講義資料PDFが講義のページからダウンロード可能です。印刷物はありません

ゲノム情報解析基礎
～ Rで塩基配列解析2 ～

[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/

全てPC使用予定です

講義予定

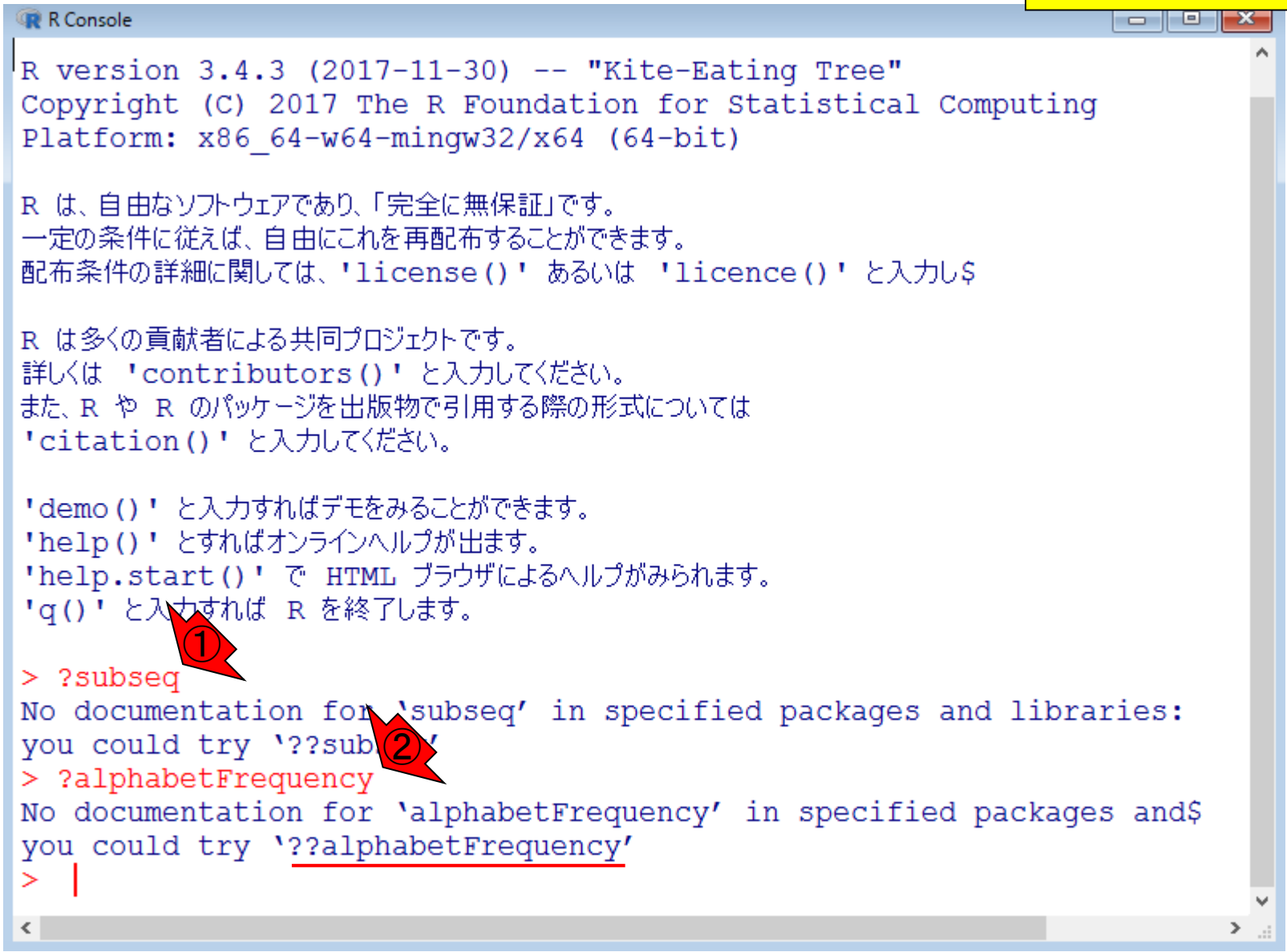
- 04月16日月曜日（17:15–20:30）
  - 嶋田透：ゲノムからの遺伝子予測
  - 門田幸二：バイオインフォマティクス基礎知識、Rのイントロダクション
- 04月23日月曜日（17:15–20:30）
  - 門田幸二：Rで塩基配列解析1、multi-FASTAファイルの各種解析
- 05月07日月曜日（17:15–20:30）
  - 嶋田透：ゲノムアノテーション、遺伝子の機能推定、RNA-seqなどによる発現解析、比較ゲノム解析
  - 門田幸二：Rで塩基配列解析2、Rパッケージ、k-mer解析の基礎
- 05月14日月曜日（17:15–19:00頃）
  - 勝間進：非コードRNA、小分子RNA、エピジェネティクス
  - 講義後、小テスト

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
  - 仮想データ
  - 実データ(課題)
  - 作図

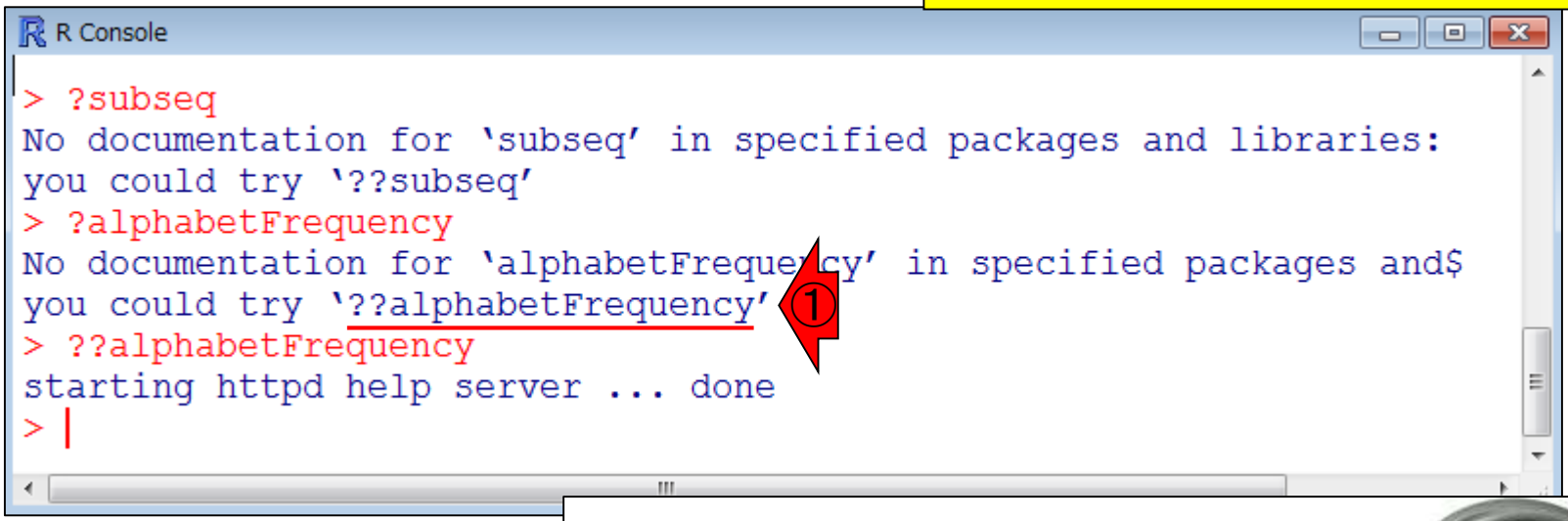
パッケージ

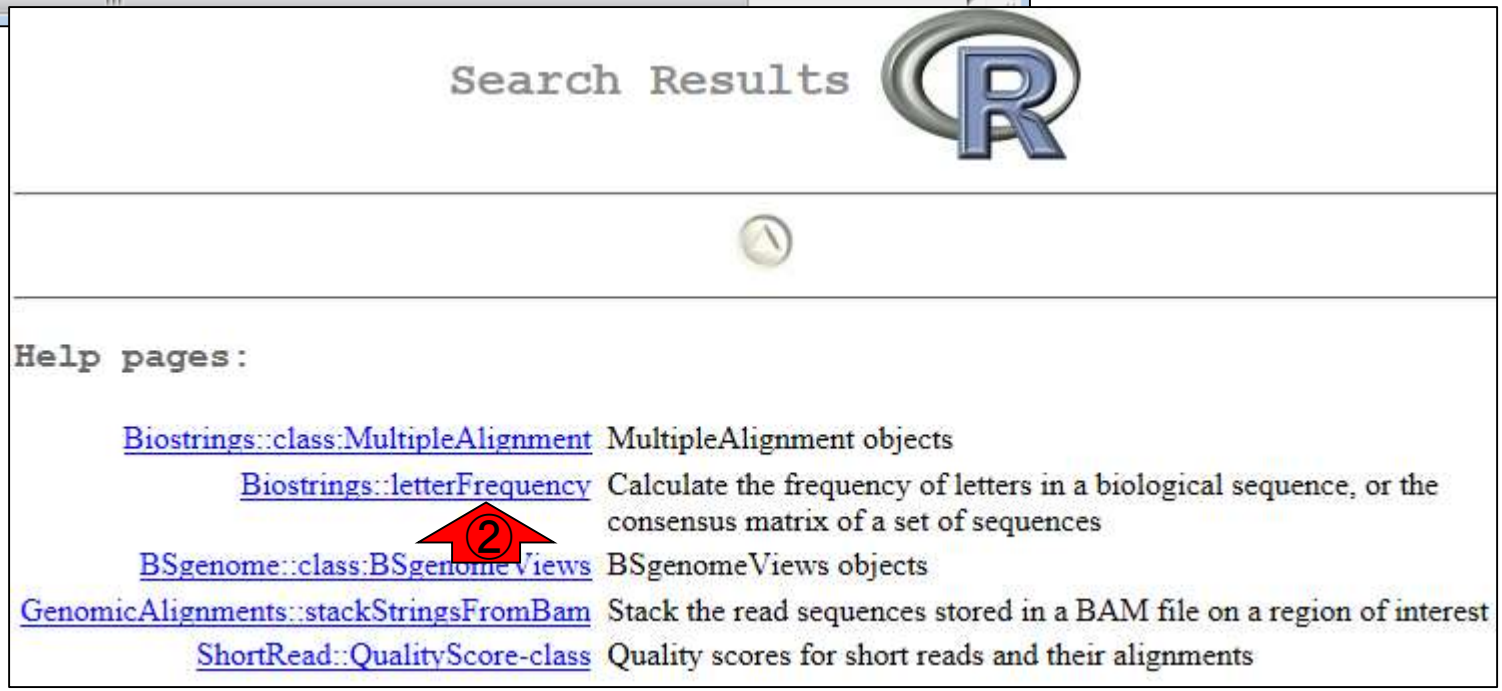
R起動直後に「**?関数名**」と打ち込んでも、使用法を記したウェブページが開かずにエラーが出ることがあります



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





パッケージ

①alphabetFrequency関数は②Biostrings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

letterFrequency {Biostrings} ②

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

Description

①

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)
```

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。③citation("パッケージ名")で引用すべき論文がわかります

- 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[illegible]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)
- イントロ | 一般 | 逆[illegible]
- イントロ | 一般 | 2[illegible]
- イントロ | 一般 | 3[illegible]
- イントロ | 一般 | 任[illegible]

①

②

①

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-F[illegible]

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

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

②

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[illegible]ate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[illegible]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[illegible]氏提供情報)を示します。
[illegible]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。

[illegible]レ(sample1.fasta)の場合:

[illegible]い single-FASTA形式ファイルです。

```
[illegible]1.fasta"                    #入力ファイル名を指定してin_fに格納
[illegible].fasta"                     #出力ファイル名を指定してout_fに格納

[illegible]ジをロード
                                       #パッケージの読み込み

[illegible]読み込み
[illegible]sta(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
```

R Console

```
> library(seqinr)
> citation("seqinr")

To cite seqinR in publications use:

  Charif, D. and Lobry, J.R. (2007)

A BibTeX entry for LaTeX users is
```

③

パッケージ

①や②の部分でパッケージをロードしている。これで、ロードしたパッケージが提供する関数群を利用可能になる

- 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)
- イントロ | 一般 | 逆
- イントロ | 一般 | 2
- イントロ | 一般 | 3
- イントロ | 一般 | 任

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-F

```
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 <- readDNAString
```

①

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
```

②

パッケージ

例えば①Biostrings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

①

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)
```

パッケージ

ここでは、①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る。実際には1回のみで大丈夫。②「?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 IRange  をロード中です
 要求されたパッケージ XVecto  ード中です
> library(Biostrings)
> ?alphabetFrequency
```

①

②

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。
  - 各種解析を行うパッケージ（またはライブラリ）をインストールして利用します。

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
  - 仮想データ
  - 実データ(課題)
  - 作図

CRANとBioconductor

- Rパッケージの2大リポジトリ(貯蔵庫)
  - CRAN：12,413パッケージ
  - Bioconductor：1,477パッケージ

②CRAN (The Comprehensive R Archive Network)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、③主にBioconductor提供パッケージを利用

- 作図 | ROC曲線 | 基礎編 | 7. 図の重ね書き(new) (last modified 2015/02/15) NEW
- 作図 | ROC曲線 | 基礎編 | 8. 凡例を追加(legend) (last modified 2015/02/15) NEW
- 作図 | ROC曲線 | 応用編 (last modified 2015/02/07) NEW
- 作図 | SplicingGraphs (last modified 2013/08/07)
- パイプライン |について (last modified 2013/10/17)
- パイプライン | ゲノム | 発現変動 | 2群間 | 対応なし | 複製あり | SRP017142(Neyr
- パイプライン | ゲノム | 機能解析 | 2群間 | 対応なし | 複製あり | SRP017142(Neyr
- パイプライン | ゲノム | 機能解析 | 2群間 | 対応なし | 複製あり | SRP011435(Huan
- パイプライン | ゲノム | small RNA | SRP016842(Nie 2013) (last modified 2014/06
- リンク集 (last modified 2012/03/29)

①

リンク集

③

- R
- Bioconductor: Gentleman et al., Genome Biol., 2004
- CRAN
- RjpWi
- R Tips(竹澤様)
- BioEdit(フリーの配列編集ソフト)
- BioMart: Smedley et al., BMC Genomics, 2009
- DDBJ Read Annotation Pipeline: Nagasaki et al., DNA Res., 2013
- EMBOSS explorer (EMBOSSのウェブ版)
- Biostar: Parnell et al., PLoS Comput Biol., 2011
- SEQanswers: Li et al., Bioinformatics, 2012
- NGS WikiBook: Li et al., Brief Bioinform., 2013
- HT Sequence Analysis with R and Bioconductor

②

約1年前の状況。着実に増えていますね

CRANとBioconductor

- Rパッケージの2大リポジトリ(貯蔵庫)
  - CRAN：10,430パッケージ
  - Bioconductor：1,294パッケージ

- 作図 | ROC曲線 | 基礎編 | 7. 図の重ね書き(new) (last modified 2015/02/15) NEW
- 作図 | ROC曲線 | 基礎編 | 8. 凡例を追加(legend) (last modified 2015/02/15) NEW
- 作図 | ROC曲線 | 応用編 (last modified 2015/02/07) NEW
- 作図 | SplicingGraphs (last modified 2013/08/07)
- パイプライン |について (last modified 2013/10/17)
- パイプライン | ゲノム | 発現変動 | 2群間 | 対応なし | 複製あり | SRP017142(Neyr
- パイプライン | ゲノム | 機能解析 | 2群間 | 対応なし | 複製あり | SRP017142(Neyr
- パイプライン | ゲノム | 機能解析 | 2群間 | 対応なし | 複製あり | SRP011435(Huan
- パイプライン | ゲノム | small RNA | SRP016842(Nie_2013) (last modified 2014/06
- リンク集 (last modified 2012/03/29)

①

リンク集

③

- R
- Bioconductor: Gentleman et al., Genome Biol., 2004
- CRAN
- RjpWi
- R Tips(竹澤様)
- BioEdit(フリーの配列編集ソフト)
- BioMart: Smedley et al., BMC Genomics, 2009
- DDBJ Read Annotation Pipeline: Nagasaki et al., DNA Res., 2013
- EMBOSS explorer (EMBOSSのウェブ版)
- Biostar: Parnell et al., PLoS Comput Biol., 2011
- SEQanswers: Li et al., Bioinformatics, 2012
- NGS WikiBook: Li et al., Brief Bioinform., 2013
- HT Sequence Analysis with R and Bioconductor

②

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。基本的に、①最新版の利用をお勧め

- 近年のリリース頻度
  - R本体 (http://www.r-project.org/)
    - 2018-04-23にver. 3.5.0をリリース ①
    - 2018-03-15にver. 3.4.4をリリース
    - 2017-04-21にver. 3.4.0をリリース
    - 2017-03-06にver. 3.3.3をリリース
    - 2016-04-14にver. 3.2.5をリリース
    - …
  - Bioconductor (http://bioconductor.org/)は半年ごとにリリース
    - 2017-10にver. 3.6をリリース (R ver. 3.4.Xで動作確認)、提供パッケージ数:1,477 ①
    - 2017-04にver. 3.5をリリース (R ver. 3.4.Xで動作確認)、提供パッケージ数:1,381
    - 2016-10にver. 3.4をリリース (R ver. 3.3.Xで動作確認)、提供パッケージ数:1,294
    - 2016-05にver. 3.3をリリース (R ver. 3.3.Xで動作確認)、提供パッケージ数:1,211
    - 2015-10にver. 3.2をリリース (R ver. 3.2.Xで動作確認)、提供パッケージ数:1,104
    - 2015-04にver. 3.1をリリース (R ver. 3.2.Xで動作確認)、提供パッケージ数:1,024
    - 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
    - …

定期的にバージョンアップ

①R本体に関しては、ここで書いている以外にも頻繁にリリースされている。例えば②の間にver. 3.4.3 (2017-11-30)や、ver. 3.4.2 (2017-09-28)などもある

- 近年のリリース頻度
  - R本体 (http://www.r-project.org/)
    - 2018-04-23にver. 3.5.0をリリース
    - 2018-03-15にver. 3.4.4をリリース
    - 2017-04-21にver. 3.4.0をリリース
    - 2017-03-06にver. 3.3.3をリリース
    - 2016-04-14にver. 3.2.5をリリース
    - …
  - Bioconductor (http://bioconductor.org/)は半年ごとにリリース
    - 2017-10にver. 3.6をリリース (R ver. 3.4.Xで動作確認)、提供パッケージ数:1,477
    - 2017-04にver. 3.5をリリース (R ver. 3.4.Xで動作確認)、提供パッケージ数:1,381
    - 2016-10にver. 3.4をリリース (R ver. 3.3.Xで動作確認)、提供パッケージ数:1,294
    - 2016-05にver. 3.3をリリース (R ver. 3.3.Xで動作確認)、提供パッケージ数:1,211
    - 2015-10にver. 3.2をリリース (R ver. 3.2.Xで動作確認)、提供パッケージ数:1,104
    - 2015-04にver. 3.1をリリース (R ver. 3.2.Xで動作確認)、提供パッケージ数:1,024
    - 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
    - …

定期的にバージョンアップ

基本的には、①でも書いているが、Bioconductorのリリースに合わせて、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。R本体のバージョンが新しくないと、最新版のBioconductorのパッケージがインストールされないので注意してください

- 近年のリリース頻度
  - R本体 (http://www.r-project.org/)
    - 2018-04-23にver. 3.5.0をリリース
    - 2018-03-15にver. 3.4.4をリリース
    - 2017-04-21にver. 3.4.0をリリース
    - 2017-03-06にver. 3.3.3をリリース
    - 2016-04-14にver. 3.2.5をリリース
    - …
  - Bioconductor (http://bioconductor.org/)は半年ごとにリリース
    - 2017-10にver. 3.6をリリース (R ver. 3.4.Xで動作確認)、提供パッケージ数:1,477
    - 2017-04にver. 3.5をリリース (R ver. 3.4.Xで動作確認)、提供パッケージ数:1,381
    - 2016-10にver. 3.4をリリース (R ver. 3.3.Xで動作確認)、提供パッケージ数:1,294
    - 2016-05にver. 3.3をリリース (R ver. 3.3.Xで動作確認)、提供パッケージ数:1,211
    - 2015-10にver. 3.2をリリース (R ver. 3.2.Xで動作確認)、提供パッケージ数:1,104
    - 2015-04にver. 3.1をリリース (R ver. 3.2.Xで動作確認)、提供パッケージ数:1,024
    - 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
    - …

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

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

①「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、「(Rで)塩基配列解析」で利用する多くの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)
- 参考資料 | 書籍、学会誌 (last modified 2017/03/13)
- 参考資料 | 講習会、講義、講演資料 (last modified 2016/12/07)
- 過去のお知らせ (last modified 2017/04/10) NEW
- インストール | について (last modified 2015/11/12)
- インストール | R本体 | 最新版 | Win用 (last modified 2015/03/22)推奨
- インストール | R本体 | 最新版 | Mac用 (last modified 2015/04/22)推奨
- インストール | R本体 | 過去版 | Win用 (last modified 2015/03/22)
- インストール | R本体 | 過去版 | Mac用 (last modified 2015/03/22)
- インストール | Rパッケージ | ほぼ全て(20GB以上?!) (last modified 2015/05/25)
- インストール | Rパッケージ | 必要最小限プラスアルファ(数GB?!) (last modified 2017/03/13)推奨
- インストール | Rパッケージ | 必要最小限プラスアルファ(アグリバイオ居室のみ) (last modified 2015/06/16)
- インストール | Rパッケージ | 必要最小限(数GB?!) (last modified 2015/05/25)
- インストール | Rパッケージ | 個別 (last modified 2015/06/10)
- (削除予定)Rのインストールと起動 (last modified 2016/02/21)
- (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)
- サンプルデータ (last modified 2016/10/05)
- 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2016 (last mo
- 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2015 (last mo
- 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2014 (last modified 2015/
- 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 2016/03/17)

パッケージのインストール

①これらはCRANから提供されているものたち。②「バイオスタティスティクス基礎論」で利用予定のパッケージは、ここに書き込んでいる

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。「options(repos="http://cran.ism.ac.jp/")」が使えなくなっているという指摘を受けたので2016.04.11にコメントアウトしました。

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 ①

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。

```
#options(repos="http://cran.ism.ac.jp/")#利用するリポジトリを指定(統計数理研究所の場合。使え
#(Rで)塩基配列解析で主に利用
install.packages("openxlsx")            #EXCELファイル(.xlsx)を直接読み込むためのパッケージ。
install.packages("PoissonSeq")
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
```

②

パッケージのインス

①ゲノム配列のパッケージ(BSgenome…)はBioconductorから提供されている。ここでは計6パッケージをインストールしている。例えば②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)に相当する。③biocLite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る

```
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)
```

③ ②

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
  - 仮想データ
  - 実データ(課題)
  - 作図

BSgenome利用の意義

ゲノム配列情報はUCSC、Ensembl、Illumina iGenomes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。①ChIP-seq解析用パッケージの②MEDIPSは、BSgenomeを利用

- 解析 | 機能解析 | パスウェイ(Pathway)解析 | SeqGSEA(Wang 2014) (last modified
- 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 (last 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

①

解析 | ChIP-seq | につ

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

②

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

①「…ゲノム配列 | BSgenome」

BSgenome

- (削除予定)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 | 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 2015/04/22)
- 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 2016/02/
- 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
- イントロ | 一般 | 配列取得 | トランスクリプトーム配列
- イントロ | 一般 | 読み込み | xlsx形式 | openxlsx (last

①

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

①黒枠部分のコードをコピペ。R ver. 3.4.3 (Bioconductor ver. 3.4)で利用可能な生物種のパッケージ名をリストアップ。②87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 NE

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(ンパンジー(P.troglodytes)、ラット(R.norvegicus)、出芽酵母(S.cerevisiae)、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。した以外に「chr1_random」や「chrUn_random」なども等価に取扱っている点に注「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

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

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

#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
available.genomes()                        #このパッケージ中

#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

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

①

R Console

```
[68]  "BSgenome.Ptroglodytes.UCSC.panTro3"
[69]  "BSgenome.Ptroglodytes.UCSC.panTro3.masked"
[70]  "BSgenome.Ptroglodytes.UCSC.panTro5"
[71]  "BSgenome.Rnorvegicus.UCSC.rn4"
[72]  "BSgenome.Rnorvegicus.UCSC.rn4.masked"
[73]  "BSgenome.Rnorvegicus.UCSC.rn5"
[74]  "BSgenome.Rnorvegicus.UCSC.rn5.masked"
[75]  "BSgenome.Rnorvegicus.UCSC.rn6"
[76]  "BSgenome.Scerevisiae.UCSC.sacCer1"
[77]  "BSgenome.Scerevisiae.UCSC.sacCer2"
[78]  "BSgenome.Scerevisiae.UCSC.sacCer3"
[79]  "BSgenome.Sscrofa.UCSC.susScr3"
[80]  "BSgenome.Sscrofa.UCSC.susScr3.masked"
[81]  "BSgenome.Tgondii.ToxoDB.7.0"
[82]  "BSgenome.Tguttata.UCSC.taeGut1"
[83]  "BSgenome.Tguttata.UCSC.taeGut1.masked"
[84]  "BSgenome.Tguttata.UCSC.taeGut2"
[85]  "BSgenome.Vvinifera.URGI.IGGP12Xv0"
[86]  "BSgenome.Vvinifera.URGI.IGGP12Xv2"
[87]  "BSgenome.Vvinifera.URGI.IGGP8X"
> |
```

②

BSgenome

①2014年4月リリースのゼブラフィッシュ(*Danio rerio*; danRer10)のパッケージもある。②ヒトゲノムはこのあたり。様々なバージョン(hg17, hg18, hg19, hg38)のゲノム配列が提供されていることがわかる

R Console

```
> available.genomes()                          #このパ$
 [1] "BSgenome.Alyrata.JGI.v1"
 [2] "BSgenome.Amellifera.BeeBase.assembly4"
 [3] "BSgenome.Amellifera.UCSC.apiMel2"
 [4] "BSgenome.Amellifera.UCSC.apiMel2.masked"
 [5] "BSgenome.Athaliana.TAIR.04232008"
 [6] "BSgenome.Athaliana.TAIR.TAIR9"
 [7] "BSgenome.Btaurus.UCSC.bosTau3"
 [8] "BSgenome.Btaurus.UCSC.bosTau3.masked"
 [9] "BSgenome.Btaurus.UCSC.bosTau4"
[10] "BSgenome.Btaurus.UCSC.bosTau4.masked"
[11] "BSgenome.Btaurus.UCSC.bosTau6"
[12] "BSgenome.Btaurus.UCSC.bosTau6.masked"
[13] "BSgenome.Btaurus.UCSC.bosTau8"
[14] "BSgenome.Celegans.UCSC.ce10"
[15] "BSgenome.Celegans.UCSC.ce11"
[16] "BSgenome.Celegans.UCSC.ce2"
[17] "BSgenome.Celegans.UCSC.ce6"
[18] "BSgenome.Cfamiliaris.UCSC.canFam2"
[19] "BSgenome.Cfamiliaris.UCSC.canFam2.masked"
[20] "BSgenome.Cfamiliaris.UCSC.canFam3"
[21] "BSgenome.Cfamiliaris.UCSC.canFam3.masked"
[22] "BSgenome.Dmelanogaster.UCSC.dm2"
[23] "BSgenome.Dmelanogaster.UCSC.dm2.masked"
[24] "BSgenome.Dmelanogaster.UCSC.dm3"
[25] "BSgenome.Dmelanogaster.UCSC.dm3.masked"
[26] "BSgenome.Dmelanogaster.UCSC.dm6"
[27] "BSgenome.Drerio.UCSC.danRer10"
[28] "BSgenome.Drerio.UCSC.danRer5"
[29] "BSgenome.Drerio.UCSC.danRer5.masked"
[30] "BSgenome.Drerio.UCSC.danRer6"
[31] "BSgenome.Drerio.UCSC.danRer6.masked"
[32] "BSgenome.Drerio.UCSC.danRer7"
[33] "BSgenome.Drerio.UCSC.danRer7.masked"
[34] "BSgenome.Ecoli.NCBI.20080805"
[35] "BSgenome.Gaculeatus.UCSC.gasAcu1"
[36] "BSgenome.Gaculeatus.UCSC.gasAcu1.masked"
[37] "BSgenome.Ggallus.UCSC.galGal3"
[38] "BSgenome.Ggallus.UCSC.galGal3.masked"
[39] "BSgenome.Ggallus.UCSC.galGal4"
[40] "BSgenome.Ggallus.UCSC.galGal4.masked"
[41] "BSgenome.Ggallus.UCSC.galGal5"
[42] "BSgenome.Hsapiens.1000genomes.hs37d5"
[43] "BSgenome.Hsapiens.NCBI.GRCh38"
[44] "BSgenome.Hsapiens.UCSC.hg17"
[45] "BSgenome.Hsapiens.UCSC.hg17.masked"
[46] "BSgenome.Hsapiens.UCSC.hg18"
[47] "BSgenome.Hsapiens.UCSC.hg18.masked"
[48] "BSgenome.Hsapiens.UCSC.hg19"
[49] "BSgenome.Hsapiens.UCSC.hg19.masked"
[50] "BSgenome.Hsapiens.UCSC.hg38"
[51] "BSgenome.Hsapiens.UCSC.hg38.masked"
[52] "BSgenome.Mfascicularis.NCBI.5.0"
[53] "BSgenome.Mfuro.UCSC.musFur1"
[54] "BSgenome.Mmulatta.UCSC.rheMac2"
[55] "BSgenome.Mmulatta.UCSC.rheMac2.masked"
[56] "BSgenome.Mmulatta.UCSC.rheMac3"
[57] "BSgenome.Mmulatta.UCSC.rheMac3.masked"
[58] "BSgenome.Mmulatta.UCSC.rheMac8"
[59] "BSgenome.Mmusculus.UCSC.mm10"
[60] "BSgenome.Mmusculus.UCSC.mm10.masked"
[61] "BSgenome.Mmusculus.UCSC.mm8"
[62] "BSgenome.Mmusculus.UCSC.mm8.masked"
[63] "BSgenome.Mmusculus.UCSC.mm9"
```

①

②

BSgenome

①実際にインストール済みのものを調べる。②このPC環境では、7パッケージであることがわかる。③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。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

イントロ | 一般 | 配列取得 | ゲノム配列 | BSgenom

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
lyrata)、セイヨウミツバチ (A. mellifera)、シロイヌナズナ(A.thaliana)、ウシ(
(C.familiaris)、キイロショウジョウバエ(D.melanogaster)、ゼブラフィッシュ(
(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

#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

#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

#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
```

①

R Console

```
[83] "BSgenome.Tguttata.UCSC.taeGut1.masked"
[84] "BSgenome.Tguttata.UCSC.taeGut2"
[85] "BSgenome.Vvinifera.URGI.IGGP12Xv0"
[86] "BSgenome.Vvinifera.URGI.IGGP12Xv2"
[87] "BSgenome.Vvinifera.URGI.IGGP8X"
> installed.genomes()
[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"
>
```

②

③

個別インストール

①パッケージの個別インストール方法。②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。
例：BSgenome.Athaliana.TAIR.TAIR9

- インストール | R本体 | 過去版 | Win用 (last modified 2015/03/22)
- インストール | R本体 | 過去版 | Mac用 (last modified 2015/03/22)
- インストール | Rパッケージ | ほぼ全て(20GB以上?!) (last modified 2015/05/25)
- インストール | Rパッケージ | 必要最小限プラスアルファ(数GB?!) (last modified 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(アグリバイオ居室のみ)
- インストール | Rパッケージ | 必要最小限(数GB?!) (last modified 2015/05/25)
- インストール | Rパッケージ | 個別 (last modified 2015/06/10)
- (削除予定)Rのインストールと起動 (last modified 2016/02/21)
- (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)
- サンプルデータ (last modified 2015/06/15)

①

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
```

②

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
  - 仮想データ
  - 実データ(課題)
  - 作図

BSgenome

①例題9。②ヒトゲノム(GRCh38)のRパッケージを入力、③multi-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。(動作が遅くなるので)**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

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
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
「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
library(BSgenome

#本番（利用可能な
available.genomes

#本番（インストール
installed.genomes

#後処理（パッケージ
installed.genomes
```

①

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。

②

③

```
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

①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。②慣れればfastaオブジェクトの中身をR上で直接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
param <- "BSgenome.Hsapiens.NCBI.GRCh38"#ハ

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

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

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

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

①

②

R Console

```
# a given sequence)
>
> #本番
> fasta <- getSeq(genome)                       #ゲノム塩基$
> names(fasta) <- seqnames(genome)              #description$
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", $
> fasta
  A DNAStringSet instance of length 455
          width seq                     names               $
   [1] 248956422 NNNNNNNN...NNNNNNNN 1
   [2] 242193529 NNNNNNNN...NNNNNNNN 2
   [3] 198295559 NNNNNNNN...NNNNNNNN 3
   [4] 190214555 NNNNNNNN...NNNNNNNN 4
   [5] 181538259 NNNNNNNN...NNNNNNNN 5
   ...       ... ...
 [451]    200773 TCTACTCT...GGGAATTC HSCHR19KIR_FH08_B$
 [452]    170148 TTTCTTTC...GGGAATTC HSCHR19KIR_FH13_A$
 [453]    215732 TGTGGTGA...GGGAATTC HSCHR19KIR_FH13_B$
 [454]    170537 TCTACTCT...GGGAATTC HSCHR19KIR_FH15_A$
 [455]    177381 GATCTATC...GGGAATTC HSCHR19KIR_RP5_B_$
> |
```

BSgenome

①1～22番染色体のみ取扱いたい場合。
②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

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)

#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, s
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
```

①

```
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
>
```

②

BSgenome

①X, Y, およびミトコンドリア配列も含めたい場合。②配列の並びの確認は試行錯誤。③最初から25番目の要素がMT（ミトコンドリア）だとわかっていたわけではありません

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)

#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, s
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
```

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

①X, Y, およびミトコンドリア配列までのサブセットをhoge10.fastaという名前で保存したい場合。②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、③水色下線部分の2か所を変更すればよい

9. インストール済みのヒト("BSgenome.Hsapiens.NCBI.GRCh38")の

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, s
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
```

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$
```

①

②

③

③

参考

BSgenome

①こんな感じで変更して実行。やらなくてよい。実行後にhoge9.fastaよりも若干ファイルサイズの小さい②hoge10.fastaが生成されているのが確認できます。**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

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, se
genome <- eval(parse(text=tmp))
genome

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

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

R Console

```
[22]  50818468 NNNNNNNNNN...NNNNNNNNN 22
> fasta[1:25]
  A DNAStringSet instance of length 25
      width seq                     names
 [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$
>
```

② hoge10.fasta 2017/04/12 13:33 3,136,542 KB FASTA ファイル

hoge9.fasta 2017/04/12 13:24 3,259,441 KB FASTA ファイル

① writeXStringSet(fasta[1:25], file="hoge10.fasta", forma$

参考

①例題10。様々な記述形式があります。やらなくてよい

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. 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.trogl
な生物種が利用可
あたりにリストアップさ
以外に「chr1_random」
「ファイル」-「ディレクト

1. 利用可能な生物種と

```
#必要なパッケージ
library(BSgenome

#本番（利用可能な
available.genome

#本番（インストー
installed.genome

#後処理（パッケー
installed.genom
```

①

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)           #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

#後処理(フィルタリング)
obj <- param_range                         #抽出したいリードの位置情報を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
```

BSgenome

①26番目以降の配列は、ヒトゲノムの一部ではあるものの、②おそらく割り当てられる染色体が定まっていないものなどです。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.GRCh

2013年12月にリリースされたGenome Reference Consortium
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ブジ
#tmp <- unlist(strsplit(param, ".", fix
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, forma
```

R Console

```
> fasta
  A DNAStringSet instance of length 455
          width seq                    names
   [1] 248956422 NNNNNNNN...NNNNNNNN 1
   [2] 242193529 NNNNNNNN...NNNNNNNN 2
   [3] 198295559 NNNNNNNN...NNNNNNNN 3
   [4] 190214555 NNNNNNNN...NNNNNNNN 4
   [5] 181538259 NNNNNNNN...NNNNNNNN 5
   ...       ... ...
 [451]    200773 TCTACTCT...GGGAATTC HSCHR19KIR_FH08_B...
 [452]    170148 TTTCTTTC...GGGAATTC HSCHR19KIR_FH13_A...
 [453]    215732 TGTGGTGA...GGGAATTC HSCHR19KIR_FH13_B...
 [454]    170537 TCTACTCT...GGGAATTC HSCHR19KIR_FH15_A...
 [455]    177381 GATCTATC...GGGAATTC HSCHR19KIR_RP5_B_...
> names(fasta[26:27])
[1] "HSCHR1_CTG1_UNLOCALIZED" "HSCHR1_CTG2_UNLOCALIZED"
> names(fasta[454:455])
[1] "HSCHR19KIR_FH15_A_HAP_CTG3_1"
[2] "HSCHR19KIR_RP5_B_HAP_CTG3_1"
>
```

①

②

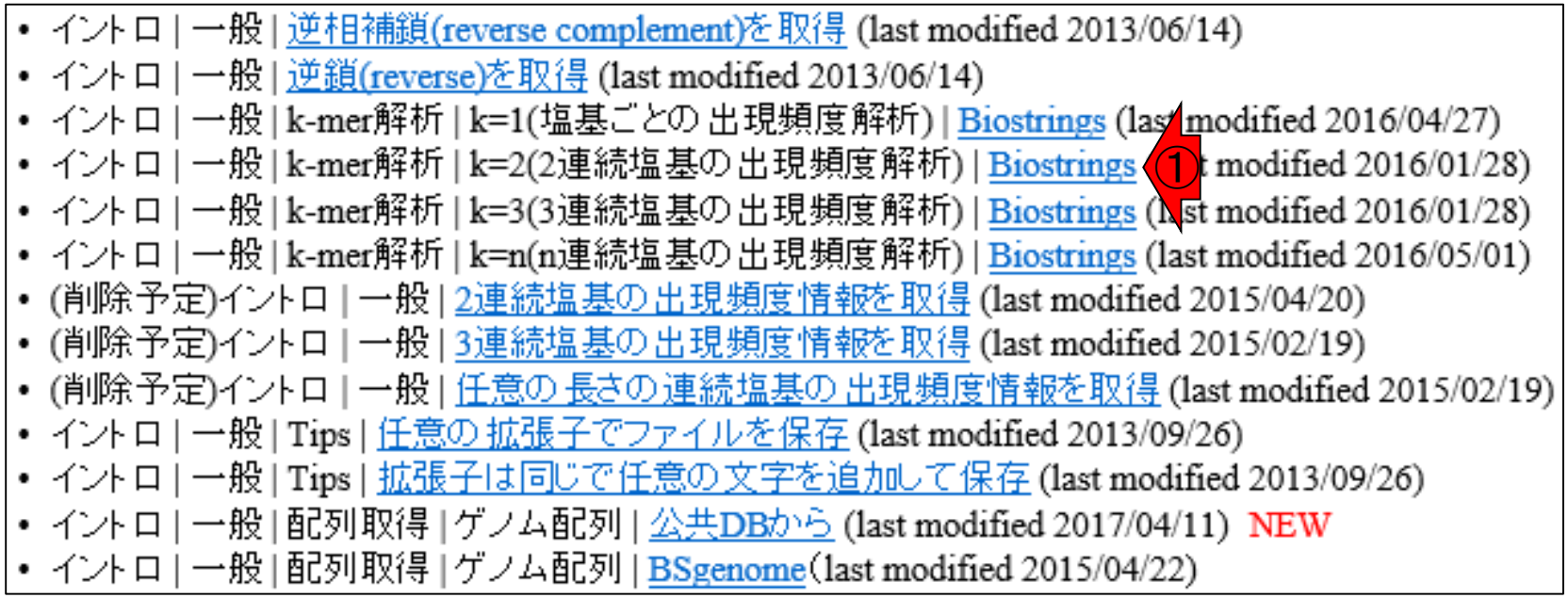
Contents

- 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
  - 仮想データ
  - 実データ(課題)
  - 作図

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%



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

2連続塩基の出現頻度

①例題1。全貌を把握可能な②hoge4.fa を作業ディレクトリに保存して実行

イントロ | 一般 | k-mer解析 | k=2(2連続塩基の出現頻度解析) | Biostrings

Biostringsパッケージを用いて、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。k-mer解析のk=2の場合に相当します。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

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:

②

タイトル通りの出現頻度です。

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

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

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

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

#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=
```

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
```

2連続塩基の出現頻度

①右クリックで保存し、②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!

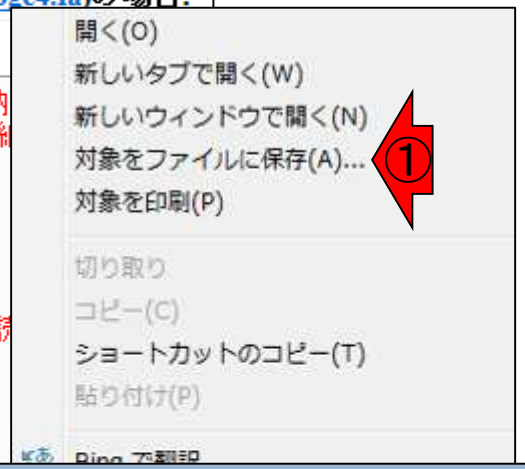
イントロ | 一般 | k-mer解析 | k=2(2連続塩基の出現頻度解析) | Biostrin

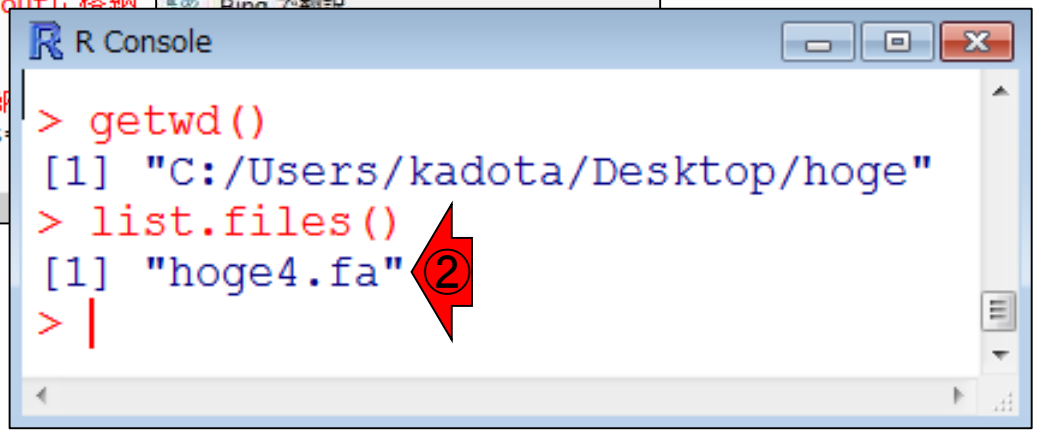
Biostringsパッケージを用いて、multi-FASTA形式ファイルを読み込んで、"AA", "AC", "AG", "AT", "CA", "CC", "CG", "CT", "GA", "GC", "GG", "GT", "TA", "TC", "TG", "TT" の計$4^2 = 16$通りの2連続塩基の出現頻度を調べるやり方を示します。k-mer解析のk=2の場合に相当します。ヒトゲノムで"CG"の割合が期待値よりも低い(Lander et al., 2001; Saxonov et al., 2006)ですが、それを簡単に検証できます。
「ファイル」-「ディレクトリの変更」で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に格納
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読
fasta                                       #確認してるだけです

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

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





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。①出力ファイルの中身は②tmpオブジェクトの中身と同じ

イントロ | 一般 | k-mer解析 | k=2(2連続塩基の出現頻度解析)

Biostringsパッケージを用いて、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。k-mer解析のk=2の場合に相当します。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
out_f <- "hoge1.txt"                        #出力ファイル名

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

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

#本番
out <- dinucleotideFrequency(fasta)         #連続塩基の出現

#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,
```

①

②

R Console

```
> out <- dinucleotideFrequency(fasta)     #連続塩$
>
> #ファイルに保存
> tmp <- cbind(names(fasta), out)         #保存し$
> write.table(tmp, out_f, sep="\t", append=F, quo$
> 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

①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 | k-mer解析 | k=2(2連続塩基の出現頻度解析) | Biostrings

Biostringsパッケージを用いて、multi-FASTA形式ファイルを読み込A
"CG", "CT", "GA", "GC", "GG", "GT", "TA", "TC", "TG", "TT" の計
るやり方を示します。k-mer解析のk=2の 場合に相当します。ヒトゲノ
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

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

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")#in
fasta                                 #確認してる

#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(na
write.table(tmp
```

①

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

①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 as.probオプションをTRUEにしているだけ

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(

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)#

#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, quo
```

①

②

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%

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
  - 仮想データ
  - 実データ(課題)
  - 作図

2連続塩基の出現確率

①例題7。②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、③fasta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。（孫 建強氏提供情報）。実行時間は約3分

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

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(in_f,
fasta

#本番
out <- dinucleotideFrequency(fast

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

①

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

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

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

#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
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

#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
```

②

③

2連続塩基の出現確率

①例題9は、②例題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。③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:

②

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

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

#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
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)
names(fasta) <- seqnames(geno
fasta

#本番
out <- dinucleotideFrequency(fas

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

①

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
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
```

③

2連続塩基の出現確率

例題7実行結果ファイル。CGの連続塩基が他に比べて確かに低いことがわかる

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%

 参考

2連続塩基の出現頻度と確率

①例題8。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。②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
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

#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
```

②

②

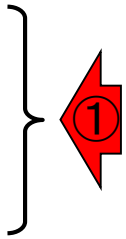
R Console

```
        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(k-mer)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。①の話の一部は、平成28年度NGSハンズオン講習会7月20日の講義資料中にもあります

- 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=25～200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

multi-FASTAファイル(kadai_20180507.fasta)について、各種基本情報、および2連続塩基の出現確率を調べ、得られた結果について考察せよ

講義日程（平成30年度）

1. 平成30年04月16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；完全版)
   講義資料PDF(Mac版；Rの説明部分のみ)
2. 平成30年04月23日（PC使用）
   講師：門田幸二
   講義資料PDF
   (Rで)塩基配列解析
   hoge7.fa (課題用)
3. 平成30年05月07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
   kadai_20180507.fasta ①(課題用)
4. 平成30年05月14日（PC使用）
   講師：勝間 進

http://www.iu.a.u-tokyo.ac.jp/~kadota/R_seq/kadai_20180507.fasta

iu.a.u-tokyo.ac.jp

```
>seq1
TTTTCTTATTTTTATATAACCAAATGTGTTTCCTTGATAACTTTGTTTGC
TCAAACACTCAAAACCAAGAATTGTTCAATTGAAGTTAACTTACTCATTT
CAACTCTGCAGTCCATAATGTCTGTCTTCCTATTTAATGTCATGTCCAAC
CATCTGTATGACTTTTGGCTGCATTTAGTAGGAATCTGAATACAATGGCT
TATTGAAACAGAGATATATTTCTCACGCAGAAAGAAATTCAGAGGCAG
CAAGTGCATGGACTCATTCCACTATGTCATAAGGGACTCAGTCTTCTTTT
CTTTCTGCCTCTTTAGCTTGTGGCTGTGTCTTCATTCTGGCCATATGGTT
GCTACATTGGCAGTCACAATTGACAAGTTGGGAAGATGCTGACCATTTGT
CTGAACTGTCTGTTTTTACTTTTTAAACAGGAAAAGGATAGCTTTTCCAG
AAATCTTACAGATTTGATTTAAAGAATAGTCTTTCATATATATTTAATTA
TCTGTAACTTGGTCAAAGGGCAATTGATTGCCACAAGTGAGTACTTGCAA
GGTATTTGCAAATATACACTTTTCTAACTCCAAACAAAATGGGGTCTGTT
GGAAAAGCAGATAATGATCTGAATACTAAGGAGATAATGAGAAGATATAT
TTGGATTTGAATCATGATACTTTGGTTTACTTACAGAGACTTTTAATAGA
TTTTTCTAGTCACTTTGAACAGATCTTCTCATCTCCTAAATGTTGAAATA
GCATCTATTTATCTGCATGCTTATGAGAAGTTAATAAGACAATATCAATC
ATTTAGTGAAATAGCTGAATCCTATCATGTTATTATTACATGAAGGGTTA
```

入力が①BSgenomeのRパッケージでもGC含量を含む基本情報を取得することができる。②例題1

ヒトゲノムの場合

- イントロ | NGS | アノテーション情報取得 | TxDb | 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 | TxDb | TxDb.*から (last modified 2015/02/19)
- イントロ | NGS | アノテーション情報取得 | TxDb | GenomicFeatures(Lawrence 2013) (last
- イントロ | NGS | アノテーション情報取得 | TxDb | GFF/GTF形式ファイルから (last modif
- イントロ | NGS | 読み込み | BSgenome | 基本情報を取得 ① modified 2015/09/12)
- イントロ | NGS | 読み込み | FASTA形式 | 基本情報を取得 last modified 2016/04/21) N
- イントロ | NGS | 読み込み | FASTA形式 | description行の記述を整形 (last modified 201
- イントロ | NGS | 読み込み | FASTQ形式 | 基礎 (last modified 2015/07/26)
- イントロ | NGS | 読み込み | FASTQ形式 | 応用 (last modified 2015/06/18)
- イントロ | NGS | 読み込み | FASTQ形式 | descr
- イントロ | NGS | 読み込み | Illuminaの* seq.txt
- イントロ | NGS | 読み込み | Illuminaの* qseq.tx
- イントロ | ファイル形式の変換 | について (last n
- イントロ | ファイル形式の変換 | BAM --> BED (

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。マウスやヒトゲノム解析の場合に「整数オーバーフロー」問題が生じていましたが、Total_lenのところで「sum(width(fasta))」を「sum(as.numeric(width(fasta)))」に、そしてsortedのところで「rev(sort(width(fasta)))」を「rev(sort(as.numeric(width(fasta))))」と書き換えることで回避可能であるという情報をいただきましたので、2015年5月27日にそのように変更しました(野間口達洋 氏提供情報)。

「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。

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

GC含量は約41%となります。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系のゲノムパ

#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
```

ヒトゲノムの場合

コピペ実行結果。①入力がBSgenomeのパッケージの場合、入力”ファイル”はない。②出力ファイルの中身に相当するtmpを表示。③配列数は455個、④全体のGC含量は約41%

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

GC含量は約41%となります。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系のゲノムパ

#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
tmp <- ls(paste("package", param_bsg
genome <- eval(parse(text=tmp))
fasta <- getSeq(genome)
names(fasta) <- seqnames(genome)
fasta

#本番(基本情報取得)
Total_len <- sum(as.numeric(width(
Number_of_contigs <- length(fasta)
Average_len <- mean(width(fasta))
Median_len <- median(width(fasta))
Max_len <- max(width(fasta))
Min_len <- min(width(fasta))
```

①

②

R Console

```
> tmp <- rbind(tmp, c("Median length", Median_len))
> tmp <- rbind(tmp, c("Max length", Max_len))
> tmp <- rbind(tmp, c("Min length", Min_len))
> tmp <- rbind(tmp, c("N50", N50))
> tmp <- rbind(tmp, c("GC content", GC_content))
> write.table(tmp, out_f, sep="\t", append=F, quote=F, ro$
> tmp
     [,1]                  [,2]
[1,] "Total length (bp)" "3209286105"
[2,] "Number of contigs" "455"
[3,] "Average length"    "7053376.05494506"
[4,] "Median length"     "161218"
[5,] "Max length"        "248956422"
[6,] "Min length"        "970"
[7,] "N50"               "145138636"
[8,] "GC content"        "0.409948515653618"
>
```

③

④

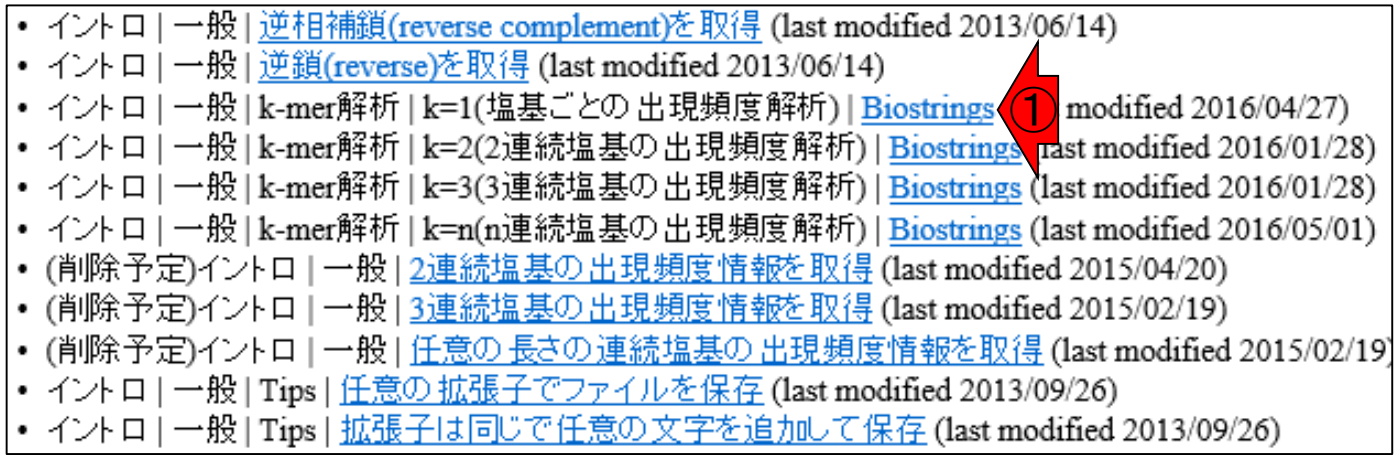
ヒトゲノムの場合

(1) 配列数は455個(BSgenome…の場合)

(2) 全体のGC含量は約41%

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

(2) 全体のGC含量は0.410だった。これはCとGの出現確率の合計が0.410ということを意味する。(3)それゆえ、各々の確率に分割すると、0.410/2 = 0.205となる。もちろん①などを参考にして塩基ごとの出現確率の実測値を求めてもよい(が結果はほぼ同じなのでやらなくてよい)



課題の(1)～(3)は

①のコードをテンプレートにして、②GC含量情報を取得した後、③を計算すればよい

(1) 配列数は455個(BSgenome…の場合)

(2) 全体のGC含量は約41% ②

(3) 各塩基(A, C, G, T)の出現確率 : (0.295, 0.205, 0.205, 0.295) ③

- 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 | TxDb | 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 | TxDb | TxDb.*から (last modified 2015/02/19)
- イントロ | NGS | アノテーション情報取得 | TxDb | GenomicFeatures(Lawrence 2013) (last modified 2015/
- イントロ | NGS | アノテーション情報取得 | TxDb | GFF/GTF形式ファイルから (last modified 2016/02/09)
- イントロ | NGS | 読み込み | BSgenome | 基本情報を取得 (last modified 2016/04/22)
- イントロ | NGS | 読み込み | FASTA形式 | 基本情報を取得 ① modified 2016/04/22)
- イントロ | NGS | 読み込み | FASTA形式 | description行の記述を整形 (last modified 2014/04/05)
- イントロ | NGS | 読み込み | FASTQ形式 | 基礎 (last modified 2015/07/26)
- イントロ | NGS | 読み込み | FASTQ形式 | 応用 (last modified 2015/06/18)
- イントロ | NGS | 読み込み | FASTQ形式 | description行の記述を整形 (last modified 2014/08/21)
- イントロ | NGS | 読み込み | SAM/BAM形式 (last modified 2016/09/14)
- イントロ | NGS | 読み込み | Illuminaの* seq.txt (last modified 2013/06/13)
- イントロ | NGS | 読み込み | Illuminaの* qseq.txt (last modified 2013/06/17)
- イントロ | ファイル形式の変換 | について (last modified 2014/06/09)
- イントロ | ファイル形式の変換 | BAM --> BED (last modified 2014/06/21)

ヒトゲノムの場合

①AとTの出現確率の合計は、GC含量(0.410)から、1 − 0.410 = 0.590となる。それゆえ、AとT各々の確率に分割すると、0.590/2 = 0.295となる。2連続塩基の出現確率は、各塩基の出現確率の掛け算で計算可能。AA, AT, TA, TTの出現確率の**期待値**(expected)は、どれも0.295×0.295 = 0.087025 (約8.7%)となる。他も同様

(1) 配列数は455個(BSgenome…の場合)

(2) 全体のGC含量は約41%

(3) 各塩基(A, C, G, T)の出現確率: (0.295, 0.205, 0.205, 0.295) ①

(4) AA, AT, TA, TTの出現確率の期待値 = 0.295×0.295 = 8.7% ②

(5) CC, **CG**, GC, GGの出現確率の期待値 = 0.205×0.205 = 4.2%

(6) AC, AG, CA, CT , GA , GT , TC , TGの出現確率の期待値 = 0.205×0.295 = 6.0%

①の情報を用いて計算すればよい

課題の(4)～(6)は

(1) 配列数は455個(BSgenome…の場合)

(2) 全体のGC含量は約41%

(3) 各塩基(A, C, G, T)の出現確率 : (0.295, 0.205, 0.205, 0.295) ①

(4) AA, AT, TA, TTの出現確率の期待値 = 0.295 × 0.295 = 8.7%

(5) CC, **CG**, GC, GGの出現確率の期待値 = 0.205 × 0.205 = 4.2%

(6) AC, AG, CA, CT , GA , GT , TC , TGの出現確率の期待値 = 0.205 × 0.295 = 6.0%

確率の期待値?!

毎年「確率の期待値って何ですか?」という質問を数名程度から受けるので、これについて説明します

(1) 配列数は455個(BSgenome…の場合)

(2) 全体のGC含量は約41%

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

(4) AA, AT, TA, TTの出現確率の期待値 = 0.295 × 0.295 = 8.7%

(5) CC, **CG**, GC, GGの出現確率の期待値 = 0.205 × 0.205 = 4.2%

(6) AC, AG, CA, CT , GA , GT , TC , TGの出現確率の期待値 = 0.205 × 0.295 = 6.0%

まず、①の計算結果は、
②の情報を用いています

確率の期待値?!

(1) 配列数は455個(BSgenome…の場合)

(2) 全体のGC含量は約41%

(3) 各塩基(A, C, G, T)の出現確率 : (0.295, 0.205, 0.205, 0.295) ②

(4) AA, AT, TA, TTの出現確率の期待値 = 0.295 × 0.295 = 8.7%

(5) CC, **CG**, GC, GGの出現確率の期待値 = 0.205 × 0.205 = 4.2%

(6) AC, AG, CA, CT , GA , GT , TC , TGの出現確率の期待値 = 0.205 × 0.295 = 6.0%

①

確率の期待値?!

で、②は、③の実測値（約41%という結果）を元に、「C = G and A = Tなので、このように割り振られるはず」という値です

(1) 配列数は455個(BSgenome…の場合)

(2) 全体のGC含量は約41%

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

(4) AA, AT, TA, TTの出現確率の期待値 = 0.295 × 0.295 = 8.7%

(5) CC, **CG**, GC, GGの出現確率の期待値 = 0.205 × 0.205 = 4.2%

(6) AC, AG, CA, CT , GA , GT , TC , TGの出現確率の期待値 = 0.205 × 0.295 = 6.0%

確率の期待値?!

①は、CCとCGとGCとGGという2連続塩基の出現確率が同じだと仮定した場合、これらの出現確率は理論上4.2%で同じはずなので、「出現確率の期待値」と書いています

(1) 配列数は455個(BSgenome…の場合)

(2) 全体のGC含量は約41%

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

(4) AA, AT, TA, TTの出現確率の期待値 = 0.295 × 0.295 = 8.7%

(5) CC, **CG**, GC, GGの出現確率の期待値 = 0.205 × 0.205 = 4.2% ①

(6) AC, AG, CA, CT , GA , GT , TC , TGの出現確率の期待値 = 0.205 × 0.295 = 6.0%

考察の基本的な考え方

①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CC, GC, GG）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 ①
  - 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)

    (0.00, 0.00, 0.00, 0.00, 0.00, 0.25, 0.25, 0.00, 0.00, 0.25, 0.25, 0.00, 0.00, 0.00, 0.00, 0.00)

考察の基本的な考え方

①GC含量100%の場合は、CとGの出現確率はそれぞれ0.5。よって、②CC, CG, GC, GGの出現確率は全て0.5×0.5 = 0.25となる。これが期待値。もし出現確率の実測値が例えばCCのみ高い(or低い)だったら、何かその生物にとって意味のあることなのだろう。これが「差分に関する議論が重要」という意味です

- 目的:2連続塩基の出現頻度(or 確率)
  - ヒトゲノムはCGという連続塩基の出現頻度
    と言われており、大まかにその傾向は確認
    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)

    ①
  - 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)

    (0.00, 0.00, 0.00, 0.00, 0.00, 0.25, 0.25, 0.00, 0.00, 0.25, 0.25, 0.00, 0.00, 0.00, 0.00, 0.00)

    ② ②

ヒトゲノムの結果

再び②例題7の、③2連続塩基の出現確率計算結果ファイル(hoge7.txt)の考察に戻る

- 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 | k-mer解析 | k=1(塩基ごとの 出現頻度解析) | Biostrings (last modified 2016/02/07)
- イントロ | 一般 | k-mer解析 | k=2(2連続塩基の出現頻度解析) | Biostrings (last modified 2016/01/28)
- イントロ | 一般 | k-mer解析 | k=3(3連続塩基の出現頻度解析) | Biostrings (last modified 2016/01/28)
- イントロ | 一般 | k-mer解析 | k=n(n連続塩基の出現頻度解析) | Biostrings (last modified 2016/01/28)
- (削除予定)イン
- (削除予定)イン
- (削除予定)イン
- イントロ | 一般
- イントロ | 一般
- イントロ | 一般
- イントロ | 一般

① ②

イントロ | 一般 | k-mer解析 | k=2(2連続塩基の出現頻度解析) | Biostrings

Biostringsパッケージを用い

"CG", "CT", "GA", "GC",
やり方を示します。k-mer解析(
al., 2001; Saxonov et al., 2006)
「ファイル」-「ディレクトリの変更

1. イントロ | 一般 | ランダムな

タイトル通りの 出現頻度です

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

#必要なパッケージをロー
library(Biostrings)

#入力ファイルの読み込み
fasta <- readDNAStrin
fasta
```

③

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 enome Reference Consortium GRCh38です。出力は出現確率です。

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

#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
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

#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
```

ヒトゲノムの結果

①赤枠の数値が、②出力ファイル(hoge7.txt)中のAA, AT, TA, TTの出現確率の**実測値**(observed)。概ね期待値(8.7%)周辺の値になっていることがわかる。考察(discussion)としては、同一種類の連続塩基(AA and TT)のほうが、異なる種類の連続塩基(AT and TA)に比べて出現確率が高めである、が言えるのでは…。

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

2013年12月にリリースされたGenome Reference Consortium GRCh38で

②

```
out_f <- "hoge7.txt"                    #出力ファイル名
param_bsgenome <- "BSgenome.Hsapiens.NCBI.GRCh38"#パ

#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
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
```

期待値	8.7	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%

① ① ① ①

ヒトゲノムの場合

①CC, CG, GC, GGの出現確率の期待値は4.2%。②CGの出現確率の実測値(約1.0%)は、期待値(約4.2%)よりもかなり低いことは明らか。それ以外にもいくつか考察できると思います

(1) 配列数は455個(BSgenome…の場合)

(2) 全体のGC含量は約41%

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

(4) AA, AT, TA, TTの出現確率の期待値 = 0.295 × 0.295 = 8.7%

(5) CC, **CG**, GC, GGの出現確率の期待値 = 0.205 × 0.205 = 4.2% ①

(6) AC, AG, CA, CT , GA , GT , TC , TGの出現確率の期待値 = 0.205 × 0.295 = 6.0%

期待値	8.7	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%

②

課題の(7)は

①の、②例題2をテンプレートとして用いれば、③**2連続塩基の出現確率の実測値**が得られます

- 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 | k-mer解析 | k=1(塩基ごとの 出現頻度解析) | Biostrings (last modified 2016/02/07)
- イントロ | 一般 | k-mer解析 | k=2(2連続塩基の出現頻度解析) | Biostrings (last modified 2016/01/28)
- イントロ | 一般 | k-mer解析 | k=3(3連続塩基の出現頻度解析) | Biostrings (last modified 2016/01/28)
- イントロ | 一般 | k-mer解析 | k=n(n連続塩基の出現頻度解析) | Biostrings (last modified 2016/01/28)
- (削除予定)イントロ
- (削除予定)イントロ
- (削除予定)イントロ
- 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

①

イントロ | 一般 | k-mer解析 | k=2(2連続塩基の出現頻度解析) | Biostrings

Biostringsパッケージを用いて、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。k-mer解析のk=2の場合に相当します。ヒトゲノムで"CG"の割合が期待値よりも低い(Lander et al., 2001; Saxonov

②

「ファイル」−「ディレ

1. イントロ | 一般 | ラン

タイトル通りの 出現頻

```
in_f <- "hoge4.f
out_f <- "hoge1.

#必要なパッケージ
library(Biostrin

#入力ファイルの読
fasta <- readDNA
fasta
```

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。

③

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

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

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

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

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

③

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
  - 仮想データ
  - 実データ(課題)
  - 作図

作図(box plot):基本形

②例題10。box plotをPNG形式ファイルで出力するやり方の基本形

- 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 | k-mer解析 | k=1(塩基ごとの出現頻度解析) | Biostrings (last modified 2016/02/07)
- イントロ | 一般 | k-mer解析 | k=2(2連続塩基の出現頻度解析) | Biostrings (last modified 2016/01/28)
- イントロ | 一般 | k-mer解析 | k=3(3連続塩基の出現頻度解析) | Biostrings (last modified 2016/01/28)
- イントロ | 一般 | k-mer解析 | k=n(n連続塩基の出現頻度解析) | Biostrings (last modified 2016/01/28)
- (削除予定)イン
- (削除予定)イン
- (削除予定)イン
- イントロ | 一般
- イントロ | 一般
- イントロ | 一般
- イントロ | 一般

①

イントロ | 一般 | k-mer解析 | k=2(2連続塩基の出現頻度解析) | Biostrings

Biostringsパッケージを用いて、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。k-mer解析のk=2の場合に相当します。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

②

1. イントロ | 一般 | ランダム

タイトル通りの出現頻度で

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

#必要なパッケージをロ
library(Biostrings

#入力ファイルの読み込
fasta <- readDNASt
fasta
```

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

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

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

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

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

①PNGファイルのサイズを指定するところ

作図(box plot):基本形

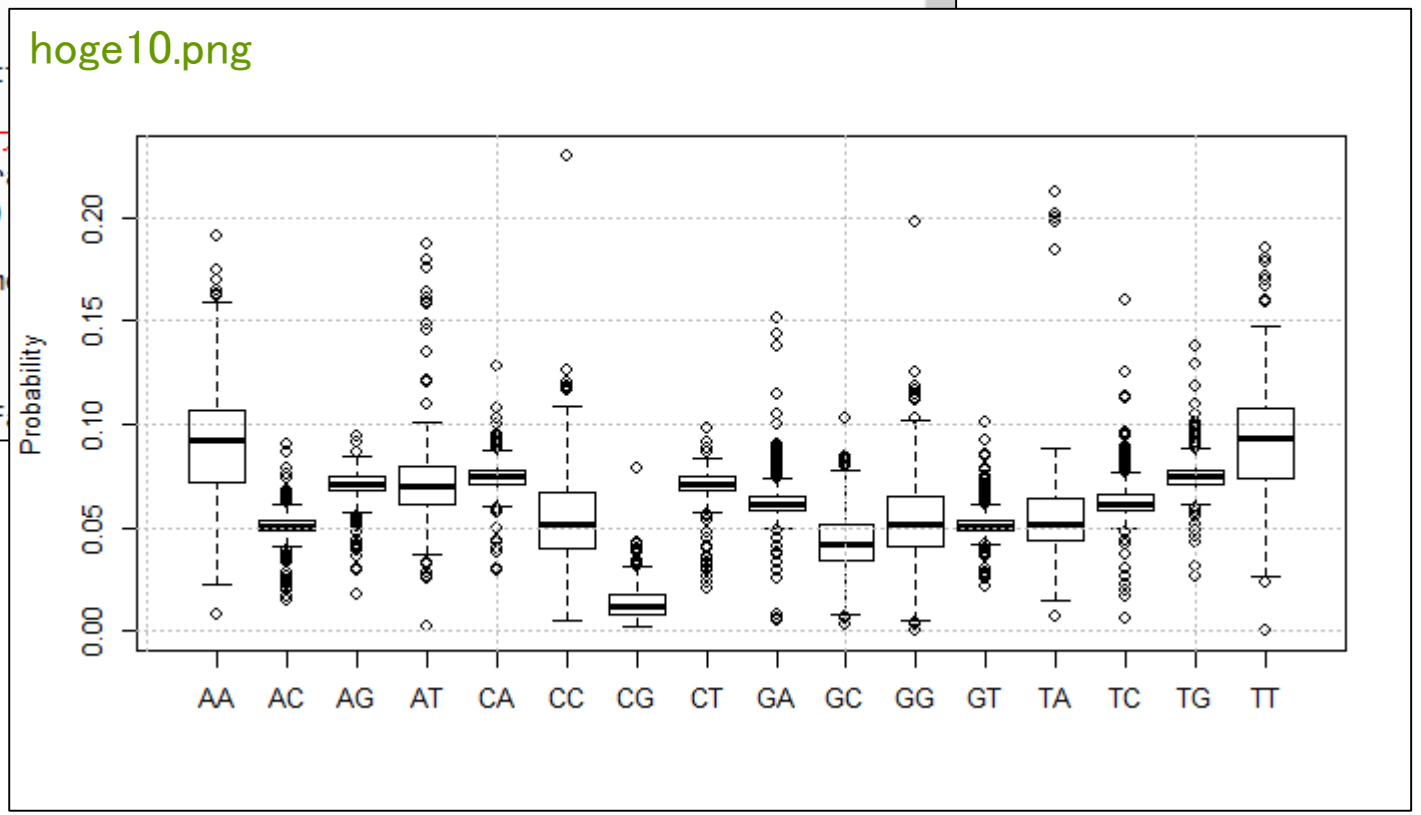
10.ヒトゲノム配列パッケージ(BSgenome.Hsapiens.NCBI.GRCh38)の場合:

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
```

①

hoge10.png



400 pixels

700 pixels

作図(box plot):色づけ

①例題11。②color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た、③タブ区切りファイル(human_2mer.txt)を与えて利用。④このファイルの情報を利用しているのは、コードの下のほう

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
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

①boxplot関数実行時の②colオプション部分で③color列の情報を利用していることがわかる。④expected列情報は、例題11では利用していない

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

#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, 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
```

①

②

③

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

④

作図(box plot):色づけ

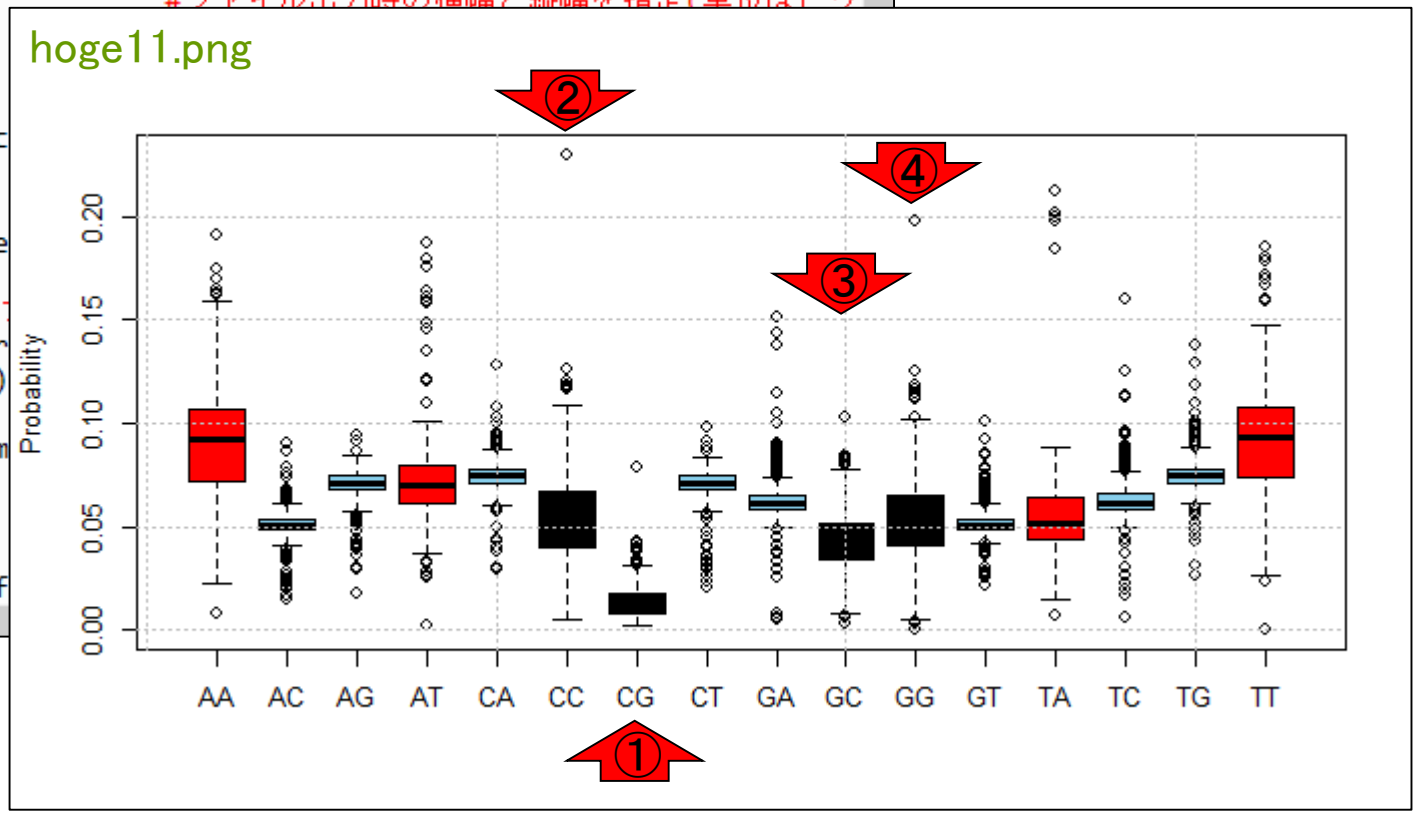
①CGの出現確率が期待値(4.2%)より少ないのは、②CC, ③GC, ④GGとの相対的な関係からも明白

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



作図(box plot):発展形

期待値との差分を評価すべく、①縦軸をlog(観測値/期待値)としてプロット。②0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。この縦軸のような表現方法は一般的です

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
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
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(ge
fasta

#本番
out <- dinucleotideFrequency(f
```

hoge12.png