

Download Files

Genbank Flat File : [annotation.gbk](#)
GFF3-formatted ... : [annotation.gff](#)
Genome Fasta F... : [genome.fna](#)
Protein Fasta File : [protein.faa](#)
CDS Fasta File : [cds.fna](#)
RNA Fasta File : [rna.fna](#)
Feature Table : [features.tsv](#)
Genome Statisti... : [statistics.txt](#)
Zip Archive : [annotation.zip](#)



DFAST Help

ざっと調べた限り、原著論文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 - Job Result

https://dfast.nig.ac.jp/analysis/annotation/cf3db3d9-65d1-4494-80ce-ddfe...



Analysis

Archive

DFAST-core

API

Download

FAQ

Help

Remember the current URL to access this page. The result will be deleted 30 days after y

Delete this job now. => Delete

Title :

JobID : cf3db3d9-65d1-4494-80ce-ddfe125dde16

Status : COMPLETE

About DFAST and DAGA

DFAST Help

DAGA Help

[2019-04-17 16:42:49.372511] Job submitted.

[2019-04-17 16:42:49.402296] Job started.

[2019-04-17 16:44:32.417969] Job completed.

Result

Features

DDBJ Submission

Log

Genome Statistics

Total Length (bp) 2,348,706

No. of Sequences 61

GC Content (%) 38.1%

N50 92,304

Gap Ratio (%) 0.0%

No. of CDSs 2,311

Download Files

Genbank Flat File : [annotation.gbk](#)

GFF3-formatted ... : [annotation.gff](#)

Genome Fasta F... : [genome.fna](#)

Protein Fasta File : [protein.faa](#)

CDS Fasta File : [cds.fna](#)

RNA Fasta File : [rna.fna](#)

Feature Table : [features.tsv](#)

Genome Statisti... : [statistics.txt](#)

Zip Archive : [annotation.zip](#)

DFAST Help

ざっと調べた限り、原著論文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。

The screenshot shows the DFAST Help page in a web browser. The browser's address bar displays the URL https://dfast.nig.ac.jp/help_annotation. The page features a navigation menu with the following items: Analysis, Archive, DFAST-core, API, Download, FAQ, and Help. The main content area is titled "DFAST Help" and is divided into sections. The first section is "1. Overview of DFAST." which contains text about the DFAST pipeline and its integration with quality and taxonomy assessment methods. The second section is "2. Submit a new job." which includes a sub-section "i. Query File" with instructions on file format and size. Red callout boxes with numbers 3, 4, and 5 are overlaid on the page. Callout 3 points to the "DFAST Help" title. Callout 4 points to the "2. Submit a new job." section header. Callout 5 points to the vertical scrollbar on the right side of the page.

課題1

DFASTへの入力として与えたout_gapClosed.fasta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。

DFAST: DDBJ Fast Annotation and ...
https://dfast.nig.ac.jp/help_annotation

2. Submit a new job.



DFAST is an annotation platform for bacterial genomes. Its core annotation process is based on PROKKA and custom reference databases tailored to specific organisms. It also generates DDBJ-compliant submission files for Mass Submission System (MSS) at DDBJ.

Query File (Fasta format) **i.** Job Title

Mail Address

Specify metadata and parameters.
These data other than minimum contig length can be altered later. Reference Databases for genera other than Lactobacillus and Peptostreptococcus are not fully supported.

Genus **ii.** Species Strain

Locus Tag Prefix **iii.** Minimum Contig Length **iv.**

Perform Genome Assessment (optional)

Perform CheckM Calculation for Genome Quality Assessment **v.**

Select a Taxonomic Group for CheckM Calculation:

Rank: Genus

Perform AMI Calculation for Taxonomic Assessment **vi.**

Select Target Group for AMI Calculation:

Run

i. Query File

Only fasta-formatted file (<10Mbyte) is acceptable. Compressed files (.zip, .gz, or .bz2) are not acceptable.

ii. Organism Name

Genus, Species, and Strain are required, and can be modified later.

iii. Locus Tag Prefix

Required, and this can be modified later. Locus_tags are identifiers that are systematically applied to every gene in a genome. You need to register locus_tag prefix before submitting the genome to INSDC. Please refer to the guideline of DDBJ for more information. [Japanese](#) / [English](#)

iv. Minimum Contig Length

Contigs shorter than this length will be eliminated. The default value of 200 bp is the recommendation of INSDC. Please refer to the [NCBI WGS submission guideline](#) for more information.

v. Check here to perform Genome Quality

Assessment using CheckM

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。

(Rで)塩基配列解析

(last modified 2019/04/15, since 2010)

このウェブページのR関連部分は、[インストーラ \(Macintosh2018.11.27版\)](#)に従ってインストールされています。初心者の方は[基本的な利用法](#)をご覧ください。**2018年7月に(Rで)塩基配列解析の更新** (2018/07/18)

What's new? (過去のお知らせはこちら)

- 「カウント情報取得 | シミュレーション」を追加しました。(2019/04/11) **NEW**
- 「カウント情報取得 | シミュレーション」を追加しました。(2019/04/11) **NEW**
- 削除予定としていた「インストール」
- 削除予定としていた「インストール」

①

②

- 前処理 | トリミング | [指定した末端塩基数だけ除去](#) (last modified 2017/11/08)
- 前処理 | [フィルタリング](#) | [について](#) (last modified 2018/08/06)
- 前処理 | [フィルタリング](#) | [PHREDスコアが低い塩基をNに置換](#) (last modified 2014/03/03)
- 前処理 | [フィルタリング](#) | [PHREDスコアが低い配列\(リード\)を除去](#) (last modified 2014/08/27)
- 前処理 | [フィルタリング](#) | [ACGTのみからなる配列を抽出](#) (last modified 2015/09/12)
- 前処理 | [フィルタリング](#) | [ACGT以外のcharacter "-"をNに変換](#) (last modified 2013/06/18)
- 前処理 | [フィルタリング](#) | [ACGT以外の文字数が閾値以下の配列を抽出](#) (last modified 2015/09/12)
- 前処理 | [フィルタリング](#) | [重複のない配列セットを作成](#) (last modified 2013/06/18)
- 前処理 | [フィルタリング](#) | [指定した長さ以上の配列を抽出](#) (last modified 2016/02/08)
- 前処理 | [フィルタリング](#) | [任意のリード\(サブセット\)を抽出](#) (last modified 2014/08/21)
- 前処理 | [フィルタリング](#) | [指定した長さの範囲の配列を抽出](#) (last modified 2015/02/26)
- 前処理 | [フィルタリング](#) | [任意のIDを含む配列を抽出](#) (last modified 2013/06/18)
- 前処理 | [フィルタリング](#) | [Illuminaのpass filtering](#) (last modified 2013/06/19)
- 前処理 | [フィルタリング](#) | [GFF/GTF形式ファイル](#) (last modified 2013/10/10)
- 前処理 | [フィルタリング](#) | [組合せ | ACGTのみ & 指定した長さの範囲の配列](#) (last modified 2015/09/12)

[トップページへ](#)

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。

DFAST結果ファイル

講義日程 (2019年度)

- 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
- 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
- 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
- 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

- 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trand情報付きのリストファイルの作成、例題6のコード解説
 - スtrand情報反映戦略を練り、コードに修正を加えて実行(例題7)

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。

おさらい

■ 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

比較ゲノム解析

■ オースログの同定

- オースログとは、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

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
- ▶ 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に近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)にして他方をそのような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。

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trand情報付きのリストファイルの作成、例題6のコード解説
 - スtrand情報反映戦略を練り、コードに修正を加えて実行(例題7)

ファイルの準備

■ オースログの同定

- オースログとは、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。

ファイルの準備

1. 生物配列解析基礎

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
- ▶ 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。

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？

問題設定

■ オースログの同定

- オースログとは、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。



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…

DFAST - Job Result

https://dfast.nig.ac.jp/analysis/annotation/cf3db3d9-65d1-4494-8

DFAST



analysis Archive DFAST-core API Download FAQ Help

Remember the current URL to access this page. The result will be deleted 30 days after your last visit.

Delete this job now. => This procedure cannot be undone.

Title :

JobID : cf3db3d9-65d1-4494-80ce-ddfe125dde16

Status : COMPLETE

```
[2019-04-17 16:42:49.372511] Job submitted.
[2019-04-17 16:42:49.402296] Job started.
[2019-04-17 16:44:32.417969] Job completed.
```

Result Features DDBJ Submission Log

Genome Statistics

Total Length (bp)	2,348,706
No. of Sequences	61
GC Content (%)	38.1%
N50	92,304
Gap Ratio (%)	0.0%
No. of CDSs	2,311

Download Files

Genbank Flat File : [annotation.gbk](#)
GFF3-formatted ... : [annotation.gff](#)
Genome Fasta F... : [genome.fna](#)
Protein Fasta File : [protein.faa](#)
CDS Fasta File : [cds.fna](#) ②
RNA Fasta File : [rna.fna](#)
Feature Table : [features.tsv](#)
Genome Statisti... : [statistics.txt](#)
Zip Archive : [annotation.zip](#)



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



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

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trand情報付きのリストファイルの作成、例題6のコード解説
 - スtrand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。

DFAST - Job Result

https://dfast.nig.ac.jp/analysis/annotation/cf3db3d9-65d1-4494-80ce-ddfe...

DFAST analysis Archive DFAST-core API Download FAQ Help

Remember the [current URL](#) to access this page. The result will be deleted 30 days after your last visit.

Delete this job now. => This procedure cannot be undone.

Title :
JobID : cf3db3d9-65d1-4494-80ce-ddfe125dde16
Status : COMPLETE

[2019-04-17 16:42:49.372511] Job submitted.
[2019-04-17 16:42:49.402296] Job started.
[2019-04-17 16:44:32.417969] Job completed.

Result Features DDBJ Submission Log

Genome Statistics

Total Length (bp)	2,348,706
No. of Sequences	61
GC Content (%)	38.1%
N50	92,304
Gap Ratio (%)	0.0%
No. of CDSs	2,311

Download Files

- Genbank Flat File : [annotation.gbk](#)
- GFF3-formatted ... : [annotation.gff](#)
- Genome Fasta F... : [genome.fna](#)
- Protein Fasta File : [protein.faa](#)
- CDS Fasta File : [cds.fna](#)
- RNA Fasta File : [rna.fna](#)
- Feature Table : [features.tsv](#)
- Genome Statisti... : [statistics.txt](#)
- Zip Archive : [annotation.zip](#)

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](#)

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

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

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。

(Rで)塩基配列解析

(last modified 2019/04/15, since 2010)

このウェブページのR関連部分は、[インストーラ \(Macintosh2018.11.27版\)](#)に従ってインストールされています。初心者の方は[基本的な利用法](#)をご覧ください。
2018年7月に(Rで)塩基配列解析の更新を行いました。(2018/07/18)

What's new? (過去のお知らせはこちら)

- 「カウント情報取得 | シミュレーション」を追加しました。(2019/04/11) **NEW**
- 「カウント情報取得 | シミュレーション」を追加しました。(2019/04/11) **NEW**
- 削除予定としていた「インストール」を削除しました。
- 削除予定としていた「インストール」を削除しました。

①

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

- 前処理 | トリミング | [指定した末端塩基数だけ除去](#) (last modified 2017/11/08)
- 前処理 | [フィルタリング](#) | [について](#) (last modified 2018/08/06)
- 前処理 | [フィルタリング](#) | [PHREDスコアが低い塩基をNに置換](#) (last modified 2014/03/03)
- 前処理 | [フィルタリング](#) | [PHREDスコアが低い配列\(リード\)を除去](#) (last modified 2014/08/27)
- 前処理 | [フィルタリング](#) | [ACGTのみからなる配列を抽出](#) (last modified 2015/09/12)
- 前処理 | [フィルタリング](#) | [ACGT以外のcharacter "-"をNに変換](#) (last modified 2013/06/18)
- 前処理 | [フィルタリング](#) | [ACGT以外の文字数が閾値以下の配列を抽出](#) (last modified 2015/09/12)
- 前処理 | [フィルタリング](#) | [重複のない配列セットを作成](#) (last modified 2013/06/18)
- 前処理 | [フィルタリング](#) | [指定した長さ以上の配列を抽出](#) (last modified 2016/02/08)
- 前処理 | [フィルタリング](#) | [任意のリード\(サブセット\)を抽出](#) (last modified 2014/08/21)
- 前処理 | [フィルタリング](#) | [指定した長さの範囲の配列を抽出](#) (last modified 2015/02/26)
- 前処理 | [フィルタリング](#) | [任意のIDを含む配列を抽出](#) (last modified 2013/06/18)
- 前処理 | [フィルタリング](#) | [Illuminaのpass filtering](#) (last modified 2013/06/19)
- 前処理 | [フィルタリング](#) | [GFF/GTF形式ファイル](#) (last modified 2013/10/10)
- 前処理 | [フィルタリング](#) | [組合せ | ACGTのみ & 指定した長さの範囲の配列](#) (last modified 2015/09/12)

[トップページへ](#)

②

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に変更してコピー実行し、⑤を眺めればよい。

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、指定した配列長さ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ー。

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"
out_f <- "hoge1.fasta"
param_length <- 50

#必要なパッケージをロード
library(Biostrings)

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

#本番
obj <- as.logical(width(fasta) >= param_length)
```

#入力ファイル名を指定してin_fに格納
#出力ファイル名を指定してout_fに格納
#配列長の閾値を指定

#パッケージの読み込み

#in_fで指定したファイルの読み込み
#確認してるだけです

[トップページへ](#)

答えを眺めておく

The screenshot shows the RGui interface. The R Console displays the output of the `fasta` command, which is a `DNASTringSet` instance of length 2311. The output lists genomic coordinates and DNA sequences for several loci. A red arrow points to a code editor window titled "無題 - RIデータ" which contains the R code used to generate the output. The code defines input and output files, sets a parameter length, loads the `Biostrings` library, and reads the input FASTA file into a `fasta` object.

```
> fasta #確認してるだけです
A DNASTringSet instance of length 2311
  width seq
[1] 489 ATGTTAAAACAAATTAGT...TATAGTATTGTTAAATAG LOCUS_23100 $
[2] 741 TTGAAAATTACAGTTCTA...AAGGTTATTGAAATCTAG LOCUS_23110 $
[3] 402 ATGGTTACACTTTATACA...CAGGCCAACTTGTTTTAA LOCUS_23120 $
[4] 705 ATGGAAATGGAACGAATT...TTTTATTTTTTCTAAGTGA LOCUS_23130 $
[5] 1170 ATGAAATATTTAGATGAA...ATGTCCGTGTTTGGGTAA LOCUS_23140 $
...
[2307] 234 ATGGACGAAGTTAAAAGT...TTTTTAAAACCTAAACTAA LOCUS_23150 $
[2308] 447 TTGGGAAGTCGTAACAGT...CAAATTAATTGTTCTAG LOCUS_23160 $
[2309] 1356 ATGTCTACTATACAACAT...ACTACAAATAGATGTTAA LOCUS_23170 $
[2310] 219 GTGTCAGTTGAAAATGGC...GAGTCAGACGGTGAGTGA LOCUS_23180 $
[2311] 162 GTGTCAGGTAGGCAGTTT...GACGAAAGTCGGACTION LOCUS_23190 $
> |
```

```
in_f <- "cds.fna"
out_f <- "hogel.fasta"
param_length <- 50
#必要なパッケージをロード
library(Biostrings)
#入力ファイルの読み込み
fasta <- readDNASTringSet(in_f,
                           out_f,
                           param_length)
fasta
```

答えを眺めておく

赤枠内で、①cds.fnaに変更してコピー実行した結果。
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。

RGui (64-bit)

ファイル 編集 パッケージ ウィンドウ ヘルプ

R Console

```
> fasta #確認してるだけです
A DNASTringSet instance of length 2311
  width seq
[1] 489 ATGTTAAAACAAATTAGT...TATAGTATTGTTAAATAG LOCUS_23100 $
[2] 741 TTGAAAATTACAGTTCTA...AAGGTTATTGAAATCTAG LOCUS_23100 $
[3] 402 ATGGTTACACTTTATACA...CAGGCCAACTTGTTTTAA LOCUS_23100 $
[4] 705 ATGGAAATGGAACGAATT...TTTTATTTTCTAAGTGA LOCUS_23100 $
[5] 1170 ATGAAATATTTAGATGAA...ATGTC CGTGTTTGGGTAA LOCUS_23100 $
... ..
[2307] 234 ATGGACGAAGTTAAAAGT...TTTTTAAAAC TAAACTAA LOCUS_23100 $
[2308] 447 TTGGGAAGTCGTAACAGT...CAAATTA AATTGTTCTAG LOCUS_23100 $
[2309] 1356 ATGTCTACTATACAACAT...ACTACAAATAGATGTTAA LOCUS_23100 $
[2310] 219 GTGTCAGTTGAAAATGGC...GAGTCAGACGGTGAGTGA LOCUS_23100 $
[2311] 162 GTGTCAGGTAGGCAGTTT...GACGAAAGTCGGACTTAG LOCUS_23110 $
> |
```

R 無題 - RIデータ

```
in_f <- "cds.fna"
out_f <- "hogel.fasta"
param_length <- 50
#必要なパッケージをロード
library(Biostrings)
#入力ファイルの読み込み
fasta <- readDNASTringSet(in_f)
fasta
```

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④ sequence logos。

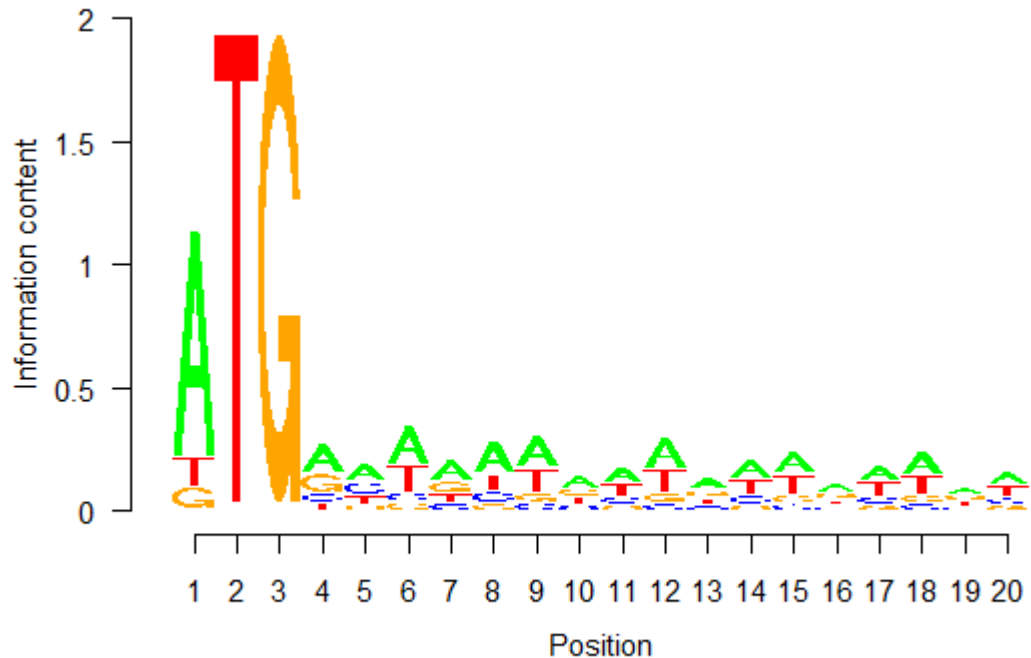
RGui (64-bit)

ファイル 編集 パッケージ ウィンドウ 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
```



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④ sequence logos。⑤ATGが多いことが分かりますね。

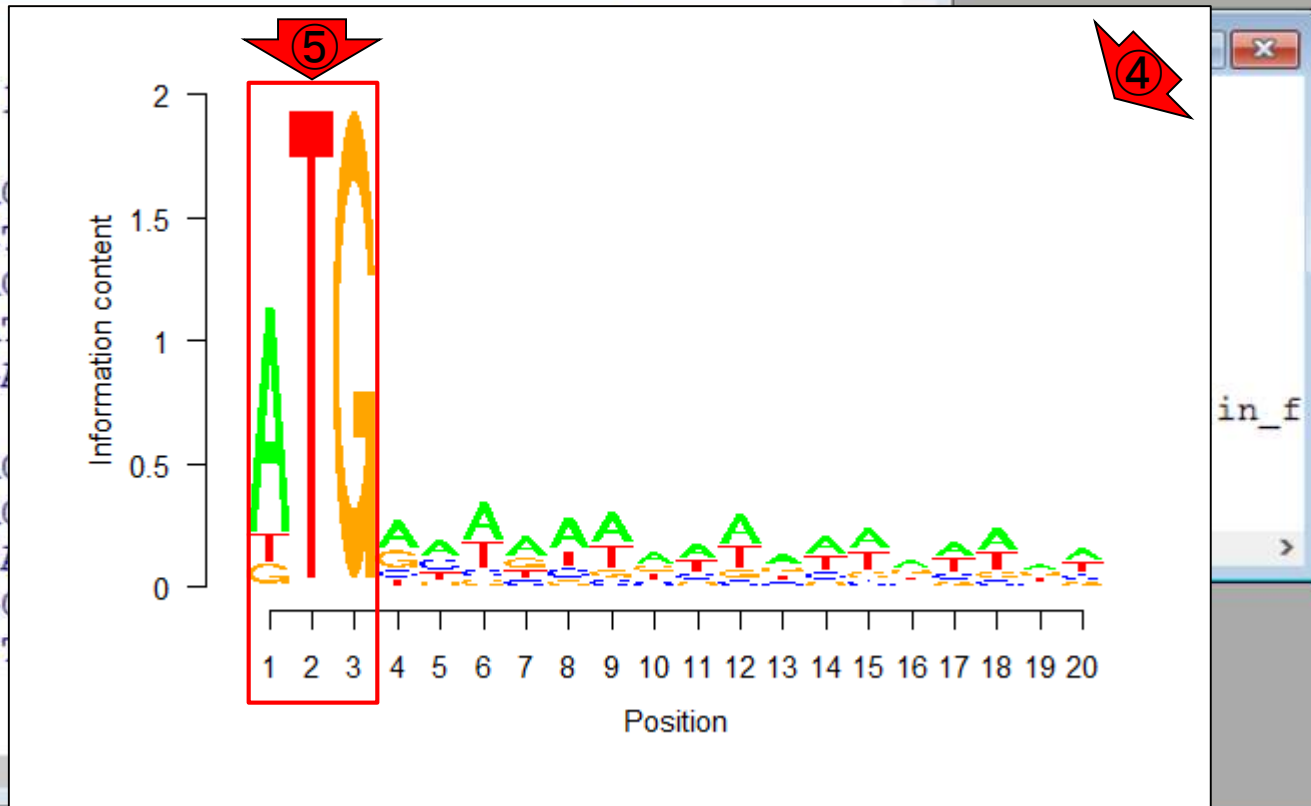
RGui (64-bit)

ファイル 編集 パッケージ ウィンドウ 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
```



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trand情報付きのリストファイルの作成、例題6のコード解説
 - スtrand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。

Sequence logos

RGui (64-bit)

ファイル 編集 パッケージ ウィンドウ ヘルプ

R Console

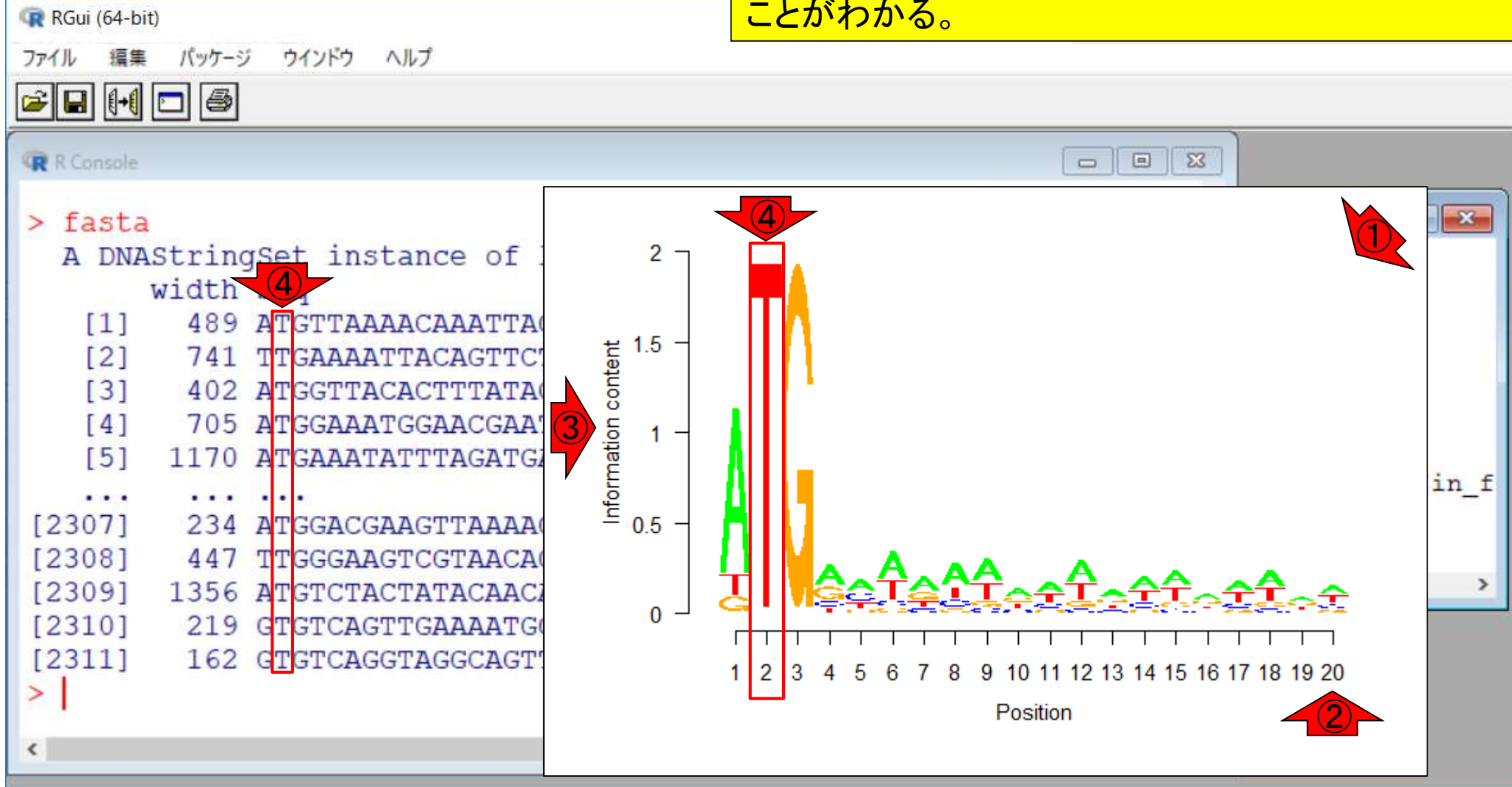
```
> fasta
A DNASTringSet instance of
width seq
[1] 489 ATGTTAAAACAAATTA
[2] 741 TTGAAAATTACAGTTC
[3] 402 ATGGTTACACTTTATA
[4] 705 ATGGAAATGGAACGAA
[5] 1170 ATGAAATATTTAGATG
...
[2307] 234 ATGGACGAAGTTAAAA
[2308] 447 TTGGGAAGTCGTAACA
[2309] 1356 ATGTCTACTATACAAC
[2310] 219 GTGTCAGTTGAAAATG
[2311] 162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

Information content

Position

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。



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。

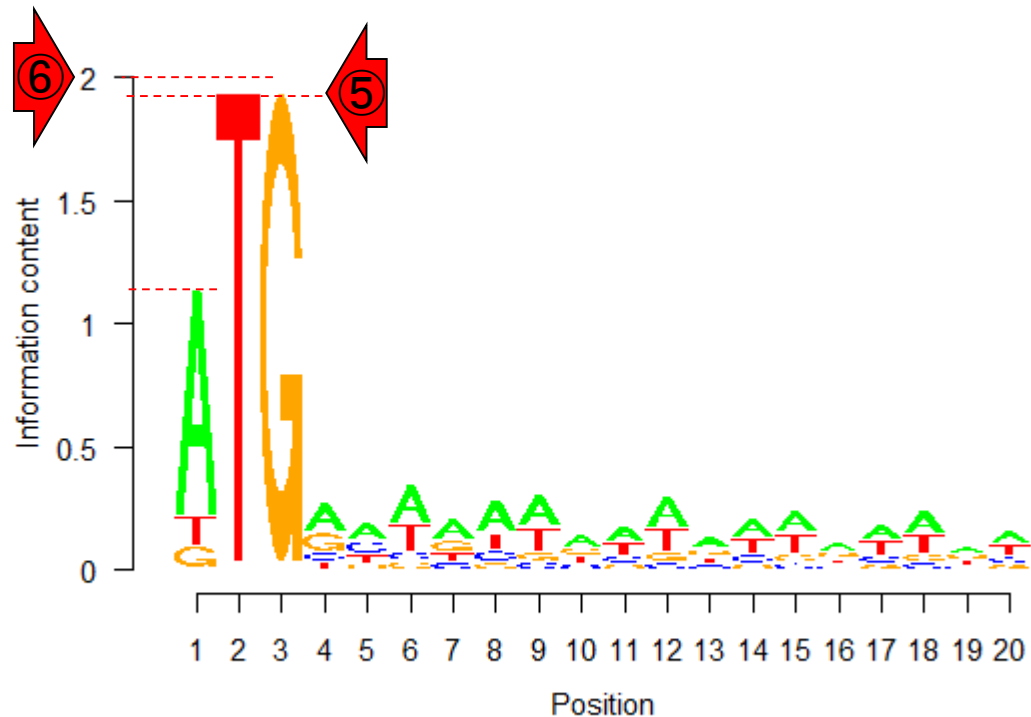
RGui (64-bit)

ファイル 編集 パッケージ ウィンドウ ヘルプ



R Console

```
> fasta
A DNASTringSet instance of
  width 1
 [1] 489 ATGTTAAAACAAATTAC
 [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
> |
```

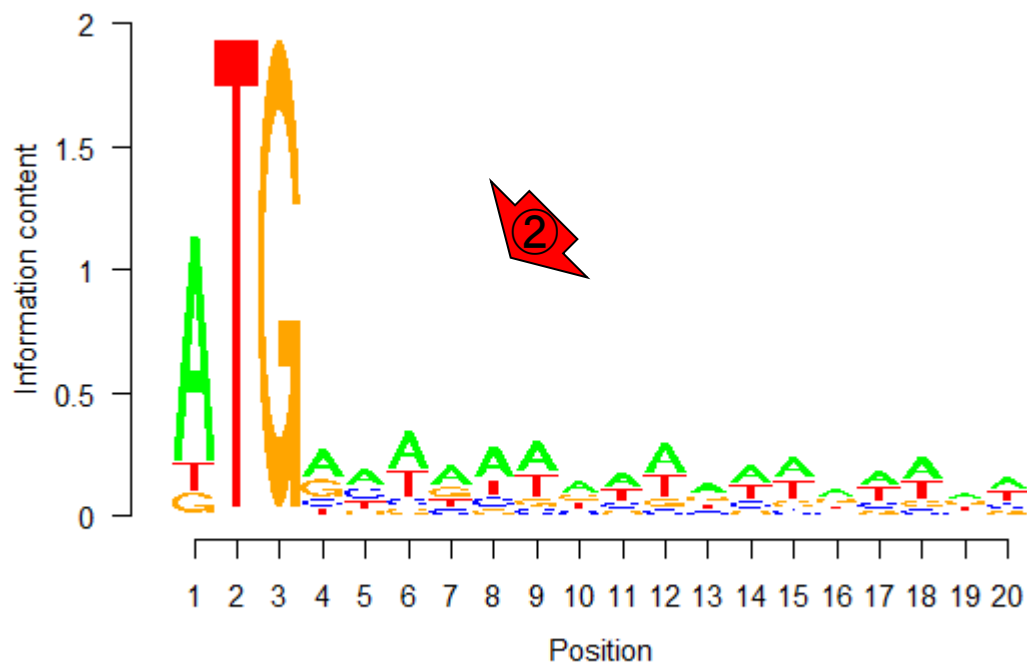
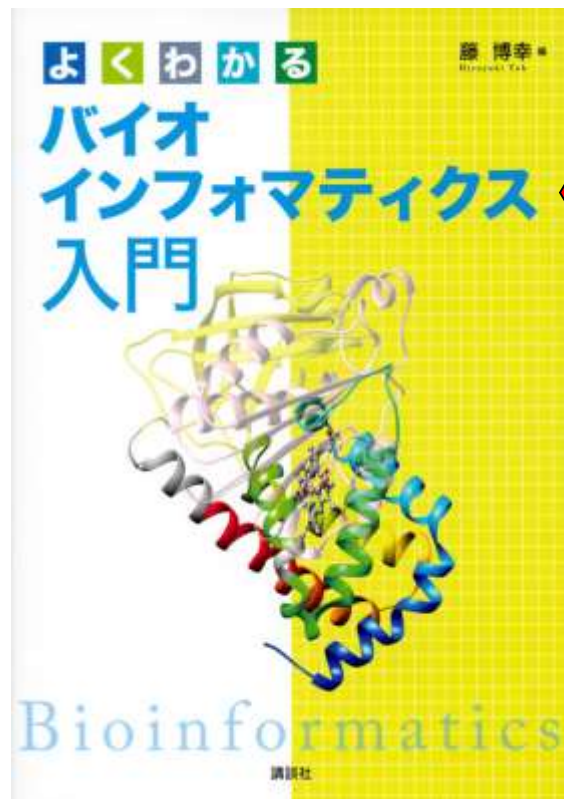


in_f

>

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…

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…

Sequence logos

(Rで)塩基配列解析

(last modified 2019/04/15, since 2010)

このウ
Macin
います
2018
(2018

What

・「ナ
加し
・「ナ
加し
・削除
・削除
「

- 解析 | 一般 | アラインメント | ペアワイス | 応用 | [Biostrings](#) (last modified 2016/12/29)
- 解析 | 一般 | アラインメント | ペアワイス | ドットプロット | [seqinr\(Charif_2005\)](#) (last modified 2019/04/15) **NEW**
- [解析 | 一般 | アラインメント | マルチプル | について](#) (last modified 2019/04/05) **NEW**
- 解析 | 一般 | アラインメント | マルチプル | [DECIPHER\(Wright_2015\)](#) (last modified 2016/12/29)
- 解析 | 一般 | アラインメント | マルチプル | [msa\(Bodenhofer_2015\)](#) (last modified 2016/12/29)
- 解析 | 一般 | [Silhouette scores\(シルエットスコア\)](#) (last modified 2019/02/28)
- 解析 | 一般 | [パターンマッチング](#) (last modified 2013/06/19)
- 解析 | 一般 | [GC含量\(GC contents\)](#) (last modified 2015/09/12)
- [解析 | 一般 | Sequence logos | について](#) (last modified 2018/06/29)
- 解析 | 一般 | Sequence logos | [seqLogo](#) (last modified 2019/04/21) **NEW**
- 解析 | 一般 | Sequence logos | [ggseqlogo\(Wagih_2017\)](#) (last modified 2018/06/29)
- 解析 | 一般 | 上流配列解析 | [LDSS\(Yamamoto_2007\)](#) (last modified 2015/02/19)
- 解析 | 一般 | 上流配列解析 | [Relative Appearance Ratio\(Yamamoto_2011\)](#) (last modified 2012/07/17) [トップページへ](#)
- 解析 | 基礎 | k-mer | ゲノムサイズ推定(基礎) | [arac](#) (last modified 2016/01/06)

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。

Sequence logos

④ 解析 | 一般 | Sequence logos | seqLogo NEW

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

1. 入力ファイルがmulti-FASTA形式の場合 :

```
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 <- readDNASTringSet(in_f)

#本番(sequence logoを実行)
hoge <- consensusMatrix(fasta, as.prob=T, baseOnly=T)
out <- makePWM(hoge[1:4, ], param_length=20, param_fig=c(600, 400))
seqLogo(out)
```

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"
out_f <- "hoge11.png"
param_length <- 20
param_fig <- c(600, 400)

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

#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

#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, ], param_length=20, param_fig=c(600, 400))#hogeはACGT以外の塩基(例えばN)のprobability: トップページへ
```

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。

④ 解析 | 一般 | Sequence logos | seqLogo NEW

seqLogoパッケージを用いて
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-FASTA形式の場合 :

```
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 <- readDNASTringSet(in_f)

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

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"
out_f <- "hoge11.png"
param_length <- 20
param_fig <- c(600, 100)
```

#入力ファイル名を指定してin_fに格納
#出力ファイル名を指定してout_fに格納
#最初の塩基数を指定
#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)
library(seqLogo)
```

#パッケージの読み込み
#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probability: 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trand情報付きのリストファイルの作成、例題6のコード解説
 - スtrand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。

(Rで)塩基配列解析

(last modified 2019/04/15, since 2010)

このウェブページ
[Macintosh2018.](#)
います。初心者
2018年7月に(R
(2018/07/18)

What's new? (

- 「カウント情報
加しました。(
- 「カウント情報
加しました。(
- 削除予定として
- 削除予定として

- イントロ | 一般 | 配列取得 | プロモーター配列 | [公共DBから](#) (last modified 2017/04/11)
- イントロ | 一般 | 配列取得 | プロモーター配列 | [BSgenomeとTxDbから](#) (last modified 2015/02/20)
- イントロ | 一般 | 配列取得 | プロモーター配列 | [GenomicFeatures\(Lawrence 2013\)](#) (last modified 2017/06/23)
- イントロ | 一般 | 配列取得 | トランスクリプトーム配列 | [公共DBから](#) (last modified 2015/05/09)
- イントロ | 一般 | 配列取得 | トランスクリプトーム配列 | [GenomicFeatures\(Lawrence 2013\)](#) (last modified 2016/02/10)
- イントロ | 一般 | 配列取得 | トランスクリプトーム配列 | [biomaRt\(Durink 2009\)](#) (last modified 2015/02/20)
- イントロ | 一般 | 配列取得 | CDS | [GenomicFeatures\(Lawrence 2013\)](#) (last modified 2019/04/19) **NEW**
- イントロ | 一般 | ExpressionSet | 1から作成 | [Biobase](#) (last modified 2018/08/01)
- イントロ | 一般 | ExpressionSet | 1から作成 | [NOISeq\(Tarazona 2015\)](#) (last modified 2018/08/02)
- イントロ | NGS | [様々なプラットフォーム](#) (last modified 2016/03/24)
- イントロ | NGS | [qPCRやmicroarrayなどとの比較](#) (last modified 2014/11/12)
- イントロ | NGS | [可視化\(ゲノムブラウザやViewer\)](#) (last modified 2016/12/22)
- イントロ | NGS | 配列取得 | FASTQ or SRA | [公共DBから](#) (last modified 2019/02/01)
- イントロ | NGS | 配列取得 | FASTQ or SRA | [SRADB\(Zhu 2013\)](#) (last modified 2019/02/01)
- [イントロ](#) | NGS | 配列取得 | [シミュレーションデータ](#) | [について](#) (last modified 2015/01/18) [トップページへ](#)
- イントロ | NGS | 配列取得 | [シミュレーションデータ](#) | [ランダムな塩基配列の生成から](#) (last modified 2015/01/18)

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

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ー。

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](#) ③。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_f1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
```

[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ファイルの読み込みは、⑤で行っています。

(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#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ー。

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.GCA_000309565.2.25.chromosome.Chromosome.gff3。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" #GFF3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
```

[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ファイルの読み込みは、⑤で行っています。赤枠内をコピー実行して、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ー。

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. ③。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" ④ 入力ファイル名を指定して
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.chromosome.Chromosome.gff3" ④ 入力ファイル名を指定してin
out_f <- "hoge1.fasta" ③ #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 ⑤
```

#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

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ファイルの読み込みは、⑤で行っています。赤枠内をコピー実行して、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(タンパク質)の取得方法について解説します。
「ファイル」 - 「ディレクトリの変更」でファイルを開く

1. GFF3形式のアノテーションファイルとFASTA形式の配列ファイル

GFF3形式ファイル(Lactobacillus casei 12a.GCA_000309565.2.25.gff3) (Lactobacillus casei 12a.GCA_000309565.2.25.gff3) (Zerbino et al., Nucleic Acids Res., 2018)から提供

```
in_f1 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.gff3"
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.fasta"
out_f <- "hoge1.fasta" #出力ファイル名
```

#必要なパッケージをロード

```
library(Rsamtools) #パッケージロード
library(GenomicFeatures) #パッケージロード
library(Biostrings) #パッケージロード
```

#入力ファイルの読み込み

```
txdb <- makeTxDbFromGFF(in_f2, format="auto")
txdb #確認
```

#前例(欲しい領域の座標情報取得)

```
hoge <- cds(txdb) #指定領域の座標情報取得
hoge #確認
```

#本番(配列取得)

```
fasta <- getSeq(FaFile(in_f1), hoge) #配列取得
fasta #確認
```

RGU (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ 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。

(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(タンパク質)の位置を「ファイル」-「ディレクトリの変更」でファイル名を指定して取得する。

1. GFF3形式のアノテーションファイルとFASTA形式の配列ファイル

GFF3形式ファイル(Lactobacillus casei 12a.GCA_000309565.2.25.gff3)とFASTA形式の配列ファイル(Lactobacillus casei 12a.GCA_000309565.2.25.fasta) (Zerbino et al., Nucleic Acids Res., 2018)から提供

```
in_f1 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.fasta"
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.gff3"
out_f <- "hoge1.fasta" #出力ファイル名
```

#必要なパッケージをロード

```
library(Rsamtools) #パッケージロード
library(GenomicFeatures) #パッケージロード
library(Biostrings) #パッケージロード
```

#入力ファイルの読み込み

```
txdb <- makeTxDbFromGFF(in_f2, format="auto") #確認
txdb
```

#前処理(欲しい領域の座標情報取得)

```
hoge <- cds(txdb) #指定領域の座標情報取得
hoge
```

#本番(配列取得)

```
fasta <- getSeq(FaFile(in_f1), hoge) #自己生成のFASTA形式の配列取得
fasta
```

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ 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。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(タンパク質)の配列取得
「ファイル」 - 「ディレクトリの変更」でファイルを選択

1. GFF3形式のアノテーションファイルとFASTA形式の配列ファイル

GFF3形式ファイル(Lactobacillus casei 12a.GCA_000309565.2.25.1.gff3) (Lactobacillus casei 12a.GCA_000309565.2.25.1.gff3) (Zerbino et al., Nucleic Acids Res., 2018)から提供

```
in_f1 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.1.gff3"
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.1.gff3"
out_f <- "hoge1.fasta" #出力ファイル名
```

#必要なパッケージをロード

```
library(Rsamtools) #パッケージロード
library(GenomicFeatures) #パッケージロード
library(Biostrings) #パッケージロード
```

#入力ファイルの読み込み

```
txdb <- makeTxDbFromGFF(in_f2, format="auto") #確認
txdb
```

#前処理(欲しい領域の座標情報取得)

```
hoge <- cds(txdb) #指定領域の座標情報取得
hoge
```

#本番(配列取得)

```
fasta <- getSeq(FaFile(in_f1), hoge) #自己生成のFASTA形式の配列取得
fasta
```

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 #5
# exon_nrow: 2800
# cds_nrow: 2681 #4
# Db created by: GenomicFeatures package from Bioconductor
# Creation time: 2019-04-22 13:31:19 +0900 (Mon, 22 Apr 2019)
# 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_0617
4	Chromosome	ena	gene	1523	2662	.	+	ID=gene:LCA12A_0618
5	Chromosome	ena	gene	3240	3452	.	+	ID=gene:LCA12A_0619
6	Chromosome	ena	gene	3449	4564	.	+	ID=gene:LCA12A_0620
7	Chromosome	ena	gene	4817	6778	.	+	ID=gene:LCA12A_0621
8	Chromosome	ena	gene	6840	9461	.	+	ID=gene:LCA12A_0622
9	Chromosome	ena	gene	9566	10270	.	-	ID=gene:LCA12A_0623
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_0617
4	Chromosome	ena	gene	1523	2662	.	+	ID=gene:LCA12A_0618
5	Chromosome	ena	gene	3240	3452	.	+	ID=gene:LCA12A_0619
6	Chromosome	ena	gene	3449	4564	.	+	ID=gene:LCA12A_0620
7	Chromosome	ena	gene	4817	6778	.	+	ID=gene:LCA12A_0621
8	Chromosome	ena	gene	6840	9461	.	+	ID=gene:LCA12A_0622
9	Chromosome	ena	gene	9566	10270	.	-	ID=gene:LCA12A_0623
10	Chromosome	ensembl	CDS	1	1350	.	+	ID=CDS:EKP96483;Parent=EK
11	Chromosome	ensembl	exon					Name=EKP96483-1;Parent=EK
12	Chromosome	ensembl	CDS	1523	2662	.	+	ID=CDS:EKP96484;Parent=EK
13	Chromosome	ensembl	exon	1523	2662	.	+	Name=EKP96484-1;Parent=EK

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。同様に、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。

	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_0617
4	Chromosome	ena	gene	1523	2662	. + . ID=gene:LCA12A_0618
5	Chromosome	ena	gene	3240	3452	. + . ID=gene:LCA12A_0619
6	Chromosome	ena	gene	3449	4564	. + . ID=gene:LCA12A_0620
7	Chromosome	ena	gene	4817	6778	. + . ID=gene:LCA12A_0621
8	Chromosome	ena	gene	6840	9461	. + . ID=gene:LCA12A_0622
9	Chromosome	ena	gene	9566	10270	. - . ID=gene:LCA12A_0623
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

```
> txdb
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
> |
```

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trand情報付きのリストファイルの作成、例題6のコード解説
 - スtrand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…

The screenshot shows the RGui interface with a GFF file open in the editor and the R Console displaying the output of the `txdb` function. The GFF file content is as follows:

A	B	C	D
##gff-version 3			
##sequence-region	Chromosome 1	2907892	
Chromosome	ena	gene	
Chromosome	ena	gene	152
Chromosome	ena	gene	324
Chromosome	ena	gene	344
Chromosome	ena	gene	481
Chromosome	ena	gene	684
Chromosome	ena	gene	956
Chromosome	ensembl	CDS	
Chromosome	ensembl	exon	
Chromosome	ensembl	CDS	152
Chromosome	ensembl	exon	152

The R Console output is as follows:

```
> txdb
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 Bioconductor
# Creation time: 2019-04-22 13:31:19 +0900 (Mon, 22 Apr 2019)
# GenomicFeatures version at creation time: 1.34.1
# RSQLite version at creation time: 2.1.1
# DBSCHEMAVERSION: 1.2
```

Red arrows and a red circle with the number 1 highlight the 'CDS' entries in the GFF file and the 'cds_nrow: 2681' line in the R Console output. A red note '#確認してるだ\$' is also present in the console.

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。ここをコピー実行。

(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ー。

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
```

[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。ここをコピー実行。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

イントロ | 一般 | 配列取得 | CDS

GenomicFeaturesパッケージを主に用いてCDS(タンパク質)の座標情報を取得する。[ファイル] - [ディレクトリの変更]でファイルを開く。

1. GFF3形式のアノテーションファイルとFASTA形式の配列ファイル

GFF3形式ファイル(Lactobacillus casei 12a.GCA_000309565.2.25.1.gff3) (Zerbino et al., Nucleic Acids Res., 2018)から提供

```
in_f1 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.1.gff3"
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.1.fasta"
out_f <- "hoge1.fasta" #出力ファイル名
```

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

```
#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="auto") #確認
```

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

```
#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge) #自己確認
fasta #確認
```

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

イントロ | 一般 | 配列取得 | CDS

GenomicFeaturesパッケージを主に用いてCDS(タンパク質コード領域)の取得
「ファイル」 - 「ディレクトリの変更」でファイルを変更

1. GFF3形式のアノテーションファイルとFASTA形式の配列取得

GFF3形式ファイル(Lactobacillus casei 12a.GC000309565.2.20180101.gff) (Zerbino et al., Nucleic Acids Res., 2018)から提供

```
in_f1 <- "Lactobacillus_casei_12a.GCA_000309565.2.20180101.gff"
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.20180101.fasta"
out_f <- "hoge1.fasta" #出力ファイル名

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

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

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

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

RGui (64-bit)

ファイル 編集

```
R Console
> hoge
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(タンパク質コード領域)の取得。「ファイル」 - 「ディレクトリの変更」でファイルを選択

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_000309565.2.25.1.gff3) (Zerbino et al., Nucleic Acids Res., 2018)から提供

```
in_f1 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.1.gff3"
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.1.fasta"
out_f <- "hoge1.fasta" #出力ファイル名
```

#必要なパッケージをロード

```
library(Rsamtools) #パッケージロード
library(GenomicFeatures) #パッケージロード
library(Biostrings) #パッケージロード
```

#入力ファイルの読み込み

```
txdb <- makeTxDbFromGFF(in_f2, format="auto")
txdb #確認
```

#前処理(欲しい領域の座標情報取得)

```
hoge <- cds(txdb) #指定領域の取得
hoge #確認
```

#本番(配列取得)

```
fasta <- getSeq(FaFile(in_f1), hoge) #自己定義関数の実行
fasta #確認
```

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。

The screenshot shows a spreadsheet with the following data:

	A	B	C	D	E	F
1	##gff-version 3					
2	##sequence-region	Chromosome 1	2907892			
3	Chromosome	ena	gene	1	1350	+
4	Chromosome	ena	gene	1523	2662	+
5	Chromosome	ena	gene	3240	3452	+
6	Chromosome	ena	gene	3449	4564	+
7	Chromosome	ena	gene	4817	6778	+
8	Chromosome	ena	gene	6840	9461	+
9	Chromosome	ena	gene	9566	10270	-
10	Chromosome	ensembl	CDS	1	1350	+
11	Chromosome	ensembl	exon	1	1350	+
12	Chromosome	ensembl	CDS	1523	2662	+
13	Chromosome	ensembl	exon	1523	2662	+

```
#確認してるか？
strand 1 metadata column:
genes 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$
> |
```

コード全体を最後までコピー実行すると、こんな感じになります。

CDS座標情報取得

イントロ | 一般 | 配列取得 | CDS

GenomicFeaturesパッケージを主に用いてCDS(タンパク質コード領域)の座標情報を取得する。 「ファイル」 - 「ディレクトリの変更」でファイルの場所を変更する。

1. GFF3形式のアノテーションファイルとFASTA形式の配列ファイル

GFF3形式ファイル(Lactobacillus casei 12a.GCA_000309565.2.25.1.gff3) (Zerbino et al., Nucleic Acids Res., 2018)から提供

```
in_f1 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.1.gff3"
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.1.gff3"
out_f <- "hoge1.fasta" #出力ファイル名

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

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

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

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

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

```
R Console
> fasta #確認してるだ$
A DNAStringSet instance of length 2681
      width seq
[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 TTGTTTGGC...GTCATCTGA Chromosome_290614...
[2681] 141 ATGACAACC...TCTGCCTAA Chromosome_290683...
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", width=width(fasta))
> |
```

CDS座標情報取得

コード全体を最後までコピー実行すると、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(タンパク質)の座標情報を取得する方法を説明します。

②「ファイル」 - 「ディレクトリの変更」でファイルの場所を変更します。

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_000309565.2.25.01.gff) (Zerbino et al., Nucleic Acids Res., 2018)から提供

```
in_f1 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.01.gff"
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.01.gff"
out_f <- "hoge1.fasta" #出力ファイル名
```

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

```
#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="auto")
txdb #確認
```

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

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

RGui (64-bit)

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

R Console

```
> fasta #確認してるだ$
A DNAString instance of length 2681
      width seq
[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 TTGTTTTCG...GTCATCTGA Chromosome_290614...
[2681] 141 ATGACAACC...TCTGCCTAA Chromosome_290683...
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", width=width(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trand情報付きのリストファイルの作成、例題6のコード解説
 - スtrand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です。①を押して行って、例題2までページ下部に移動。

失敗例は例題2

The screenshot shows a web browser window with the URL `www.iu.a.u-tokyo.ac.jp/~kadota/r_seq.html#intro_general_getseq_CDS...`. The page title is "イントロ | 一般 | 配列取得 | CDS | GenomicFeatures(Lawrence_2013) NEW". The main content includes a paragraph about the `GenomicFeatures` package and a section titled "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:". Below this, there is R code for loading packages, reading a GFF3 file, and using `getSeq` to retrieve sequences. A keyboard image is overlaid on the right side of the code, with a red arrow pointing to the `PgDn` key, labeled with a circled "1".

```
in_f1 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.dna.chromosome1.fasta"
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.chromosome1.gff3"
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オブジェクトを作成
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
```

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

失敗例は例題2

(Rで塩基配列解析

x +

- □ ×

② 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...

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のところですか。gffをgff3に変えとか、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を押して行って、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…

(Rで)塩基配列解析

②

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)** **④**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のところでも、gffをgff3に変えろとか、fnaをfaにしろ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。

③
`in_f1 <- "genome.fna"`
`in_f2 <- "annotation.gff"`
`out_f <- "hoge2.fasta"`

#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)
#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)
#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を押して行って、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



txdbの作成まで

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

(Rで)塩基配列解析

①

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のところですか。gffをgff3に変えとか、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)
```

②

③

④

[トップページへ](#)

txdbの作成まで

①例題2は、②GFFファイルの読み込みのところがうまくいっていないことが分かっています。③赤枠内をコピー実行して、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.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形式ファイルです。エラーは出ていないものの、GFFファイルを変えると、fnaをfaにするとかそういう問題で代

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

```
#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="auto")
txdb
```

```
#前外(欲しい領域の座標情報取得)
hoge <- cds(txdb)
hoge
```

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

```
#後処理(description部分を変更)
```

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ 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ファイルの読み込みのところがうまくいっていないことが分かっています。③赤枠内をコピー実行して、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形式ファイルです。エラーは出ていないものの、GFFファイルを変えると、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"

#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="auto")
txdb

#前外に欲しい領域の座標情報取得
hoge <- cds(txdb)
hoge

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

#後処理(description部分を変更)
```

RGui (64-bit)

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

R Console

```
> txdb
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ファイルの読み込みのところがうまくいっていないことが分かっています。③赤枠内をコピー実行して、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個となってしまうのが厄介なところ。

(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形式ファイルです。エラーは出ていないものの、GFFファイルを変えると、fnaをfaにするとかそういう問題では

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

#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="auto")
txdb
```

```
#前外に欲しい領域の座標情報取得
hoge <- cds(txdb)
hoge
```

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

```
#後処理(description部分を変更)
```

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
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$
```

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.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形式ファイルです。エラーは出ていないものの、GFFファイルを変えると、fnaをfaにするとかそういう問題では

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

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

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

#前処理(欲しい領域の座標情報取得)
hoge <- cds(txdb) #指定した範囲
hoge #確認

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

#後処理(description部分を変更)
```



RGui (64-bit)



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) #指定した範囲
> 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$

> |
```



Red box highlighting the R console output from the `hoge <- cds(txdb)` command, showing that the number of CDS ranges is 0.

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

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形式ファイルです。エラーは出ていないものの、GFFファイルを変えると、fnaをfaにするとかそういう問題では

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

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

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

#前処理(欲しい領域の座標情報取得)
hoge <- cds(txdb) #指定した範囲
hoge #確認

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

#後処理(description部分を変更)
```

RGU (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)
> hoge
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$

> |
```

CDS座標情報取得

当然最後までコピー実行しても、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形式ファイルです。エラーは出ていないものの、GFFファイルを変えると、fnaをfaにするとかそういう問題では

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

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

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

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

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

#後処理(description部分を変更)
```



RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ 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trand情報付きのリストファイルの作成、例題6のコード解説
 - スtrand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。

(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...

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のところですか。gffをgff3に変えとか、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](#)

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。

DFAST - Job Result

https://dfast.nig.ac.jp/analysis/annotation/cf3db3d9-65d1-4494-8

DFAST

③

analysis

Archive

DFAST-core

API

Remember the current URL to access this page. The result will be deleted 30 days after

Delete this job now. => Delete This procedure cannot be undone.

Title :

JobID : cf3db3d9-65d1-4494-80ce-ddfe125dde16

Status : COMPLETE

[2019-04-17 16:42:49.372511] Job submitted.
[2019-04-17 16:42:49.402296] Job started.
[2019-04-17 16:44:32.417969] Job completed.

Result

Features

DDBJ Submission

Log

Genome Statistics

Total Length (bp)	2,348,706
No. of Sequences	61
GC Content (%)	38.1%
N50	92,304
Gap Ratio (%)	0.0%
No. of CDSs	2,311

③

Download Files

Genbank Flat File : [annotation.gbk](#)

GFF3-formatted ... : [annotation.gff](#)

Genome Fasta F... : [genome.fna](#)

Protein Fasta File : [protein.faa](#)

CDS Fasta File : [cds.fna](#)

RNA Fasta File : [rna.fna](#)

Feature Table : [features.tsv](#)

Genome Statisti... : [statistics.txt](#)

Zip Archive : [annotation.zip](#)

②

④

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…

annotation.gff - Excel

ファイル ホーム 挿入 ページレイアウト 数式 データ 校閲 表示 ヘルプ 実行したい作業を入力してください 共有

A1 : X ✓ fx ##gff-version 3

	A	B	C	D	E	F	G	H	I
1	##gff-version 3								
2	sequence01	Barnap:0	rRNA	70	182	.	+	.	ID=LOCUS_r00010;product=5S ribosomal RNA;inference=
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。

annotation.gff

ff-version 3

	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;in
23	sequence01	ab initio p	CDS	2637	3377	.	+	0	ID=LOCUS_00020;product=MBL fold metallo-hy
24	sequence01	ab initio p	CDS	3685	4086	.	+	0	ID=LOCUS_00030;product=regulatory protein Sp

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

なぜ読み込めない?!

(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形式ファイルです。エラーは出ていないものの、GFFファイルを変えると、fnaをfaにするとかそういう問題で代

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

#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="auto")
txdb

#前外に欲しい領域の座標情報取得
hoge <- cds(txdb)
hoge

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

#後処理(description部分を変更)
```

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ 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
> |
```

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trand情報付きのリストファイルの作成、例題6のコード解説
 - スtrand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。

(Rで)塩基配列解析

(last modified 2019/04/15, since 2010)

このウェブページのR関連部分は、[インストール \(Macintosh2018.11.27版\)](#)に従ってフリーにあります。初心者の方は[基本的な利用法](#)([Wiki](#))
2018年7月に(Rで)塩基配列解析の一部 (2018/07/18)

What's new? (過去のお知らせはこちら)

- 「カウント情報取得 | シミュレーション」を追加しました。(2019/04/11) **NEW**
- 「カウント情報取得 | シミュレーション」を追加しました。(2019/04/11) **NEW**
- 削除予定としていた「インストール | R」
- 削除予定としていた「インストール | R」

(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

- [基本的な利用法](#) (last modified 2019/03/12)
- [サンプルデータ](#) (last modified 2018/06/09)
- イントロ | 一般 | [ランダムに行を抽出](#) (last modified 2014/07/17)
- イントロ | 一般 | [任意の文字列を行の最初に挿入](#) (last modified 2014/07/17)
- イントロ | 一般 | [任意のキーワードを含む行を抽出\(基礎\)](#) (last modified 2016/04/20)
- イントロ | 一般 | [ランダムな塩基配列を生成](#) (last modified 2014/06/16)
- イントロ | 一般 | [任意の長さの可能な全ての塩基配列を作成](#) (last modified 2015/02/19)
- イントロ | 一般 | [任意の位置の塩基を置換](#) (last modified 2013/09/12)
- イントロ | 一般 | [指定した範囲の配列を取得](#) (last modified 2015/04/06)
- イントロ | 一般 | [指定したID\(染色体やdescription\)の配列を取得](#) (last modified 2014/03/10)
- イントロ | 一般 | [翻訳配列\(translate\)を取得\(基礎\) | Biostrings](#) (last modified 2015/09/12)
- イントロ | 一般 | [翻訳配列\(translate\)を取得\(応用\) | seqinr\(Charif_2005\)](#) (last modified 2015/03/09)
- イントロ | 一般 | [相補鎖\(complement\)を取得](#) (last modified 2019/03/10)
- イントロ | 一般 | [逆相補鎖\(reverse complement\)を取得](#) (last modified 2019/03/10)
- イントロ | 一般 | [逆鎖\(reverse\)を取得](#) (last modified 2019/03/10)
- イントロ | 一般 | [k-mer解析 | k=1\(塩基ごとの出現頻度解析\) | Biostrings](#) (last modified 2016/04/27)

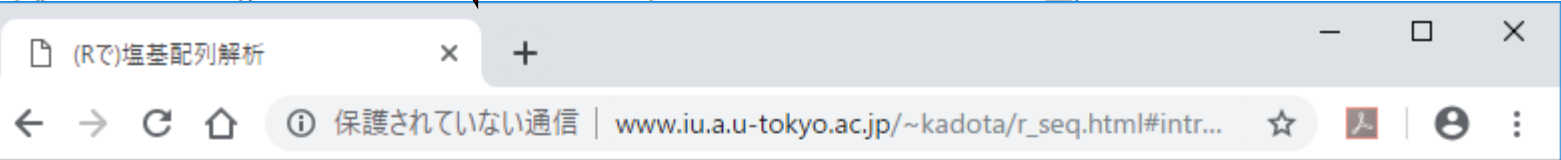
[トップページへ](#)

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。



single-FASTA形式やmulti-FASTA形式は、「この染色体の、ここから、ここまでの範囲の配列のみ抽出」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。



```
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ファイルを取得するやり方です。

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

```
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
```

[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す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。

イントロ | 一般 | 指定した範囲の配列を取得

single-FASTA形式やmulti-FASTA形式は、「この染色体の、ここから、ここまでの配列のみ抽出」というようにロード時に、*.fastaという拡張子で「ファイル」-「ディレクトリ」のパスを指定する。

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
```

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ファイルを取得するやり方です。

```
in_f1 <- "ref_genome.fa"
in_f2 <- "list_sub3.txt"
out_f <- "hoge5.fasta"
```

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

```
#入力ファイルの読み込み
```

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

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。

The screenshot shows a spreadsheet with the following data:

	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=
21	sequence01	Aragorn:1 tRNA		1827	1902	.	+	.	ID=LOCUS_t00190;product=
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=

The resulting list file (step 5) contains the following data:

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。

The screenshot shows a spreadsheet with columns A through I. Red callouts are placed as follows:

- ①: Points to the filename 'annotation.gff' in the top right.
- ②: Points to the 'CDS' text in column C of row 22.
- ③: Points to the 'sequence01' text in column A of row 22.
- ④: Points to the start and end coordinates (2054, 2542) in columns D and E of row 22.
- ⑤: Points to the 'sequence01' text in column A of the first row of the inset table.
- ⑥: Points to the '-' sign in column F of row 22.

	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=
21	sequence01	Aragorn:1 tRNA		1827	1902	.	+	.	ID=LOCUS_t00190;product=
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=

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trand情報付きのリストファイルの作成、例題6のコード解説
 - スtrand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	TTTAAGGACCCTACACATTTGATTTGTCGAACTTTGTTTCAGTTTTCAAAGGTCTACTTTGTCGCTATAI								
2379	TTAGTGCTTCTAATTGTAAAATTGCTTTTTTTGTCATAATTTAAGTCCTCCTAGATTTGTTTTTCCTTATI								
2380	TCCTGATTGTATTAATCTTTGCCTGTAATCAAATGCGCCAAACGTCACTAGTTTTTCATTACCGCTTATGC								

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	.
2374	##FASTA					
2375	>sequence01					
2376	ATCGATATTATTTAGTTTTGAGAGCGCAAGTTCTCATCATAGTTATAGCTATGAATAAGCACTATAGTG					
2377	>sequence02					
2378	TTAAGGACCCTACACATTTGATTTGTCGAACTTTGTTTCAGTTTTCAAAGGTCTACTTTGTCGCTATAT					
2379	TTAGTGCTTCTAATTGTAAAATTGCTTTTTTTGTCATAATTTAAGTCCTCCTAGATTTGTTTTTCCTTATT					
2380	TCCTGATTGTATTAATCTTTGCCTGTAATCAAATGCGCCAAACGTCAGTTTTCATTACCGCTTATGC					

CDSを含む行の抽出

(Rで)塩基配列解析

(last modified 2019/04/15, since 2010)

このウェブページのR関連部分は、[インストール \(Macintosh2018.11.27版\)](#)に従ってフリー
 います。初心者の方は[基本的な利用法](#)([Wiki](#)
 2018年7月に[\(Rで\)塩基配列解析の一部](#) (2018/07/18)

What's new? (過去のお知らせはこちら)

- 「カウント情報取得 | シミュレーション」に追加しました。(2019/04/11) **NEW**
- 「カウント情報取得 | シミュレーション」に追加しました。(2019/04/11) **NEW**
- 削除予定としていた「インストール | R」
- 削除予定としていた「インストール | R」

- [基本的な利用法](#) (last modified 2019/03/12)
- [サンプルデータ](#) (last modified 2018/06/09)
- イントロ | 一般 | [ランダムに行を抽出](#) (last modified 2014/07/17)
- イントロ | 一般 | [任意の文字列を行の最初に挿入](#) (last modified 2014/07/17)
- イントロ | 一般 | [任意のキーワードを含む行を抽出\(基礎\)](#) (last modified 2016/04/20)
- イントロ | 一般 | [ランダムな塩基配列を生成](#) (last modified 2014/06/16)
- イントロ | 一般 | [任意の長さの可能な全ての塩基配列を作成](#) (last modified 2015/02/19)
- イントロ | 一般 | [任意の位置の塩基を置換](#) (last modified 2013/09/12)
- イントロ | 一般 | [指定した範囲の配列を取得](#) (last modified 2015/04/06)
- イントロ | 一般 | [指定したID\(染色体やdescription\)の配列を取得](#) (last modified 2014/03/10)
- イントロ | 一般 | [翻訳配列\(translate\)を取得\(基礎\)](#) | [Biostrings](#) (last modified 2015/09/12)
- イントロ | 一般 | [翻訳配列\(translate\)を取得\(応用\)](#) | [seqinr\(Charif_2005\)](#) (last modified 2015/03/09)
- イントロ | 一般 | [相補鎖\(complement\)を取得](#) (last modified 2019/03/10)
- イントロ | 一般 | [逆相補鎖\(reverse complement\)を取得](#) (last modified 2019/03/10)
- イントロ | 一般 | [逆鎖\(reverse\)を取得](#) (last modified 2019/03/10)
- イントロ | 一般 | [k-mer解析 | k=1\(塩基ごとの出現頻度解析\)](#) | [Biostrings](#) (last modified 2016/04/27)

[トップページへ](#)

CDSを含む行の抽出

(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#int... ☆ 👤 ⋮

イントロ | 一般 | 任意のキーワードを含む行を抽出(基礎) ②

例えばタブ区切りテキストファイルのやり方を示します。Linux (UNIX)の「ファイル」-「ディレクトリ」の操作は、

1. 目的のタブ区切りテキストファイル (genelist1.txt)中のものが

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

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

#本番
```

(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... ☆ 👤 ⋮

15. GFF3形式ファイル(annotation.gff)に対して、"CDS"という文字列が含まれる行全体を出力したい場合 : 2019年5月13日の講義で利用したファイルです。

```
in_f <- "annotation.gff"
out_f <- "hoge15.txt"
param <- "CDS"

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

#本番(paramで指定した文字列と一致する行番号情報を得て、その行のみ出力)
hoge <- sapply(param, grep, x=data)
hoge <- unique(unlist(hoge))
out <- data[hoge]
length(out)

#ファイルに保存
writelines(out, out_f)
```

#入力ファイル名を指定してin_fに格納
#出力ファイル名を指定してout_fに格納
#検索したい文字列を指定

#in_fで指定したファイルの読み込み
#オブジェクトdataの要素数を表示

#paramで指定した文字列と一致する行番号情報を得ている
#得られるhogeベクトルは重複している可能性があるため重複を
#hogeで指定したもののみ抽出した結果をoutに格納(dataオブジェクトの要素数を表示)

#outの中身を指定したファイル名で保存

[トップページへ](#)

②で目的を達成できそう。③例題15。コピー実行後の状態。

CDSを含む行の抽出

The image shows a screenshot of the RGui (64-bit) interface. The R Console window displays the following code and its output:

```
> 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) #オブジェクト$
[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) #オブジェクト$
[1] 2311
>
> #ファイルに保存
> writeLines(out, out_f) #outの中身を$
> |
```

The R Script Editor window shows the following code:

```
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報を得$
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
```

A red circle with the number 3 is overlaid on the browser address bar.

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

CDSを含む行の抽出

The image shows a screenshot of the RGui (64-bit) interface. The main window displays R code for processing a GFF3 file. The R Console window shows the execution of the code, with the number of lines extracted (2311) highlighted by a red arrow labeled '4'.

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報)
hoge <- sapply(param, grep, x=data) #
hoge <- unique(unlist(hoge)) #
out <- data[hoge] #h
length(out) #オ

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

R Console Output:

```
> 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) #オブジェクト$
[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) #オブジェクト$
[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。

The screenshot shows the RGui interface with a script editor on the left and an R Console on the right. The script editor contains R code for reading a GFF file, filtering for CDS entries, and saving the results. The R Console shows the execution of this code, with the total number of lines (2496) and the number of CDS lines (2311) displayed. Red arrows with circled numbers 1 through 5 point to specific parts of the code and output.

```
in_f <- "annotation.gff"
out_f <- "hoge15.txt"
param <- "CDS"

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

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

#ファイルに保存
writeLines(out, out_f)
```

```
> param <- "CDS"
>
> #入力ファイルの読み込み
> data <- readLines(in_f)
> length(data)
[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)
> hoge <- unique(unlist(hoge))
> out <- data[hoge]
> length(out)
[1] 2311
>
> #ファイルに保存
> writeLines(out, out_f)
> |
```

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trand情報付きのリストファイルの作成、例題6のコード解説
 - スtrand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。

補足説明

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"
out_f <- "hoge15.txt"
param <- "CDS"

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

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

#ファイルに保存
writeLines(out, out_f)
```

R Console

```
> param <- "CDS"
>
> #入力ファイルの読み込み
> data <- readLines(in_f)
> length(data)
[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る)
> hoge <- sapply(param, grep, x=data)
> hoge <- unique(unlist(hoge))
> out <- data[hoge]
> length(out)
[1] 2311
>
> #ファイルに保存
> writeLines(out, out_f)
> |
```

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。

The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H	I
2541	TTAATCATATCCGCTGTGTTTCTCATGGTAGTCATTTGACCAATCTTACTAACCACACTTTTGAAATATC								
2542	>sequence58								
2543	ACCATTGTCAGTGATGATTAAAGTATACCGATCAATTCAAAAATCAACGTCCAAAAGCAATATAGTTT								
2544	>sequence59								
2545	CCAACCTATCTACCTCATCATCTCTGAGGGGTCTTACTCTTTCCGAAGAAAGATGGGAAATCTCATCTT								
2546	>sequence60								
2547	CGGGAGTATGCTTTTGCCATTGCCCCTATGGGGCTACTCGTTGCTGAATCAACAGTAATATTACCAATI								
2548	>sequence61								
2549	ATGGGGTCTTTCCGTCCTGTCGCGGGTAACCTGCATCTTCACAGGTACTTCAATTTACCGAGTCTCTC								
2550									
2551									
2552									

準備完了 平均: 5266.75 データの個数: 16 合計: 84268

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、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。

	A	B	C	D	E	F
2374	##FASTA					
2375	>sequence01					
2376	ATCGATATTATTAGTTTTGAGAGCGCAAGTTCTCATCATAGTTATAGCTATGAATAAGCACTATAGTG					
2377	>sequence02					
2378	TTAAGGACCTACACATTTGATTTGTCGAAACTTTGTTTCAGTTTCAAAGGTCTACTTTGTCGCTATA					
2379	TTAGTGCTTCTAATTGTAATAATTGCTTTTTTTGTCATAATTTAAGTCCTCCTAGATTTGTTTTTCCTTAT					
2380	TCCTGATTGTATTAATCTTTGCCTGTAATCAAATGCGCCAAACGTCACTAGTTTTTCATTACCGCTTATGC					
2381	TGCTGCTTGAAATTCCTTGTCGTTCAATAACATCTTTTCGATGATTAGTATAGACACCATTTCATGTGA					
2382	GTGTTTATTGTATTATACTGCATTTTAAACGTTAAAATCAGTAGATATTGCAGTATAGTGCAATATTATAA					
2383	>sequence03					
2384	CACCGTATCAGGTTATGCTGACAAATTTGTCCAAAATTCGGATTAAATTTGCAATCTACCATATACTA					
2385	AGCCGGAACCCTATTTTGAACATAAGTGCGGAAATTTAGCTAATCAGCAAAAATTCGCCGCGAAATTAC					

補足説明

Windows用の高機能エディタの1つである、①
EmEditor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。

```
C:\Users\kadota\Desktop\annotation.gff - EmEditor
ファイル(F) 編集(E) 検索(S) 表示(V) ツール(T) ウィンドウ(W) ヘルプ(H)
annotation.gff x
sequence46 ab initio prediction:MetaGeneAnnotator CDS 718 ^
sequence46 ab initio prediction:MetaGeneAnnotator CDS 1314
sequence48 ab initio prediction:MetaGeneAnnotator CDS 49
sequence49 ab initio prediction:MetaGeneAnnotator CDS 212
sequence50 ab initio prediction:MetaGeneAnnotator CDS 12
sequence56 ab initio prediction:MetaGeneAnnotator CDS 327
sequence59 ab initio prediction:MetaGeneAnnotator CDS 455
##FASTA↓
>sequence01↓
ATCGATATTATTTATTTTGAGAGCGCAAGTTCTCATCATAGTTATAGCTATGAATAAGCACTATAGT
>sequence02↓
TTTAAGGACCCTACACATTTGATTTGTCGAAACTTTGTTTCAGTTTCAAAGGTCTACTTTGTCGCTAT
>sequence03↓
CACCGTATCAGGTTATGCTGACAAATTTGTCCAAAAATTCGGATTAATTTGCAATCTACCATATACT
>sequence04↓
3.06 MB (3,218,315 バイト), 2,497 Text 2,374行, 8桁 日本語 (シフト JIS) 0文字 0/2,497 行
```


補足説明

Windows用の高機能エディタの1つである、①
EmEditor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…

C:\Users\kadota\Desktop\annotation.gff - EmEditor

ファイル(F) 編集(E) 検索(S) 表示(V) ツール(T) ウィンドウ(W)

annotation.gff x

```
sequence46      ab initio prediction:MetaGeneAnnotator  CDS      718 ^
sequence46      ab initio prediction:MetaGeneAnnotator  CDS     1314
sequence48      ab initio prediction:MetaGeneAnnotator  CDS      49
sequence49      ab initio prediction:MetaGeneAnnotator  CDS     212
sequence50      ab initio prediction:MetaGeneAnnotator  CDS      12
sequence56      ab initio prediction:MetaGeneAnnotator  CDS     327
sequence59      ab initio prediction:MetaGeneAnnotator  CDS     455
##FASTA↓
>sequence01↓
ATCGATATTATTTATTTTGAGAGCGCAAGTTCTCATCATAGTTATAGCTATGAATAAGCACTATAGT
>sequence02↓
TTTAAGGACCCTACACATTTGATTTGTCGAAACTTTGTTTCAGTTTCAAAGGTCTACTTTGTCGCTAT
>sequence03↓
CACCGTATCAGGTTATGCTGACAAATTTGTCCAAAAATTCGGATTAATTTGCAATCTACCATATACT
>sequence04↓
```

3.06 MB (3,218,315 バイト), 2,497 Text 2,374行, 8桁 日本語 (シフト JIS) 0文字 0/2,497 行

補足説明

Windows用の高機能エディタの1つである、① EmEditor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。

```
>sequence55↓
AAATAAATGTATACATATTCCTGAATAAGCAAGAATTTTCGTGCATTTATAAATACTATCGGAGGTGT
>sequence56↓
TTCTCACCTGTCTTTCGCTACTCATACCGGCATTCTCACTTCTAAGCGCTCCACCAGTCCTCACGGTC
>sequence57↓
TTAATCATATCCGCTGTGTTTCTCATGGTAGTCATTTGACCAATCTTACTAACCACACTTTTGAAATA
>sequence58↓
ACCATTGTCAGTGATGATTAAGTATACCGATCAATTCAAAAATCAACGTCCAAAAGCAATATAGTT
>sequence59↓
CCAACCTATCTACCTCATCATCTCTGAGGGGTCTTACTCTTTCGGAAGAAAGATGGGAAATCTCATCT
>sequence60↓
CGGGAGTATGCTTTTGCCATTGCCCTATGGGGCTACTCGTTGCTGAATCAACAGTAATATTACCAAT
>sequence61↓
ATGGGGTCTTTCGTCCTGTGCGGGTAACCTGCATCTTACAGGTACTTCAATTTACCGAGTCTCT
```

3.06 MB (3,218,315 バイト), 2,497 行。Text 2,497行, 1桁 日本語 (シフト JIS) 0 文字 0/2,497 行

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trand情報付きのリストファイルの作成、例題6のコード解説
 - スtrand情報反映戦略を練り、コードに修正を加えて実行(例題7)

Excelで確認

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
```

R Console

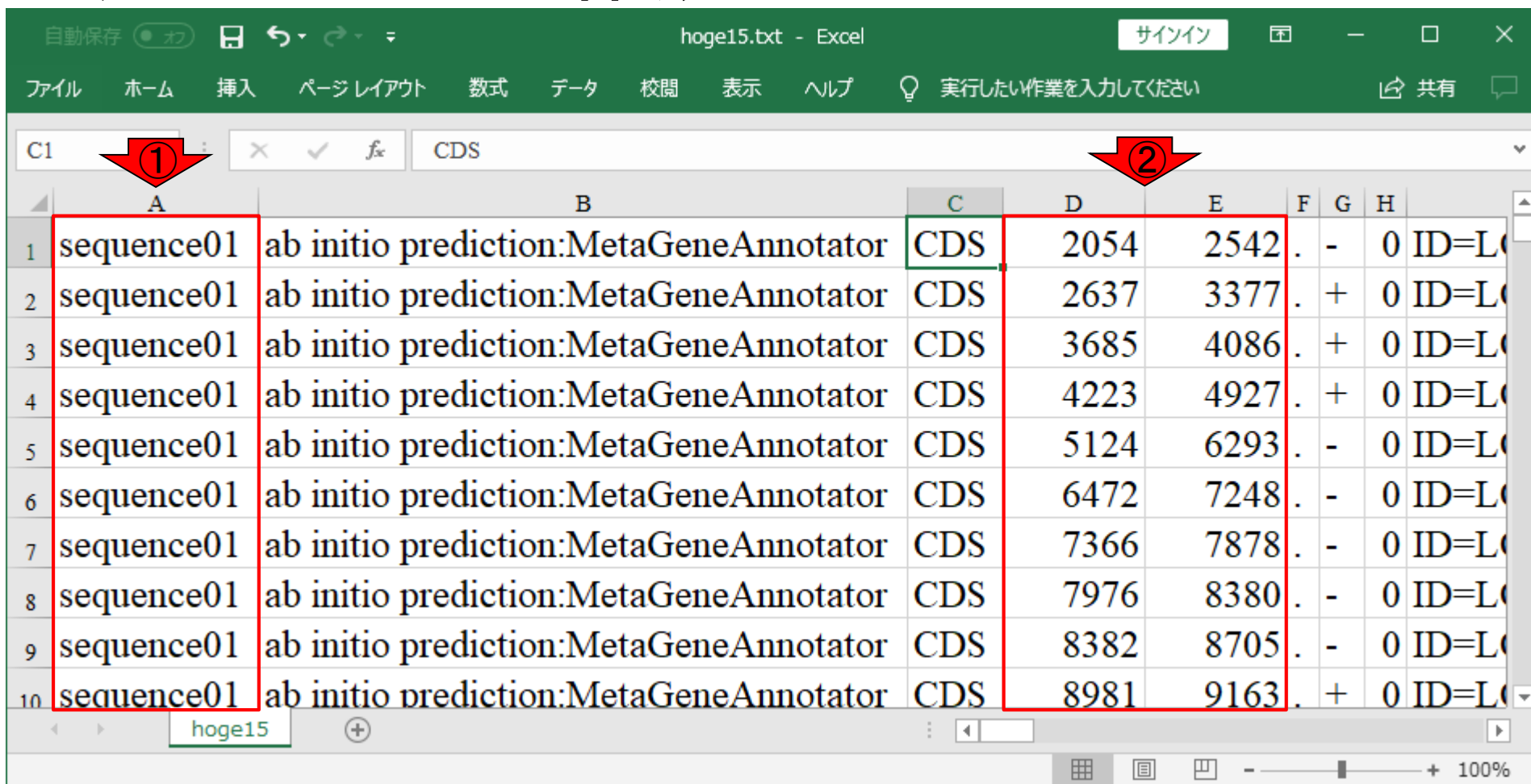
```
> param <- "CDS" #検索したい$  
>  
> #入力ファイルの読み込み  
> data <- readLines(in_f) #in_fで指定$  
> length(data) #オブジェクト$  
[1] 2496  
>  
> #本番(paramで指定した文字列と一致する行番号情報を得$  
> hoge <- sapply(param, grep, x=data) #paramで指定$  
> hoge <- unique(unlist(hoge)) #得られるhog$  
> out <- data[hoge] #hogeで指定$  
> length(out)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

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

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③ list_20190513.txtとしました。

The screenshot shows a spreadsheet with the following data:

	A	B	C	D	E	F
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			

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trand情報付きのリストファイルの作成、例題6のコード解説
 - スtrand情報反映戦略を練り、コードに修正を加えて実行(例題7)

基本テクで実行

(Rで)塩基配列解析

(last modified 2019/04/15, since 2010)

このウェブページのR関連部分は、[インストール \(Macintosh2018.11.27版\)](#)に従ってフリー
います。初心者の方は[基本的な利用法](#)([Wiki](#))
2018年7月に(Rで)塩基配列解析の一部 (2018/07/18)

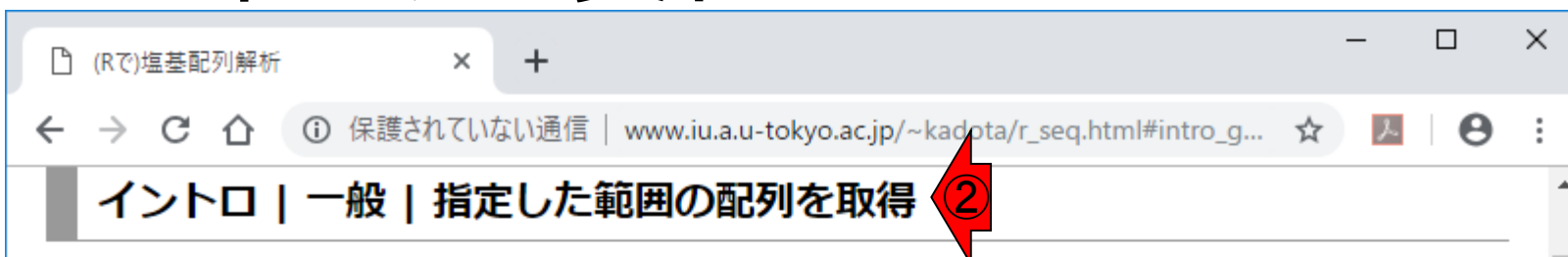
What's new? (過去のお知らせはこちら)

- 「カウント情報取得 | シミュレーション」に追加しました。(2019/04/11) **NEW**
- 「カウント情報取得 | シミュレーション」に追加しました。(2019/04/11) **NEW**
- 削除予定としていた「インストール | R」
- 削除予定としていた「インストール | R」

- [基本的な利用法](#) (last modified 2019/03/12)
- [サンプルデータ](#) (last modified 2018/06/09)
- イントロ | 一般 | [ランダムに行を抽出](#) (last modified 2014/07/17)
- イントロ | 一般 | [任意の文字列を行の最初に挿入](#) (last modified 2014/07/17)
- イントロ | 一般 | [任意のキーワードを含む行を抽出\(基礎\)](#) (last modified 2016/04/20)
- イントロ | 一般 | [ランダムな塩基配列を生成](#) (last modified 2014/06/16)
- イントロ | 一般 | [任意の長さの可能な全ての塩基配列を作成](#) (last modified 2015/02/19)
- イントロ | 一般 | [任意の位置の塩基を置換](#) (last modified 2013/09/12)
- イントロ | 一般 | [指定した範囲の配列を取得](#) (last modified 2015/04/06)
- イントロ | 一般 | [指定したID\(染色体やdescription\)の配列を取得](#) (last modified 2014/03/10)
- イントロ | 一般 | [翻訳配列\(translate\)を取得\(基礎\) | Biostrings](#) (last modified 2015/09/12)
- イントロ | 一般 | [翻訳配列\(translate\)を取得\(応用\) | seqinr\(Charif_2005\)](#) (last modified 2015/03/09)
- イントロ | 一般 | [相補鎖\(complement\)を取得](#) (last modified 2019/03/10)
- イントロ | 一般 | [逆相補鎖\(reverse complement\)を取得](#) (last modified 2019/03/10)
- イントロ | 一般 | [逆鎖\(reverse\)を取得](#) (last modified 2019/03/10)
- イントロ | 一般 | [k-mer解析 | k=1\(塩基ごとの出現頻度解析\) | Biostrings](#) (last modified 2016/04/27)

[トップページへ](#)

基本テクで実行



single-FASTA形式やmulti-FASTA形式の配列を抽出する。この染色体の、ここから、ここまでに、chr3の20000から50000、chr8の配列のみ抽出といった場合に、*.fastaという拡張子が*.txtと異なる。[ファイル] - [ディレクトリ]

1. (single-)FASTA形式ファイルの場合:
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
fasta <- readDNASTringSet(in_f, format="fasta")
```

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ファイルを取得するやり方です。

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

#必要なパッケージをロード
library(Biostrings)

#入力ファイルの読み込み
fasta <- readDNASTringSet(in_f1, format="fasta")#in_f1で指定したファイルの読み込み
posi <- read.table(in_f2)
fasta

#本番
hoge <- NULL
for(i in 1:nrow(posi)){
  obj <- names(fasta) == posi[i,1]
  tmp <- subseq(fasta[obj], start=posi[i,2], end=posi[i,3])#subseq関数を用いてobjがTRUEとな
```

#入力ファイル名を指定してin_f1に格納(multi-FASTAファイル)
#入力ファイル名を指定してin_f2に格納(リストファイル)
#出力ファイル名を指定してout_fに格納

#パッケージの読み込み

#in_f2で指定したファイルの読み込み
#確認してるだけです

#最終的に得る結果を格納するためのブレースホルダhogeを作成
#length(posi)回だけループを回す
#条件を満たすかどうかを判定した結果をobjに格納する
[トップページへ](#)

基本テクで実行

6. multi-FASTA形式のファイル (`genome.fna`) フォーマット

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ルからなるmulti-FASTAファイルを取得するやり方

```

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="fasta") #入
posi <- read.table(in_f2) #in
fasta #確

#本番
hoge <- NULL #最
for(i in 1:nrow(posi)){ #1レ
  obj <- names(fasta) == posi[i,1] #条
  tmp <- subseq(fasta[obj], start=posi[i,2], end=posi[i,3]) #出
}
writeXStringSet(tmp, file=out_f, format="fasta") #出

```

R Console Output:

```

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...TATTTTCAT 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", $
> |

```

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

6. multi-FASTA形式のファイル (genome.fna) フ

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ル
からなるmulti-FASTAファイルを取得するやり方

```
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="fasta") #入
posi <- read.table(in_f2) #in
fasta #確

#本番
hoge <- NULL #最
for(i in 1:nrow(posi)){ #1レ
  obj <- names(fasta) == posi[i,1] #条
  tmp <- subseq(fasta[obj], start=posi[i,2], end=posi[i,3]) #行
```

RGU (64-bit)

ファイル 編集 関



R Console

```
A DNAStringSet instance of length 2311
width 5 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...TATTCAT 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

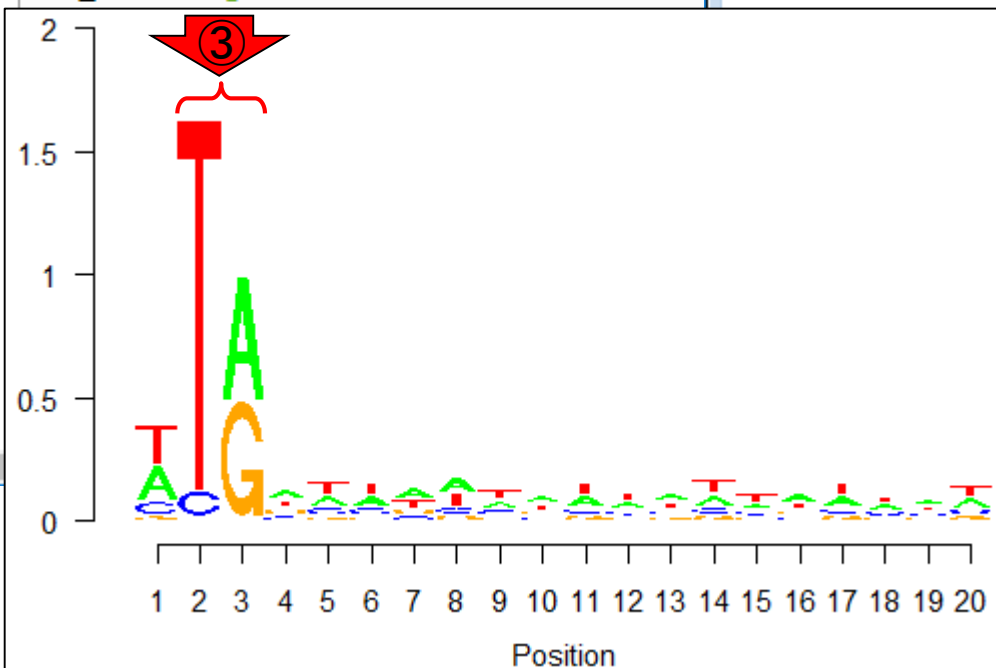
①例題6のコピペ実行結果。今抽出している部分配列は、CDS(タンパク質コード領域の配列; coding sequence)。したがって、メチオニンをコードするATG(やGTGやTTG)が多くみられるはずだが、②2番目と3番目のポジションを眺めた段階で、「何かミスっている」ことに気づく。Sequence logosで確認してもよい。③こんな感じの結果が得られるはずです。

(Rで)塩基配列解析

6. multi-FASTA形式のファイル (genome.fna) フ

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ルからなるmulti-FASTAファイルを取得するやり方

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



A DNASTringSet instance of length 2311
width 5 names

②

```
CTATTTAAC...TTTAACAT sequence01_2054_2542  
TTGAAAATT...AAATCTAG sequence01_2637_3377  
ATGGTTACA...TGTTTTAA sequence01_3685_4086  
ATGGAAATG...CTAAGTGA sequence01_4223_4927  
TTACCCAAA...TATTTTCAT 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trand情報付きのリストファイルの作成、例題6のコード解説
 - スtrand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…

The screenshot shows a spreadsheet application window titled 'hoge15.txt'. The spreadsheet has 10 rows and 8 columns labeled A through H. The data in the first five columns is as follows:

	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	E	F	G	H	I
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。

(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ファイルを取得するやり方です。

```
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.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  <- "hoge6.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
fast  <- fasta                   #確認してるだけです

#本番
hoge <- NULL                   #最終的に得る結果を格納するためのプレースホルダhogeを作る
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とな
```

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。

(Rで)塩基配列解析

x +

① 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
fast #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スホルダhogeを作
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とな
  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" "で結合したも
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](#)

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

x +

保護されていない通信 | www.iu.a.u-tokyo.ac.jp/~kadota/seq/ind...
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ファイルを取得するやり方です。

```
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
```

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

トップページへ

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ファイルを取得するやり方です。

```
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  
for(i in 1:nrow(posi)){ #最終的に得る結果を格納するためのプレースホルダhoge  
  obj <- names(fasta) == posi[i,1] #length(posi)回だけループを回す  
  tmp <- subseq(fasta[obj], start=posi[i,2], end=posi[i,3]) #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, tmp) #subseq関数を用いてobjがTRUE #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  
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
```

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/seq/indm/indm_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 ④ ①
hoge <- NULL
for(i in 1:nrow(posi)){ #最終的に得る結果を格納するためのプレースホルダhoge
  obj <- names(fasta) == posi[i,1] #length(posi)回だけループを回す
  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
```

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

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。

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。予め用意した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ファイルを取得するやり方です。

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

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](#)

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…

コードの解説

```

(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ファイルを取得するやり方です。

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]) #seq関数を用いてobjがTRUE
  hoge <- append(hoge, tmp) #tmpの情報をhogeに追加格納
}
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

```

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、③`start`位置の要素に相当する*i*行2列の要素③`posi[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、⑤`subseq`関数を用いて部分配列を抽出している。

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ファイルを取得するやり方です。

```
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:length(posi)){ #length(posi)回だけループを回す
  obj <- 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(obj, start=posi[i,2], end=posi[i,3]) #subseq関数を用いてobjがTRUEの時tmpの情報をhogeに追加格納
  hoge <- append(hoge, tmp)
}
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
```

sequence01	2054	2542
sequence01	2537	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。

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。予め用意した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を読み込む
posi <- read.table(in_f2) #in_f2で指定したファイルを読み込む
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の要素を抽出
  hoge <- append(hoge, tmp) #tmpの情報をhogeに追加で格納
}
fasta #hogeの中身をfastaに格納
#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
```

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](#)

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trand情報付きのリストファイルの作成、例題6のコード解説
 - スtrand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。

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]) #seq関数を用いてobjがTRUE
  hoge <- append(hoge, tmp) #tmpの情報をhogeに追加格納
}
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
```

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](#)

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

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"
in_f2 <- "list_20190513_strand.txt"
out_f <- "hoge7.fasta"

#必要なパッケージをロード
library(Biostrings)

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

#本番
hoge <- NULL
for(i in 1:nrow(posi)){
  obj <- names(fasta) == posi[i,1]
  tmp <- subseq(fasta[obj], start=posi[i,2], end=posi[i,3])
  if(posi[i,4] == "-"){tmp <- reverseComplement(tmp)}
  hoge <- append(hoge, tmp)
}
fasta <- hoge
```

```
#入力ファイル名を指定してin_f1に格納(multi-FASTA形式)
#入力ファイル名を指定してin_f2に格納(リストファイル)
#出力ファイル名を指定してout_fに格納

#パッケージの読み込み

#in_f1で指定したファイルの読み込み
#in_f2で指定したファイルの読み込み
#確認してるだけです

#最終的に得る結果を格納するためのプレースホルダー
#length(posi)回だけループを回す
#条件を満たすかどうかを判定した結果をobjに格納
#subseq関数を用いてobjから配列を抽出
#ストランドが”-”の場合は逆相補鎖を作成
#tmpの情報をhogeに追加で格納

#hogeの中身をfastaに格納
#確認してるだけです
```

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	8081	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

x +

① 保護されていない通信 | www.iu.a.u-tokyo.ac.jp/~kad

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"
in_f2 <- "list_20190513_strand.txt"
out_f <- "hoge7.fasta"

#必要なパッケージをロード
library(Biostrings)

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

#本番
hoge <- NULL
for(i in 1:nrow(posi)){
  obj <- names(fasta) == posi[i,1]
  tmp <- subseq(fasta[obj], start=posi[i,2], end=posi[i,3])
  if(posi[i,4] == "-"){tmp <- reverseComplement(tmp)}
  hoge <- append(hoge, tmp)
}
fasta <- hoge
```

② #入力ファイル名を指定してin_f1に格納(multi-FASTA形式) / #入力ファイル名を指定してin_f2に格納(リストファイル) / #出力ファイル名を指定してout_fに格納

#パッケージの読み込み

#in_f1で指定したファイルの読み込み / #in_f2で指定したファイルの読み込み / #確認してるだけです

#最終的に得る結果を格納するためのプレースホルダー / #length(posi)回だけループを回す / #条件を満たすかどうかを判定した結果をobjに格納 / #subseq関数を用いてobjからストランドが“-”の場合は逆相補鎖を作成 / #tmpの情報をhogeに追加で格納

#hogeの中身をfastaに格納 / #確認してるだけです

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

例題7

①

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

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読み込ませて、ストランドを適切に反映させた2,311行の出力形式です。

```

in_f1 <- "genome.fna" #入力ファイル名
in_f2 <- "list_20190513_strand.txt" #入力ファイル名
out_f <- "hoge7.fasta" #出力ファイル名

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

#入力ファイルの読み込み
fasta <- readDNAStringSet(in_f1, format="fasta") #FASTA形式で読み込み
posi <- read.table(in_f2) #ストランド情報を読み込み
fasta #確認

#本番
hoge <- NULL #初期値
for(i in 1:nrow(posi)){ #ループ
  obj <- names(fasta) == posi[i,1] #対象のFASTAエントリを特定
  tmp <- subseq(fasta[obj], start=posi[i,2], end=posi[i,3]) #指定範囲のシーケンスを抽出
  if(posi[i,4] == "-"){tmp <- reverseComplement(tmp)} #逆相補を生成
  hoge <- append(hoge, tmp) #結果を結合
}
fasta <- hoge #最終結果
writeXStringSet(fasta, file=out_f, format="fasta") #ファイルに保存

```

RGui (64-bit)

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", $
> |

```

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、うまくいってい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

7. multi-FASTA形式のファイル (genome.fna)の読み込み

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読み込ませて、ストランドを適切に反映させた2,311行の出力です。

```
in_f1 <- "genome.fna" #入力ファイル名
in_f2 <- "list_20190513_strand.txt" #ストランド情報ファイル名
out_f <- "hoge7.fasta" #出力ファイル名

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

#入力ファイルの読み込み
fasta <- readDNAStringSet(in_f1, format="fasta") #FASTA形式で読み込み
posi <- read.table(in_f2) #ストランド情報を読み込み
fasta #確認

#本番
hoge <- NULL #初期値
for(i in 1:nrow(posi)){ #ストランド情報に基づいて反転処理
  obj <- names(fasta) == posi[i,1] #対象のFASTAレコードを抽出
  tmp <- subseq(fasta[obj], start=posi[i,2], end=posi[i,3]) #指定範囲を抽出
  if(posi[i,4] == "-"){tmp <- reverseComplement(tmp)} #逆転写
  hoge <- append(hoge, tmp) #結果に追加
}
fasta <- hoge #最終結果
fasta #確認
```

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ



R Console

```
A DNAStringSet instance of length 2311
      width 5 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", $
> |
```