ゲノム情報解析基礎
～ Rで塩基配列解析2 ～

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
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/

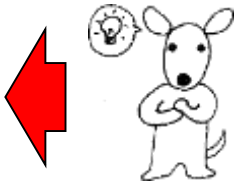
「感想やコメント」へのコメント

多くのヒトが感想を述べられて
いました。ありがとうございます。

- Pythonを使った解析をやりたい → NGSハンズオン講習会(7/28)でプログラミングをやる予定
- MacでFASTAファイルをダウンロードすると”hoge4.fa.txt”などのようになったよ → (Rで)塩基配列解析中の「基本的な利用法(Mac版)」にも記載あり。や講義中でも説明しましたよ。
- 講義の進度(分かりやすいや理解しやすいを除く)
  - もう少し早くてもよい、作業の待ち時間が長い:4名
  - ちょうどよい、問題ない:6名
- 発展問題があればなおよい(3名) → 前向きに検討させていただきま、、、した。
- ホチキスの位置を… → 気をつけますm(_ _)m…
- library(Biostrings)でパッケージをダウンロードしておかないとFASTAファイルを読み込めない？alphabetFrequency()が使えない？ → 正解です。今日の講義で話します。
- 2限目は75分ほどで終わらせて残りはRの雑学などを… → 無理です(笑)
- 基礎なので仕方ないかもしれないが教わった内容を応用できる気がしないのが残念 → (TbT)

「感想やコメント」へのコメント

有用な情報ですので、ぜひ共有させてくださいm(_ _)m



- 講義の後半は疲れているので比重を前半にしてほしい → 気持ちはよくわかるので善処します
- Tm (melting temperature)計算がalphabetFrequency関数で簡単にできそう → 確かにTm値はGC含量と配列長で決まりますね!
- (Rで)塩基配列解析のウェブページで「・」ではなく「Number」にするほうがforeign peopleにとっていいのでは? → It is actually difficult to use numbers instead of dots for items or subitems in the lists because I frequently add (and delete) them, giving unfixed numbering for individual (sub)items.
- N50が大きければ大きいほどコンティグに含まれる配列長が長く、コンティグ数が少なくて精度が高いことを意味するのか？ → (ほぼ)正解です
- 疲れました → お疲れ様です!
- 使いこなせるようになりたい!、頑張ります! → 頑張ってください!

NGSハンズオン講習会

4/20から受講申込開始。部分受講可能。**アグリバイオの講義とは無関係!本気で学びたいヒトが大勢います!**受講申込だけとりあえずしておこう的な軽い気持ちの人はご遠慮ください。もちろん定員に余裕があるなら参加したい的な控えめな申込みはアリ(個人の感想です)。

(Rで)塩基配列解析

~NGS、RNA-seq、ゲノム、トランスクリプトーム、正規化、発現変動、統計、モデル、バイオインフォ

(last modified 2015/04/14, since 2010)

What's new?

- このウェブページはインストール | について の推奨手順 (Windows2015.04.04版とMacintosh2015.04.03版)に従って フリーソフト Rと必要なパッケージをインストール済みであるという前提で記述しています。初心者の方は基本的な利用法(Windows2015.04.03版と Macintosh2015.04.03版)で自習してください。本ウェブページを体系的にまとめた書籍もあります。(2015/04/03) NEW
- 平成27年度NGSハンズオン講習会を2015年7月22日-8月6日の11日間で実施します。受講申込開始は4/20予定です。昨年度の「NGS速習コース」同様、オブザーバー(TA)募集も並行して行いますので可能な範囲でご協力をお願いしますm(_ _)m(2015/04/15) NEW
- 参考資料

- はじめに
- 参考資料
- 過去のお
- インストー

NBDC National Bioscience Database Center

-散在するデータベースを、まとめて、使い易く-

バイオサイエンスデータベースセンター

国立研究開発法人 科学技術振興機構

文字サイズ変更 大 中 小

English サイトマップ サイト内検索 検索... 検索

ホーム | NBDCについて | 研究開発 | 公募情報 | 採用情報 | 広報 | 人材支援 | お問い合わせ | リンク

Home ▸ 人材支援 ▸ 支援 ▸ 講習会 ▸ 平成27年度NGSハンズオン講習会

平成27年度NGSハンズオン講習会（2015年7月22日~8月6日（7月31日および土日は除く））

バイオインフォマティクス人材育成カリキュラム
次世代シークエンサ(NGS)ハンズオン講習会を開催します。

NGSハンズオン講習会

受講生の要望に応えて…①講師数を減らし項目間の連携強化。②Python追加。③NGS解析部分を増加(2.5から4日分)。④統計解析追加。⑤ハンズオンのみ。⑥予備日の確保。

- 7月22日(水):Bio-Linux 8とRのインストール状況確認。主にPC持込者を対象。基本自習(門田)
- 7月23日(木):Linux基礎。LinuxコマンドなどUNIXの基礎の理解(門田)
- 7月24日(金):スクリプト言語。シェルスクリプト(アメリエフ株式会社 服部恵美先生)

- 7月27日(月):スクリプト言語。Perl(アメリエフ 服部恵美先生)
- 7月28日(火):スクリプト言語。**Python**(アメリエフ 服部恵美先生)
- 7月29日(水):データ解析環境R(門田)
- 7月30日(木):データ解析環境R(門田)

- 8月3日(月):NGS解析。基礎(アメリエフ 山口昌雄先生)
- 8月4日(火):NGS解析。ゲノムReseq、変異解析(アメリエフ 山口昌雄先生)
- 8月5日(水):NGS解析。RNA-seq、統計解析(前半:アメリエフ 山口昌雄先生、後半:門田)
- 8月6日(木):NGS解析。ChIP-seq(東京医科歯科大学 森岡勝樹先生)

- 8月26日(水):予備日
- 8月27日(木):予備日
- 8月28日(金):予備日

参考

AJACSで補完

DDBJ, DBCLS, NBDCの昨年度講師の先生方は、統合データベース講習会(AJACS)の枠組みでも活動しています。DBCLS主催のハンズオン講習会(AJACSadvanced)もあり。2015年のAJACSは以下の通り。

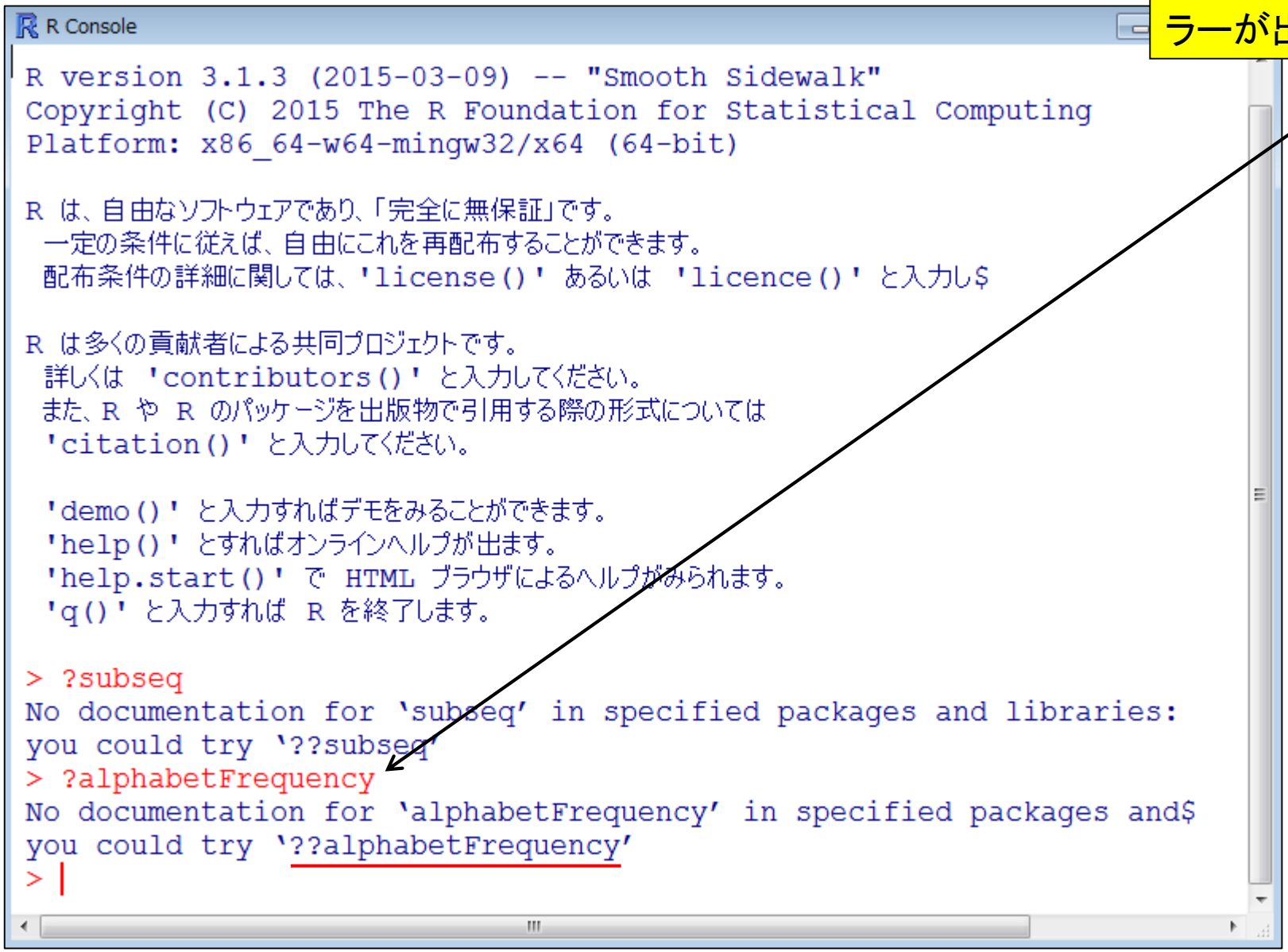
- 東京医科歯科大学　(2015年5月20日-21日開催予定)
- 大阪大学　(2015年6月8日-9日開催予定)
- 鳥取大学　(2015年8月4日開催予定)
- 弘前大学　(2015年9月3日-4日開催予定)
- 愛媛大学　(2015年9月25日開催予定)
- 鹿児島大学　(2016年1月26日-27日開催予定)

Contents

- パッケージ
  - CRANとBioconductor
  - 推奨パッケージインストール手順のおさらい
  - ゲノム情報パッケージBSgenomeの概観
  - ヒトゲノム情報パッケージの解析
- 2連続塩基出現頻度解析(CpG解析)、k-mer解析
  - 課題
  - GC含量の違いを考慮(連続塩基の種類ごとに期待値が異なる)
  - 作図(box plot)
- その他
  - 数式の感覚を理解
  - Sequence logos (Schneider and Stephens, 1990)
  - プロモーター配列取得

パッケージ

R起動直後に「**?関数名**」と打ち込んでも、使用法を記したウェブページが開かずにエラーが出ることがあります。



パッケージ

①「??alphabetFrequency」と打ち込むように勧められているので打ってみる。検索結果のウェブページが表示されるので、②それっぽい関数名のところをクリック。

R Console

```
> ?subseq
No documentation for 'subseq' in specified packages and libraries:
you could try '??subseq'
> ?alphabetFrequency
No documentation for 'alphabetFrequency' in specified packages and$
you could try '??alphabetFrequency'
> ??alphabetFrequency
starting httpd help server ... done
>
```

①

Search Results

The search string was **"alphabetFrequency"**

Help pages:

Biostrings::class:MultipleAlignment	MultipleAlignment objects
Biostrings::letterFrequency	Calculate the frequency of letters in a biological sequence, or the consensus matrix of a set of sequences
GenomicAlignments::stackStringsFromBam	Stack the read sequences stored in a BAM file on a region of interest
ShortRead::QualityScore-class	Quality scores for short reads and their alignments

②

パッケージ

letterFrequency {Biostrings} R Documentation

Calculate the frequency of letters in a biological sequence, or the consensus matrix of a set of sequences

Description

Given a biological sequence (or a set of biological sequences), the `alphabetFrequency` function computes the frequency of each letter of the relevant alphabet.

`letterFrequency` is similar, but more compact if one is only interested in certain letters. It can also tabulate letters "in common".

`letterFrequencyInSlidingView` is a more specialized version of `letterFrequency` for (non-masked) XString objects. It tallys the requested letter frequencies for a fixed-width view, or window, that is conceptually slid along the entire input sequence.

The `consensusMatrix` function computes the consensus matrix of a set of sequences, and the `consensusString` function creates the consensus sequence from the consensus matrix based upon specified criteria.

In this man page we call "DNA input" (or "RNA input") an XString, XStringSet, XStringViews or MaskedXString object of base type DNA (or RNA).

Usage

```
alphabetFrequency(x, as.prob=FALSE, ...)
hasOnlyBaseLetters(x)
uniqueLetters(x)
```

alphabetFrequency関数はBiostringsというパッケージから提供されているものだと読み解く。「??関数名」は、関数名は既知だがどのパッケージから提供されているものかを知りたい場合などに利用する。

パッケージ

multi-FASTAファイルを読み込んで様々な解析ができるのは、①Biostringsや②seqinrなどの塩基配列解析用パッケージのおかげです。

- 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) NEW
- 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 2015/03/09)
- 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 2013/06/14)
- イントロ | 一般 | 逆相補鎖(reverse complement)を取得 (last modified 2013/06/14)

①

②

イントロ | 一般 | 翻訳配列(translate)を取得(基礎) | Biostrings

Biostringsパッケージを用いて塩基配列を読み込んでアミノ酸配列に翻訳するやり方を示します。翻訳のための遺伝コード(genetic code)は、Standard Genetic Codeだそうです。もちろん生物種?!によって多少違い(variants)があるようで、"Standard", "SGC0", "Vertebrate Mitochondrial", "SGC1"などいろいろ選べるようです。
「ファイル」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピペ。

1. FASTA形式ファイル(sample1.fasta)の場合:

multi-FASTAではないsingle-FAST

```
in_f <- "sample1.fasta"
out_f <- "hoge1.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
```

イントロ | 一般 | 翻訳配列(translate)を取得(応用) | seqinr(Charif_2005)

seqinrパッケージを用いて塩基配列を読み込んでアミノ酸配列に翻訳するやり方を示します。本気で翻訳配列を取得する場合にはこちらの利用をお勧めします。翻訳できないコドンはアミノ酸X(不明なアミノ酸)に変換してくれたり、translate関数のオプションとしてambiguous=Tとすると、翻訳できるものは可能な限り翻訳してくれます(高橋 広夫 氏提供情報)。lapply関数を用いるやり方(高橋 広夫 氏提供情報)とsapply関数を用いるやり方(甲斐 政親 氏提供情報)を示します。
「ファイル」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピペ。

1. FASTA形式ファイル(sample1.fasta)の場合:

multi-FASTAではないsingle-FASTA形式ファイルです。

```
in_f <- "sample1.fasta"                    #入力ファイル名を指定してin_fに格納
out_f <- "hoge1.fasta"                     #出力ファイル名を指定してout_fに格納

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

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

パッケージ

Biostringsというパッケージをlibrary関数を用いて読み込むことによって、alphabetFrequencyのようなBiostringsが提供する関数群を利用できるのです。ここでは、意図的に「library(Biostrings)」を2回実行して、2回目は何も表示されないということを思い出させています。実際には1回のみで大丈夫です。「?alp」まで打ってからTabキーを押すなどして「タブ補完」テクを有効利用。

R Console

```
> library(Biostrings)
 要求されたパッケージ BiocGenerics をロード中です
 要求されたパッケージ parallel をロード中です

 次のパッケージを付け加えます: 'BiocGenerics'

The following objects are masked from 'package:parallel':

    clusterApply, clusterApplyLB, clusterCall, cluste
    clusterExport, clusterMap, parApply, parCapply,
    parLapply, parLapplyLB, parRapply, parSapply, par

The following object is masked from 'package:stats':

    xtabs

The following objects are masked from 'package:base':

    anyDuplicated, append, as.data.frame, as.vector,
    colnames, do.call, duplicated, eval, evalq, Filte
    get, intersect, is.unsorted, lapply, Map, mapply,
    mget, order, paste, pmax, pmax.int, pmin, pmin.in
    Position, rank, rbind, Reduce, rep.int, rownames,
    setdiff, sort, table, tapply, union, unique, unli
    unsplit

 要求されたパッケージ S4Vectors をロード中です
 要求されたパッケージ stats4 をロード中です
 要求されたパッケージ IRanges をロード中です
 要求されたパッケージ XVector をロード中です
> library(Biostrings)
> ?alphabetFrequency
```

letterFrequency {Biostrings} R Documentation

Calculate the frequency of letters in a biological sequence, or the consensus matrix of a set of sequences

Description

Given a biological sequence (or a set of biological sequences), the `alphabetFrequency` function computes the frequency of each letter of the relevant alphabet.

`letterFrequency` is similar, but more compact if one is only interested in certain letters. It can also tabulate letters "in common".

`letterFrequencyInSlidingView` is a more specialized version of `letterFrequency` for (non-masked) XString objects. It tallys the requested letter frequencies for a fixed-width view, or window, that is conceptually slid along the entire input sequence.

The `consensusMatrix` function computes the consensus matrix of a set of sequences, and the `consensusString` function creates the consensus sequence from the consensus matrix based upon specified criteria.

In this man page we call "DNA input" (or "RNA input") an XString, XStringSet, XStringViews or MaskedXString object of base type DNA (or RNA).

Usage

```
alphabetFrequency(x, as.prob=FALSE, ...)
hasOnlyBaseLetters(x)
uniqueLetters(x)
```

R本体とパッケージの関係

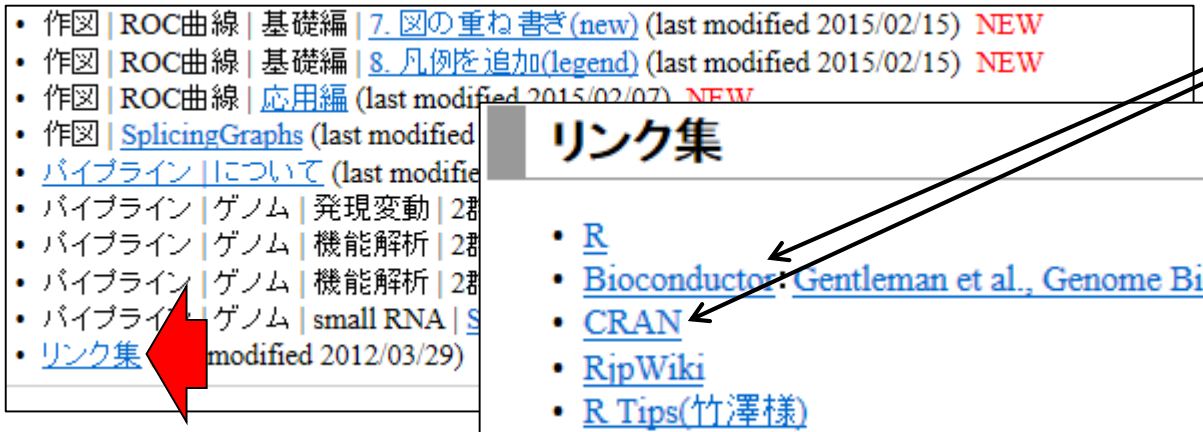
「R本体」と「パッケージ」の関係は、「パソコン」と「ソフト」、「Microsoft EXCEL」と「アドイン」、「Cytoscape」と「プラグイン」のようなものという理解でよい。

- パソコンを購入しただけの状態では、できることが限られています。
  - 通常は、Officeやウイルス撃退ソフトなどをインストールして利用します。
- Linuxをインストールしただけの状態では、できることが限られています。
  - 通常は、マッピングなど各種プログラムをインストールして利用します。
- R本体をインストールしただけの状態では、できることが限られています。
  - NGS解析を行う各種パッケージ（またはライブラリ）をインストールして利用します。

CRANとBioconductor

CRAN提供パッケージは生命科学を含む様々な分野で利用される。NGS解析は、主にBioconductor提供パッケージを利用。

- R上で利用可能なパッケージの2大リポジトリ(貯蔵庫)
  - CRAN (The Comprehensive R Archive Network):6,328パッケージ
  - Bioconductor:934パッケージ



定期的にバージョンアップ

バグの修正や新たな機能がどんどん追加されている。最新版の利用をお勧め。毎年5月と11月ごろにバージョンアップするとよいだろう。

- 近年のリリース頻度
  - R本体 (http://www.r-project.org/)
    - 2015-04-16にver. 3.2.0をリリース
    - 2015-03-09にver. 3.1.3をリリース
    - 2014-10-31にver. 3.1.2をリリース
    - 2014-07-10にver. 3.1.1をリリース
    - …
    - 2012-03-30にver. 2.15.0をリリース
    - …
  - Bioconductor (http://bioconductor.org/)は半年ごとにリリース
    - 2014-10にver. 3.0をリリース (R ver. 3.1.1で動作確認)、提供パッケージ数:934
    - 2014-04にver. 2.14をリリース (R ver. 3.1.0で動作確認)、提供パッケージ数:824
    - 2013-10にver. 2.13をリリース (R ver. 3.0で動作確認)、提供パッケージ数:750
    - 2013-04にver. 2.12をリリース (R ver. 3.0で動作確認)、提供パッケージ数:672
    - 2012-10にver. 2.11をリリース (R ver. 2.15.1で動作確認)、提供パッケージ数:608
    - 2012-04にver. 2.10をリリース (R ver. 2.15.0で動作確認)、提供パッケージ数:553
    - 2011-11にver. 2.9をリリース (R ver. 2.14.0で動作確認)、提供パッケージ数:517
    - …

Bioconductor

Bioconductorに関する総説(Review)。ゲノム配列やアノテーションパッケージもBioconductorから提供されており、それらに関する言及もあり。

2. ゲノム情報解析基礎

授業の目標・

次世代シーケン
るスキルが要求
フリーソフトウ
を目指した実習
また、ウェブツ
は解析するため

担当教員

嶋田 透 （東大
勝間 進 （東大
門田幸二 （東

参考図書

講義日程（平成27年度）

1. 平成27年04月07日（PC使用）
   講師：嶋田 透
   講師：門田幸二
   バイオインフォマティクス基礎知識
   講義資料PDF(Win版)
   講義資料PDF(Mac版)
2. 平成27年04月14日（PC使用）
   講師：門田幸二
   講義資料PDF
   (Rで)塩基配列解析
   rcode_20140908.txt
   sample4.fasta
3. 平成27年04月21日（PC使用）
   講師：嶋田 透
   講師：門田幸二
   講義資料PDF
   (Rで)塩基配列解析
   Huber et al., Nat Methods, 2015
4. 平成27年04月[illegible]（PC使用）
   講師：勝間 進

Nat Methods. 2015 Feb;12(2):115-21. doi: 10.1038/nmeth.3252.

Orchestrating high-throughput genomic analysis with Bioconductor.

Huber W[1], Carey VJ[2], Gentleman R[3], Anders S[1], Carlson M[4], Carvalho BS[5], Bravo HC[6], Davis S[7], Gatto L[8], Girke T[9], Gottardo R[10], Hahne F[11], Hansen KD[12], Irizarry RA[13], Lawrence M[3], Love MI[13], MacDonald J[14], Obenchain V[4], Oleś AK[1], Pagès H[4], Reyes A[1], Shannon P[4], Smyth GK[15], Tenenbaum D[4], Waldron L[16], Morgan M[4].

Author information

Abstract

Bioconductor is an open-source, open-development software project for the analysis and comprehension of high-throughput data in genomics and molecular biology. The project aims to enable interdisciplinary research, collaboration and rapid development of scientific software. Based on the statistical programming language R, Bioconductor comprises 934 interoperable packages contributed by a large, diverse community of scientists. Packages cover a range of bioinformatic and statistical applications. They undergo formal initial review and continuous automated testing. We present an overview for prospective users and contributors.

パッケージのインストール

「必要最小限プラスアルファ」の推奨インストール手順を行えば、バイオスタティスティクス基礎論で用いるmapsパッケージや、いくつかのゲノム配列(BSgenome)パッケージがインストールされます。

- はじめに (last modified 2015/03/31) NEW
- 参考資料(講義、講習会、本など) (last modified 2015/03/09) NEW
- 過去のお知らせ (last modified 2015/03/31) NEW
- インストール | について (last modified 2015/03/22) NEW
- インストール | R本体 | 最新版 | Win用 (last modified 2015/03/22)推奨 NEW
- インストール | R本体 | 最新版 | Mac用 (last modified 2015/03/22)推奨 NEW
- インストール | R本体 | 過去版 | Win用 (last modified 2015/03/22) NEW
- インストール | R本体 | 過去版 | Mac用 (last modified 2015/03/22) NEW
- インストール | Rパッケージ | ほぼ全て(20GB以上?!) (last modified 2015/03/22) NEW
- インストール | Rパッケージ | 必要最小限プラスアルファ(数GB?!) (last modified 2015/03/27)推奨 NEW
- インストール | Rパッケージ | 必要最小限(数GB?!) (last modified 2015/03/23) NEW
- インストール | Rパッケージ | 個別 (last modified 2015/03/23) NEW
- (削除予定)Rのインストールと起動 (last modified 2015/03/27) NEW
- (削除予定)個別パッケージのインストール (last modified 2015/02/20)
- 基本的な利用法 (last modified 2015/01/16)
- サンプルデータ (last modified 2015/02/15)
- バイオインフォマティクス人材育成カリキュラム(次世代シークエンサ) | 速習コース (last modified 2015/0
- 書籍 | トランスクリプトーム解析 | について (last modified 2014/05/12)
- 書籍 | トランスクリプトーム解析 | 2.3.1 RNA-seqデータ(FASTQファイル) (last modified 2014/04/15)

パッケージのインストール

インストール | Rパッケージ | 必要最小限プラスアルファ(数GB?!) NEW

(Rで)塩基配列解析、(Rで)マイクロアレイデータ解析中で利用するパッケージ、プラスアルファのパッケージをインストールするやり方です。Rパッケージの2大リポジトリであるCRANとBioconductorから提供されているパッケージ群のうち、一部のインストールに相当しますので、相当短時間でインストールが完了します。

1. R本体を起動

2. CRANから提供されているパッケージ群のインストール

以下を「R コンソール画面上」でコピー&ペースト。どこからダウンロードするか？と聞かれるので、その場合は自分のいる場所から近いサイトを指定しましょう。

```
#(Rで)塩基配列解析で主に利用
install.packages("limma")                #(Rで)マイクロアレイデータ解析でも利用
install.packages("samr")                 #(Rで)マイクロアレイデータ解析でも利用
install.packages("seqinr")               #(Rで)マイクロアレイデータ解析でも利用

#(Rで)マイクロアレイデータ解析で利用
install.packages("cclust")
install.packages("class")
install.packages("e1071")
install.packages("GeneCycle")
install.packages("gptk")
install.packages("GSA")
install.packages("mixOmics")
install.packages("pvclust")
install.packages("RobLoxBioC")
install.packages("som")
install.packages("st")
install.packages("varSelRF")

#アグリバイオの他の講義科目で利用予定
install.packages("ape")
install.packages("cluster")
install.packages("fields")
```

①「バイオスタティスティクス基礎論」で用いるmapsやmapdataパッケージのインストールは、この段階で行っています。②これらはCRANから提供されているものたちです。

②

①

パッケージのインストール

```
biocLite("DESeq", suppressUpdates=TRUE)
biocLite("DESeq2", suppressUpdates=TRUE)
biocLite("DiffBind", suppressUpdates=TRUE)
biocLite("doMC", suppressUpdates=TRUE)
biocLite("EBSeq", suppressUpdates=TRUE)
biocLite("EDASeq", suppressUpdates=TRUE)
biocLite("edgeR", suppressUpdates=TRUE)
biocLite("GenomicAlignments", suppressUpdates=TRUE)
```

①

4. Bioconductorから提供されているパッケージ群のインストール

ゲノム配列パッケージです。一つ一つの容量が尋常でないため、必要に応じてテキストエディタなどに予めコピペしておき、いらないゲノムパッケージを削除してからお使いください。

```
source("http://bioconductor.org/biocLite.R")#おまじない
biocLite("BSgenome.Athaliana.TAIR.TAIR9", suppressUpdates=TRUE)#シロイヌナズナゲノム
biocLite("BSgenome.Celegans.UCSC.ce6", suppressUpdates=TRUE)#線虫ゲノム
biocLite("BSgenome.Drerio.UCSC.danRer7", suppressUpdates=TRUE)#ゼブラフィッシュゲノム
biocLite("BSgenome.Hsapiens.NCBI.GRCh38", suppressUpdates=TRUE)#ヒトゲノム(GRCh38)
biocLite("BSgenome.Hsapiens.UCSC.hg19", suppressUpdates=TRUE)#ヒトゲノム(hg19)
biocLite("BSgenome.Mmusculus.UCSC.mm10", suppressUpdates=TRUE)#マウスゲノム(mm10)
```

②

①ゲノム情報のパッケージ群(BSgenome…)はBioconductorから提供されています。ここでは計6パッケージをインストールしています。②例えば赤線部分は、マウスのmm10というバージョンのゲノム配列情報を含むパッケージの名前(BSgenome.Mmusculus.UCSC.mm10)に相当します。biocLiteという関数を用いて該当パッケージをインストールしています。

BSgenome利用の意義

ゲノム配列情報はUCSCやEnsemblなどのウェブサイトから取得するのが一般的ではあるが、Rの生物種ごとに提供されているBSgenomeで取得、あるいは取り扱うことも可能。ChIP-seq用パッケージMEDIPSはBSgenomeを利用。

- 解析 | 機能解析 | パスウェイ(Pathway)解析 | SeqGSEA(Wang 2014) (last modified 2014/
- 解析 | 菌叢解析 | について (last modified 2014/12/19)
- 解析 | 菌叢解析 | phyloseq(McMurdie 2013) (last modified 2014/05/29)
- 解析 | エクソーム解析 | について (last modified 2
- 解析 | ChIP-seq | について modified 2015/02
- 解析 | ChIP-seq | DiffBind(Ross-Innes 2012) (las
- 解析 | ChIP-seq | ChIPseqR(Humburg 2011) (last
- 解析 | ChIP-seq | chipseq (last modified 2011/12/1
- 解析 | ChIP-seq | PICS(Zhang 2011) (last modifie

解析 | ChIP-seq | について

このあたりはほとんどノータッチです。SraTailor (Oki et al., 2014)は、実験医学2014年12月号の「Close Up実験法」中で日本語による解説記事があります(沖 真弥氏提供情報)。
2015年2月に調査した結果をリストアップします。

R用:

- ChIPsim: Zhang et al., PLoS Comput. Biol., 2008
- PeakSeq法: Rozowsky et al., Nat Biotechnol., 2009
- CSAR: Kaufmann et al., PLoS Biol., 2009
- rMAT: Droit et al., Bioinformatics, 2010
- ChIPpeakAnno: Zhu et al., BMC Bioinformatics, 2010
- PICS: Zhang et al., Biometrics, 2011
- ChIPseqR: Humburg et al., BMC Bioinformatics, 2011
- DiffBind: Ross-Innes et al., Nature, 2012
- MEDIPS: Lienhard et al., Bioinformatics, 2014
- DSS: Feng et al., Nucleic Acids Res., 2014
- methylSig: Park et al., Bioinformatics, 2014

R以外:

- bwtool: Pohl and Beato, Bioinformatics, 2014
- SraTailor: Oki et al., Genes Cells., 2014

Review、ガイドライン、パイプライン系:

- ガイドライン: Bailey et al., PLoS Comput Biol., 2013
- Review: Robinson et al., Front Genet., 2014

BSgenome

プロモーター配列取得ではなく、
ゲノム配列取得のほうです！

- イントロ | 一般 | 任意の長さの連続塩基の出現頻度情報を取得 (last modified 2013/06/14)
- イントロ | 一般 | Tips | 任意の拡張子でファイルを保存 (last modified 2013/09/26)
- イントロ | 一般 | Tips | 拡張子は同じで任意の文字を追加して保存 (last modified 2013/09/26)
- イントロ | 一般 | 配列取得 | ゲノム配列 | 公共DBから (last modified 2014/05/28)
- イントロ | 一般 | 配列取得 | ゲノム配列 | BSgenome (last modified 2015/02/18) NEW
- 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 2014/04/02)
- イントロ | 一般 | 配列取得 | プロモーター配列 | BSgenome (last modified 2014/04/25)
- イントロ | 一般 | 配列取得 | プロモーター配列 | GenomicFeatures(Lawrence 2013) (last modified 2014/04/23)
- イントロ | 一般 | 配列取得 | トランスクリ
- イントロ | 一般 | 配列取得 | トランスクリ
- イントロ | NGS | 様々なプラットフォーム
- イントロ | NGS | qPCRやmicroarrayなど

イントロ | 一般 | 配列取得 | ゲノム配列 | BSgenome NEW

BSgenomeパッケージを用いて様々な生物種のゲノム配列を取得するやり方を示します。ミヤマハタザオ (A. lyrata)、セイヨウミツバチ (A. mellifera)、シロイヌナズナ(A.thaliana)、ウシ(B.taurus)、線虫(C.elegans)、犬(C.familiaris)、キイロショウジョウバエ(D.melanogaster)、ゼブラフィッシュ(D.rerio)、大腸菌(E.coli)、イトヨ(G.aculeatus)、セキショクヤケイ(G.gallus)、ヒト(H.sapiens)、アカゲザル(M.mulatta)、マウス(M.musculus)、チンパンジー(P.troglodytes)、ラット(R.norvegicus)、出芽酵母(S.cerevisiae)、トキソプラズマ(T.gondii)と実に様々な生物種が利用可能であることがわかります。getSeq関数はBSgenomeオブジェクト 中の「single sequences」というあたりにリストアップされているchr...というものを全て抽出しています。したがって、例えばマウスゲノムは「chr1」以外に「chr1_random」や「chrUn_random」なども等価に取扱っている点に注意してください。
「ファイル」-「ディレクトリの変更」でファイルを保存したいディレクトリに移動し以下をコピペ。

1. 利用可能な生物種とRにインストール済みの生物種をリストアップしたい場合:

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

#本番（利用可能なパッケージをリストアップ；インストール済みとは限らない）
available.genomes()                        #このパッケージ中で利用可能なゲノムをリストアップ

#本番（インストール済みの生物種をリストアップ）
installed.genomes()                        #インストール済みの生物種をリストアップ

#後処理（パッケージ名でだいたいわかるがproviderやversionを分割して表示したい場合）
installed.genomes(splitNameParts=TRUE) #インストール済みの生物種をリストアップ
```

BSgenome

黒枠部分のコードをコピペ。R ver. 3.1.3 (Bioconductor ver. 3.0)で利用可能な生物種のパッケージ名をリストアップ。71個あることが分かる。Rのバージョンが古いとパッケージ数は少なくなる。

イントロ | 一般 | 配列取得 | ゲノム配列 | BSgenome NEW

BSgenomeパッケージを用いて様々な生物種のゲノム配列を取得するやり方を示します。ミヤマハタ
lyrata)、セイヨウミツバチ (A. mellifera)、シロイヌナズナ(A.thaliana)、ウシ(B.taurus)、線虫(C.elegans)、犬
(C.familiaris)、キイロショウジョウバエ(D.melanogaster)、ゼブラフィッシュ(D.rerio)、大腸菌(E.coli)、イトヨ
(G.aculeatus)、セキショクヤケイ(G.gallus)、ヒト(H.sapiens)、アカゲザル(M.mulatta)、マウス(M.musculus)、チ
ンパンジー(P.troglodytes)、ラット(R.norvegicus)、出芽酵母(S.cerevisiae)、トキソプラズマ(T.gondii)と実に様々
な生物種が利用可能であることがわかります。getSeq関数はBSgenomeオブジェクト 中の「single sequences」という
あたりにリストアップされているchr...というものを全て抽出しています。したがって、例えばマウスゲノムは「chr1」
以外に「chr1_random」や「chrUn_random」なども等価に取扱っている
「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

1. 利用可能な生物種とRにインストール済みの生物種をリストアップ

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

#本番（利用可能なパッケージをリストアップ；インストール
available.genomes()                                  #このパッケー

#本番（インストール済みの生物種をリストアップ）
installed.genomes()                                  #インストー

#後処理（パッケージ名でだいたいわかるがproviderやversi
installed.genomes(splitNameParts=TRUE) #インストー
```

R Console

```
[56] "BSgenome.Ptroglodytes.UCSC.panTro2"
[57] "BSgenome.Ptroglodytes.UCSC.panTro2.masked"
[58] "BSgenome.Ptroglodytes.UCSC.panTro3"
[59] "BSgenome.Ptroglodytes.UCSC.panTro3.masked"
[60] "BSgenome.Rnorvegicus.UCSC.rn4"
[61] "BSgenome.Rnorvegicus.UCSC.rn4.masked"
[62] "BSgenome.Rnorvegicus.UCSC.rn5"
[63] "BSgenome.Rnorvegicus.UCSC.rn5.masked"
[64] "BSgenome.Scerevisiae.UCSC.sacCer1"
[65] "BSgenome.Scerevisiae.UCSC.sacCer2"
[66] "BSgenome.Scerevisiae.UCSC.sacCer3"
[67] "BSgenome.Sscrofa.UCSC.susScr3"
[68] "BSgenome.Sscrofa.UCSC.susScr3.masked"
[69] "BSgenome.Tgondii.ToxoDB.7.0"
[70] "BSgenome.Tguttata.UCSC.taeGut1"
[71] "BSgenome.Tguttata.UCSC.taeGut1.masked"
> |
```

R Console

```
> available.genomes()
 [1] "BSgenome.Alyrata.JGI.v1"
 [2] "BSgenome.Amellifera.BeeBase.assem
 [3] "BSgenome.Amellifera.UCSC.apiMel2"
 [4] "BSgenome.Amellifera.UCSC.apiMel2.
 [5] "BSgenome.Athaliana.TAIR.04232008"
 [6] "BSgenome.Athaliana.TAIR.TAIR9"
 [7] "BSgenome.Btaurus.UCSC.bosTau3"
 [8] "BSgenome.Btaurus.UCSC.bosTau3.mas
 [9] "BSgenome.Btaurus.UCSC.bosTau4"
[10] "BSgenome.Btaurus.UCSC.bosTau4.mas
[11] "BSgenome.Btaurus.UCSC.bosTau6"
[12] "BSgenome.Btaurus.UCSC.bosTau6.mas
[13] "BSgenome.Celegans.UCSC.ce10"
[14] "BSgenome.Celegans.UCSC.ce2"
[15] "BSgenome.Celegans.UCSC.ce6"
[16] "BSgenome.Cfamiliaris.UCSC.canFam2
[17] "BSgenome.Cfamiliaris.UCSC.canFam2
[18] "BSgenome.Cfamiliaris.UCSC.canFam3
[19] "BSgenome.Cfamiliaris.UCSC.canFam3
[20] "BSgenome.Dmelanogaster.UCSC.dm2"
[21] "BSgenome.Dmelanogaster.UCSC.dm2.m
[22] "BSgenome.Dmelanogaster.UCSC.dm3"
[23] "BSgenome.Dmelanogaster.UCSC.dm3.m
[24] "BSgenome.Drerio.UCSC.danRer5"
[25] "BSgenome.Drerio.UCSC.danRer5.mask
[26] "BSgenome.Drerio.UCSC.danRer6"
[27] "BSgenome.Drerio.UCSC.danRer6.mask
[28] "BSgenome.Drerio.UCSC.danRer7"
```

R Console

```
[29] "BSgenome.Drerio.UCSC.danRer7.masked"
[30] "BSgenome.Ecoli.NCBI.20080805"
[31] "BSgenome.Gaculeatus.UCSC.gasAcu1"
[32] "BSgenome.Gaculeatus.UCSC.gasAcu1.masked"
[33] "BSgenome.Ggallus.UCSC.galGal3"
[34] "BSgenome.Ggallus.UCSC.galGal3.masked"
[35] "BSgenome.Ggallus.UCSC.galGal4"
[36] "BSgenome.Ggallus.UCSC.galGal4.masked"
[37] "BSgenome.Hsapiens.NCBI.GRCh38"
[38] "BSgenome.Hsapiens.UCSC.hg17"
[39] "BSgenome.Hsapiens.UCSC.hg17.masked"
[40] "BSgenome.Hsapiens.UCSC.hg18"
[41] "BSgenome.Hsapiens.UCSC.hg18.masked"
[42] "BSgenome.Hsapiens.UCSC.hg19"
[43] "BSgenome.Hsapiens.UCSC.hg19.masked"
[44] "BSgenome.Mfuro.UCSC.musFur1"
[45] "BSgenome.Mmulatta.UCSC.rheMac2"
[46] "BSgenome.Mmulatta.UCSC.rheMac2.masked"
[47] "BSgenome.Mmulatta.UCSC.rheMac3"
[48] "BSgenome.Mmulatta.UCSC.rheMac3.masked"
[49] "BSgenome.Mmusculus.UCSC.mm10"
[50] "BSgenome.Mmusculus.UCSC.mm10.masked"
[51] "BSgenome.Mmusculus.UCSC.mm8"
[52] "BSgenome.Mmusculus.UCSC.mm8.masked"
[53] "BSgenome.Mmusculus.UCSC.mm9"
[54] "BSgenome.Mmusculus.UCSC.mm9.masked"
[55] "BSgenome.Osativa.MSU.MSU7"
```

2013年12月にリリースされたヒトゲノム最新版(GRCh38)のRパッケージも利用可能です。

BSgenome

イントロ | 一般 | 配列取得 | ゲノム配列 | BSgenome NEW

BSgenomeパッケージを用いて様々な生物種のゲノム配列を取得するやり方を示します。ミヤマハ
lyrata)、セイヨウミツバチ (A. mellifera)、シロイヌナズナ(A.thaliana)、ウシ(B.taurus)、線虫(C.eleg
(C.familiaris)、キイロショウジョウバエ(D.melanogaster)、ゼブラフィッシュ(D.rerio)、大腸菌(E.co
(G.aculeatus)、セキショクヤケイ(G.gallus)、ヒト(H.sapiens)、アカゲザル(M.mulatta)、マウス(M.
ンパンジー(P.troglodytes)、ラット(R.norvegicus)、出芽酵母(S.cerevisiae)、トキソプラズマ(T.gond
な生物種が利用可能であることがわかります。getSeq関数はBSgenomeオブジェクト中の「single se
あたりにリストアップされているchr...というものを全て抽出しています。したがって、例えばマウスゲ
以外に「chr1_random」や「chrUn_random」なども等価に取扱っている点に注意してください。
「ファイル」-「ディレクトリの変更」でファイルを保存したいディレクトリに移動し以下をコピペ。

1. 利用可能な生物種とRにインストール済みの生物種をリストアップしたい場合:

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

#本番（利用可能なパッケージをリストアップ；インストール済みと
available.genomes()                        #このパッケージ中で

#本番（インストール済みの生物種をリストアップ）
installed.genomes()                        #インストール済みの

#後処理（パッケージ名でだいたいわかるがproviderやversionを分
installed.genomes(splitNameParts=TRUE) #インストール済みの
```

黒枠部分のコードをコピペ。数分程度かかります。実際にインストール済みのものは(このPC環境では)7パッケージであることがわかる。植物のシロイヌナズナ(*Arabidopsis thaliana*)のパッケージは推奨手順通りにインストール作業をしたヒトは存在するはずです。が、私もインストールされてなかったり…しますので、なければ個別インストールで対応してください。

R Console

```
[68] "BSgenome.Sscrofa.UCSC.susScr3.masked"
[69] "BSgenome.Tgondii.ToxoDB.7.0"
[70] "BSgenome.Tguttata.UCSC.taeGut1"
[71] "BSgenome.Tguttata.UCSC.taeGut1.masked"
> installed.genomes()                      #イン$
[1] "BSgenome.Athaliana.TAIR.TAIR9"
[2] "BSgenome.Celegans.UCSC.ce2"
[3] "BSgenome.Celegans.UCSC.ce6"
[4] "BSgenome.Drerio.UCSC.danRer7"
[5] "BSgenome.Hsapiens.NCBI.GRCh38"
[6] "BSgenome.Hsapiens.UCSC.hg19"
[7] "BSgenome.Mmusculus.UCSC.mm10"
>
```

個別インストール

パッケージの個別インストール方法。パッケージ名部分を変更すれば、基本どのパッケージのインストールにも対応可能です。
例：BSgenome.Athaliana.TAIR.TAIR9。

- インストール | R本体 | 最新版 | Win用 (last modified 2015/03/22)推奨 NEW
- インストール | R本体 | 最新版 | Mac用 (last modified 2015/04/01)推奨 NEW
- インストール | R本体 | 過去版 | Win用 (last modified 2015/03/22) NEW
- インストール | R本体 | 過去版 | Mac用 (last modified 2015/03/22) NEW
- インストール | Rパッケージ | ほぼ全て(20GB以上?!) (last modified 2015/03/2
- インストール | Rパッケージ | 必要最小限プラスアルファ(数GB?!) (last modifie
- インストール | Rパッケージ | 必要最小限(数GB?!) (last modified 2015/03/23)
- インストール | Rパッケージ | 個別 modified 2015/03/23) NEW
- (削除予定)Rのインストールと起動 (last modified 2015/04/02) NEW
- (削除予定)個別パッケージのインストール (last modified 2015/02/20)
- 基本的な利用法 (last modified 2015/04/03) NEW
- サンプルデータ (last modified 2015/02/15)

インストール | Rパッケージ | 個別 NEW

多くのBSgenome系パッケージやTxDb系のパッケージは、「ほぼ全て」の手順ではインストールされません。理由は、BSgenomeはゲノム配列情報のパッケージなので、ヒトゲノムの様々なバージョン、マウスゲノム、ラットゲノムなどを全部入れると大変なことになるからです。それでもピンポイントで必要に迫られる局面もあると思いますので、ここではRのパッケージを個別にインストールするやり方を示します。

1. ゼブラフィッシュゲノムのパッケージ(BSgenome.Drerio.UCSC.danRer7)をインストールしたい場合:

400MB程度あります....。

```
param <- "BSgenome.Drerio.UCSC.danRer7"#パッケージ名を指定

#本番
source("http://bioconductor.org/biocLite.R")#おまじない
biocLite(param, suppressUpdates=TRUE)  #おまじない
```

Contents

- パッケージ
  - CRANとBioconductor
  - 推奨パッケージインストール手順のおさらい
  - ゲノム情報パッケージBSgenomeの概観
  - ヒトゲノム情報パッケージの解析
- 2連続塩基出現頻度解析(CpG解析)、k-mer解析
  - 課題
  - GC含量の違いを考慮(連続塩基の種類ごとに期待値が異なる)
  - 作図(box plot)
- その他
  - 数式の感覚を理解
  - Sequence logos (Schneider and Stephens, 1990)
  - プロモーター配列取得

BSgenome

2013年12月にリリースされたヒトゲノム最新版(GRCh38)のRパッケージを入力、multi-FASTAファイルを出力として得る。作業ディレクトリはどこでもよいが基本はデスクトップ上のhoge。数分かかるが、約3.3GBのファイルが生成される。**決してテキストエディタで開かないで!**

イントロ | 一般 | 配列取得 | ゲノム配列 | BSgenome NEW

BSgenomeパッケージを用いて様々な生物種のゲノム配列を取得するやり方を示します。ミヤマハタザ
lyrata)、セイヨウミツバチ (A. mellifera)、シロイヌナズナ(A.thaliana)、ウシ(B.taurus)、線虫(C.elegans
(C.familiaris)、キイロショウジョウバエ(D.melanogaster)、ゼブラフィッシュ(D.rerio)、大腸菌(E.coli)、
(G.aculeatus)、セキショクヤケイ(G.gallus)、ヒト(H.sapiens)、アカゲザル(M.mulatta)、マウス(M.mus
ンパンジー(P.troglodytes)、ラット(R.norvegicus)、出芽酵母(S.cerevisiae)、トキソプラズマ(T.gondii)
な生物種が利用可能であることがわかります。getSeq関数はBSgenomeオブジェクト 中の「single sequences」という
あたりにリストアップされているchr、というものを全て抽出しています。したがって、例えばマウスゲノムは「chr1」
以外に「chr1_random
「ファイル」-「ディレクト

1. 利用可能な生物種とR

```
#必要なパッケージを
library(BSgenome)

#本番（利用可能なパ
available.genomes

#本番（インストール
installed.genomes

#後処理（パッケージ
installed.genomes
```

9. インストール済みのヒト("BSgenome.Hsapiens.NCBI.GRCh38")のゲノム配列をmulti-FASTAファイルで保存したい場合:

2013年12月にリリースされたGenome Reference Consortium GRCh38です。R ver. 3.1.0とBioconductor ver. 2.14以上の環境で実行可能です。

```
out_f <- "hoge9.fasta"                    #出力ファイル名を指定してout_fに格納
param <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定

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

#前処理(paramで指定したパッケージ中のオブジェクト名をgenomeに統一)
#tmp <- unlist(strsplit(param, ".", fixed=TRUE))[2]#paramで指定した文字列からオブジェクト名を取得し
tmp <- ls(paste("package", param, sep=":"))#paramで指定したパッケージで利用可能なオブジェクト名を取得
genome <- eval(parse(text=tmp))           #文字列tmpをRオブジェクトとしてgenomeに格納(パッケージ中には
genome                                    #確認してるだけです

#本番
fasta <- getSeq(genome)                   #ゲノム塩基配列情報を抽出した結果をfastaに格納
names(fasta) <- seqnames(genome)          #description情報を追加している

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

BSgenome

出力ファイルの内容はfastaオブジェクトに格納されている。慣れればfastaオブジェクトの中身を眺めるほうが全体像をつかみやすい。

9. インストール済みのヒト("BSgenome.Hsapiens.NCBI.GRCh38")のゲノム配列をmulti-FASTA

2013年12月にリリースされたGenome Reference Consortium GRCh38です。R ver. 3.1.0とBioconductor ver. 2.14以上の環境で実行可能です。

```
out_f <- "hoge9.fasta"                    #出力ファイル名を指定してout_fに格納
param <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定

#必要なパッケージをロード
library(param, character.only=T)

#前処理(paramで指定したパッケージ中のオ
#tmp <- unlist(strsplit(param, "."),
tmp <- ls(paste("package", param, sep
genome <- eval(parse(text=tmp))
genome

#本番
fasta <- getSeq(genome)
names(fasta) <- seqnames(genome)

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

R Console

```
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", widt$
> fasta
  A DNAStringSet instance of length 455
          width seq                    names
  [1] 248956422 NNNNNNNNN...NNNNNNNNN 1
  [2] 242193529 NNNNNNNNN...NNNNNNNNN 2
  [3] 198295559 NNNNNNNNN...NNNNNNNNN 3
  [4] 190214555 NNNNNNNNN...NNNNNNNNN 4
  [5] 181538259 NNNNNNNNN...NNNNNNNNN 5
  ...       ... ...
[451]    200773 TCTACTCTC...GGGGAATTC HSCHR19KIR_FH08_B...
[452]    170148 TTTCTTTCT...GGGGAATTC HSCHR19KIR_FH13_A...
[453]    215732 TGTGGTGAG...GGGGAATTC HSCHR19KIR_FH13_B...
[454]    170537 TCTACTCTC...GGGGAATTC HSCHR19KIR_FH15_A...
[455]    177381 GATCTATCT...GGGGAATTC HSCHR19KIR_RP5_B_...
> |
```

BSgenome

1～22番染色体のみ取扱いたい場合。染色体番号の数が大きくなるほど配列長が短くなっている傾向が一目瞭然ですね。

9. インストール済みのヒト("BSgenome.Hsapiens.NCBI.GRCh38")のゲノム配列をmulti-FASTAファイ

2013年12月にリリースされたGenome Reference Consortium GRCh38です。R ver. 3.1.0とBioconductor ver. 2.14以上の環境で実行可能です。

```
out_f <- "hoge9.fasta"                    #出力ファイル名を指定してout_fに格納
param <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定

#必要なパッケージをロード
library(param, character.only=T)

#前処理(paramで指定したパッケージ中のオ
#tmp <- unlist(strsplit(param, "."),
tmp <- ls(paste("package", param, sep
genome <- eval(parse(text=tmp))
genome

#本番
fasta <- getSeq(genome)
names(fasta) <- seqnames(genome)

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

R Console

```
[454]    170537 TCTACTCTC...GGGGAATTC HSCHR19KIR_FH15_A...
[455]    177381 GATCTATCT...GGGGAATTC HSCHR19KIR_RP5_B_...
> fasta[1:22]
  A DNAStringSet instance of length 22
      width seq                    names
 [1] 248956422 NNNNNNNNNN...NNNNNNNNN 1
 [2] 242193529 NNNNNNNNNN...NNNNNNNNN 2
 [3] 198295559 NNNNNNNNNN...NNNNNNNNN 3
 [4] 190214555 NNNNNNNNNN...NNNNNNNNN 4
 [5] 181538259 NNNNNNNNNN...NNNNNNNNN 5
 ...       ... ...
[18]  80373285 NNNNNNNNNN...NNNNNNNNN 18
[19]  58617616 NNNNNNNNNN...NNNNNNNNN 19
[20]  64444167 NNNNNNNNNN...NNNNNNNNN 20
[21]  46709983 NNNNNNNNNN...NNNNNNNNN 21
[22]  50818468 NNNNNNNNNN...NNNNNNNNN 22
>
```

BSgenome

9. インストール済みのヒト("BSgenome.Hsapiens.NCBI.GRCh38")のゲノム配列をmulti-FASTAファイル

2013年12月にリリースされたGenome Reference Consortium GRCh38です。R ver. 3.1.0とBioconductor v
実行可能です。

```
out_f <- "hoge9.fasta"                    #出力ファイル名を指定してout_fに格納
param <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定

#必要なパッケージをロード
library(param, character.only=T)

#前処理(paramで指定したパッケージ中のオ
#tmp <- unlist(strsplit(param, "."),
tmp <- ls(paste("package", param, sep
genome <- eval(parse(text=tmp))
genome

#本番
fasta <- getSeq(genome)
names(fasta) <- seqnames(genome)

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

X, Y, およびミトコンドリア配列も含めたい場合。配列の並びの確認は試行錯誤。最初からわかっていたわけではありません。R画面上で眺めるほうが、全体像を把握しやすい。

R Console

```
[21]  46709983 NNNNNNNNNN...NNNNNNNNN 21
[22]  50818468 NNNNNNNNNN...NNNNNNNNN 22
> fasta[1:25]
  A DNAStringSet instance of length 25
         width seq                    names
 [1] 248956422 NNNNNNNNNN...NNNNNNNNN 1
 [2] 242193529 NNNNNNNNNN...NNNNNNNNN 2
 [3] 198295559 NNNNNNNNNN...NNNNNNNNN 3
 [4] 190214555 NNNNNNNNNN...NNNNNNNNN 4
 [5] 181538259 NNNNNNNNNN...NNNNNNNNN 5
 ...       ... ...
[21]  46709983 NNNNNNNNNN...NNNNNNNNN 21
[22]  50818468 NNNNNNNNNN...NNNNNNNNN 22
[23] 156040895 NNNNNNNNNN...NNNNNNNNN X
[24]  57227415 NNNNNNNNNN...NNNNNNNNN Y
[25]     16569 GATCACAGGT...ATCACGATG MT
> |
```

BSgenome

X, Y, およびミトコンドリア配列までのサブセットをhoge10.fastaで保存したい場合。①上矢印キーを何回か押して、ファイルに保存するためのコマンドを出し、水色下線部分を変更すればよい。

9. インストール済みのヒト("BSgenome.Hsapiens.NCBI.GRCh38")のゲノム配列をmulti-FASTAフ

2013年12月にリリースされたGenome Reference Consortium GRCh38です。R ver. 3.1.0とBioconduct
実行可能です。

```
out_f <- "hoge9.fasta"                    #出力ファイル名を指定してout_fに格納
param <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定

#必要なパッケージをロード
library(param, character.only=T)

#前処理(paramで指定したパッケージ中のオ
#tmp <- unlist(strsplit(param, "."),
tmp <- ls(paste("package", param, sep
genome <- eval(parse(text=tmp))
genome

#本番
fasta <- getSeq(genome)
names(fasta) <- seqnames(genome)

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

①

R Console

```
[21]  46709983 NNNNNNNNNN...NNNNNNNNN 21
[22]  50818468 NNNNNNNNNN...NNNNNNNNN 22
> fasta[1:25]
  A DNAStringSet instance of length 25
      width seq                    names
 [1] 248956422 NNNNNNNNNN...NNNNNNNNN 1
 [2] 242193529 NNNNNNNNNN...NNNNNNNNN 2
 [3] 198295559 NNNNNNNNNN...NNNNNNNNN 3
 [4] 190214555 NNNNNNNNNN...NNNNNNNNN 4
 [5] 181538259 NNNNNNNNNN...NNNNNNNNN 5
 ...       ... ...
[21]  46709983 NNNNNNNNNN...NNNNNNNNN 21
[22]  50818468 NNNNNNNNNN...NNNNNNNNN 22
[23] 156040895 NNNNNNNNNN...NNNNNNNNN X
[24]  57227415 NNNNNNNNNN...NNNNNNNNN Y
[25]     16569 GATCACAGGT...ATCACGATG MT
> writeXStringSet(fasta, file=out_f, format="fasta", widt$
```

BSgenome

実行後にhoge9.fastaよりも若干ファイルサイズの小さいhoge10.fastaが生成されていることが確認できるはず。**決してテキストエディタで開かないで!**

9. インストール済みのヒト("BSgenome.Hsapiens.NCBI.GRCh38")のゲノム配列をmulti-FASTAファイル

2013年12月にリリースされたGenome Reference Consortium GRCh38です。R ver. 3.1.0とBioconductor
実行可能です。

```
out_f <- "hoge9.fasta"                    #出力ファイル名を指定してout_fに格納
param <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定

#必要なパッケージをロード
library(param, character.only=T)

#前処理(paramで指定したパッケージ中のオ
#tmp <- unlist(strsplit(param, "."),
tmp <- ls(paste("package", param, sep
genome <- eval(parse(text=tmp))
genome

#本番
fasta <- getSeq(genome)
names(fasta) <- seqnames(genome)

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

R Console

```
[22]  50818468 NNNNNNNNNN...NNNNNNNNN 22
> fasta[1:25]
  A DNAStringSet instance of length 25
         width seq                    names
 [1] 248956422 NNNNNNNNNN...NNNNNNNNN 1
 [2] 242193529 NNNNNNNNNN...NNNNNNNNN 2
 [3] 198295559 NNNNNNNNNN...NNNNNNNNN 3
 [4] 190214555 NNNNNNNNNN...NNNNNNNNN 4
 [5] 181538259 NNNNNNNNNN...NNNNNNNNN 5
 ...       ... ...
[21]  46709983 NNNNNNNNNN...NNNNNNNNN 21
[22]  50818468 NNNNNNNNNN...NNNNNNNNN 22
[23] 156040895 NNNNNNNNNN...NNNNNNNNN X
[24]  57227415 NNNNNNNNNN...NNNNNNNNN Y
[25]     16569 GATCACAGGT...ATCACGATG MT
> writeXStringSet(fasta[1:25], file="hoge10.fasta", forma$
>
```

参考

様々な記述形式があります。やらなくていいです。
決してテキストエディタで開かないで!

BSgenome

イントロ | 一般 | 配列取得 | ゲノム配列 | BSgenome NEW

BSgenomeパッケージを用いて様々な生物種のゲノム配列を取得するやり方を示します。ミヤマハタザオ (A. lyr… セイヨウミツバチ (A. mellifera)、シロイヌナズナ(A.thaliana)、ウシ(B.taurus)、線虫(C.elegans)、ボ… (C… (G.a… ンパンジ… な生物… あたり… 以外に… 「ファイ…

10. インストール済みのヒト("BSgenome.Hsapiens.NCBI.GRCh38")のゲノム配列のmulti-FASTAファイルで保存したい場合:

一部を抽出して保存するやり方です。このパッケージ中の染色体の並びが既知(chr1, 2, ..., chr22, chrX, chrY, and MT)であるという前提です。

```
out_f <- "hoge10.fasta"                  #出力ファイル名を指定してout_fに格納
param <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定
param_range <- 1:25                      #抽出したい範囲を指定

#必要なパッケージをロード
library(param, character.only=T)

#前処理(paramで指定したパッケージ中…
#tmp <- unlist(strsplit(param, "…
tmp <- ls(paste("package", param,
genome <- eval(parse(text=tmp))
genome

#本番
fasta <- getSeq(genome)
names(fasta) <- seqnames(genome)

#後処理(フィルタリング)
obj <- param_range
fasta <- fasta[obj]
fasta
```

1. 利用…

```
#必…
libr…

#本…
avai…

#本…
inst…

#後…
inst…
```

R Console

```
          width seq                                    names
 [1] 248956422 NNNNNNNNNNN...NNNNNNNNNN 1
 [2] 242193529 NNNNNNNNNNN...NNNNNNNNNN 2
 [3] 198295559 NNNNNNNNNNN...NNNNNNNNNN 3
 [4] 190214555 NNNNNNNNNNN...NNNNNNNNNN 4
 [5] 181538259 NNNNNNNNNNN...NNNNNNNNNN 5
 ...       ... ...
[21]  46709983 NNNNNNNNNNN...NNNNNNNNNN 21
[22]  50818468 NNNNNNNNNNN...NNNNNNNNNN 22
[23] 156040895 NNNNNNNNNNN...NNNNNNNNNN X
[24]  57227415 NNNNNNNNNNN...NNNNNNNNNN Y
[25]     16569 GATCACAGGTC...CATCACGATG MT
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", width=$
>
```

BSgenome

26番目以降の配列は、ヒトゲノムの一部ではあるものの、まだ割り当てられる染色体が定まっていないものたちです。メタゲノム解析などでヒトゲノムにマップされないリードのみ取扱いたい場合には、利用可能な全配列をマッピング時のリファレンスとして用いるのが自然だと思います。

9. インストール済みのヒト("BSgenome.Hsapiens.NCBI.GRCh38")のゲノム配列をmul

2013年12月にリリースされたGenome Reference Consortium GRCh38です。R ver. 3.1.
実行可能です。

```
out_f <- "hoge9.fasta"                    #出力ファイル名を指定してou
param <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定

#必要なパッケージをロード
library(param, character.only=T)

#前処理(paramで指定したパッケージ中のオ
#tmp <- unlist(strsplit(param, "."),
tmp <- ls(paste("package", param, sep
genome <- eval(parse(text=tmp))
genome

#本番
fasta <- getSeq(genome)
names(fasta) <- seqnames(genome)

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

R Console

```
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", widt$
> fasta
  A DNAStringSet instance of length 455
          width seq                    names
  [1] 248956422 NNNNNNNNN...NNNNNNNNN 1
  [2] 242193529 NNNNNNNNN...NNNNNNNNN 2
  [3] 198295559 NNNNNNNNN...NNNNNNNNN 3
  [4] 190214555 NNNNNNNNN...NNNNNNNNN 4
  [5] 181538259 NNNNNNNNN...NNNNNNNNN 5
  ...       ... ...
[451]    200773 TCTACTCTC...GGGGAATTC HSCHR19KIR_FH08_B...
[452]    170148 TTTCTTTCT...GGGGAATTC HSCHR19KIR_FH13_A...
[453]    215732 TGTGGTGAG...GGGGAATTC HSCHR19KIR_FH13_B...
[454]    170537 TCTACTCTC...GGGGAATTC HSCHR19KIR_FH15_A...
[455]    177381 GATCTATCT...GGGGAATTC HSCHR19KIR_RP5_B_...
> |
```

Contents

- パッケージ
  - CRANとBioconductor
  - 推奨パッケージインストール手順のおさらい
  - ゲノム情報パッケージBSgenomeの概観
  - ヒトゲノム情報パッケージの解析
- 2連続塩基出現頻度解析(CpG解析)、k-mer解析
  - 課題
  - GC含量の違いを考慮(連続塩基の種類ごとに期待値が異なる)
  - 作図(box plot)
- その他
  - 数式の感覚を理解
  - Sequence logos (Schneider and Stephens, 1990)
  - プロモーター配列取得

Rで調べることができます

ヒトゲノム中のCpG出現確率は低い

- 全部で16通りの2連続塩基の出現頻度分布を調べると、CGとなる確率の実測値(0.986%)は期待値(4.2%)よりもかなり低い
- 期待値
  - ゲノム中のGC含量を考慮した場合：約41%(A:0.295, C:0.205, G: 0.205, T:0.295)なので、0.205×0.205= 4.2%
  - ゲノム中のGC含量を考慮しない場合：50%(A:0.25, C:0.25, G: 0.25, T:0.25)なので、0.25×0.25= 6.25%

2連続塩基の出現頻度

全貌を把握可能なhoge4.faを作業ディレクトリにダウンロードして実行しましょう。

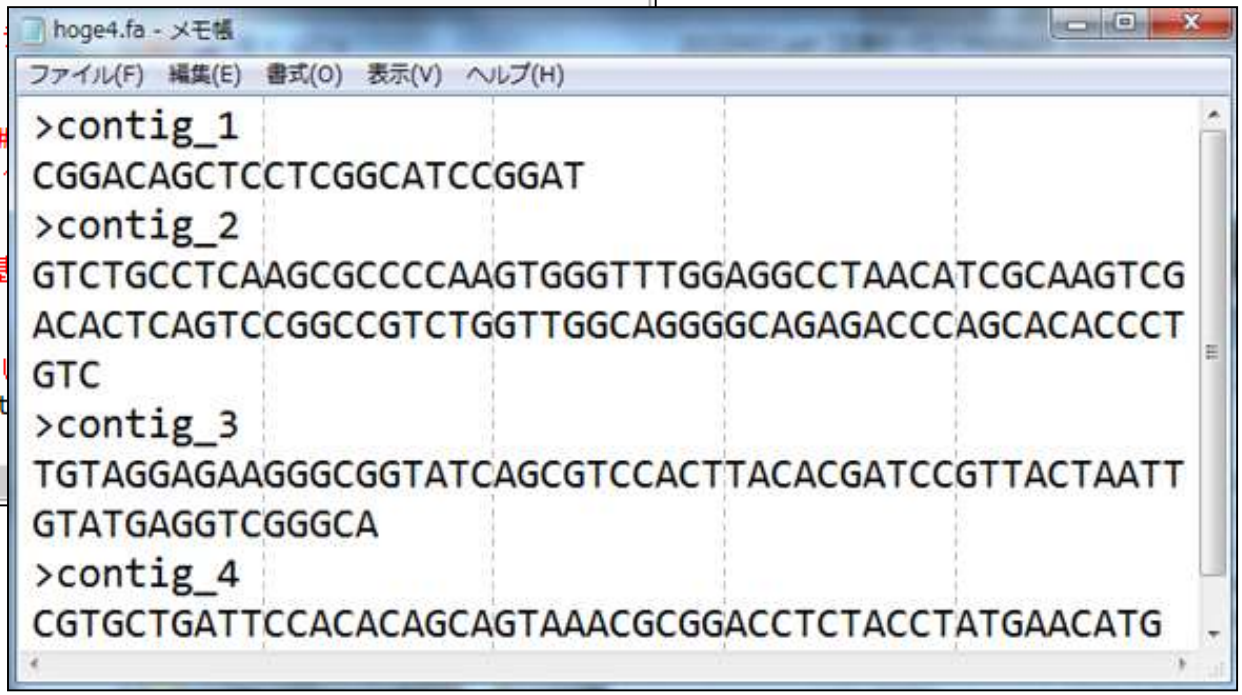
イントロ | 一般 | 2連続塩基の出現頻度情報を取得 NEW

multi-FASTA形式ファイルを読み込んで、"AA", "AC", "AG", "AT", "CA", "CC", "CG", "CT", "GA", "GC", "GG", "GT", "TA", "TC", "TG", "TT" の計$4^2 = 16$通りの2連続塩基の出現頻度を調べるやり方を示します。ヒトゲノムで"CG"の割合が期待値よりも低い(Lander et al., 2001; Saxonov et al., 2006)ですが、それを簡単に検証できます。「ファイル」ー「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピペ。

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

タイトル通りの出現頻度です。

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

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

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

#本番
out <- dinucleotideFrequency(fasta)         #2連続塩基

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



2連続塩基の出現頻度

右クリックでダウンロードし、作業ディレクトリ中にhoge4.faがあることを確認。Macのヒトは.txtが付与されてしまう拡張子問題の解決も忘れずに!

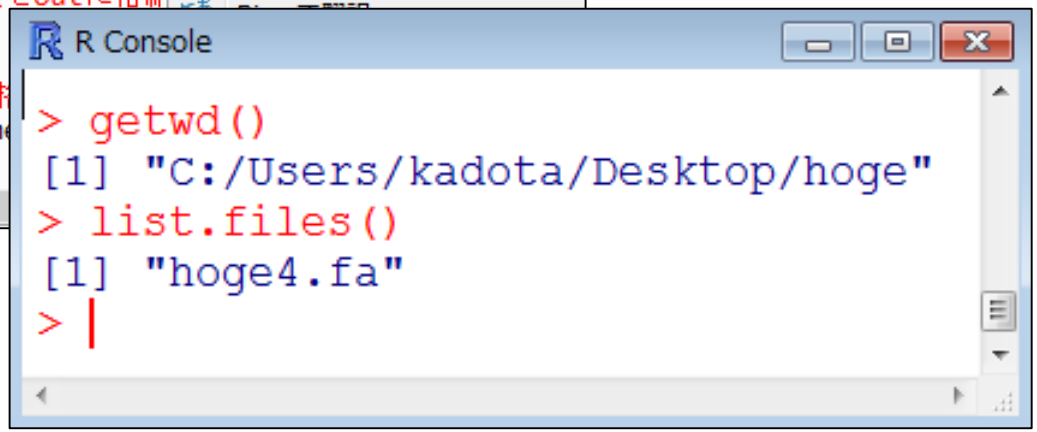
イントロ | 一般 | 2連続塩基の出現頻度情報を取得

multi-FASTA形式ファイルを読み込んで、"AA", "AC", "AG", "AT", "CA", "CC", "CG", "CT", "GA", "GC", "GG", "GT", "TA", "TC", "TG", "TT" の計$4^2 = 16$通りの2連続塩基の出現頻度を調べるやり方を示します。ヒトゲノムで"CG"の割合が期待値よりも低い(Lander et al., 2001; Saxonov et al., 2006)ですが、それを簡単に検証できます。
「ファイル」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピペ。

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

タイトル通りの出現頻度です。

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

#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
fasta                                       #確認してるだけです

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

#ファイルに保存
tmp <- cbind(names(fasta), out)             #保存したい情報をtmpに格
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.name
```



2連続塩基の出現頻度

Internet ExplorerのヒトはCTRLとALTキーを押しながらコードの枠内で**左クリック**すると全選択できます。基本はコピペ。

イントロ | 一般 | 2連続塩基の出現頻度情報を取得 NEW

multi-FASTA形式ファイルを読み込んで、"AA", "AC", "AG", "AT", "CA", "CC", "CG", "CT", "GA", "GC", "GG", "GT", "TA", "TC", "TG", "TT" の計$4^2 = 16$通りの2連続塩基の出現頻度を調べるやり方を示します。ヒトゲノムで"CG"の割合が期待値よりも低い(Lander et al., 2001; Saxonov et al., 2006)ですが、それを簡単に検証できます。
「ファイル」ー「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピペ。

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

タイトル通りの出現頻度です。

```
in_f <- "hoge4.fa"                          #入力ファイ
out_f <- "hoge1.txt"                        #出力ファイ

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

#入力ファイルの読み込み
fasta <- readDNAStringSet(in_f, format="fasta")#i
fasta                                       #確認してる

#本番
out <- dinucleotideFrequency(fasta)     #2連続塩基(

#ファイルに保存
tmp <- cbind(names(fasta), out)             #保存したい
write.table(tmp, out_f, sep="\t", append=F, quote
```

R Console

```
> out <- dinucleotideFrequency(fasta)     #連続塩基$
>
> #ファイルに保存
> tmp <- cbind(names(fasta), out)         #保存した$
> write.table(tmp, out_f, sep="\t", append=F, quot$
> tmp
                AA  AC  AG  AT  CA  CC  CG  CT
[1,] "contig_1" "0" "1" "1" "2" "2"  "2"  "3" "2"
[2,] "contig_2" "4" "6" "9" "1" "11" "11" "5" "6"
[3,] "contig_3" "2" "4" "5" "4" "4"  "2"  "5" "2"
[4,] "contig_4" "3" "6" "2" "3" "5"  "3"  "3" "4"
     GA  GC  GG   GT  TA  TC  TG  TT
[1,] "2" "2" "3"  "0" "0" "3" "0" "0"
[2,] "4" "9" "10" "8" "1" "8" "6" "3"
[3,] "4" "3" "7"  "6" "6" "4" "3" "3"
[4,] "3" "3" "1"  "2" "3" "2" "4" "1"
>
```

2連続塩基の出現頻度

出力ファイルは、配列ごと(この場合コンティグごと)に16種類の2連続塩基の出現頻度をカウントしたものです。

イントロ | 一般 | 2連続塩基の出現頻度情報を取得 NEW

multi-FASTA形式ファイルを読み込んで、"AA", "AC", "AG", "AT
"GT", "TA", "TC", "TG", "TT" の計$4^2 = 16$通りの2連続塩基の出
で"CG"の割合が期待値よりも低い(Lander et al., 2001; Saxonov et
「ファイル」ー「ディレクトリの変更」で解析したいファイルを置いてあ

1. イントロ | 一般 | ランダムな塩基配列を作成の4.を実行して得ら

タイトル通りの出現頻度です。

```
in_f <- "hoge4.fa"                          #入力ファ
out_f <- "hoge1.txt"                        #出力ファ

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

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

#本番
out <- dinucleotideFrequency(fasta)     #2連続塩基の出現頻度情報をoutに格納

#ファイルに保存
tmp <- cbind(na
write.table(tmp
```

hoge4.fa - メモ帳

ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

```
>contig_1
CGGACAGCTCCTCGGCATCCGGAT
>contig_2
GTCTGCCTCAAGCGCCCCAAGTGGGTTTGGAGGCCTAACATCGCAAGTCG
ACACTCAGTCCGGCCGTCTGGTTGGCAGGGGCAGAGACCCAGCACACCCT
GTC
>contig_3
TGTAGGAGAAGGGCGGTATCAGCGTCCACTTACACGATCCGTTACTAATT
GTATGAGGTCGGGCA
>contig_4
CGTGCTGATTCCACACAGCAGTAAACGCGGACCTCTACCTATGAACATG
```

出力:hoge1.txt

	AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
contig_1	0	1	1	2	2	2	3	2	2	2	3	0	0	3	0	0
contig_2	4	6	9	1	11	11	5	6	4	9	10	8	1	8	6	3
contig_3	2	4	5	4	4	2	5	2	4	3	7	6	6	4	3	3
contig_4	3	6	2	3	5	3	3	4	3	3	1	2	3	2	4	1

2連続塩基の出現確率

出力ファイルは、配列ごと(この場合コンティグごと)に16種類の2連続塩基の出現確率をカウントしたものです。

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

出現頻度ではなく、出現確率を得るやり方です。

```
in_f <- "hoge4.fa"                          #入力ファ
out_f <- "hoge2.txt"                        #出力ファ

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

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

#本番
out <- dinucleotideFrequency(fasta, as.prob=T)#2

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

hoge4.fa - メモ帳

ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

```
>contig_1
CGGACAGCTCCTCGGCATCCGGAT
>contig_2
GTCTGCCTCAAGCGCCCCAAGTGGGTTTGGAGGCCTAACATCGCAAGTCG
ACACTCAGTCCGGCCGTCTGGTTGGCAGGGGCAGAGACCCAGCACACCCT
GTC
>contig_3
TGTAGGAGAAGGGCGGTATCAGCGTCCACTTACACGATCCGTTACTAATT
GTATGAGGTCGGGCA
>contig_4
CGTGCTGATTCCACACAGCAGTAAACGCGGACCTCTACCTATGAACATG
```

出力:hoge2.txt

	AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
contig_1	0.0%	4.3%	4.3%	8.7%	8.7%	8.7%	13.0%	8.7%	8.7%	8.7%	13.0%	0.0%	0.0%	13.0%	0.0%	0.0%
contig_2	3.9%	5.9%	8.8%	1.0%	10.8%	10.8%	4.9%	5.9%	3.9%	8.8%	9.8%	7.8%	1.0%	7.8%	5.9%	2.9%
contig_3	3.1%	6.3%	7.8%	6.3%	6.3%	3.1%	7.8%	3.1%	6.3%	4.7%	10.9%	9.4%	9.4%	6.3%	4.7%	4.7%
contig_4	6.3%	12.5%	4.2%	6.3%	10.4%	6.3%	6.3%	8.3%	6.3%	6.3%	2.1%	4.2%	6.3%	4.2%	8.3%	2.1%

2連続塩基の出現確率

ヒトゲノムRパッケージを入力とすることもできます。一見ややこしいですが、fastaオブジェクトの作成までを「お約束の手順」だと思えばいいのです。

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

出現頻度ではなく、出現確率を得るやり方です。

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

#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(i
fasta

#本番
out <- dinucleotideFrequenc

#ファイルに保存
tmp <- cbind(names(fasta),
write.table(tmp, out_f, sep
```

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

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

```
out_f <- "hoge7.txt"                          #出力ファイル名を指定してout_fに格納
param <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定

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

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

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

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

2連続塩基の出現確率

9.は、7.の記述が気になるヒト用。パッケージ名をベタで書いています。9.のtmpの中身はBSgenome.Hsapiens.NCBI.GRCh38中で利用可能なオブジェクト名です。

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

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

```
out_f <- "hoge7.txt"                          #出力ファイル名を指定してout_fに格納
param <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定

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

#前処理(paramで指定したパッケージ中のオブジェクト名をgenomeに統一)
tmp <- ls(paste("package", param, sep=":"))#paramで指定したパッケージで利用可能なオブジェク
genome <- eval(parse(text=tmp))               #文字列tmpをRオブジェクトとしてgenomeに格納(パッケ
fasta <- getSeq(genome)
names(fasta) <- seqnames(ge
fasta

#本番
out <- dinucleotideFrequency(f

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

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

基本的に7.と同じです。7.の手順がややこしいと思う人向けの解説用です。簡単に言えば、パッケージ名を2回書かなくて済むテクニックを用いているだけです。もう少し詳細に書くと、BSgenomeパッケージはlibrary関数で読み込んだ後にパッケージ名と同じ名前のオブジェクトを利用できるようになります。例えばBSgenome.Hsapiens.NCBI.GRCh38パッケージの場合は、BSgenome.Hsapiens.NCBI.GRCh38という名前のオブジェクトを利用できるようになります。ベタで書くと2回BSgenome.Hsapiens.NCBI.GRCh38を記述する必要性があるため、間違う確率が上昇します。7.のように一見ややこしく書けば、結果的に一度のみの記述で済むのです。

```
out_f <- "hoge9.txt"                          #出力ファイル名を指定してout_fに格納

#必要なパッケージをロード
library(Biostrings)                           #パッケージの読み込み
library(BSgenome.Hsapiens.NCBI.GRCh38) #パッケージの読み込み

#前処理(paramで指定したパッケージ中のオブジェクト名をgenomeに統一)
tmp <- ls("package:BSgenome.Hsapiens.NCBI.GRCh38")#指定したパッケージで利用可能なオブジェク
genome <- eval(parse(text=tmp))               #文字列tmpをRオブジェクトとしてgenomeに格納(パッケ
fasta <- getSeq(genome)                       #ゲノム塩基配列情報を抽出した結果をfastaに格納
names(fasta) <- seqnames(genome)              #description情報を追加している
fasta                                         #確認してるだけです
```

2連続塩基の出現確率

2分強かかります。CGの連続塩基が他に比べて確かに低いことがわかります。

出力：hoge7.txt

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

2連続塩基の出現頻度と確率

染色体ごとではなく、全てをひとまとめにするやり方です。連続塩基の出現頻度順にソートしてCGが少ないことを確かめています。

8. BSgenomeパッケージ中のヒトゲノム配列("BSgenome.Hsapiens.NCBI.GRCh38")の場合:

全配列を合算して、連続塩基ごとの出現頻度(frequency)と出現確率(probability)を出力するやり方です。
dinucleotideFrequency関数中の「simplify.as="collapsed"」オプションでも一応実行できますが、桁が多くなりすぎて「整数オーバーフロー」問題が起きたのでやめてます。

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

#前処理(paramで指定したパッケージ中のオブジェクト名をgenomeに統一)
tmp <- ls(paste("package", param, sep=":"))#paramで指定したパッケージで利用可能なオブジ
genome <- eval(parse(text=tmp))        #文字列
fasta <- getSeq(genome)                #ゲノム
names(fasta) <- seqnames(genome)       #descr
fasta                                  #確認し

#本番
hoge <- dinucleotideFrequency(fasta, as.prob=
frequency <- colSums(hoge)             #列ごと
probability <- frequency / sum(frequency)#出現
frequency                              #中身を
sort(frequency, decreasing=F)          #値の小

#ファイルに保存
tmp <- cbind(names(frequency), frequency, pro
write.table(tmp, out_f, sep="\t", append=F, q
```

R Console

```
       TT
299351073
> sort(frequency, decreasing=F)            #値の小さい$
       CG        GC        AC        GT        CC
 30979743 130065644 153830681 154194068 158048073
       GG        TC        GA        TA        CT
159302235 181782675 182772932 197567087 213517855
       AG        CA        TG        AT        AA
213785914 221181041 222266728 233904527 296763858
       TT
299351073
>
> #ファイルに保存
> tmp <- cbind(names(frequency), frequency, probabil$
> write.table(tmp, out_f, sep="\t", append=F, quote=$
>
```

k連続塩基解析

2連続塩基の解析は、k=2のときのk連続塩基の解析(k-mer解析)と同じです。

- 比較ゲノム解析
  - k=3 or 4付近の値を用いてゲノムごとの頻度情報を取得し、類似性尺度として利用
- アセンブル(ゲノムやトランスクリプトーム)
  - k=25～50付近の値を用いてde Bruijnグラフを作成
  - k-mer頻度グラフを作成して眺め、Heterozygosityの有無などを調査
- モチーフ解析
  - 転写開始点の上流配列解析。古細菌の上流50塩基に絞ってk=4で出現頻度解析すると、おそらくTATAが上位にランクイン
- 発現量推定
  - RNA-seq解析で、リファレンスにリードをマップしてリード数をカウントするのが主流だが、マッピング作業をすっ飛ばしてk-merに基づく方法で定量。Sailfish (Patro et al., *Nat Biotechnol.*, 2014)やRNA-Skim (Zhang and Wang, *Bioinformatics*, 2014)。

課題

任意の生物種のパッケージについて2連続塩基の出現確率を調べ、得られた結果について簡単に考察せよ(7.のヒトゲノムやhoge4.faを除く)。「どのパッケージ(あるいは生物種)を解析し、どういう結果(期待値と実測値)が得られ、例えばヒトゲノムの場合と比べてどうだったか」という程度でよい。

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

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

```
out_f <- "hoge7.txt"                          #出力ファイル名を指定してout_fに格納
param <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定

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

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

#本番
out <- dinucleotideFrequency(fasta, as.prob=T)#2連続塩

#ファイルに保存
tmp <- cbind(names(fasta), out)               #保存したい情報
write.table(tmp, out_f, sep="\t", append=F, quote=F,
```

R Console

```
[68] "BSgenome.Sscrofa.UCSC.susScr3.masked"
[69] "BSgenome.Tgondii.ToxoDB.7.0"
[70] "BSgenome.Tguttata.UCSC.taeGut1"
[71] "BSgenome.Tguttata.UCSC.taeGut1.masked"
> installed.genomes()                         #インス
[1] "BSgenome.Athaliana.TAIR.TAIR9"
[2] "BSgenome.Celegans.UCSC.ce2"
[3] "BSgenome.Celegans.UCSC.ce6"
[4] "BSgenome.Drerio.UCSC.danRer7"
[5] "BSgenome.Hsapiens.NCBI.GRCh38"
[6] "BSgenome.Hsapiens.UCSC.hg19"
[7] "BSgenome.Mmusculus.UCSC.mm10"
>
```

課題の基本的な考え方

解析する生物種のGC含量を把握し、期待値からの差分に関する議論が重要。

- 目的：2連続塩基の出現頻度（or 確率）を調べ、偏りの有無を調査
  - ヒトゲノムはCGという連続塩基の出現頻度が他（特にGG, CC, GC）に比べて少ないと言われており、大まかにその傾向は確認済み。他の生物種ではどういう傾向にあるのか？ということに興味をもち調べようとしている。
- 注意点：生物種ごとにGC含量が異なる。
  - GC含量が高いということは、CとGの出現頻度が高いことを意味する。それは、AとTの出現頻度の相対的な低下を意味する。
  - GC含量50%の生物種の場合、A, C, G, Tの出現確率は等しい(0.25, 0.25, 0.25, 0.25)。それゆえ、計16種類の2連続塩基の出現確率の期待値は全て0.25×0.25 = 1/16。

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

    (1/16, 1/16, 1/16, 1/16, 1/16, 1/16, 1/16, 1/16, 1/16, 1/16, 1/16, 1/16, 1/16, 1/16, 1/16, 1/16)
  - 極端な例として、全てCまたはGのみからなるGC含量100%の生物種の場合、(A, C, G, T)の出現確率は(0.0, 0.5, 0.5, 0.0)となる。この2連続塩基出現確率の期待値：

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

    (0.0, 0.0, 0.0, 0.0, 0.0, 0.25, 0.25, 0.0, 0.0, 0.25, 0.25, 0.0, 0.0, 0.0, 0.0, 0.0)

GC含量情報を把握

入力データがFASTA形式でも
BSgenomeのRパッケージでもGC含量を
計算することができます。

- イントロ | NGS | アノテーション情報取得 | biomaRt(Durinck 2009) (last modified 2013/09/26)
- イントロ | NGS | アノテーション情報取得 | TranscriptDb | について (last modified 2014/03/28)
- イントロ | NGS | アノテーション情報取得 | TranscriptDb | TxDb.*から (last modified 2015/02/19)
- イントロ | NGS | アノテーション情報取得 | TranscriptDb | GenomicFeatures(Lawrence 2013) (last
- イントロ | NGS | アノテーション情報取得 | Tra Db | GFF/GTF形式ファイルから (last modifie
- イントロ | NGS | 読み込み | BSgenome | 基本情報を取得 (last modified 2015/04/18) NEW
- イントロ | NGS | 読み込み | FASTA形式 | 基本情報を取得 (last modified 2014/08/18)
- イントロ | NGS | 読み込み | FAST
- イントロ | NGS | 読み込み | FAST
- イントロ | NGS | 読み込み | FAST
- イントロ | NGS | 読み込み | Illum
- イントロ | NGS | 読み込み | Illum
- イントロ | ファイル形式の変換 |
- イントロ | ファイル形式の変換 | B
- イントロ | ファイル形式の変換 | F

イントロ | NGS | 読み込み | BSgenome | 基本情報を取得 NEW

BSgenomeパッケージを読み込んで、Total lengthやaverage lengthなどの各種情報取得を行うためのやり方を示します。パッケージがインストールされていない場合は、インストール | Rパッケージ | 個別を参考にしてインストールしておく必要があります。
「ファイル」-「ディレクトリの変更」で出力結果ファイルを保存したいディレクトリに移動し以下をコピペ。

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

「整数オーバーフロー」問題のためにゲノムサイズ(Total_length)やN50の情報が得られませんが、GC含量は約41%という値が得られています。これは、GとCが各20.5%を占め、残りのAとTが各29.5%を占めることを意味します。

```
out_f <- "hoge1.txt"                          #出力ファイル名を指定してout_fに格納
param_bsgenome <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定(BSgenome系のゲノム

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

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

GC含量情報を把握

イントロ | NGS | 読み込み | BSgenome | 基本情報を取得 NEW

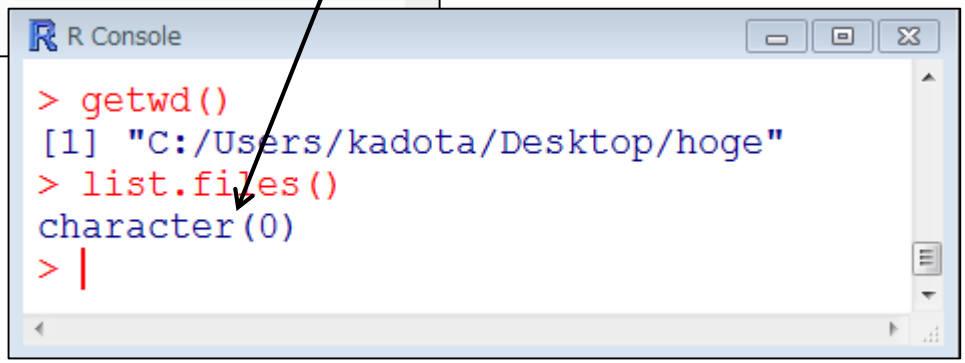
BSgenomeパッケージを読み込んで、Total lengthやaverage lengthなどの各種情報取得を行うための
す。パッケージがインストールされていない場合は、インストール | Rパッケージ | 個別を参考にして
おく必要があります。
「ファイル」-「ディレクトリの変更」で出力結果ファイルを保存したいディレクトリに移動し以下をコピペ。

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

「整数オーバーフロー」問題のためにゲノムサイズ(Total_length)やN50の情報が得られませんが、GC含量は約41%という値が得られています。これは、GとCが各20.5%を占め、残りのAとTが各29.5%を占めることを意味します。

```
out_f <- "hoge1.txt"                        #出力ファイル名を指定してout_fに格納
param_bsgenome <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定(BSgenome系のゲノム

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

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

入力がBSgenomeのパッケージの場合、入力ファイルというものはない。getwd()で確認するのは、出力ファイルが作成される場所。list.files()で「character(0)」と表示されているが、これは「何もない」という意味。



GC含量情報を把握

コピペ後の途中経過。警告メッセージ部分を表示。ヒトゲノムレベルの配列長はsum(width(fasta))の結果を整数では格納できない。hoge4.faのような小さい入力ファイルの場合はうまくいくが、非常に大きいデータのときにうまくいかないことがある、という例。

イントロ | NGS | 読み込み | BSgenome | 基本情報を取得

BSgenomeパッケージを読み込んで、Total lengthやaverage lengthなどの各種情報取得
す。パッケージがインストールされていない場合は、インストール | Rパッケージ | 個別
おく必要があります。
「ファイル」-「ディレクトリの変更」で出

1. ヒトゲノム配列パッケージ(BSgen

「整数オーバーフロー」問題のため
という値が得られています。これは

```
out_f <- "hoge1.txt"
param_bsgenome <- "BSgenor

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

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

R Console

```
[454]    170537 TCTACTCTC...GGGGAATTC HSCHR19KIR_FH15_A...
[455]    177381 GATCTATCT...GGGGAATTC HSCHR19KIR_RP5_B_...
>
> #本番(基本情報取得)
> Total_len <- sum(width(fasta))          #コンティグの「トータルの$
 警告メッセージ:
In sum(width(fasta)) :
   整数のオーバーフローがありました。sum(as.numeric(.)) を使います
> Number_of_contigs <- length(fasta)      #「コンティグ数」を取得
> Average_len <- mean(width(fasta))       #コンティグの「平均長」を$
> Median_len <- median(width(fasta))      #コンティグの「中央値」を$
> Max_len <- max(width(fasta))            #コンティグの長さの「最大$
> Min_len <- min(width(fasta))            #コンティグの長さの「最小$
>
> #本番(N50情報取得)
> sorted <- rev(sort(width(fasta)))       #長さ情報を降順にソートし$
> obj <- (cumsum(sorted) >= Total_len*0.5)#条件を満たすかどうかを判$
 警告メッセージ:
 'cumsum' 関数において整数のオーバーフローがありました。'cumsum(as.$
> N50 <- sorted[obj][1]                   #objがTRUEとなる1番最初の$
```

GC含量情報を把握

これは警告メッセージを見逃してはいけない例。**発展問題**:この整数オーバーフロー問題の解決策を示し、門田に報告せよ。**報酬?!**:「〇〇氏提供情報」として、(Rで)塩基配列解析のウェブページ上で公開。他のヒトにも有効利用してもらいます。ちなみにGC含量は約41%。

イントロ | NGS | 読み込み | BSgenome | 基本情報を取得 NEW

BSgenomeパッケージを読み込んで、Total lengthやaverage lengthなどの各種情報取得を行う
す。パッケージがインストールされていない場合は、インストール | Rパッケージ | 個別 を参考
おく必要があります。
「ファイル」-「ディレクトリの変更」で出力結果ファイルを保存したいディレクトリに移動し以下を

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

「整数オーバーフロー」問題のためにゲノムサイズ(Total_length)やN50の情報が得られませんが、GC含量は約41%という値が得られています。これは、GとCが各20.5%を占め、残りのAとTが各29.5%を占めることを意味します。

```
out_f <- "hoge1.txt"                          #出力ファイル名を指定してout_fに格納
param_bsgenome <- "BSgenome.Hsapiens.NCBI.GRCh3

#必要なパッケージをロード
library(Biostrings)                           #パッケー
library(param_bsgenome, character.only=T)#指定し

#前処理(指定したパッケージ中のオブジェクト名をgenom
tmp <- ls(paste("package", param_bsgenome, sep=
genome <- eval(parse(text=tmp))               #文字列tm
fasta <- getSeq(genome)                       #ゲノム塩
names(fasta) <- seqnames(genome)              #descrip
fasta                                         #確認して
```

R Console

```
> tmp <- rbind(tmp, c("N50", N50))
> tmp <- rbind(tmp, c("GC content", GC_content))
> write.table(tmp, out_f, sep="\t", append=F, quot$
> tmp
      [,1]                  [,2]
[1,] "Total length (bp)" NA
[2,] "Number of contigs" "455"
[3,] "Average length"    "7053376.05494506"
[4,] "Median length"     "161218"
[5,] "Max length"        "248956422"
[6,] "Min length"        "970"
[7,] "N50"               NA
[8,] "GC content"        "0.409948515653618"
>
```

ヒトゲノムのCpG解析

解析する生物種のGC含量を把握し、期待値からの差分に関する議論が重要。出現確率は染色体（コンティグ）ごとにばらつきがある。私ならbox plotを採用。

① 解析したパッケージ名：BSgenome.Hsapiens.NCBI.GRCh38

② ヒトゲノムの全体のGC含量：約41%

各塩基(A, C, G, T)の出現確率：(0.295, 0.205, 0.205, 0.295)

③ AA, AT, TA, TTの期待値 = 0.295 × 0.295 = 8.7%

④ CC, **CG**, GC, GGの期待値 = 0.205 × 0.205 = 4.2%

⑤ AC, AG, CA, CT, GA, GT, TC, TGの期待値 = 0.205 × 0.295 = 6.0%

期待値	8.8	6.0	6.0	8.7	6.0	4.2	**4.2**	6.0	6.0	4.2	4.2	6.0	8.7	6.0	6.0	8.7
連続塩基	AA	AC	AG	AT	CA	CC	**CG**	CT	GA	GC	GG	GT	TA	TC	TG	TT
1	9.5%	5.0%	7.1%	7.4%	7.3%	5.4%	**1.0%**	7.1%	6.0%	4.4%	5.4%	5.0%	6.3%	6.0%	7.3%	9.6%
2	10.0%	5.0%	7.0%	7.9%	7.2%	5.0%	**0.9%**	7.0%	5.9%	4.1%	5.0%	5.0%	6.7%	5.9%	7.2%	10.0%
3	10.1%	5.0%	6.9%	8.0%	7.2%	4.9%	**0.8%**	6.9%	5.9%	4.0%	4.9%	5.0%	6.9%	5.9%	7.2%	10.2%
4	10.6%	5.0%	6.7%	8.5%	7.1%	4.5%	**0.8%**	6.7%	5.9%	3.8%	4.5%	5.0%	7.3%	5.8%	7.1%	10.6%
5	10.2%	5.0%	6.9%	8.1%	7.2%	4.8%	**0.9%**	6.9%	5.9%	4.0%	4.8%	5.1%	6.9%	5.9%	7.2%	10.3%
6	10.2%	5.0%	6.9%	8.1%	7.2%	4.8%	**0.9%**	6.9%	5.9%	4.0%	4.9%	5.0%	6.9%	5.9%	7.2%	10.2%
7	9.8%	5.0%	7.0%	7.7%	7.3%	5.1%	**1.0%**	7.0%	6.0%	4.2%	5.1%	5.1%	6.5%	5.9%	7.3%	10.0%
8	10.0%	5.1%	6.9%	7.9%	7.2%	5.0%	**0.9%**	6.9%	6.0%	4.1%	5.0%	5.0%	6.7%	5.9%	7.2%	10.0%
9	9.7%	5.1%	7.0%	7.6%	7.3%	5.3%	**1.0%**	7.0%	6.0%	4.3%	5.3%	5.0%	6.4%	6.0%	7.3%	9.7%
10	9.6%	5.0%	7.1%	7.5%	7.3%	5.3%	**1.0%**	7.1%	6.0%	4.4%	5.3%	5.1%	6.3%	6.0%	7.4%	9.7%
11	9.5%	5.1%	7.1%	7.5%	7.3%	5.3%	**1.0%**	7.1%	6.1%	4.3%	5.4%	5.0%	6.3%	6.0%	7.3%	9.6%

Contents

- パッケージ
  - CRANとBioconductor
  - 推奨パッケージインストール手順のおさらい
  - ゲノム情報パッケージBSgenomeの概観
  - ヒトゲノム情報パッケージの解析
- 2連続塩基出現頻度解析(CpG解析)、k-mer解析
  - 課題
  - GC含量の違いを考慮(連続塩基の種類ごとに期待値が異なる)
  - 作図(box plot)
- その他
  - 数式の感覚を理解
  - Sequence logos (Schneider and Stephens, 1990)
  - プロモーター配列取得

box plotをPNG形式ファイルで出力するやり方の基本形です

作図(box plot):基本形

- イントロ | 一般 | 逆相補鎖(reverse complement)を取得 (last modified 2013/06/14)
- イントロ | 一般 | 逆鎖(reverse)を取得 (last modified 2013/06/14)
- イントロ | 一般 | 2連続塩基の出現頻度情報を取得 (last modified 2015/04/20) NEW
- イントロ | 一般 | 3連続塩基の出現頻度情報を取得 (last modified 2015/02/19)
- イントロ | 一般 | 任意の長さの連続塩基の出現頻度情報を取得 (last modified 2015/02/19)
- イントロ | 一般 | T
- イントロ | 一般 | T
- イントロ | 一般 | 配
- イントロ | 一般 | 配
- イントロ | 一般 | 配

イントロ | 一般 | 2連続塩基の出現頻度情報を取得 NEW

multi-FASTA形式ファイルを読み込んで、"AA", "AC", "AG", "AT", "CA", "CC", "CG", "CT", "GA", "GC", "GG", "GT", "TA", "TC", "TG", "TT" の計4^2 = 16通りの2連続塩基の出現頻度を調べるやり方を示します。ヒトゲノムで"CG"の割合が期待値よりも低い(Lander et al., 2001; Saxonov et al., 2006)ですが、それを簡単に検証できます。
「ファイル」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピペ。

1. イントロ | 一般 | ヌ

タイトル通りの出現頻度

```
in_f <- "hoge4.fa
out_f <- "hoge1.t

#必要なパッケージを
library(Biostring

#入力ファイルの読み
fasta <- readDNAS
fasta

#本番
out <- dinucleot
```

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

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

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

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

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

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

作図(box plot):基本形

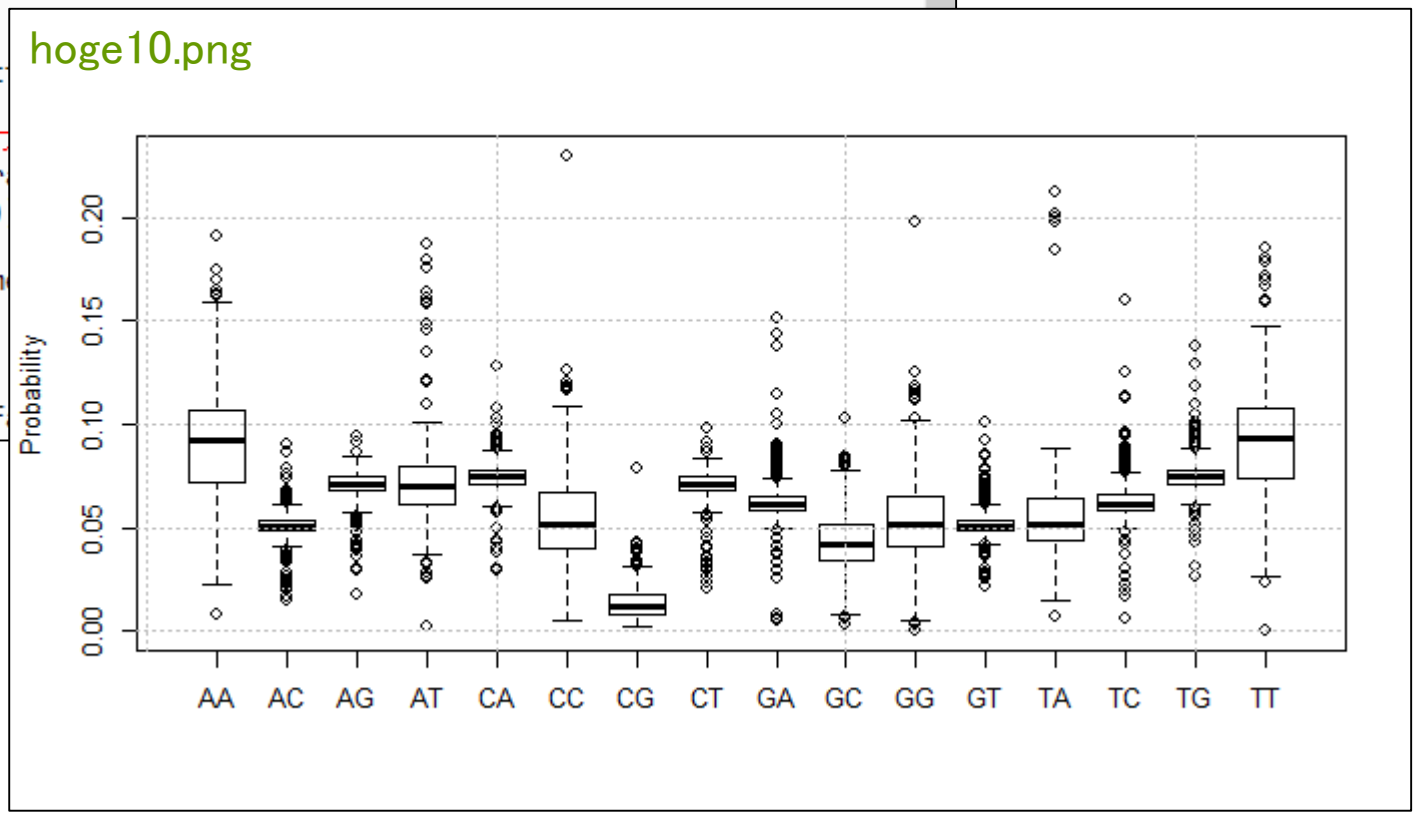
PNGファイルのサイズを指定するところです。

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

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

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

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

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

#本番
out <- dinucleotideFrequency(f
```

hoge10.png



400 pixels

700 pixels

作図(box plot):色づけ

タブ区切りテキストファイル(human_2mer.txt)を用意しておき、colorという列名のところに2連続塩基の種類ごとに色を指定しています。

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

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

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

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

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

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

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

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

作図(box plot):色づけ

boxplot関数実行時のcolオプション部分で利用していることがわかります

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

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

```
data <- read.table(in_f, header=TRUE, row.names=1, sep="\t", quote="")#in_fで指

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

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

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

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

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

作図(box plot):色づけ

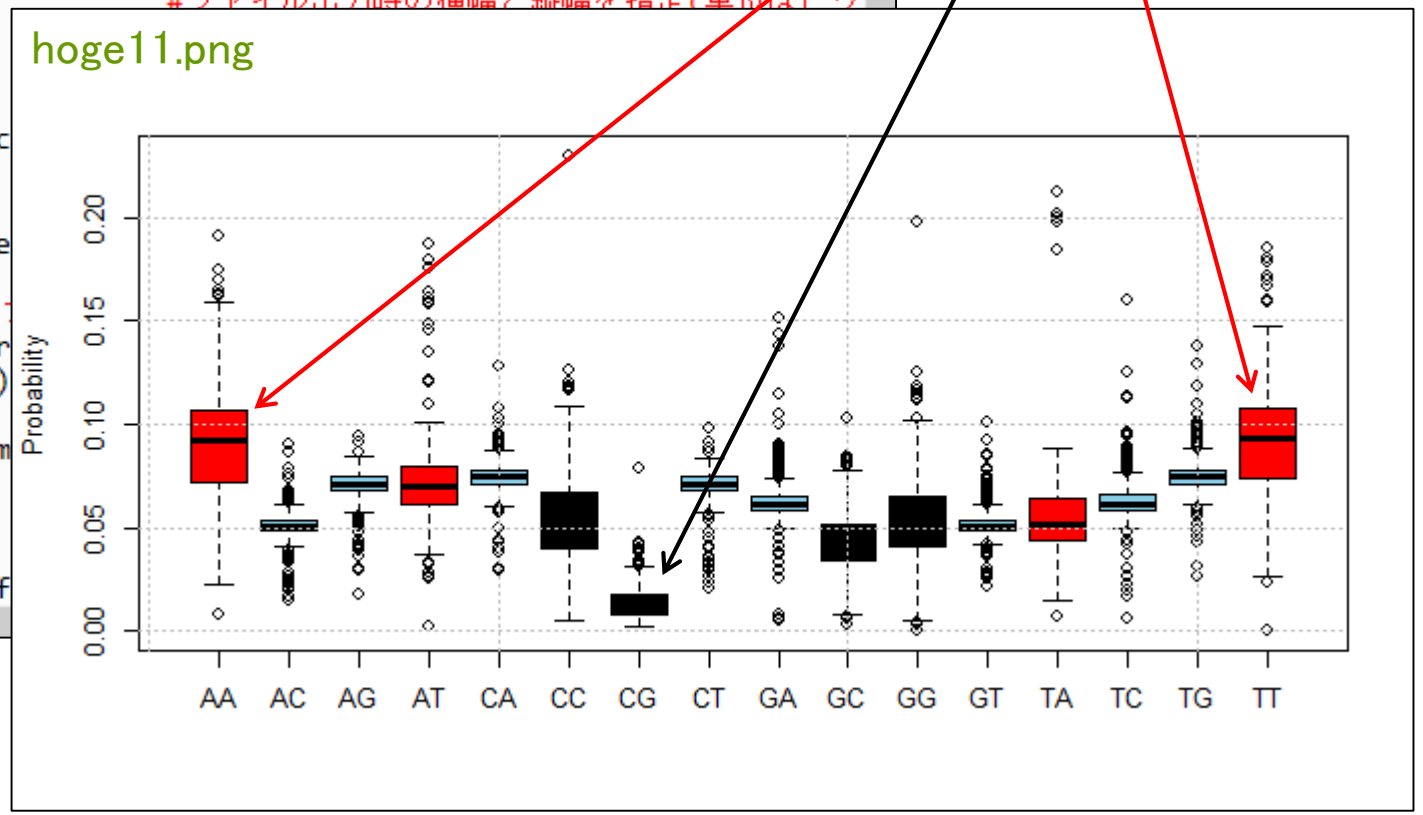
CGの出現確率が期待値(4.2%)より少ないのはCC, GC, GGとの相対的な関係からも明白。AAやTTが多いのは、期待値(8.7%)も高いためであろうと判断。

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

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

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

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

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

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

#本番
out <- dinucleotideFrequency(f
```

hoge11.png



400 pixels

700 pixels

作図(box plot):発展形

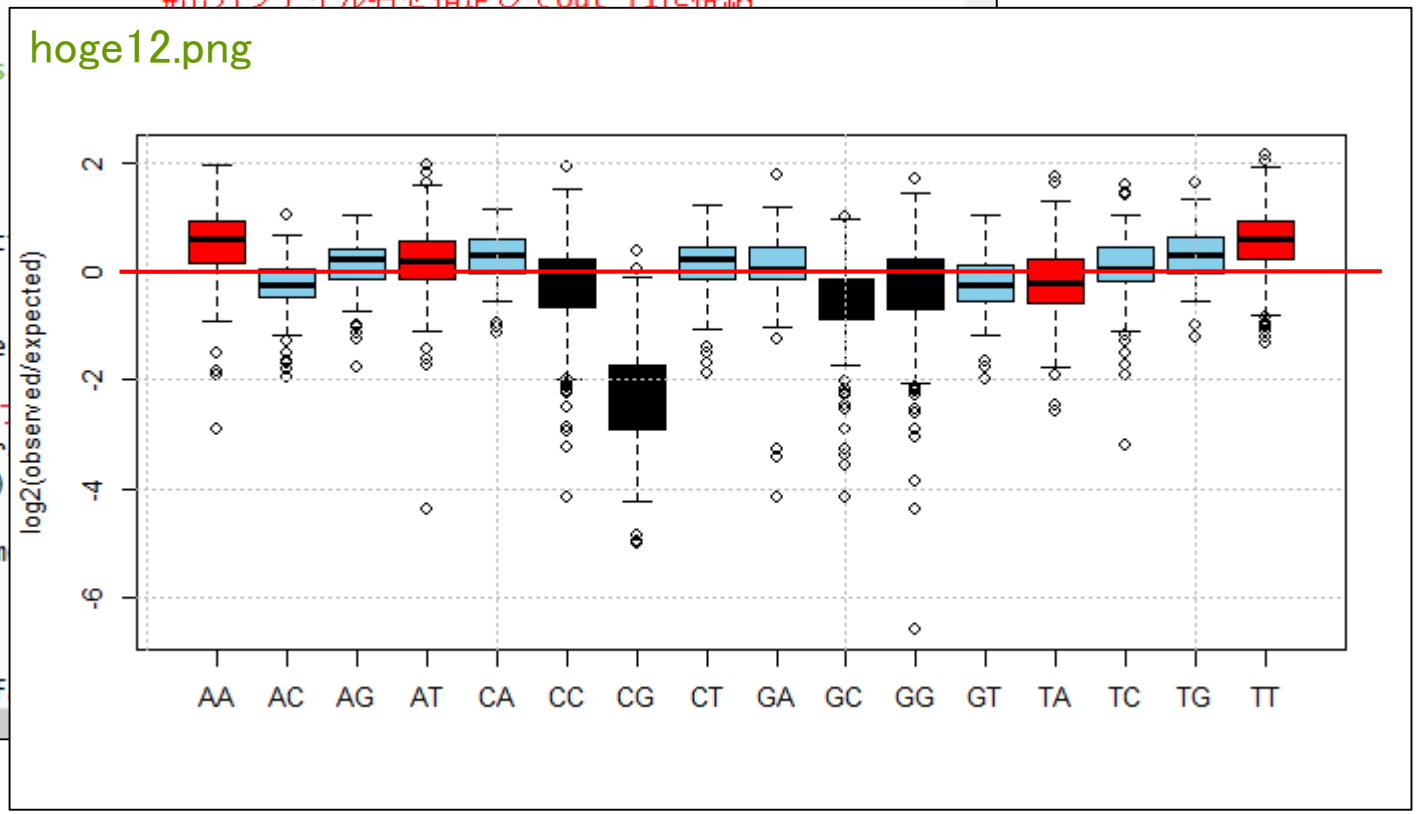
期待値との差分を評価すべく、縦軸にlog(観測値/期待値)をプロット。0付近にある2連続塩基は、観測値(実測された出現確率)が期待値とほぼ同じことを意味する。

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

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

```
in_f <- "human_2mer.txt"                    #入力ファイル名を指定してin_fに格納
out_f1 <- "hoge12.txt"                      #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge12.png"
param_bsgenome <- "BSgenome.Hs
param_fig <- c(700, 400)

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

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

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

#本番
out <- dinucleotideFrequency(f
```

hoge12.png



Contents

- パッケージ
  - CRANとBioconductor
  - 推奨パッケージインストール手順のおさらい
  - ゲノム情報パッケージBSgenomeの概観
  - ヒトゲノム情報パッケージの解析
- 2連続塩基出現頻度解析(CpG解析)、k-mer解析
  - 課題
  - GC含量の違いを考慮(連続塩基の種類ごとに期待値が異なる)
  - 作図(box plot)
- その他
  - 数式の感覚を理解
  - プロモーター配列取得
  - Sequence logos (Schneider and Stephens, 1990)

数式の感覚を理解

ニュースレター中の参考書紹介記事（の一部）。log(観測値/期待値)そのものの解説ではないが、数式が出ると混乱しがちなヒト向けに「重みつき平均」や「エントロピー」の具体的な計算手順を示しながら解説。

本書の主な存在意義は、講義や講習会の枠組みでは伝授が難しい数式の意味やアルゴリズム（問題解決のためのやり方や工夫）をじっくり学べる点であろう。なぜ二乗を含む数式にはないが三乗を含む数式には絶対値があるのか？なぜ数式の分母に非常に小さい数値が足されているのか？なぜ最大値で割るのか？なぜ log をとるのか？探索範囲を限定するとどういう結果になりうるのか？などバイ

オインフォマティクス的なものの考え方や注意点を述べている箇所の理解がオススメポイントである。具体的には、2.2.3（p45-62）や 4.2.3（p182-188）である。2.2.4（p62-70）や 3.2.2（p107-111）で述べている内容も、中長期的には役立つ考え方であろう。全てが「RNA-seq」ではなく「マイクロアレイ」の項目であるが、RNA-seq 部分のみ読むつもりであれば考えを改めるか購入しなくてもよいだろう。

COMPLEX ADAPTIVE TRAITS Newsletter Vol. 5 No. 7
発　行：2015年3月30日
発行者：新学術研究領域「複合適応形質進化の遺伝子基盤解明」（領域代表者　長谷部光泰）
編　集：COMPLEX ADAPTIVE TRAITS Newsletter 編集委員会（編集責任者　深津武馬）
領域URL：http://staff.aist.go.jp/t-fukatsu/SGJHome.html

自習用教材情報

ここから門田関連の講義、講習会、講演資料、執筆原稿などの情報を辿れます。

(Rで)塩基配列解析

～NGS、RNA-seq、ゲノム、トランスクリプトーム、正規化、発現変動、統計、モデル、バイオインフォマティクス～
(last modified 2015/04/20, since 2010)

What's new?

- このウェブページはインストール | について の推奨手順 (Windows2015.04.04版とMacintosh2015.04.03版)に従って フリーソフトRと必要なパッケージをインストール済みであるという前提で記述しています。初心者の方は基本的な利用法(Windows2015.04.03版とMacintosh2015.04.03版)で自習してください。本ウェブページを体系的にまとめた書籍もあります。(2015/04/03) NEW

- 日本乳酸菌学会誌のNGS関連連載の第3回分PDFを公開しました。(2015/04/20) NEW
- 下記NGSハンズオン講習会の申込み受付サイトへのリンクが張られていませんでしたm(_ _)m ここから受講申込ボタンにたどり着けます。4/20の14:00ごろオープンしています。失礼しましたm(_ _)m(2015/04/20) NEW
- 平成27年度NGSハンズオン講習会を2015年7月22日-8月6日の11日間で実施します。受講申込開始は4/20予定です。昨年度の「NGS速習コース」同様、オブザーバー(TA)募集も並行して行いますので可能な範囲でご協力をお願いしますm(_ _)m(2015/04/15) NEW
- 参考資料(講義、講習会、本など)の項目を更新しました。(2015/04/20) NEW

- はじめに (last modified 2015/03/31) NEW
- 参考資料(講義、講習会、本など) (last modified 2015/04/20) NEW
- 過去のお知らせ (last modified 2015/04/14) NEW
- インストール | について (last modified 2015/04/04) NEW
- インストール | R本体 | 最新版 | Win用 (last modified 2015/03/22)推奨 NEW

自習用教材情報

参考資料(講義、講習会、本など) NEW

基本的に私門田の個人ページに記載してあるものです。かなり古い講演資料などの情報をもとに勉強されている方もいらっしゃるようですので、ここでは2013年秋以降の情報を載せておくとともに、大まかな内容についても述べておきます。講演予定のものについては、資料のアップは講演当日が基本です。50-100MB程度ありますがオリジナルのPowerPointファイルがほしい方はお気軽にリクエストしてください。講義資料としての利用などは事前連絡や私個人への謝辞も気にせずご自由にお使いください。

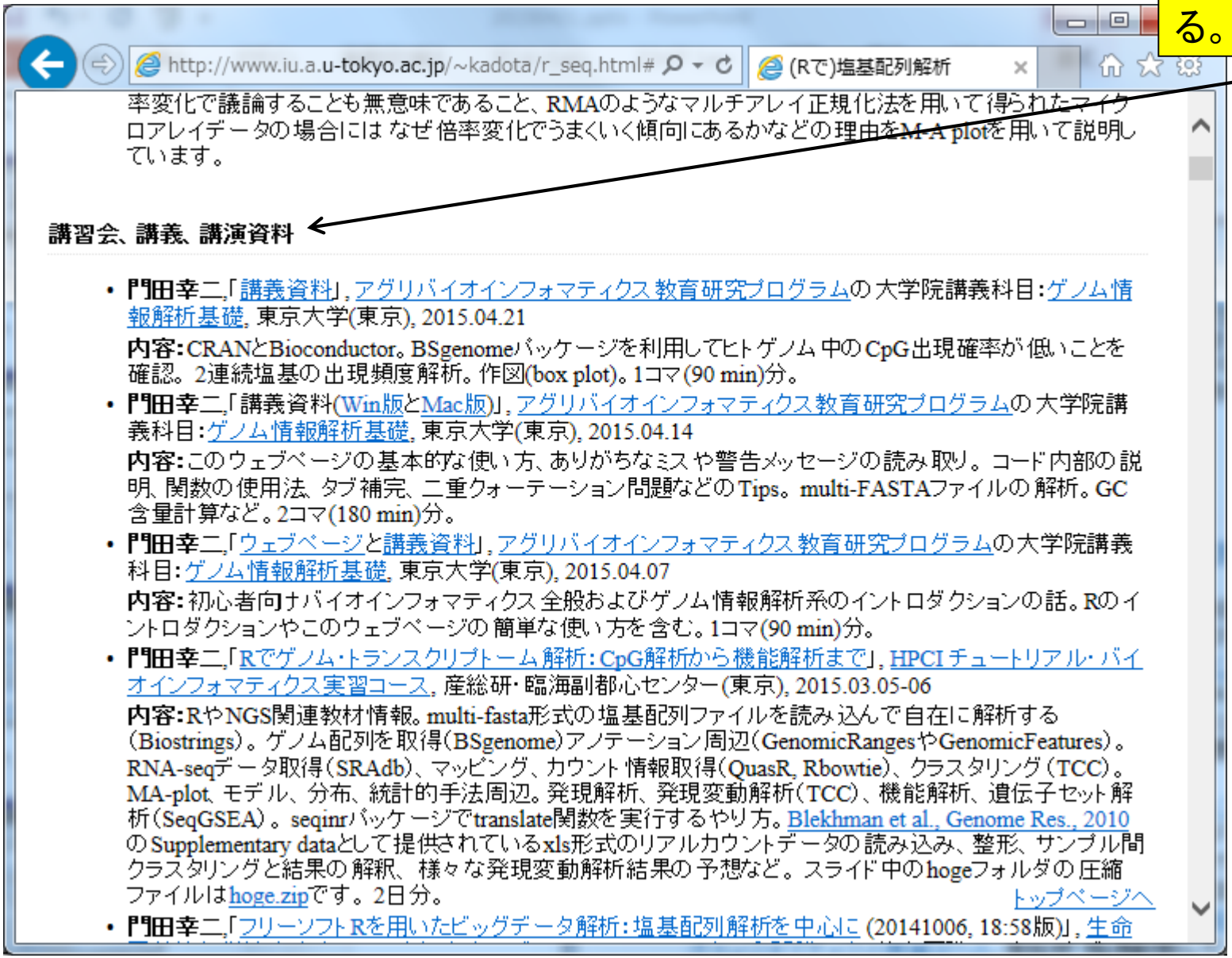
書籍、学会誌

- 孫建強, 三浦文, 清水謙多郎, **門田幸二**,「次世代シーケンサーデータの解析手法:第3回Linux環境構築からNGSデータ取得まで」, *日本乳酸菌学会誌*, **26**(1):32-41, 2015.
  内容: 日本乳酸菌学会誌のNGS関連連載の第3回分です。書籍中のリンク先やウェブ資料などは「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」の項目をご覧ください。
- 孫建強, 湯敏, 西岡輔, 清水謙多郎, **門田幸二**,「次世代シーケンサーデータの解析手法:第2回GUI環境からコマンドライン環境へ」, *日本乳酸菌学会誌*, **25**(3):166-174, 2014.
  内容: 日本乳酸菌学会誌のNGS関連連載の第2回分です。GUI環境とコマンドライン環境の違い、Windowsのコマンドプロンプトやコマンド、Macintoshのターミナルやlinuxコマンドの説明。WinとMacでコマンド名が異なること、WinはMS-DOSの流れをくんでいる。ターミナルがOS X以降に利用可能であることやUNIXの説明。基本的なLinuxコマンド(pwd, cd, ls, mv, grep, find)の説明やカレントディレクトリの概念。バイオインフォマティクス分野の常識・非常識として、ファイル名や拡張子など。「最低限必要なLinuxコマンドとは？」や「バイオインフォマティクス初級、中級、上級」などの基準は人それぞれであってないようなもの。Tipsとして、「1. タブ補完(Tabキーによる補完機能の利用)」でタイプミスを大幅に減らせること、「2. 上下左右の矢印キーの利用」で以前打ち込んだコマンドを呼び出して再利用、「3.ヒストリー機能の利用」で以前打ち込んだコマンドリストを表示して再利用。Linuxコマンドオプションや、オプションの組合せの説明。grepコマンドを利用して乳酸菌ゲノム配列ファイル中のコンティグ数情報を得る。Bio-Linuxの導入。書籍中のリンク先やウェブ資料などは「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」の項目をご覧ください。
- **門田幸二**, 孫建強, 湯敏, 西岡輔, 清水謙多郎,「次世代シーケンサーデータの解析手法:第1回イントロダクション」, *日本乳酸菌学会誌*, **25**(2), 87-94, 2014.
  内容: 日本乳酸菌学会誌のNGS関連連載の第1回分です。NGS解析関連の情報収集先、NGS用教育用

これはLinux系の教材。日本乳酸菌学会誌の連載。第3回分は2015年4月20日に公開。平成27年度NGSハンズオン講習会の7/23分はこれらを基本とする予定。

自習用教材情報

さらに1ページ分ほど下に移動すると、「講習会、講義、講演資料」のPDFが見られる。**時系列順**にリストアップ。



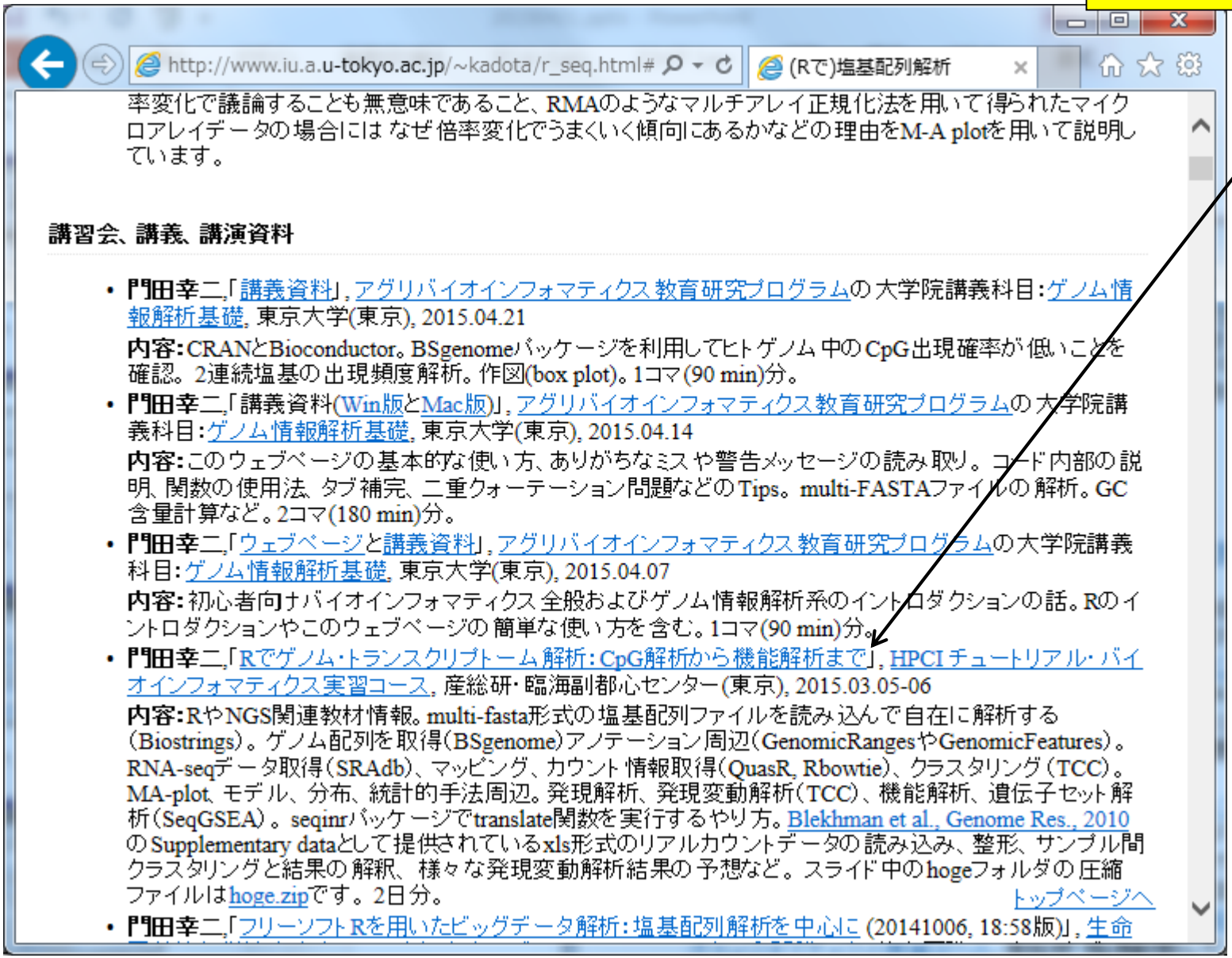
率変化で議論することも無意味であること、RMAのようなマルチアレイ正規化法を用いて得られたマイクロアレイデータの場合には なぜ倍率変化でうまくいく傾向にあるかなどの理由をM-A plotを用いて説明しています。

講習会、講義、講演資料

- **門田幸二**,「講義資料」, アグリバイオインフォマティクス教育研究プログラムの大学院講義科目:ゲノム情報解析基礎, 東京大学(東京), 2015.04.21
  内容: CRANとBioconductor。BSgenomeパッケージを利用してヒトゲノム中のCpG出現確率が低いことを確認。2連続塩基の出現頻度解析。作図(box plot)。1コマ(90 min)分。
- **門田幸二**,「講義資料(Win版とMac版)」, アグリバイオインフォマティクス教育研究プログラムの大学院講義科目:ゲノム情報解析基礎, 東京大学(東京), 2015.04.14
  内容: このウェブページの基本的な使い方、ありがちなミスや警告メッセージの読み取り。コード内部の説明、関数の使用法、タブ補完、二重クォーテーション問題などのTips。multi-FASTAファイルの解析。GC含量計算など。2コマ(180 min)分。
- **門田幸二**,「ウェブページと講義資料」, アグリバイオインフォマティクス教育研究プログラムの大学院講義科目:ゲノム情報解析基礎, 東京大学(東京), 2015.04.07
  内容: 初心者向けバイオインフォマティクス全般およびゲノム情報解析系のイントロダクションの話。Rのイントロダクションやこのウェブページの簡単な使い方を含む。1コマ(90 min)分。
- **門田幸二**,「Rでゲノム・トランスクリプトーム解析:CpG解析から機能解析まで」, HPCIチュートリアル・バイオインフォマティクス実習コース, 産総研・臨海副都心センター(東京), 2015.03.05-06
  内容: RやNGS関連教材情報。multi-fasta形式の塩基配列ファイルを読み込んで自在に解析する(Biostrings)。ゲノム配列を取得(BSgenome)アノテーション周辺(GenomicRangesやGenomicFeatures)。RNA-seqデータ取得(SRAdb)、マッピング、カウント情報取得(QuasR, Rbowtie)、クラスタリング(TCC)。MA-plot、モデル、分布、統計的手法周辺。発現解析、発現変動解析(TCC)、機能解析、遺伝子セット解析(SeqGSEA)。seqinrパッケージでtranslate関数を実行するやり方。Blekhman et al., Genome Res., 2010のSupplementary dataとして提供されているxls形式のリアルカウントデータの読み込み、整形、サンプル間クラスタリングと結果の解釈、様々な発現変動解析結果の予想など。スライド中のhogeフォルダの圧縮ファイルはhoge.zipです。2日分。 トップページへ
- **門田幸二**,「フリーソフトRを用いたビッグデータ解析:塩基配列解析を中心に (20141006, 18:58版)」, 生命

自習用教材情報

プロモーター配列取得は、2015.03.05-06の講習会PDFのスライド133-138あたりに記載あり。



率変化で議論することも無意味であること、RMAのようなマルチアレイ正規化法を用いて得られたマイクロアレイデータの場合には なぜ倍率変化でうまくいく傾向にあるかなどの理由をM-A plotを用いて説明しています。

講習会、講義、講演資料

- **門田幸二**,「講義資料」, アグリバイオインフォマティクス教育研究プログラムの大学院講義科目:ゲノム情報解析基礎, 東京大学(東京), 2015.04.21
  内容: CRANとBioconductor。BSgenomeパッケージを利用してヒトゲノム中のCpG出現確率が低いことを確認。2連続塩基の出現頻度解析。作図(box plot)。1コマ(90 min)分。
- **門田幸二**,「講義資料(Win版とMac版)」, アグリバイオインフォマティクス教育研究プログラムの大学院講義科目:ゲノム情報解析基礎, 東京大学(東京), 2015.04.14
  内容: このウェブページの基本的な使い方、ありがちなミスや警告メッセージの読み取り。コード内部の説明、関数の使用法、タブ補完、二重クォーテーション問題などのTips。multi-FASTAファイルの解析。GC含量計算など。2コマ(180 min)分。
- **門田幸二**,「ウェブページと講義資料」, アグリバイオインフォマティクス教育研究プログラムの大学院講義科目:ゲノム情報解析基礎, 東京大学(東京), 2015.04.07
  内容: 初心者向けバイオインフォマティクス全般およびゲノム情報解析系のイントロダクションの話。Rのイントロダクションやこのウェブページの簡単な使い方を含む。1コマ(90 min)分。
- **門田幸二**,「Rでゲノム・トランスクリプトーム解析:CpG解析から機能解析まで」, HPCIチュートリアル・バイオインフォマティクス実習コース, 産総研・臨海副都心センター(東京), 2015.03.05-06
  内容: RやNGS関連教材情報。multi-fasta形式の塩基配列ファイルを読み込んで自在に解析する(Biostrings)。ゲノム配列を取得(BSgenome)アノテーション周辺(GenomicRangesやGenomicFeatures)。RNA-seqデータ取得(SRAdb)、マッピング、カウント情報取得(QuasR, Rbowtie)、クラスタリング(TCC)。MA-plot、モデル、分布、統計的手法周辺。発現解析、発現変動解析(TCC)、機能解析、遺伝子セット解析(SeqGSEA)。seqinrパッケージでtranslate関数を実行するやり方。Blekhman et al., Genome Res., 2010のSupplementary dataとして提供されているxls形式のリアルカウントデータの読み込み、整形、サンプル間クラスタリングと結果の解釈、様々な発現変動解析結果の予想など。スライド中のhogeフォルダの圧縮ファイルはhoge.zipです。2日分。 トップページへ
- **門田幸二**,「フリーソフトRを用いたビッグデータ解析:塩基配列解析を中心に (20141006, 18:58版)」, 生命

プロモーター配列取得

基本はゲノム情報をアノテーション情報を読み込んで任意の範囲を指定することで目的を達成可能。しかし、2015年4月20現在のコードにはバグがあります。subseq関数のところで説明した存在しない範囲を指定することに起因するものです。**発展問題**:このバグを修正せよ。**報酬?!**:「〇〇氏提供情報」として公開。

イントロ | 一般 | 配列取得 | プロモーター配列 | BSgenomeとTxDbから

ゲノム配列(BSgenome)パッケージとアノテーション情報(TxDb)パッケージを用いて様々な生物種のプロモーター配列(転写開始点近傍配列;上流配列)を取得するやり方を示します。2014年4月リリースのBioconductor 2.14以降の推奨手順では、ゲノムのパッケージ(例:BSgenome.Hsapiens.UCSC.hg19)と対応するアノテーションパッケージ(例:TxDb.Hsapiens.UCSC.hg19.knownGene)の両方を読み込ませる必要がありますので、2015年2月に記述内容を大幅に変更しました。ヒトなどの主要なパッケージ以外はおそらくデフォルトではインストールされていませんので、「パッケージがインストールされていません」的なエラーが出た場合は、個別パッケージのインストールを参考にして予め利用したいパッケージのインストールを行ってから再挑戦してください。出力はmulti-FASTAファイルです。現状では、ゼブラフィッシュ(danRer7)はゲノムパッケージ(BSgenome.Drerio.UCSC.danRer7)は存在しますが、対応するTxDbパッケージが存在しないので、どこかからGFFファイルを取得してmakeTranscriptDbFromGFF関数などを利用してTranscriptDbオブジェクトを得るなどする必要があります。

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

1. ヒト(hg19)の場合:

ゲノムパッケージ(BSgenome.Hsapiens.UCSC.hg19)と対応するアノテーションパッケージ(TxDb.Hsapiens.UCSC.hg19.knownGene)を読み込んで、転写開始点上流1000塩基分を取得するやり方です。

```
out_f <- "hoge1.fasta"                          #出力ファイル名を指定してout_fに格納
param_bsgenome <- "BSgenome.Hsapiens.UCSC.hg19"#パッケージ名を指定(BSgenome系のゲノムパッ
param_txdb <- "TxDb.Hsapiens.UCSC.hg19.knownGene"#パッケージ名を指定(TxDB系のアノテーショ
param_upstream <- 1000                          #転写開始点上流の塩基配列数を指定

#前処理(指定したパッケージ中のオブジェクト名をgenomeおよびtxdbに統一)
library(param_bsgenome, character.only=T)#指定したパッケージの読み込み
tmp <- ls(paste("package", param_bsgenome, sep=":"))#指定したパッケージで利用可能なオブジェ
genome <- eval(parse(text=tmp))                 #文字列tmpをRオブジェクトとしてgenomeに格納
```

自習用教材情報

MA-plot、モデル、分布、統計的手法周辺。発現解析、発現変動解析(TCC)、機能解析、遺伝子セット解析(SeqGSEA)。seqinrパッケージでtranslate関数を実行するやり方。Blekhman et al., Genome Res., 2010のSupplementary dataとして提供されているxls形式のリアルカウントデータの読み込み、整形、サンプル間クラスタリングと結果の解釈、様々な発現変動解析結果の予想など。スライド中のhogeフォルダの圧縮ファイルはhoge.zipです。2日分。

- **門田幸二**,「フリーソフトRを用いたビッグデータ解析:塩基配列解析を中心に(20141006, 18:58版)」, 生命医薬情報学連合大会2014, 中級者向けバイオインフォマティクス入門講習会, 仙台国際センター(宮城), 10:50-12:20, 2014.10.04

  内容:アグリバイオインフォマティクス教育研究プログラムの大学院講義資料の紹介。その中の一部として、CpG解析を行う2連続塩基の出現頻度解析。small RNA-seqデータのマッピングおよび結果の考察からsequence logosを利用するモチベーションを論理的に説明。small RNA-seqデータのsequence logosを実行することでアダプター配列がわかることなど。動作確認用のサブセット作成手順。sequence logosの縦軸の情報量(information contents; ic)の計算手順。情報量は、内部的にはエントロピーを計算しているだけであり、エントロピーを計算しておいて、数値が大きければ大きいほどうれしいようにしたいがためにエントロピーの最大値を実際のエントロピー値から差し引いた情報量を縦軸として採用しているのがsequence logosであること。エントロピー自体は、組織特異的発現パターン検出にもそのまま利用されていること。41.Blekhman et al., Genome Res., 2010のリアルカウントデータただし、バックグラウンドレベルが高めの場合にはうまく特異的発現パターンがエントロピーの低さで表現できないので、バックグラウンドを差し引いたデータ変換を行ったのちエントロピーを計算するROKU法開発に至る思考回路の紹介。スライド中のhogeフォルダの圧縮ファイルはhoge.zip(20140929, 22:27版)です。90min分。

- **門田幸二**,「ビッグデータ解析とR (20141006, 18:51版)」, 生命医薬情報学連合大会2014, HPCIワークショップ「医療とビッグデータ解析」, 仙台国際センター(宮城), 9:00-10:30, 2014.10.04

  内容:NGSデータ解析にRがある程度利用可能である、というお話。EMBOSS、k-mer解析、wget、SAMtools、FastQC、small RNAのマッピング、sequence logos周辺がRでもできます的な話。20min分。

- **門田幸二**,「3. データ解析基礎 3-5. R bioconductor II (2014.08.28版)」, バイオインフォマティクス人材育成カリキュラム(次世代シークエンサ)速習コース, 東京大学(東京), 2014年9月9日15:00-18:15

  内容:multi-FASTAファイルからの情報抽出。GC含量計算の詳細な説明。alphabetFrequency, apply関

Sequence logosは内部的にエントロピーを計算しているだけ。情報量という小難しい単語に惑わされるべからず!
2015.10.04の講習会資料中にやさしく?!解説しています。参考書のp182-188にも記載あり。

門田死すとも教材死せず。最小限の労力で最大限の成果を!