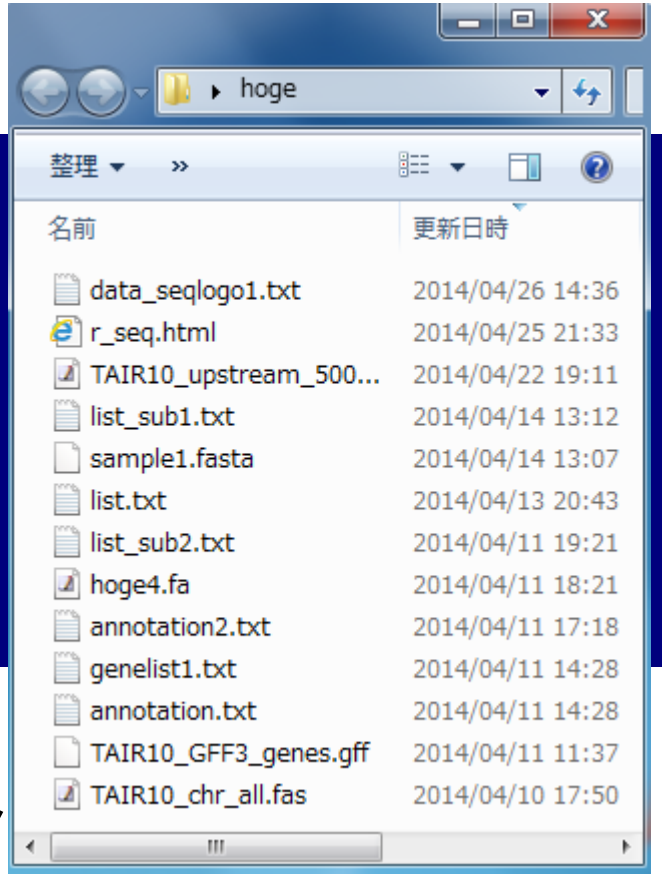
講義室後ろにあるUSBメモリ中のhogeフォルダをデスクトップにコピーしておいてください。



前回(4/23)のhogeフォルダがデスクトップに残っているかもしれないのでご注意ください。

ゲノム情報解析基礎

～ Rで塩基配列解析 ～

東京大学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ユニット
門田 幸二(かどた　こうじ)
http://www.iu.a.u-tokyo.ac.jp/~kadota/
kadota@iu.a.u-tokyo.ac.jp

fastaオブジェクトの中身

5. インストール済みのヒト("BSgenome.Hsapiens.UCSC.hg19")のゲノム配列をmulti-fastaファイルで保存したい場合:

3.0GB程度のファイルが生成されます…。ヒトゲノムは、まだ完全に22本の常染色体とX, Y染色体の計24本になっているわけではないことがわかります。

```
out_f <- "hoge5.txt"                    #出力ファイル名を指定してout_fに格納
param <- "BSgenome.Hsapiens.UCSC.hg19" #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, "
tmp <- ls(paste("package", param,
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
```

Rのほうが全体像の俯瞰が容易ですよね

R Console

```
> fasta
  A DNAStringSet instance of length 93
         width seq                                          names
  [1] 249250621 NNNNNNNNNNNNNNN...NNNNNNNNNNNNNNN chr1
  [2] 243199373 NNNNNNNNNNNNNNN...NNNNNNNNNNNNNNN chr2
  [3] 198022430 NNNNNNNNNNNNNNN...NNNNNNNNNNNNNNN chr3
  [4] 191154276 NNNNNNNNNNNNNNN...NNNNNNNNNNNNNNN chr4
  [5] 180915260 NNNNNNNNNNNNNNN...NNNNNNNNNNNNNNN chr5
  ...       ... ...
 [89]     36651 GATCAGATAGGCTTT...TTTCAGAATATGATC chrUn_gl000245
 [90]     38154 GATCTTAAGCCTTTG...CGAGGCGAGTGGATC chrUn_gl000246
 [91]     36422 GATCTAAGTTTGATT...GCTTTTCCCAAGATC chrUn_gl000247
 [92]     39786 GATCTGTCATTGTCT...TTGATACAGTTGATC chrUn_gl000248
 [93]     38502 GATCACCAAGGCTGG...AGTAGAATCTGGATC chrUn_gl000249
>
```

fastaオブジェクトの中身

R Console

```
> fasta[1:24]
  A DNAStringSet instance of length 24
         width seq                                          names
 [1] 249250621 NNNNNNNNNNNNNNN...NNNNNNNNNNNNNNN chr1
 [2] 243199373 NNNNNNNNNNNNNNN...NNNNNNNNNNNNNNN chr2
 [3] 198022430 NNNNNNNNNNNNNNN...NNNNNNNNNNNNNNN chr3
 [4] 191154276 NNNNNNNNNNNNNNN...NNNNNNNNNNNNNNN chr4
 [5] 180915260 NNNNNNNNNNNNNNN...NNNNNNNNNNNNNNN chr5
 ...       ... ...
[20]  63025520 NNNNNNNNNNNNNNN...NNNNNNNNNNNNNNN chr20
[21]  48129895 NNNNNNNNNNNNNNN...NNNNNNNNNNNNNNN chr21
[22]  51304566 NNNNNNNNNNNNNNN...NNNNNNNNNNNNNNN chr22
[23] 155270560 NNNNNNNNNNNNNNN...NNNNNNNNNNNNNNN chrX
[24]  59373566 NNNNNNNNNNNNNNN...NNNNNNNNNNNNNNN chrY
> 
```

最初の24個分を表示させたい場合

上矢印キーを押すと直前に打ったコマンドが出る。有効に利用し最小限の労力で打つ。

2連続塩基の出現確率:ヒトゲノムファイル

5. インストール済みのヒト("BSgenome.Hsapiens.UCSC.hg19")のゲノム配列をmulti-fastaファイルで保存したい場合:

3.0GB程度のファイルが生成されます...。ヒトゲノムは、まだ完全に22本の常染色体とX, Y染色体の計24本になっているわけではないことがわかります。

```
out_f <- "hoge5.txt"                    #出力ファイル名を指定してout_fに格納
param <- "BSgenome.Hsapiens.UCSC.hg19" #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
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
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
genome                                  #確認してるだけです

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

#ファイルに保存
writeXStringSet(fasta, fi
```

ヒトゲノムファイルhoge5.txtを入力ファイルとして与えるやり方でもよい

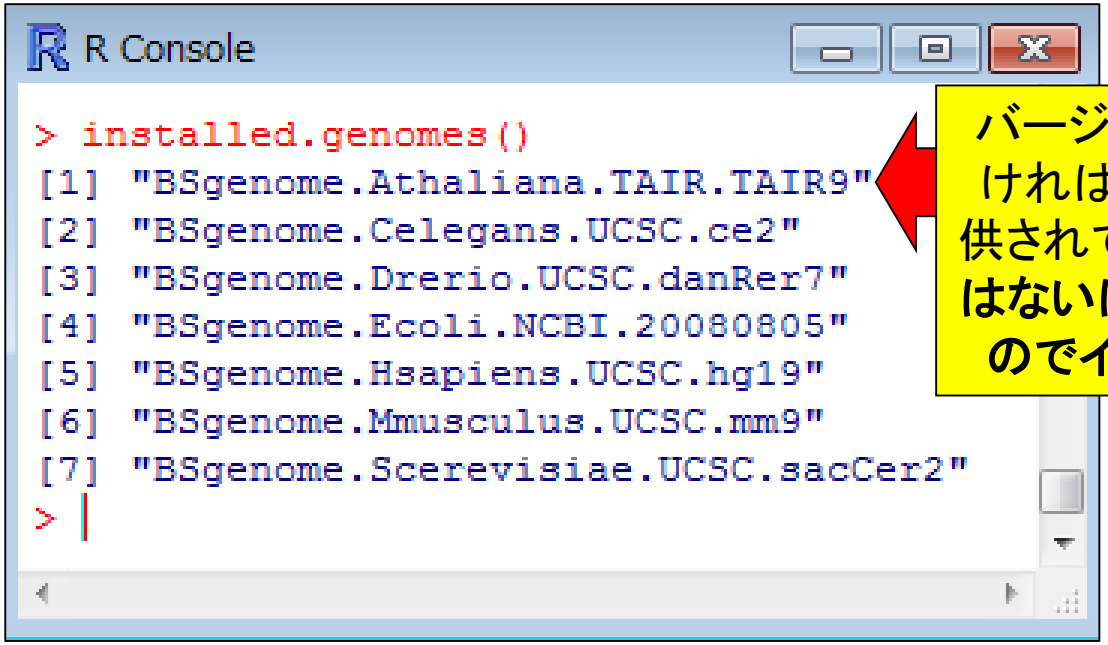
2. イントロ | 一般 | ランダムな塩基配列を作成の4.を実行して得られたmulti-fastaファイル(hoge4.fa)の場合:

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

#本番
out <- dinucleotideFrequency(fasta, as.prob=T)#2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)         #最初の列にID情報、そのあとに出現頻度情報のoutを結合したtr
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定したファイルそ
```

シロイヌナズナゲノムのCpG解析

- シロイヌナズナゲノムのGC含量は36%
  - A: 0.32, C: 0.18, G: 0.18, T: 0.32
- 期待値
  - AA, AT, TA, TTの期待値 = 0.32 × 0.32 = 0.1024
  - CC, **CG**, GC, GGの期待値 = 0.18 × 0.18 = 0.0324
  - AC, AG, CA, CT, GA, GT, TC, TGの期待値 = 0.18 × 0.32 = 0.0576



6. BSgenomeパッケージ中のシロイヌナズナゲノム配列("BSgenome.Athaliana.TAIR.TAIR9")の場合:

出現頻度ではなく、出現確率を得るやり方です。パッケージがインストールされていない場合は、イントロ | 一般 | 配列取得 | ゲノム配列 | BSgenomeを参考にしてインストールしてから再度チャレンジ。

```
out_f <- "hoge6.txt"                          #出力ファイル名を指定してout_fに格納
param <- "BSgenome.Athaliana.TAIR.TAIR9"#パッケージ名を指定

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

#前処理(paramで指定したパッ
tmp <- ls(paste("package"
genome <- eval(parse(text
fasta <- getSeq(genome)
names(fasta) <- seqnames(
fasta

#本番
out <- dinucleotideFreque

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

R Console

```
> out_f <- "hoge6.txt"                          #出力ファイル名を指$
> param <- "BSgenome.Athaliana.TAIR.TAIR9"#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$
> library(param, character.only=T)              #paramで指定したパ$
>
> #前処理(paramで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, sep=":"))#paramで指定し$
> genome <- eval(parse(text=tmp))               #文字列tmpをRオブジ$
> fasta <- getSeq(genome)                       #ゲノム塩基配列情報$
> names(fasta) <- seqnames(genome)              #description情報を$
> fasta                                         #確認してるだけです
  A DNAStringSet instance of length 7
      width seq                                 names
[1] 30427671 CCCTAAACCCTA...TAGGGTTTAGGG Chr1
[2] 19698289 NNNNNNNNNNNN...TAGGGTTTAGGG Chr2
[3] 23459830 NNNNNNNNNNNN...AACCCTAAACCC Chr3
[4] 18585056 NNNNNNNNNNNN...TTAGGGTTTAGG Chr4
[5] 26975502 TATACCATGTAC...GGTTTTTAGATC Chr5
[6]   366924 GGATCCGTTCGA...ACAAACCGGATT ChrM
[7]   154478 ATGGGCGAACGA...GTCCCGGGCATC ChrC
>
> #本番
> out <- dinucleotideFrequency(fasta, as.prob=T)#2連続塩基の$
>
> #ファイルに保存
> tmp <- cbind(names(fasta), out)               #最初の列にID情報、$
> write.table(tmp, out_f, sep="\t", append=F, quote=F, row.n$
>
```

シロイヌナズナ(A. thaliana)のRパッケージがインストールされている状態での実行結果

6. BSgenomeパッケージ中のシロイヌナズナゲノム配列("BSgenome.Athaliana.TAIR.TAIR9")の場合:

出現頻度ではなく、出現確率を得るやり方です。パッケージがインストールされていない場合は、イントロ | 一般 | 配列取得 | ゲノム配列 | BSgenomeを参考にしてインスト

```
out_f <- "hoge6.txt"                          #出力ファイ
param <- "BSgenome.Athaliana.TAIR.TAIR9"#パッケージ

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

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

R Console

```
> installed.genomes()
[1] "BSgenome.Celegans.UCSC.ce2"
[2] "BSgenome.Drerio.UCSC.danRer7"
[3] "BSgenome.Ecoli.NCBI.20080805"
[4] "BSgenome.Hsapiens.UCSC.hg19"
[5] "BSgenome.Mmusculus.UCSC.mm9"
[6] "BSgenome.Scerevisiae.UCSC.sacCer2"
>
```

シロイヌナズナ(A. thaliana)のRパッケージがインストールされていない状態での実行結果

R Console

```
> out_f <- "hoge6.txt"                          #出力ファイル名を指定してout_f$
> param <- "BSgenome.Athaliana.TAIR.TAIR9"#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$
 以下にエラー library(param, character.only = T) :
   'BSgenome.Athaliana.TAIR.TAIR9' という名前のパッケージはありません
>
> #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$
 以下にエラー as.environment(pos) :
   検索リストに "package:BSgenome.Athaliana.TAIR.TAIR9" という項目はあり$
> genome <- eval(parse(text=tmp))               #文字列tmpをRオブジェクトとして$
 以下にエラー eval(parse(text = tmp)) :
   引数 'expr' の評価中にエラーが起きました (関数 'eval' に対するメソッ$

> fasta <- getSeq(genome)                       #ゲノム塩基配列情報を抽出した結$
 以下にエラー getSeq(genome) :
   引数 'x' の評価中にエラーが起きました (関数 'getSeq' に対するメソッド$

> names(fasta) <- seqnames(genome)              #description情報を追加している
 エラー:  関数 "seqnames" を見つけることができませんでした
> fasta                                         #確認してるだけです
 エラー:  オブジェクト 'fasta' がありません
>
> #本番
> out <- dinucleotideFrequency(fasta, as.prob=T)#2連続塩基の出現確率情報$
 以下にエラー oligonucleotideFrequency(x, 2L, step = step, as.prob = as.$
   引数 'x' の評価中にエラーが起きました (関数 'oligonucleotideFrequency$

>
> #ファイルに保存
> tmp <- cbind(names(fasta), out)               #最初の列にID情報、そのあとに出$
 以下にエラー eval(expr, envir, enclos) :  オブジェクト 'fasta' がありま$
> write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F)#tmp$
 以下にエラー is.data.frame(x) :  オブジェクト 'tmp' がありません
>
```

シロイヌナズナゲノムのCpG解析

- シロイヌナズナゲノムのGC含量は36%
  - A: 0.32, C: 0.18, G: 0.18, T: 0.32
- 期待値
  - AA, AT, TA, TTの期待値 = 0.32×0.32 = 10.2%
  - CC, **CG**, GC, GGの期待値 = 0.18×0.18 = 3.2%
  - AC, AG, CA, CT, GA, GT, TC, TGの期待値 = 0.18×0.32 = 5.8%

	AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
Chr1	11.6%	5.2%	5.9%	9.3%	6.4%	3.4%	**2.3%**	5.9%	6.4%	3.0%	3.3%	5.2%	7.7%	6.4%	6.3%	11.6%
Chr2	11.6%	5.2%	5.9%	9.3%	6.3%	3.4%	**2.3%**	5.9%	6.3%	3.0%	3.4%	5.2%	7.8%	6.4%	6.3%	11.6%
Chr3	11.5%	5.2%	6.0%	9.2%	6.4%	3.4%	**2.4%**	6.0%	6.5%	3.0%	3.4%	5.2%	7.6%	6.4%	6.4%	11.4%
Chr4	11.5%	5.3%	6.0%	9.2%	6.4%	3.4%	**2.4%**	6.0%	6.4%	3.0%	3.4%	5.2%	7.6%	6.4%	6.3%	11.4%
Chr5	11.6%	5.2%	5.9%	9.3%	6.3%	3.3%	**2.3%**	5.9%	6.4%	3.0%	3.4%	5.2%	7.7%	6.4%	6.4%	11.7%
ChrM	8.8%	5.0%	7.5%	6.6%	5.8%	5.6%	**3.7%**	7.4%	7.1%	4.8%	5.5%	4.8%	6.2%	7.1%	5.5%	8.5%
ChrC	11.6%	4.4%	5.4%	10.0%	5.3%	4.5%	**3.0%**	5.7%	6.4%	2.9%	4.1%	4.5%	8.2%	6.7%	5.3%	12.1%

TAIR9のRパッケージ（BSgenome.Athaliana.TAIR.TAIR9）
を入力とした場合の結果

課題3

- シロイヌナズナのゲノム配列ファイル（TAIR10_chr_all.fas）を入力として2連続塩基の出現頻度解析を行い、得られた結果を簡単に考察せよ。
  - 期待値との比較、CpGの結果、TAIR9のRパッケージ（BSgenome.Athaliana.TAIR.TAIR9）の結果との違い、ヒトゲノムの結果との比較など数行程度でよい。

	AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
Chr1	11.6%	5.2%	5.9%	9.3%	6.4%	3.4%	**2.3%**	5.9%	6.4%	3.0%	3.3%	5.2%	7.7%	6.4%	6.3%	11.6%
Chr2	11.6%	5.2%	5.9%	9.3%	6.3%	3.4%	**2.3%**	5.9%	6.3%	3.0%	3.4%	5.2%	7.8%	6.4%	6.3%	11.6%
Chr3	11.5%	5.2%	6.0%	9.2%	6.4%	3.4%	**2.4%**	6.0%	6.5%	3.0%	3.4%	5.2%	7.6%	6.4%	6.4%	11.4%
Chr4	11.5%	5.3%	6.0%	9.2%	6.4%	3.4%	**2.4%**	6.0%	6.4%	3.0%	3.4%	5.2%	7.6%	6.4%	6.3%	11.4%
Chr5	11.6%	5.2%	5.9%	9.3%	6.3%	3.3%	**2.3%**	5.9%	6.4%	3.0%	3.4%	5.2%	7.7%	6.4%	6.4%	11.7%
ChrM	8.8%	5.0%	7.5%	6.6%	5.8%	5.6%	**3.7%**	7.4%	7.1%	4.8%	5.5%	4.8%	6.2%	7.1%	5.5%	8.5%
ChrC	11.6%	4.4%	5.4%	10.0%	5.3%	4.5%	**3.0%**	5.7%	6.4%	2.9%	4.1%	4.5%	8.2%	6.7%	5.3%	12.1%

TAIR9のRパッケージ（BSgenome.Athaliana.TAIR.TAIR9）を入力とした場合の結果

パッケージって何？

R Console

```
R version 3.0.2 (2013-09-25) -- "Frisbee Sailing"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R は、自由なソフトウェアであり、「完全に無保証」です。
一定の条件に従えば、自由にこれを再配布することができます。
配布条件の詳細に関しては、'license()' あるいは 'licence()' と入力してくださ$

R は多くの貢献者による共同プロジェクトです。
詳しくは 'contributors()' と入力してください。
また、R や R のパッケージを出版物で引用する際の形式については
'citation()' と入力してください。

'demo()' と入力すればデモをみることができます。
'help()' とすればオンラインヘルプが出ます。
'help.start()' で HTML ブラウザによるヘルプがみられます。
'q()' と入力すれば R を終了します。

> ?subseq
No documentation for 'subseq' in specified packages and libraries:
you could try '??subseq'
> ?dinucleotideFrequency
No documentation for 'dinucleotideFrequency' in specified packages and lib$
you could try '??dinucleotideFrequency'
>
```

Rを再起動した状態で**?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

パッケージって何？

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"
out_f <- "hoge2.txt"

#必要なパッケージをロード
library(Biostrings)

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

#本番
out <- dinucleotideFrequency(fasta,

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

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'

以下のオブジェクトはマスクされています (from 'package:parallel') $

    clusterApply, cluster
    clusterExport, cluster
    parLapplyLB, parRapply

以下のオブジェクトはマスクされてい

    xtabs

以下のオブジェクトはマスクされています (from 'package:base') :

    anyDuplicated, append, as.data.frame, as.vector, cbind,
    colnames, duplicated, eval, evalq, Filter, Find, get, intersect,
    is.unsorted, lapply, Map, mapply, match, mget, order, paste,
    pmax, pmax.int, pmin, pmin.int, Position, rank, rbind, Reduce,
    rep.int, rownames, sapply, setdiff, sort, table, tapply, union,
    unique, unlist

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
> ?dinucleotideFrequency
starting httpd help server ... done
>
```

*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、dinucleotideFrequencyのような*Biostrings*が提供する関数群を利用できるんです

5. BSgenomeパッケージ中のヒトゲノム配列("BSgenome.Hsapiens.UCSC.hg19")の場合:

出現頻度ではなく、出現確率を得るやり方です。

```
out_f <- "hoge5.txt"                          #出力ファ
param <- "BSgenome.Hsapiens.UCSC.hg19" #パッケー

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

#前処理(paramで指定したパッケージ中のオブジェクト名を
tmp <- ls(paste("package", param, sep=":"))#para
genome <- eval(parse(text=tmp))               #文字列tmp
fasta <- getSeq(genome)                       #ゲノム塩基
names(fasta) <- seqnames(genome)              #descript
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)               #最初の列
write.table(tmp, out_f, sep="\t", append=F, quot
```

- BioconductorのBiostringsのwebページ
- BioconductorのBSgenomeのwebページ
- Bird AP., Nucleic Acids Res., 1980
- Lander et al., Nature, 2001
- Saxonov et al., Proc Natl Acad Sci U S A., 2006

参考

• イントロ | 一般 | 2連続塩基の出現頻度情報を取得

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Home | Install | Help | Develo | Search:

Home » Bioconductor 2.12 » Software Packages » Biostrings

Biostrings

String objects representing biological sequences, and matching algorithms

Bioconductor version: Release (2.12)

Memory efficient string containers, string matching algorithms, and other utilities, for fast manipulation of large biological sequences or sets of sequences.

Author: H. Pages, P. Aboyoun, R. Gentleman, and S. DebRoy

Maintainer: H. Pages <hpages at fhcrc.org>

To install this package, start R and enter:

```
source("http://bioconductor.org/biocLite.R")
biocLite("Biostrings")
```

パッケージを個別にインストールする場合

To cite this package in a publication, start R and enter:

```
citation("Biostrings")
```

Documentation

PDF	R Script	A short presentation of the basic classes defined in Biostrin
PDF	R Script	Biostrings Quick Overview
PDF	R Script	Handling probe sequence information
PDF	R Script	Multiple Alignments
PDF	R Script	Pairwise Sequence Alignments
PDF		Reference Manual
Text		NEWS

使い方の解説記事はPDFのところをクリック

Details

biocViews	DataImport, DataRepresentation, Genetics, Infrastructure, SequenceMatching, Sequencing, Software

Biostrings Quick Overview

Hervé Pagès
Fred Hutchinson Cancer Research Center
Seattle, WA

April 3, 2013

参考

Please note that *most* but *not all* the functionalities provided by the Biostrings package are listed in document.

Function	Description
`length`	Return the number of sequences in an object.
`names`	Return the names of the sequences in an object.
`[`	Extract sequences from an object.
`head, tail`	Extract the first or last sequences from an object.
`rev`	Reverse the order of the sequences in an object.
`c`	Put in a single object the sequences from 2 or more objects.
`width, nchar`	Return the sizes (i.e. number of letters) of all the sequences in object.
`==, !=`	Element-wise comparison of the sequences in 2 objects.
`match, %in%`	Analog to `match` and `%in%` on character vectors.
`duplicated, unique`	Analog to `duplicated` and `unique` on character vectors.
`sort, order`	Analog to `sort` and `order` on character vectors, except that t ordering of DNA or Amino Acid sequences doesn't depend on t locale.
`split, relist`	Analog to `split` and `relist` on character vectors, except that t result is a *DNAStringSetList* or *AAStringSetList* object.

Table 1: Low-level manipulation of *DNAStringSet* or *AAStringSet* objects.

Function	Description
`subseq, subseq<-`	Extract or replace subsequences in a set of sequences.

`reverse`
`complement`
`reverseComplement`
	Compute the reverse, complement, or reverse-complement, of a s of DNA sequences.

`translate`	Translate a set of DNA sequences into a set of Amino Acid s quences.
`chartr`	Translate the letters in a set of sequences.
`replaceLetterAt`	Replace the letters specified by a set of positions by

Table 2: Basic transformations of sequences.

Function	Description
`alphabetFrequency` `letterFrequency`	Tabulate the letters (all the letters in the alphabet for `alphabetFrequency`, only the specified letters for `letterFrequency`) of a sequence or set of sequences.
`letterFrequencyInSlidingView`	Specialized version of `letterFrequency` that tallies the requested letter frequencies for a fixed-width view that is conceptually slid along the input sequence.
`consensusMatrix`	Computes the consensus matrix of a set of sequences.

`dinucleotideFrequency`
`trinucleotideFrequency`
`oligonucleotideFrequency`
	Fast 2-mer, 3-mer, and k-mer counting for DNA or RNA.

`nucleotideFrequencyAt`	Tallies the short sequences formed by extracting the nucleotides found at a set of fixed positions from each sequence of a set of DNA or RNA sequences.

Table 3: Counting / tabulating.

Function	Description
`matchPattern` `countPattern`	Find/count all the occurrences of a given pattern (typically short) in a reference sequence (typically long). Support mismatches and indels.
`vmatchPattern` `vcountPattern`	Find/count all the occurrences of a given pattern (typically short) in a set of reference sequences. Support mismatches and indels.

`matchPDict`
`countPDict`
`whichPDict`
	Find/count all the occurrences of a set of patterns in a reference sequence. (`whichPDict` only identifies which patterns in the set have at least one match.) Support a small number of mismatches.

`vmatchPDict`
`vcountPDict`
`vwhichPDict`
	[Note: `vmatchPDict` not implemented yet.] Find/count all the occurrences of a set of patterns in a set of reference sequences. (`whichPDict` only identifies for each reference sequence which patterns in the set have at least one match.) Support a small number of mismatches.

`pairwiseAlignment`	Solve (Needleman-Wunsch) global alignment, (Smith-Waterman) local alignment, and (ends-free) overlap alignment problems.
	sequence.

*Biostrings*中の関数を使いこなせると、他の自然言語処理系プログラミング言語（perlやruby）を改めて勉強しなくても必要な解析の大部分が可能です

参考

個別のパッケージのインストール

2. ゼブラフィッシュ("BSgenome.Drerio.UCSC.danRer7")のゲノム情報をRにインストールしたい場合:

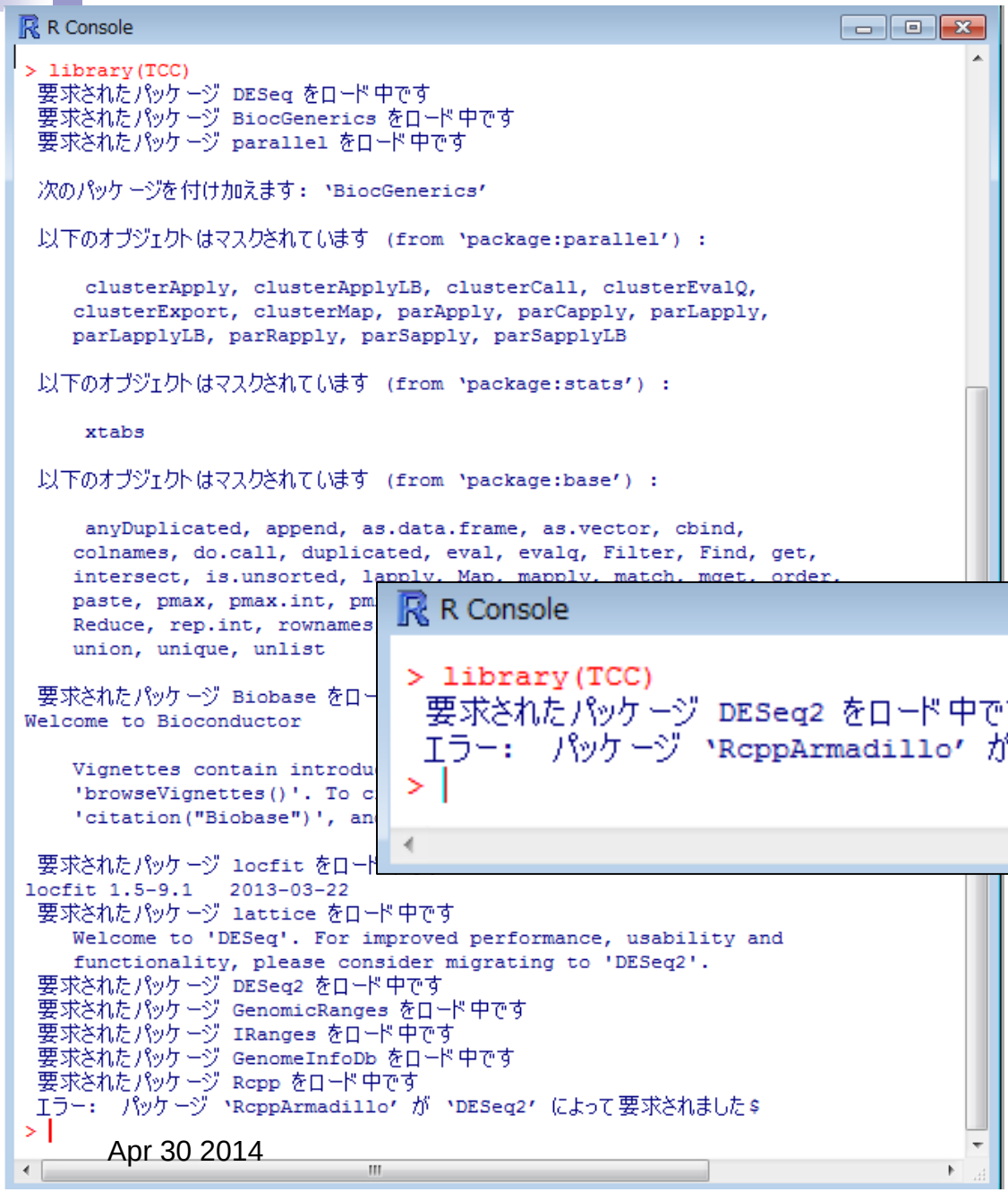
400MB程度あります...

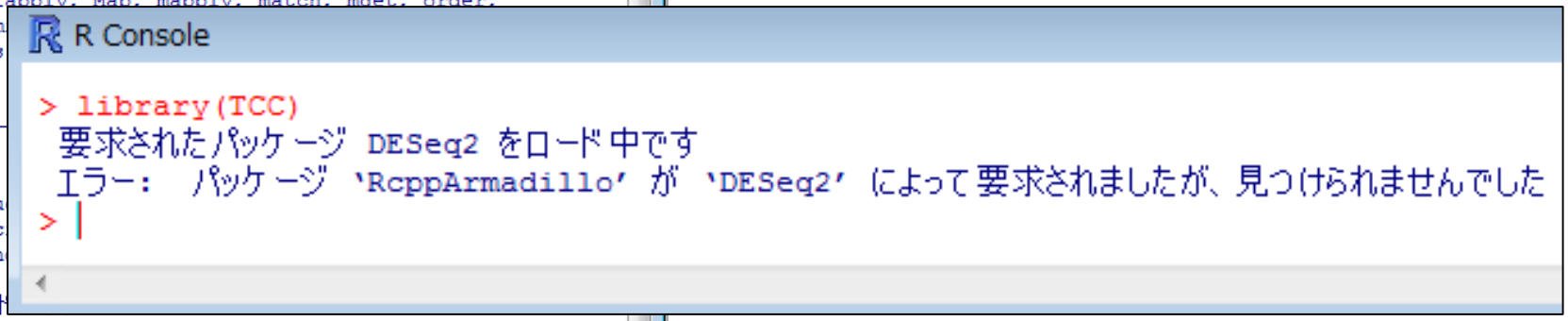
```
param <- "BSgenome.Drerio.UCSC.danRer7"#パッケージ名を指定

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

シロイヌナズナゲノム（BSgenome.Athaliana.TAIR.TAIR9）などのゲノム関連パッケージに限らず、他の様々なパッケージ（TCC, Biostring, ShortRead, DESeq2, …）のインストール手順も同じです。

エラー遭遇例とその対処法1

ときどき必要なパッケージのインストールに失敗していて、任意のパッケージ*XXX*の読み込みを行うlibrary(*XXX*)実行後にエラーが出てしまうことがあります。例はTCCパッケージが要求している「RcppArmadilloパッケージがないからダメ!」と文句を言われている例です。





エラー遭遇例とその対処法1

基本的な対処法は、文句を言われたパッケージのみインストールすることです。RcppArmadilloパッケージを個別にインストールするためのコマンドの基本形は以下のとおりです：

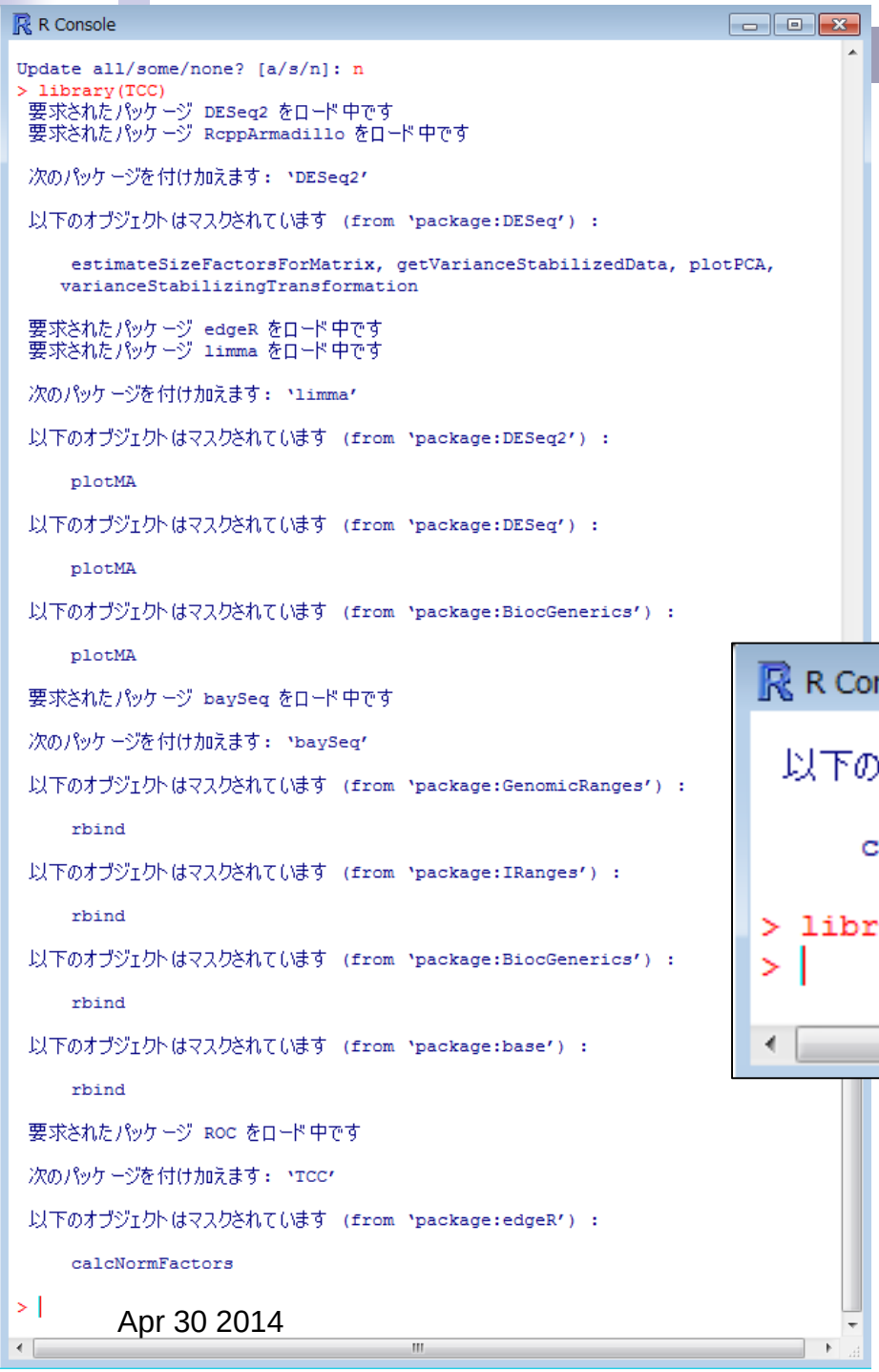
source("http://www.bioconductor.org/biocLite.R")
biocLite("RcppArmadillo")

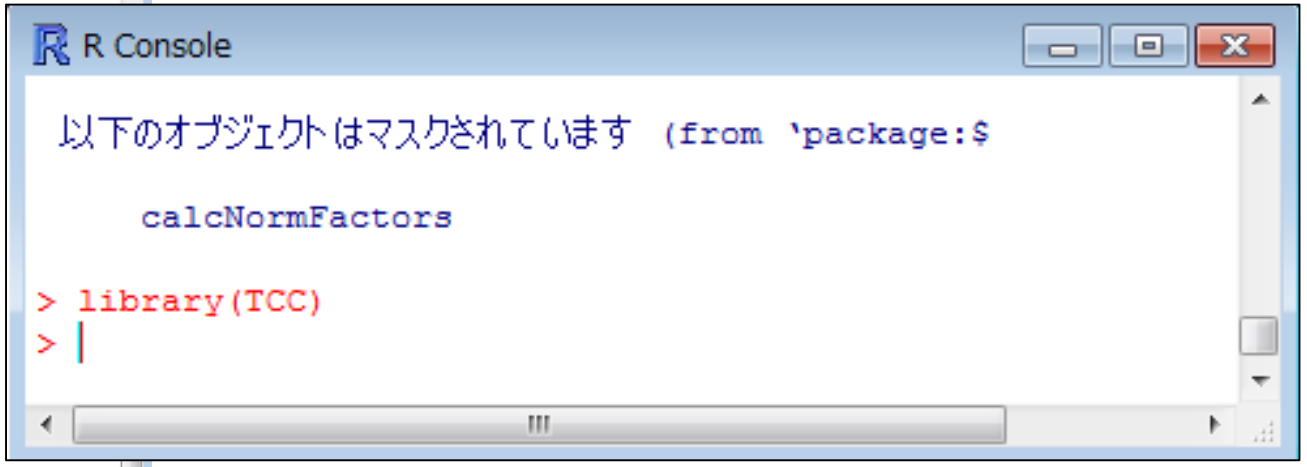
R Console

```
 要求されたパッケージ DESeq2 をロード中です
 エラー:  パッケージ 'RcppArmadillo' が 'DESeq2' によって要求されましたが、見つけられ$
> source("http://www.bioconductor.org/biocLite.R")
Bioconductor version 2.14 (BiocInstaller 1.14.1), ?biocLite for help
> biocLite("RcppArmadillo")
BioC_mirror: http://bioconductor.org
Using Bioconductor version 2.14 (BiocInstaller 1.14.1), R version 3.1.0.
Installing package(s) 'RcppArmadillo'
 URL 'http://cran.fhcrc.org/bin/windows/contrib/3.1/RcppArmadillo_0.4.200.0.zip' を試して$
Content type 'application/zip' length 1551263 bytes (1.5 Mb)
 開かれた URL
downloaded 1.5 Mb

package 'RcppArmadillo' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\kadota\AppData\Local\Temp\RtmpgNGQWx\downloaded_packages
Old packages: 'AnthropMMD', 'bayesQR', 'Bclim', 'care', 'clogitL1', 'freestats',
  'geomorph', 'gtools', 'investr', 'jsonlite', 'markovchain', 'meta', 'multicon',
  'mvtnorm', 'NLP', 'openNLP', 'PBD', 'pdfetch', 'poisson.glm.mix', 'QCA3', 'Rbitcoin',
  'regRSM', 'Reol', 'rgbif', 'Rmpi', 'rsm', 'RTextureMetrics', 'RWebLogo', 'sda',
  'SEERaBomb', 'segmented', 'seqDesign', 'seqminer', 'sjPlot', 'spcr', 'st', 'yuima'
Update all/some/none? [a/s/n]: n
>
```

Update all/some/none? [a/s/n]:
と聞かれることもありますが基本はnでいいです。

RcppArmadilloパッケージのインストール後に、もう一度library(TCC)とやって、エラーが出なくなることを確認しています。

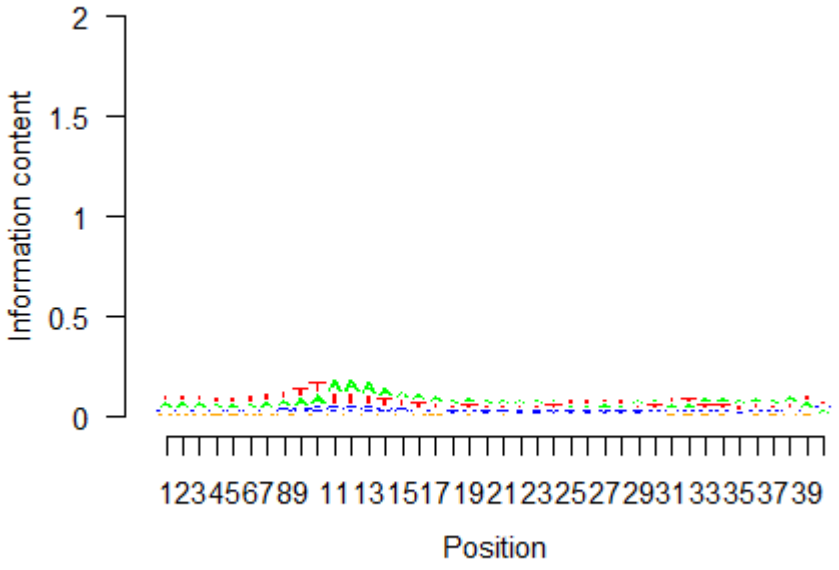


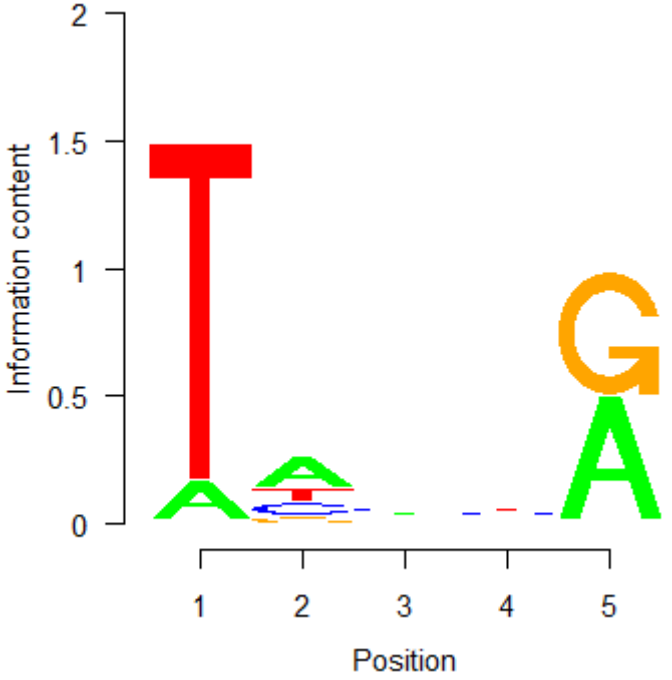


Apr 30 2014

ゲノム情報解析：sequence logos

- シロイヌナズナ（*Arabidopsis thaliana*）の上流配列解析
  - ゲノム配列ファイル（TAIR10_chr_all.fas）とアノテーションファイル（TAIR10_GFF3_genes.gff）からプロモーター配列取得（**が失敗**）
    - TAIRから取得した転写開始点上流500bpのmulti-FASTAファイル（TAIR10_upstream_500_20101028.fa）を入力としてsequence logosを実行



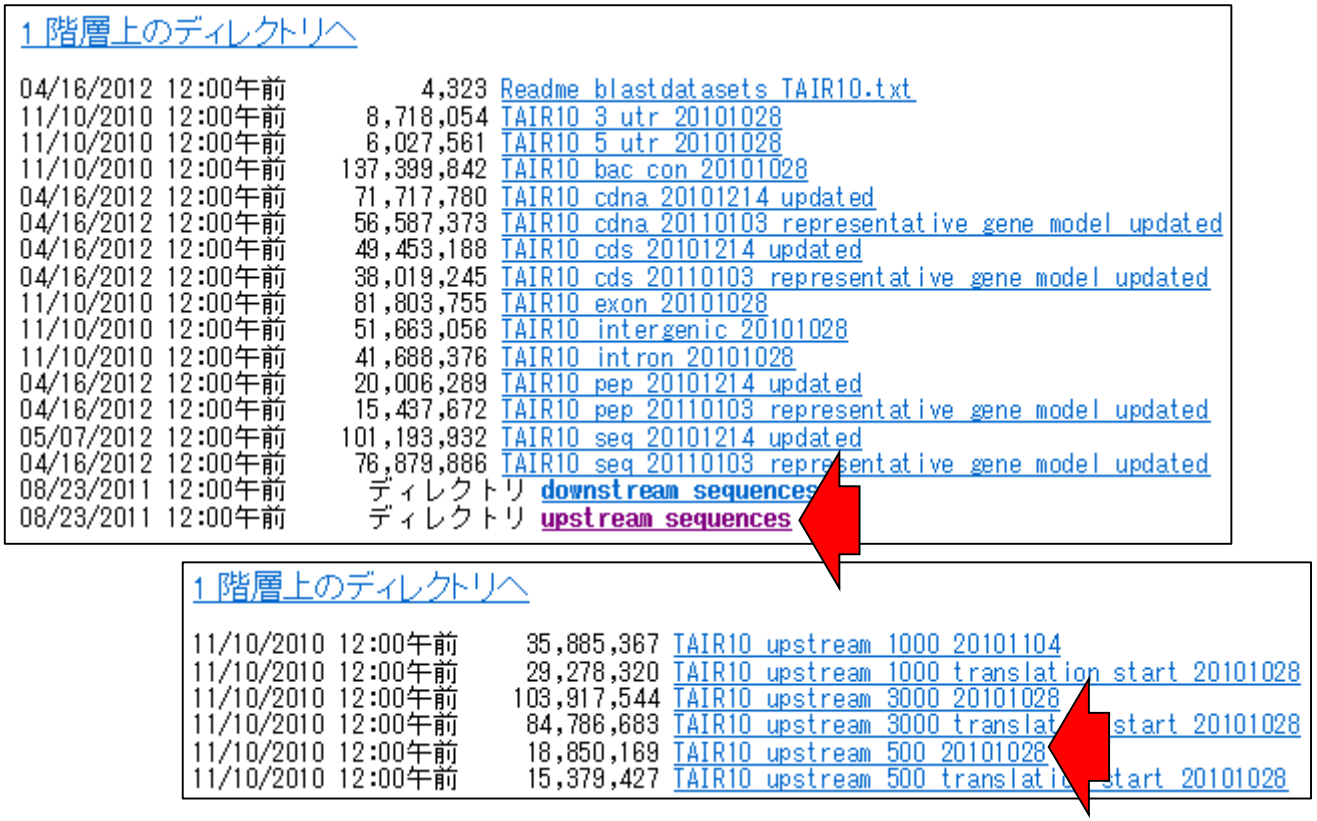


- イントロ | 一般 | 配列取得 | ゲノム配列 | 公共DBから (last modified 2014/04/10) NEW
- イントロ | 一般 | 配列取得 | ゲノム配列 | BSgenome (last modified 2014/04/22) NEW
- イントロ | 一般 | 配列取得 | プロモーター配列 | 公共DBから (last modified 2014/04/02) NEW
- イントロ | 一般 | 配列取得 | プロモーター配列 | BSgenome (last modified 2014/04/25) NEW
- イントロ | 一般 | 配列取得 | プロモーター配列 | GenomicFeatures (Lawrence, 2013) (last modified 2014/0

イントロ | 一般 | 配列取得 | プロモーター配列 | 公共DBから NEW

- **UCSC (Karolchik et al., Nucleic Acids Res., 2014)を利用する場合:**
  UCSCの Sequence and Annotation Downloadsページからリストアップされている目的の生物種を選択。**Full data set**ページの下のほうで上流1000, 2000, 5000bpの配列を取得可能。
  - ラット: Rat
  - ヒト: Human
  - マウス: Mouse
  - ゼブラフィッシュ: Zebrafish
  - ...
- **DBTSS (Yamashita et al., Nucleic Acids Res., 2012)を利用する場合:**
  左側の下のほうのDownloadをクリックし、dbtss_recentlyを選択。2012/07/06現在、以下の5つの生物種の転写開始点(Transcriptional Start Sites; TSS)の上流1000bp-下流200bpの範囲の配列を取得可能。
  - ヒト: hspromoter.tab.gz
  - マウス: mmpromoter.tab.gz
  - シアニディオシゾン (C. merolae): cmpromoter.tab.gz
  - ゼブラフィッシュ(D. rerio): drpromoter.tab.gz
  - 熱帯熱マラリア原虫(P. falciparum): pfpromoter.tab.gz
- **EPD (Dreos et al., Nucleic Acids Res., 2013)を利用する場合:**
  ヒトやマウスは「Access EPD」-「Download EPD db」ですぐに到達可能。
  それ以外のイネとかは「Access EPD」-「Download EPD db」で「Download EPD (refine selection)」の"refine selection"のところをいじれば...13046 rice sequencesがダウンロードできるはずだが...
- **個別の生物種ごとに作成されたDBを利用する場合:**
  - イネ: RAP-DB(Rice Annotation Project, Nucleic Acids Res., 2008)のダウンロードタブから上流・下流それぞれ1000, 2000, 3000bpの配列を取得可能(計6種類)。
  - シロイヌナズナ: The Arabidopsis Information Resource (TAIR) (Lamesch et al., Nucleic Acids Res., 2012)の「ダウンロード」-「Sequences」タブからblast_datasetsのTAIR10_blastsetsまでいくと、上流・下流それぞれ500, 1000, 3000bpの配列を取得可能(計6種類)。

上流配列取得



1 階層上のディレクトリへ

04/16/2012 12:00午前	4,323	Readme_blastdatasets_TAIR10.txt
11/10/2010 12:00午前	8,718,054	TAIR10_3_utr_20101028
11/10/2010 12:00午前	6,027,561	TAIR10_5_utr_20101028
11/10/2010 12:00午前	137,399,842	TAIR10_bac_con_20101028
04/16/2012 12:00午前	71,717,780	TAIR10_cdna_20101214_updated
04/16/2012 12:00午前	56,587,373	TAIR10_cdna_20110103_representative_gene_model_updated
04/16/2012 12:00午前	49,453,188	TAIR10_cds_20101214_updated
04/16/2012 12:00午前	38,019,245	TAIR10_cds_20110103_representative_gene_model_updated
11/10/2010 12:00午前	81,803,755	TAIR10_exon_20101028
11/10/2010 12:00午前	51,663,056	TAIR10_intergenic_20101028
11/10/2010 12:00午前	41,688,376	TAIR10_intron_20101028
04/16/2012 12:00午前	20,006,289	TAIR10_pep_20101214_updated
04/16/2012 12:00午前	15,437,672	TAIR10_pep_20110103_representative_gene_model_updated
05/07/2012 12:00午前	101,193,932	TAIR10_seq_20101214_updated
04/16/2012 12:00午前	76,879,886	TAIR10_seq_20110103_representative_gene_model_updated
08/23/2011 12:00午前	ディレクトリ	**downstream_sequences**
08/23/2011 12:00午前	ディレクトリ	**upstream_sequences**

1 階層上のディレクトリへ

11/10/2010 12:00午前	35,885,367	TAIR10_upstream_1000_20101104
11/10/2010 12:00午前	29,278,320	TAIR10_upstream_1000_translation_start_20101028
11/10/2010 12:00午前	103,917,544	TAIR10_upstream_3000_20101028
11/10/2010 12:00午前	84,786,683	TAIR10_upstream_3000_translat…start_20101028
11/10/2010 12:00午前	18,850,169	TAIR10_upstream_500_20101028
11/10/2010 12:00午前	15,379,427	TAIR10_upstream_500_translati…start_20101028

上流配列取得

Sequence logos

- 特徴的なパターンをもつポジションを誇示させる方法

Nucleic Acids Res. 1990 Oct 25;18(20):6097-100.

Sequence logos: a new way to display consensus sequences.

Schneider TD, Stephens RM.

National Cancer Institute, Frederick Cancer Research and Development Center, MD 21701.

Rでsequence logosができます

Abstract

A graphical method is presented for displaying the patterns in a set of aligned sequences. The characters representing the sequence are stacked on top of each other for each position in the aligned sequences. The height of each letter is made proportional to its frequency, and the letters are sorted so the most common one is on top. The height of the entire stack is then adjuste
From these 'sequence logos', one can determ
of bases and the information content (measu
both significant residues and subtle sequenc

解析 | 一般 | Sequence logos(Schneider_1990) NEW

seqLogoパッケージを用いてsequence logos (Schneider and Stephens, 1990)を実行するやり方を示します。ここでは、multi-FASTAファイルを読み込んでポジションごとの出現頻度を調べる目的で利用します。上流-35 bpにTATA boxがあることを示す目的などに利用されます。

「ファイル」-「デ

1. 入力ファイルが"m

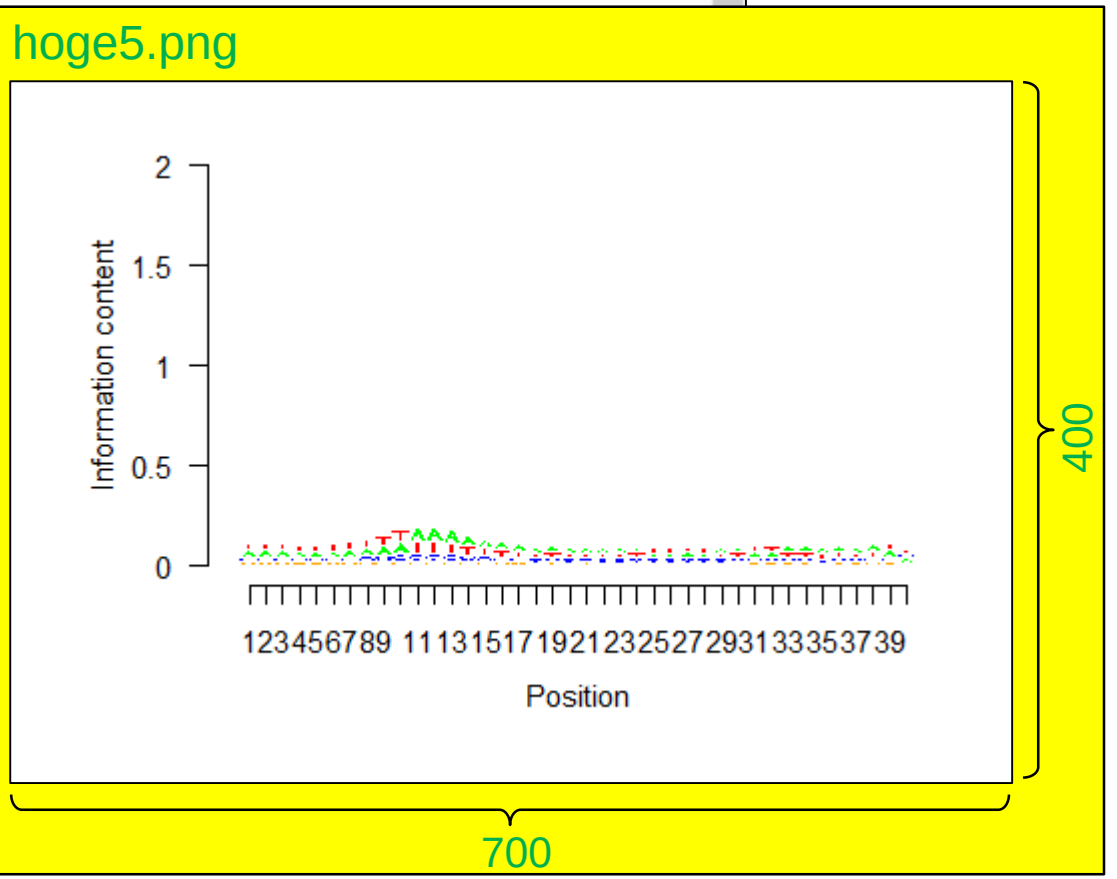
```
in_f <- "test

#必要なパッケ
library(Biost
library(seqLo

#入力ファイル
fasta <- read

#本番(sequenc
hoge <- conse
out <- makePW
```

5. 入力ファイルがmulti-FASTA形式のファイル(TAIR10_upstream_500_20101028)の場合:

Arabidopsisの上流500bpの配列セットです。500bpと長いため、461-500bpの範囲のみについて解析し、得られた図をファイルに保存するやり方です。以下はダウンロードしたファイルの拡張子として、"fa"を付加しているという前提です。

```
in_f <- "TAIR10_upstream_500_20101028.fa"#入力ファイル名を指定してin_fに格納
out_f <- "hoge5.png"                     #出力ファイル名を指定してout_fに格納
param1 <- c(461, 500)                    #抽出したい範囲の始点と終点を指定
param2 <-  500                           #配列長を指定
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(seqLogo)

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

#前処理(配列長が500bpのもののみフィルタリ
obj <- as.logical(width(fasta) == para
fasta <- fasta[obj]
fasta
fasta <- subseq(fasta, param1[1], para
fasta

#本番(sequence logoを実行)
```

hoge5.png



2013年7月以降のリニューアルで、コードのコピーがやりずらくなっています。**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。

5. 入力ファイルがmulti-FASTA形式のファイル(TAIR10_upstream_500_20101028)の場合:

Arabidopsisの上流500bpの配列セットです。500bpと長いため、461-500bpの範囲のみについて解析し、得られた図をファイルに保存するやり方です。以下はダウンロードしたファイルの拡張子として、"fa"を付加しているという前提です。

```
in_f <- "TAIR10_upstream_500_20101028.fa"#入力ファイル名を指定してin_fに格納
out_f <- "hoge5.png"                     #出力ファイル名を指定してout_fに格納
param1 <- c(461, 500)                    #開始と終点を指定
param2 <-  500
param_fig <- c(700, 400)                 #横幅と縦幅を指定(単位はピクセル)

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

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

#前処理(配列長が500bpのもの
obj <- as.logical(width(f
fasta <- fasta[obj]
fasta
fasta <- subseq(fasta, pa
fasta

#本番(sequence logoを実行)
```

切り取り(T)
コピー(C)
貼り付け
すべて選択(A)
印刷(I)...
印刷プレビュー(N)...
Bing でマップ
Bing で翻訳
Google で検索
電子メール (Win
すべてのアクセ
Send to OneNo

R Console

```
 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(seqLogo)                          #パッケージの読み込み
 要求されたパッケージ grid をロード中です
>
> #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
> fasta                                     #確認してるだけです
  A DNAStringSet instance of length 33602
        width seq                                               names
    [1]   500 TACAGATGATTATGCCTTTT...TCTTAATGTGGATAGTGCT AT1G08520 | chr1:...
    [2]   500 CCTCACAGGAAAATATTATT...TCCTGGAGAAGAGAAGACT AT1G08530 | chr1:...
    [3]   500 GAAATATCACTACAACCTGT...TTTGATTGAGATAGAAGCT AT1G08540 | chr1:...
    [4]   500 ACAAATTAGAACATAAGTTT...GAGTTGGGTGGGTATTACC AT1G36060 | chr1:...
    [5]   500 CCAGAAGCAGTAACGTCAGA...AAAAAAAATCCAAAGATTA AT1G08550 | chr1:...
    ...   ... ...
[33598]   500 ATCTTCAGGGTATGATACCT...ATAATTGAACAAAAGCGAG ATMG00420 | chrM:...
[33599]   500 GCCATAGTTTCAGCATTACG...CGCGCGATCGTATACTGAG ATMG01330 | chrM:...
[33600]   500 CGATTCACAGAGCGCCATTT...GAAAGCTTTCTTTATCTTT ATMG00070 | chrM:...
[33601]   500 TAAAAGGAGAGGTGCTTTAG...AGCCCTCTGTCAATCTCTG ATMG00130 | chrM:...
[33602]   500 AGTGAACTTTCACAATTTCT...TATCAATTGTGATAAGAAA ATMG00930 | chrM:...
>
```

• 解析 | 一般 | Sequence logos(Schneider 1990)

33,602配列からなり、見えている範囲では500bpであることがわかる

5. 入力ファイルがmulti-FASTA形式のファイル(TAIR10_upstream_500_20101028)の場合:

Arabidopsisの上流500bpの配列セットです。500bpと長いため、461-500bpの範囲のみについて解析し、得られた図をファイルに保存するやり方です。以下はダウンロードしたファイルの拡張子として、"fa"を付加しているという前提です。

```
library(seqLogo)                          #パッケージの読み込み

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

#前処理(配列長が500bpのもののみフィルタリング後、解析したいサブセットを抽出)
obj <- as.logical(width(fasta) == param2)#条件を満たすかどうかを判定した結果をobjに格
fasta <- fasta[obj]                       #objがTRUEとなるもののみ抽出した結果をfastaに
fasta                                     #確認してるだけです
fasta <- subseq(fasta, param1[1], p                点の範囲の配列を
fasta

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

#ファイルに保存
png(out_f, pointsize=13, width=para
seqLogo(out)
dev.off()
```

切り取り(T)
コピー(C)
貼り付け

R Console

```
[33599]    500 GCCATAGTTTCAGCATTACG...CGCGCGATCGTATACTGAG ATMG01330 | chrM:...
[33600]    500 CGATTCACAGAGCGCCATTT...GAAAGCTTTCTTTATCTTT ATMG00070 | chrM:...
[33601]    500 TAAAAGGAGAGGTGCTTTAG...AGCCCTCTGTCAATCTCTG ATMG00130 | chrM:...
[33602]    500 AGTGAACTTTCACAATTTCT...TATCAATTGTGATAAGAAA ATMG00930 | chrM:...
> obj <- as.logical(width(fasta) == param2)#条件を満たすかどうかを判定した結$
> fasta <- fasta[obj]                      #objがTRUEとなるもののみ抽出した結$
> fasta                                    #確認してるだけです
  A DNAStringSet instance of length 33600
        width seq                                               names
    [1]   500 TACAGATGATTATGCCTTTT...TCTTAATGTGGATAGTGCT AT1G08520 | chr1:...
    [2]   500 CCTCACAGGAAAATATTATT...TCCTGGAGAAGAGAAGACT AT1G08530 | chr1:...
    [3]   500 GAAATATCACTACAACCTGT...TTTGATTGAGATAGAAGCT AT1G08540 | chr1:...
    [4]   500 ACAAATTAGAACATAAGTTT...GAGTTGGGTGGGTATTACC AT1G36060 | chr1:...
    [5]   500 CCAGAAGCAGTAACGTCAGA...AAAAAAAATCCAAAGATTA AT1G08550 | chr1:...
    ...   ... ...
[33596]   500 ATCTTCAGGGTATGATACCT...ATAATTGAACAAAAGCGAG ATMG00420 | chrM:...
[33597]   500 GCCATAGTTTCAGCATTACG...CGCGCGATCGTATACTGAG ATMG01330 | chrM:...
[33598]   500 CGATTCACAGAGCGCCATTT...GAAAGCTTTCTTTATCTTT ATMG00070 | chrM:...
[33599]   500 TAAAAGGAGAGGTGCTTTAG...AGCCCTCTGTCAATCTCTG ATMG00130 | chrM:...
[33600]   500 AGTGAACTTTCACAATTTCT...TATCAATTGTGATAAGAAA ATMG00930 | chrM:...
>
```

• 解析 | 一般 | Sequence logos(Schneider 1990)

上流500 bpの配列セットであるはずなのに、2つの配列長が500bpではなかったことがわかる

5. 入力ファイルがmulti-FASTA形式のファイル(TAIR10_upstream_500_20101028)の場合:

Arabidopsisの上流500bpの配列セットです。500bpと長いため、461-500bpの範囲のみについて解析し、得られた図をファイルに保存するやり方です。以下はダウンロードしたファイルの拡張子として、"fa"を付加しているという前提です。

```
library(seqLogo)                          #パッケージの読み込み

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

#前処理(配列長が500bpのもののみフィルタリング後
obj <- as.logical(width(fasta) == param2)#条
fasta <- fasta[obj]                       #obj
fasta                                     #確認
fasta <- subseq(fasta, param1[1], param1[2])
fasta                                     #確認

#本番(sequ
hoge <- c
out <- ma

#ファイルI
png(out_f
seqLogo(o
dev.off()
```

切り取り(T)
コピー(C)
貼り付け
すべて選択(A)
印刷(I)...
印刷プレビュー(N)...
Bing でマップ
Bing で翻訳

- 解析 | 一般 | Sequence logos(Schneider 1990)

R Console

```
> fasta                                   #確認してるだけです
  A DNAStringSet instance of length 33600
        width seq                                               names
    [1]   500 TACAGATGATTATGCCTTTT..TCTTAATGTGGATAGTGCT AT1G08520 | chr1:...
    [2]   500 CCTCACAGGAAAATATTATT..TCCTGGAGAAGAGAAGACT AT1G08530 | chr1:...
    [3]   500 GAAATATCACTACAACCTGT..TTTGATTGAGATAGAAGCT AT1G08540 | chr1:...
    [4]   500 ACAAATTAGAACATAAGTTT..GAGTTGGGTGGGTATTACC AT1G36060 | chr1:...
    [5]   500 CCAGAAGCAGTAACGTCAGA..AAAAAAAATCCAAAGATTA AT1G08550 | chr1:...
    ...   ... ...
[33596]   500 ATCTTCAGGGTATGATACCT..ATAATTGAACAAAAGCGAG ATMG00420 | chrM:...
[33597]   500 GCCATAGTTTCAGCATTACG..CGCGCGATCGTATACTGAG ATMG01330 | chrM:...
[33598]   500 CGATTCACAGAGCGCCATTT..GAAAGCTTTCTTTATCTTT ATMG00070 | chrM:...
[33599]   500 TAAAAGGAGAGGTGCTTTAG..AGCCCTCTGTCAATCTCTG ATMG00130 | chrM:...
[33600]   500 AGTGAACTTTCACAATTTCT..TATCAATTGTGATAAGAAA ATMG00930 | chrM:...
> fasta <- subseq(fasta, param1[1], param1[2])#param1で指定した始点と終点の$
> fasta                                   #確認してるだけです
  A DNAStringSet instance of length 33600
        width seq                                               names
    [1]    40 TTTTGAGACATTTTATTGAAATCTTAATGTGGATAGTGCT AT1G08520 | chr1:...
    [2]    40 TAAAAGAAAACTGGACTTGGCTCCTGGAGAAGAGAAGACT AT1G08530 | chr1:...
    [3]    40 AAAATTAGAAGTAATGAATCCTTTGATTGAGATAGAAGCT AT1G08540 | chr1:...
    [4]    40 AGCCCTCTATATAAACGGTGAGAGTTGGGTGGGTATTACC AT1G36060 | chr1:...
    [5]    40 TCTCTTATAAGAATTTGGGAAAAAAAAAATCCAAAGATTA AT1G08550 | chr1:...
    ...   ... ...
[33596]    40 TTCTTTCTTTTATTTATATTTATAATTGAACAAAAGCGAG ATMG00420 | chrM:...
[33597]    40 TAATAGGACTCCCAGTTACTGCGCGCGATCGTATACTGAG ATMG01330 | chrM:...
[33598]    40 GAGCAAGAAGCGGAACTACAAGAAAGCTTTCTTTATCTTT ATMG00070 | chrM:...
[33599]    40 GGAGAGAGTTAATCAACGGGTAGCCCTCTGTCAATCTCTG ATMG00130 | chrM:...
[33600]    40 GAATCCTCGTCTTTACAGTCTTATCAATTGTGATAAGAAA ATMG00930 | chrM:...
>
```

全部で500bpからなる配列の最後のほうがとれていることがわかる

5. 入力ファイルがmulti-FASTA形式のファイル(TAIR10_upstream_500_20101028)の場合:

Arabidopsisの上流500bpの配列セットです。500bpと長いため、461-500bpの範囲のみについて解析し、得られた図をファイルに保存するやり方です。以下はダウンロードしたファイルの拡張子として、"fa"を付加しているという前提です。

```
library(seqLogo)                        #パッケージの読み込み

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

#前処理(配列長が500bpのもののみフィルタリング後、解析したいサブセットを抽出)
obj <- as.logical(width(fasta) == param2)#条件を満たすかどうかを判定した結果をobjに格
fasta <- fasta[obj]                     #objがTRUEとなるもののみ抽出した結果をfastaに格
fasta                                   #確認してるだけです
fasta <- subseq(fasta, param1[1], param1[2])#param1で指定した始点と終点の範囲の配列を
fasta                                   #確認してるだけです

#本番(sequence logoを実行)
hoge <- consensusMatrix(fasta, as.prob=T, baseOnly=T)#各ポジションの塩基組成（probabi
out <- makePWM(hoge[1:4,])              )のprobabilityも

#ファイルに保存
png(out_f, pointsize=13, width=pa
seqLogo(out)
dev.off()
```

切り取り(T)
コピー(C)
貼り付け
すべて
印刷
印刷

R Console

```
> hoge <- consensusMatrix(fasta, as.prob=T, baseOnly=T)#各ポジションの塩基組$
> dim(hoge)
[1]  5 40
> hoge
              [,1]         [,2]         [,3]         [,4]         [,5]
A     0.3255059524 0.3236011905 0.3252083333 0.3173809524 0.3111011905
C     0.1874107143 0.1897321429 0.1933928571 0.1929761905 0.1972321429
G     0.1522916667 0.1509821429 0.1485714286 0.1547023810 0.1562500000
T     0.3346428571 0.3355357143 0.3326785714 0.3347916667 0.3352678571
other 0.0001488095 0.0001488095 0.0001488095 0.0001488095 0.0001488095
              [,6]         [,7]         [,8]         [,9]        [,10]
A     0.3136309524 0.3165773810 0.3208333333 0.3410119048 0.3488392857
C     0.1927976190 0.1860416667 0.1801190476 0.1739285714 0.1575595238
G     0.1500000000 0.1438095238 0.1399107143 0.1302678571 0.1218154762
T     0.3434226190 0.3534226190 0.3589880952 0.3546428571 0.3716369048
other 0.0001488095 0.0001488095 0.0001488095 0.0001488095 0.0001488095
             [,11]        [,12]        [,13]        [,14]        [,15]
```

consensusMatrix関数はポジションごとの塩基組成を計算しているだけ

- 解析 | 一般 | Sequence logos(Schneider 1990)

5. 入力ファイルがmulti-FASTA形式のファイル(TAIR10_upstream_500_20101028)の場合:

Arabidopsisの上流500bpの配列セットです。500bpと長いため、461-500bpの範囲のみについて解析し、得られた図をファイルに保存するやり方です。以下はダウンロードしたファイルの拡張子として、"fa"を付加しているという前提です。

```
library(seqLogo)                        #パッケージの読み込み

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

#前処理(配列長が500bpのもののみフィルタリング後、解析したいサブセットを抽出)
obj <- as.logical(width(fasta) == param2)#条件を満たすかどうかを判定した結果をobjに格納
fasta <- fasta[obj]                     #objがTRUEとなるもののみ抽出した結果をfastaに
fasta                                   #確認
fasta <- subseq(fasta, param1[1], param1[2])
fasta                                   #確認

#本番(sequence logoを実行)
hoge <- consensusMatrix(fasta, as.prob=T, ba
out <- makePWM(hoge[1:4,])              #hoge

#ファイルに保存
png(out_f, pointsize=13, width=p
seqLogo(out)
dev.off()
```

切り取り(T)
コピー(C)
貼り付け
すべて選択(
印刷(I)...

R Console

```
> out <- makePWM(hoge[1:4,])            #hogeはACGT以外の塩基(例えばN)のpro$
> dim(out)
NULL
> out
        1      2      3      4      5      6      7      8      9     10
A 0.3255 0.3236 0.3252 0.3174 0.3111 0.3136 0.3166 0.3208 0.3410 0.3488
C 0.1874 0.1897 0.1934 0.1930 0.1972 0.1928 0.1860 0.1801 0.1739 0.1576
G 0.1523 0.1510 0.1486 0.1547 0.1562 0.1500 0.1438 0.1399 0.1303 0.1218
T 0.3346 0.3355 0.3327 0.3348 0.3353 0.3434 0.3534 0.3590 0.3546 0.3716
       11     12     13     14     15     16     17     18     19     20
A 0.3697 0.3777 0.3875 0.3781 0.3666 0.3508 0.3403 0.3284 0.3262 0.3255
C 0.1499 0.1494 0.1562 0.1624 0.1720 0.1894 0.1895 0.2044 0.2026 0.2045
G 0.1215 0.1187 0.1234 0.1388 0.1449 0.1480 0.1557 0.1588 0.1571 0.1604
T 0.3587 0.3540 0.3328 0.3205 0.3164 0.3117 0.3144 0.3083 0.3140 0.3095
       21     22     23     24     25     26     27     28     29     30
A 0.3257 0.3185 0.3178 0.3192 0.3155 0.3103 0.3149 0.3119 0.3191 0.3227
C 0.2026 0.2062 0.2057 0.2046 0.2026 0.2035 0.2091 0.2071 0.2114 0.2075
G 0.1586 0.1602 0.1610 0.1571 0.1605 0.1622 0.1541 0.1575 0.1542 0.1546
T 0.3130 0.3149 0.3154 0.3190 0.3214 0.3239 0.3217 0.3234 0.3152 0.3151
       31     32     33     34     35     36     37     38     39     40
A 0.3202 0.3182 0.3269 0.3270 0.3261 0.3266 0.3575 0.3687 0.3160 0.2708
C 0.1999 0.1880 0.1978 0.1973 0.1946 0.2054 0.2038 0.1786 0.2326 0.2785
G 0.1550 0.1673 0.1588 0.1604 0.1717 0.1534 0.1666 0.1692 0.1267 0.1448
T 0.3248 0.3263 0.3163 0.3152 0.3074 0.3144 0.2720 0.2835 0.3245 0.3059
>
```

makePWM関数実行結果のoutオブジェクトは行列形式ではなさそう

- 解析 | 一般 | Sequence logos(Schneider 1990)

• 解析 | 一般 | Sequence logos(Schneider 1990)

5. 入力ファイルがmulti-FASTA形式のファイル(TAIR10_upstream_500_20101028)の場合:

Arabidopsisの上流500bpの配列セットです。500bpと長いため、461-500bpの範囲のみについて解析し、得られた図をファイルに保存するやり方です。以下はダウンロードしたファイルの拡張子として、"fa"を付加しているという前提です。

```
library(seqLogo)                          #パッケージの読み込み

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

#前処理(配列長が500bpのもののみフィルタリング後、解析したいサブセットを抽出)
obj <- as.logical(width(fasta) == param2)#条件を満たすかどうかを判定した結果をobjに格納
fasta <- fasta[obj]                       #objがTRUEとなるもののみ抽出した結果をfastaに
fasta                                     #
fasta <- subseq(fasta, param1[1], param1[
fasta                                     #

#本番(sequence logoを実行)
hoge <- consensusMatrix(fasta, as.prob=T,
out <- makePWM(hoge[1:4,])                #

#ファイルに保存
png(out_f, pointsize=13, width=p
seqLogo(out)
dev.off()
```

R Console

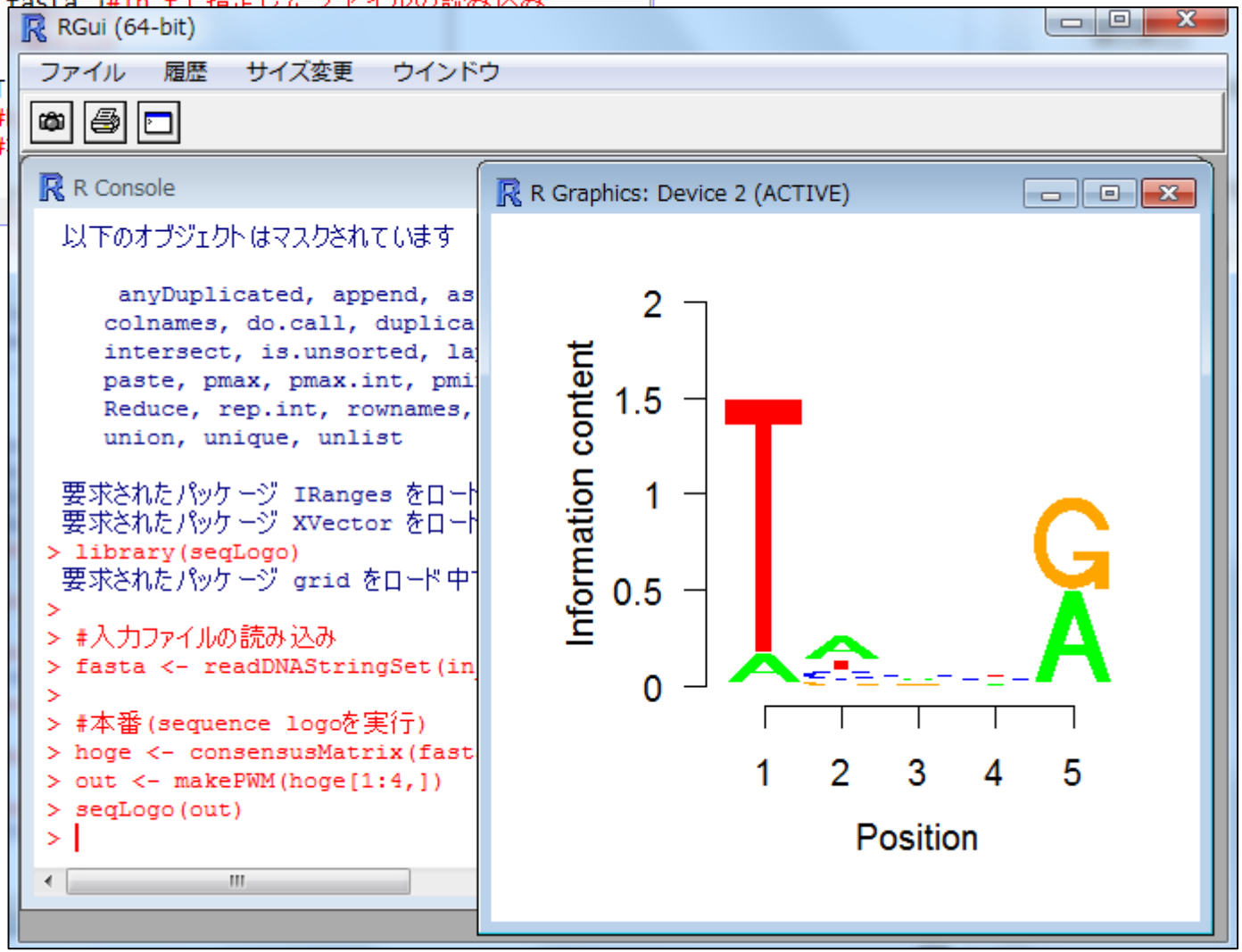
```
> str(out)
Formal class 'pwm' [package "seqLogo"] with 5 slots
  ..@ pwm      : num [1:4, 1:40] 0.326 0.187 0.152 0.335 0.324 ...
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:4] "A" "C" "G" "T"
  .. .. ..$ : chr [1:40] "1" "2" "3" "4" ...
  ..@ consensus: chr "TTTTTTTTTTAAAAAAAAAAAAAAATTTTAATTAAAAAATT"
  ..@ ic       : num [1:40] 0.0782 0.0779 0.0776 0.0714 0.067 ...
  ..@ width    : int 40
  ..@ alphabet : chr "DNA"
> pwn(out)
 エラー:  関数 "pwn" を見つけることができませんでした
> dim(out@pwn)
 エラー:  名前 "pwn" というスロットが、クラス "pwm" のこのオブジェクトには存$
> dim(out@pwm)
[1]  4 40
> dim(out@ic)
NULL
> length(out@ic)
[1] 40
> head(out@ic)
[1] 0.07820452 0.07786482 0.07764915 0.07142892 0.06697533 0.07739266
>
```

str関数を用いて、どういう情報が含まれているかを眺める

2. 入力ファイルがmulti-FASTA形式のファイル(data_seqlogo1.txt)の場合:

```
in_f <- "data_seqlogo1.txt"              #入力ファイル名を指定してin_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(seqLogo)                         #パッケージの読み込み

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

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

- 解析 | 一般 | Sequence logos(Schneider 1990)



他の解析例

Sequence logos：計算手順

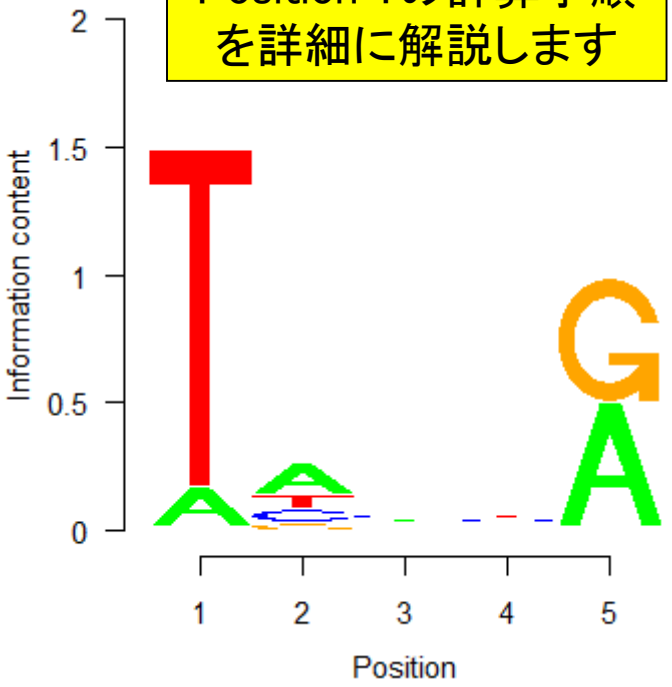
Position 1の計算手順を詳細に解説します

data_seqlogo1.txt

```
>seq1
TACGG
>seq2
TAACG
>seq3
TGTAG
>seq4
ACTTA
>seq5
TTGGA
>seq6
TCAAG
>seq7
TACTA
>seq8
TTGCA
>seq9
TAACA
>seq10
TACTG
```

	position *i*				
	1	*2*	*3*	*4*	*5*
配列 1	T	A	C	G	G
配列 2	T	A	A	C	G
配列 3	T	G	T	A	G
配列 4	A	C	T	T	A
配列 5	T	T	G	G	A
配列 6	T	C	A	A	G
配列 7	T	A	C	T	A
配列 8	T	T	G	C	A
配列 9	T	A	A	C	A
配列10	T	A	C	T	G

Sequence logos



positionごとの情報量(IC)を計算

$$IC_i = \underbrace{\log_2(N)}_{=2} - H(\boldsymbol{x}_i)$$

N：塩基の種類数= 4
Hの取りうる範囲：$0 \leqq H \leqq \log_2 N$

エントロピー $H(\boldsymbol{x}_i) = -\sum_{j=1}^{N} p_{ij} \log_2(p_{ij}),$ where $p_{ij} = x_{ij} / \sum_{j=1}^{N} x_{ij}$

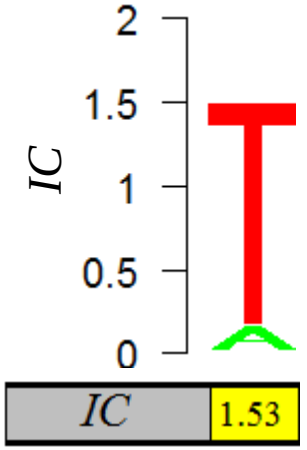
Sequence logos：計算手順

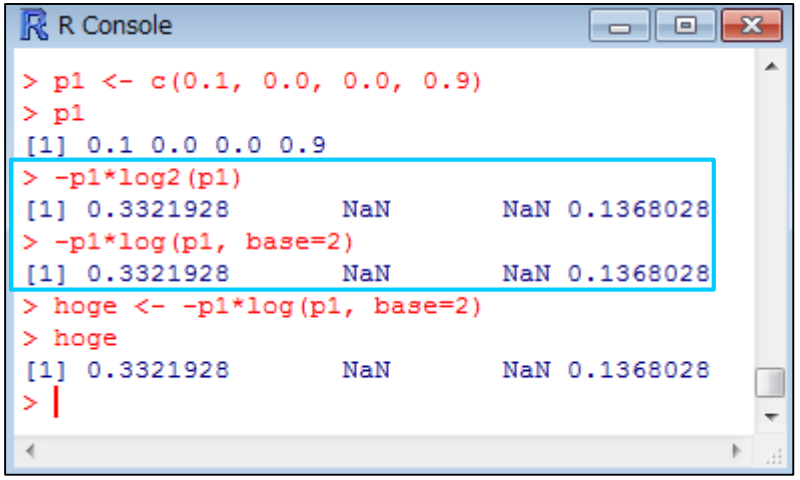
- position iの情報量 $IC_i = \log_2(N) - H(x_i)$
  - $\log_2(N) = 2$

	1
配列 1	T
配列 2	T
配列 3	T
配列 4	A
配列 5	T
配列 6	T
配列 7	T
配列 8	T
配列 9	T
配列10	T

p_{ij}	1
1	0.1
2	0.0
3	0.0
4	0.9
Σ_j	1.0

$-p_{ij}\log_2(p_{ij})$	1
1	0.33
2	0.00
3	0.00
4	0.14
$H = \Sigma_j$	0.47





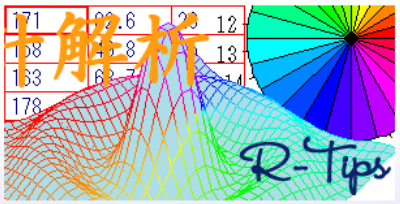
0*log(0)は計算できないので0と定義したい

R-Tipsというウェブページは有用

http://cse.naro.affrc.go.jp/takezawa/r-tips/r.html

R-Tips

フレーム：なし あり



統計解析フリーソフト R の備忘録頁 ver.3.1

R は有名な統計言語『S 言語』をオープンソースとして実装し直した統計
ざまなプラットフォーム(OS)に対応しており，誰でも自由にダウンロードす
それにも関わらず，世界中の専門家が開発に携わっており，日々新しい
付け加えられています．とにかく計算が速い上にグラフィックも充実してい
にも持ってこいです．このドキュメントは Windows 版 R と Mac OS X 版 R
R)でコマンドを調べた足跡です．

ちなみに，この頁の内容を新しくした書籍は こちら，電子書籍版は こち
ます．

● 入 門 篇 ●

リンク	The R Project	リンク	RjpWiki
リンク	PDF版 R-Tips(200頁・3Mb)	索引	この頁の索引
第01節	R のセットアップ＋参考文献	第02節	R の起動と終了
第03節	簡単な計算	第04節	R 用エディタ
第05節	オブジェクトと代入(付値)	第06節	作業ディレクト
第07節	ヘルプを見る	第08節	パッケージ・ラ
第09節	データの型	第10節	オブジェクトの
第11節	オプション		

● ベクトル篇 ●

第12節	ベクトルの作成	第13節	要素へのアク
第14節	ベクトルの計算	第15節	ベクトル要素の
第16節	種々のベクトル	第17節	文字列を操作
第18節	NULL, NA, NaN, Infの操作		

HOME

18. NULL, NA, NaN, Infの操作

● NULL, NA, NaN, Inf なのか否かを調べる

NULL (何も無い)，NA (欠損値)，NaN (非数)，Inf (無限大) は，大抵は演算を施してもそのままの値 (NA や NAN) が返ってくる．すなわち，原則として NaN にどのような演算を施しても結果は NaN になる．よって，比較演算子 == すら使えないことになる．

```
x <- c(1.0, NA, 3.0, 4.0)  # NA はどれかを調べても...
x == NA                    # NA に対する演算は全て NA となる

[1] NA NA NA
```

これら 4 つの値の検査を行う関数がそれぞれ用意されている．

命令	is.null()	is.na()	is.nan()	is.finite()	is.infinite()	complete.cases()
対象	NULLか否か	NAか否か	NaNか否か	有限か否か	無限か否か	欠損か否か

例えば，ある値が NA かどうかのテストは関数 is.na() で行なう (比較演算子 == では行なえないことに注意) が，これは NaN を代入しても TRUE が返ってしまう．そこで NaN か否か (NA か否かではない) を判定するために is.nan() という関数が用意されている．

```
is.na(NA)

[1] TRUE

is.nan(NA)

[1] FALSE

x <- c(1, NA, 3, 4)
is.na(x)

[1] FALSE TRUE FALSE FALSE
```

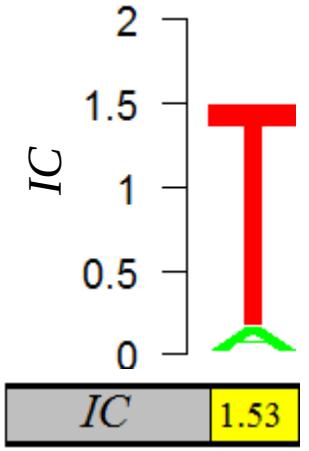
Sequence logos：計算手順

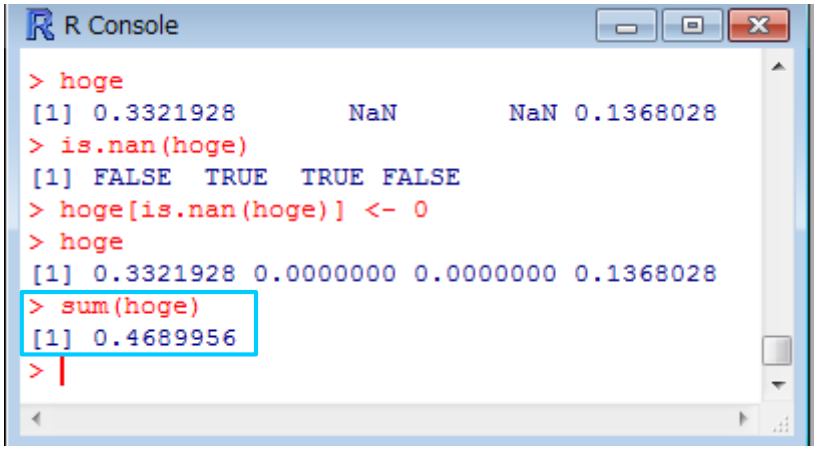
- position iの情報量 $IC_i = \log_2(N) - H(x_i)$
  $\log_2(N) = 2$

	1
配列 1	T
配列 2	T
配列 3	T
配列 4	A
配列 5	T
配列 6	T
配列 7	T
配列 8	T
配列 9	T
配列 10	T

p_{ij}	1
1	0.1
2	0.0
3	0.0
4	0.9
Σ_j	1.0

$-p_{ij}\log_2(p_{ij})$	1
1	0.33
2	0.00
3	0.00
4	0.14
$H = \Sigma_j$	0.47





これ（=0.4689956）が、この数値ベクトルのエントロピーです

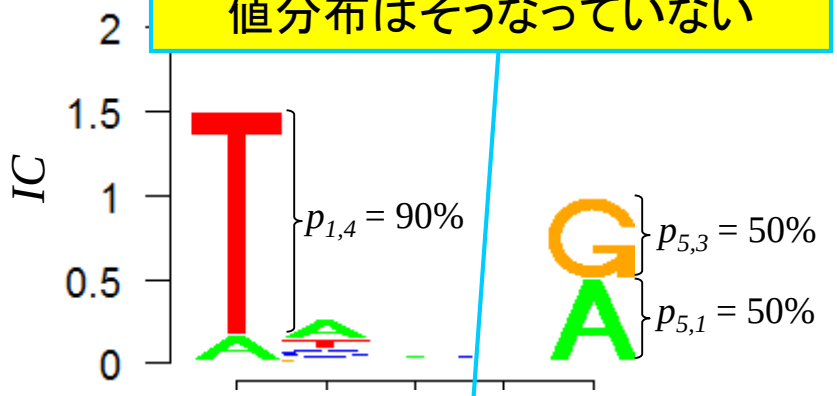
Sequence logos：計算手順

- position iの情報量 $IC_i = \log_2(N) - H(x_i)$ $= 2$

Sequence logosは、あるポジションに特定の塩基が濃縮されている状態をうまく表したい、という思想だが、エントロピーの数値分布はそうなっていない

position i

	1	2	3	4	5	…
配列 1	T	A	C	G	G	…
配列 2	T	A	A	C	G	…
配列 3	T	G	T	A	G	…
配列 4	A	C	T	T	A	…
配列 5	T	T	G	G	A	…
配列 6	T	C	A	A	G	…
配列 7	T	A	C	T	A	…
配列 8	T	T	G	C	A	…
配列 9	T	A	A	C	A	…
配列10	T	A	C	T	G	…



IC	1.53	0.24	0.03	0.03	1.00	…

x_{ij}	1	2	3	4	5	…
Aの数 (j=1)	1	5	3	2	5	…
Cの数 (j=2)	0	2	3	3	0	…
Gの数 (j=3)	0	1	2	2	5	…
Tの数 (j=4)	9	2	2	3	0	…
$\Sigma_j\ x_{ij}$	10	10	10	10	10	

j

p_{ij}	1	2	3	4	5	…
1	0.1	0.5	0.3	0.2	0.5	…
2	0.0	0.2	0.3	0.3	0.0	…
3	0.0	0.1	0.2	0.2	0.5	…
4	0.9	0.2	0.2	0.3	0.0	…
Σ_j	1.0	1.0	1.0	1.0	1.0	

$-p_{ij}\log_2(p_{ij})$	1	2	3	4	5	…
1	0.33	0.50	0.52	0.46	0.50	…
2	0.00	0.46	0.52	0.52	0.00	…
3	0.00	0.33	0.46	0.46	0.50	…
4	0.14	0.46	0.46	0.52	0.00	…
$H = \Sigma_j$	0.47	1.76	1.97	1.97	1.00	

Sequence logos：計算手順

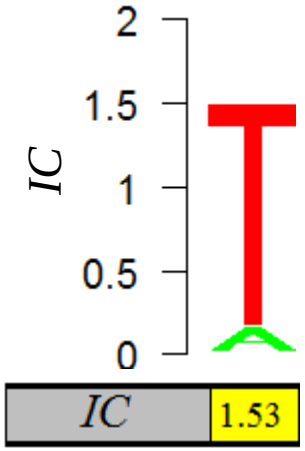
- position iの情報量 $IC_i = \log_2(N) - H(x_i)$

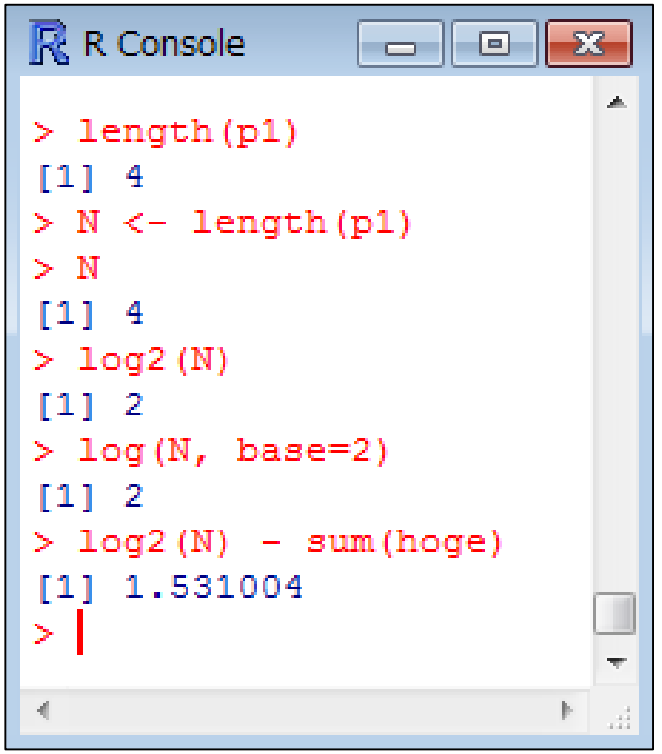
$\log_2(N) = 2$

	1
配列 1	T
配列 2	T
配列 3	T
配列 4	A
配列 5	T
配列 6	T
配列 7	T
配列 8	T
配列 9	T
配列10	T

p_{ij}	1
1	0.1
2	0.0
3	0.0
4	0.9
Σ_j	1.0

$-p_{ij}\log_2(p_{ij})$	1
1	0.33
2	0.00
3	0.00
4	0.14
$H = \Sigma_j$	0.47





length関数はベクトルの要素数を計算。p1はA, C, G, Tの出現確率からなるため、要素数N = 4。
したがって、20種類からなるアミノ酸配列のsequence logosの場合はN = 20となり、ICの最大値は約4.32となる。

5. 入力ファイルがmulti-FASTA形式のファイル(TAIR10_upstream_500_20101028)の場合:

Arabidopsisの上流500bpの配列セットです。500bpと長いため、461-500bpの範囲のみについて解析し、得られた図をファイルに保存するやり方です。以下はダウンロードしたファイルの拡張子として、"fa"を付加しているという前提です。

```
library(seqLogo)                          #パッケージの読み込み

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

#前処理(配列長が500bpのもののみフィルタリング後、解析したいサブセットを抽出)
obj <- as.logical(width(fasta) == param2)#条件を満たすかどうかを判定した結果をobjに格
fasta <- fasta[obj]                       #objがTRUEとなるもののみ抽出した結果をfastaに格
fasta
fasta <- subseq(fasta
fasta

#本番(sequence logoを
hoge <- consensusMatr
out <- makePWM(hoge[1

#ファイルに保存
png(out_f, pointsize=
seqLogo(out)
dev.off()
```

• 解析 | 一般 | Sequence logos(Schneider 1990)

配列長が500bpのもののみ抽出するための演算子

R Console

```
[33599]    500 GCCATAGTTTCAGCATTACG...CGCGCGATCGTATACTGAG ATMG01330 | chrM:...
[33600]    500 CGATTCACAGAGCGCCATTT...GAAAGCTTTCTTTATCTTT ATMG00070 | chrM:...
[33601]    500 TAAAAGGAGAGGTGCTTTAG...AGCCCTCTGTCAATCTCTG ATMG00130 | chrM:...
[33602]    500 AGTGAACTTTCACAATTTCT...TATCAATTGTGATAAGAAA ATMG00930 | chrM:...
> obj <- as.logical(width(fasta) == param2)#条件を満たすかどうかを判定した結$
> fasta <- fasta[obj]
> fasta
  A DNAStringSet instanc
        width seq
    [1]    500 TACAGATGAT
    [2]    500 CCTCACAGGA
    [3]    500 GAAATATCAC
    [4]    500 ACAAATTAGA
    [5]    500 CCAGAAGCAG
    ...    ... ...
[33596]    500 ATCTTCAGGG
[33597]    500 GCCATAGTTT
[33598]    500 CGATTCACAG
[33599]    500 TAAAAGGAGA
[33600]    500 AGTGAACTTT
>
```

http://cse.naro.affrc.go.jp/takezawa/r-tips/r/28.ht — R-Source

HOME

28. 演算子

比較演算子・比較演算関数

比較演算子には次のようなものがある. == は単なる = ではないことに注意 (= は「代入」). 以下の演算子で ASCII コード順の辞書式での文字列の比較も出来る.

記号	意味	記号	意味
==	等号	>=	≧
!=	≠	>	>
<=	≦	<	<

他にも, 比較用の関数として以下が用意されている.

命令	is.null()	is.na()	is.nan()	is.finite()	is.infinite()	complete.cases()
検査対象	NULL	NA	NaN	有限か否か	無限か否か	欠測なし？

課題4

1. TAIRから取得した転写開始点上流500bpのmulti-FASTAファイル（TAIR10_upstream_500_20101028.fa）中には500 bpでない配列が2つ含まれている。この2つの配列のみを抽出するためには以下のテンプレートコードのどこをどう変更すればよいか示せ（ここをこう変えるとよい、などでよい）。

5. 入力ファイルがmulti-FASTA形式のファイル(TAIR10_upstream_500_20101028)の場合:

Arabidopsisの上流500bpの配列セットです。500bpと長いため、461-500bpの範囲のみについて解析し、得られた図をファイルに保存するやり方です。以下はダウンロードしたファイルの拡張子として、"fa"を付加しているという前提です。

```
in_f <- "TAIR10_upstream_500_20101028.fa"#入力ファイル名を指定してin_fに格納
out_f <- "hoge5.png"                     #出力ファイル名を指定してout_fに格納
param1 <- c(461, 500)                    #抽出したい範囲の始点と終点を指定
param2 <-  500                           #配列長を指定
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(seqLogo)                         #パッケージの読み込み

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

#前処理(配列長が500bpのもののみフィルタリング後、解析したいサブセットを抽出)
obj <- as.logical(width(fasta) == param2)#条件を満たすかどうかを判定した結果をobjに格納
fasta <- fasta[obj]                      #objがTRUEとなるもののみ抽出した結果をfastaに格
fasta                                    #確認してるだけです
fasta <- subseq(fasta, param1[1], param1[2])#param1で指定した始点と終点の範囲の配列を
fasta                                    #確認してるだけです

#本番(sequence logoを実行)
```

課題4

2. TAIRから取得した転写開始点上流500bpのmulti-FASTAファイル（TAIR10_upstream_500_20101028.fa）中には500 bpでない配列が2つ含まれている。この2つの配列の配列長を示せ。
3. このコードは、上流500bpのファイルを読み込んで461番目から500番目の範囲を切り出して、sequence logosを行っている。この目的を達成するために、subseq関数のデフォルトオプションの「start=461とend=500」を利用している。一方、これは上流40bp、つまり最後から40bpを切り出していることと同義であるため、「end=500と**何か**」というオプションで表現することもできる。この**何か**を示せ。

7. 入力ファイルがmulti-FASTA形式のファイル(TAIR10_upstream_1000_20101104)の場合:

6.と同じ結果が得られますが、転写開始点上流50bpのみを切り出して解析するというオプションにしています。

```
#前処理(配列長が500bpのもののみフィルタリング後、解析したいサブセットを抽出)
obj <- as.logical(width(fasta) == param2)#条件を満たすかどうかを判定した結果をobjに格納
fasta <- fasta[obj]                      #objがTRUEとなるもののみ抽出した結果をfastaに格納
fasta                                    #確認してるだけです
fasta <- subseq(fasta, width=param1, end=param2)#解析したい範囲を切り出してfastaに格納
fasta                                    #確認してるだけです

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

png(out_f, pointsize=13, width=param_fig[1], height=param_fig[2])#出力ファイルの各種パラメータを
seqLogo(out)                             #塩基組成やicの情報を含むoutを入力としてsequence logoを描
dev.off()                                #おまじない

#おまけ(配列長分布や配列長がparam2と異なるID情報を取得)
fasta <- readDNAStringSet(in_f, format="fasta")#in_fで指定したファイルの読み込み
table(width(fasta))                      #配列長分布(ほとんどがparam2と同じであることがわかる)
obj <- as.logical(width(fasta) != param2)#条件を満たすかどうかを判定した結果をobjに格納
fasta <- fasta[obj]                      #objがTRUEとなるもののみ抽出した結果をfastaに格納
fasta                                    #確認してるだけです
```

析 | 一般 | Sequence logos(Schneider 1990)

課題4のヒント

R Console

```
> #おまけ(配列長分布や配列長がparam2と異なるID情報を取得)
> fasta <- readDNAStringSet(in_f, format="fasta")#in_fで指定したファイルの読$
> table(width(fasta))                      #配列長分布(ほとんどがparam2と同じ$

   26   224  1000
    1     1 33600
> obj <- as.logical(width(fasta) != param2)#条件を満たすかどうかを判定した結$
> fasta <- fasta[obj]                      #objがTRUEとなるもののみ抽出した結$
> fasta                                    #確認してるだけです
  A DNAStringSet instance of length 2
    width seq                                               names
[1]    26 GGGTTTAGGGTTTAGGGTTTAGGGTT                        AT3G63540 | chr3:...
[2]   224 AATCCGGTTTGTTTCCATTCTG...ATAGGTTGTTACACCCCTTCC ATMG01410 | chrM:...
>
```

プロモーター解析とRのバージョン周辺

(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 2014/04/23, since 2010)

What's new?

- このウェブページはフリーソフトR
  ンストールと起動を参考にして
  います。(2014/04/22) NEW
- 2014年9月1日～12日に「バイオ
  ます。近いうちに詳細を公開しま
- 門田幸二 著シリーズ Useful R
- 参考資料(講義、講習会、本など

- はじめに (last modified 2014/01/
- 参考資料(講義、講習会、本など
- 過去のお知らせ (last modified 2
- Rのインストールと起動 (last mod
- サンプルデータ (last modified 20
- 書籍 | について (last modified 20
- 書籍 | 第2章 データ取得 | 2.3 RN
- 書籍 | 第2章 データ取得 | 2.3 RN
- 書籍 | 第2章 データ取得 | 2.3 RN
- 書籍 | 第2章 データ取得 | 2.3 RN
- 書籍 | 第2章 データ取得 | 2.3 RN
- 書籍 | 第2章 データ取得 | 2.3 RN
- 書籍 | 第3章 データ解析(基礎) |
- 書籍 | 第3章 データ解析(基礎) |

- 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 2014/04/10) NEW
- イントロ | 一般 | 配列取得 | ゲノム配列 | BSgenome(last modified 2014/04/22) 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) NEW
- イントロ | 一般 | 配列取得 | プロモーター配列 | BSgenome(last modified 2014/04/23) NEW
- 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) NEW
- 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 2014/04/02) NEW
- イントロ | 一般 | 配列取得 | トランスクリプトーム配列 | biomaRt(Durinck 2009)(last modified 2013/09/25)
- イントロ | NGS | 様々なプラットフォーム(last modified 2013/06/12)
- イントロ | NGS | qPCRやmicroarrayなどとの比較(last modified 2010/12/16)
- イントロ | NGS | Viewer(last modified 2014/01/29)
- イントロ | NGS | 配列取得 | FASTQ or SRALite | 公共DBから(last modified 2014/03/27) NEW
- イントロ | NGS | 配列取得 | FASTQ or SRALite | SRAdb(Zhu 2013)(last modified 2014/04/01) NEW
- イントロ | NGS | アノテーション情報取得 | について(last modified 2014/03
- イントロ | NGS | アノテーション情報取得 | GFF/GTF形式ファイル(last mo
- イントロ | NGS | アノテーション情報取得 | refFlat形式ファイル(last modifi
- イントロ | NGS | アノテーション情報取得 | biomaRt(Durinck 2009)(last mo
- 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) NEW
- イントロ | NGS | アノテーション情報取得 | TranscriptDb | TxDb.*から(last modified 2013/10/08)
- イントロ | NGS | アノテーション情報取得 | TranscriptDb | GenomicFeatures(Lawrence 2013)(last modified 2013/10/13)推奨
- イントロ | NGS | アノテーション情報取得 | TranscriptDb | GFF/GTF形式ファイルから(last modified 2014/04/01) NEW
- イントロ | NGS | 読み込み | FASTA形式 | 基本情報を取得(last modified 2014/03/10)
- 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) NEW

バージョンによってできることとできないことがあるという話

プロモーター解析とRのバージョン周辺

イントロ | 一般 | 配列取得 | プロモーター配列 | BSgenome NEW

BSgenome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。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.s
(R.norvegicus)、出
わかりますが、生物種
「ファイル」-「ディレクトリ

1. 利用可能な生物種と

```
#必要なパッケージを
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
```

2. ゼブラフィッシュ("BSg

400MB程度あります...

5. インストール済みのヒト("BSgenome.Hsapiens.UCSC.hg19")の転写開始点上流配列(1000bp)をmulti-FASTAファイルで保存したい場合:

上流1000bp以外に2000bp, 5000bpもあります...

```
out_f <- "hoge5.txt"                          #出力ファイル名を指定してout_fに格納
param1 <- "BSgenome.Hsapiens.UCSC.hg19"#パッケージ名を指定
param2 <- "upstream1000"                      #上流2000bpを指定

#必要なパッケージをロード
library(param1, character.only=T)             #param1で指定したパッケージの読み込み

#前処理(param1で指定したパッケージ中のオブジェクト名をhogeに統一)
#tmp <- unlist(strsplit(param, ".", fixed=TRUE))[2]#param1で指定した文字列からオブジェク
tmp <- ls(paste("package", param1, sep=":"))#param1で指定したパッケージで利用可能なオブジ
hoge <- eval(parse(text=tmp))                 #文字列tmpをRオブジェクトとしてhogeに格納(パッケー
hoge                                          #確認してるだけ

#本番
tmp <- paste("hoge$", param2, sep="")         #param2で指定した文字列を含むオブジェクト名を作成
fasta <- eval(parse(text=tmp))                #文字列tmpをRオブジェクトとしてfastaに格納

#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
```

バージョンによってできることとできないことがあるという話

プロモーター解析とRのバージョン周辺

R ver. 3.0.3での実行例を示します。

R Console

```
R version 3.0.3 (2014-03-06) -- "Warm Puppy"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R は、自由なソフトウェアであり、「完全に無保証」です。
一定の条件に従えば、自由にこれを再配布することができます。
配布条件の詳細に関しては、'license()' あるいは 'lice

R は多くの貢献者による共同プロジェクトです。
詳しくは 'contributors()' と入力してください。
また、R や R のパッケージを出版物で引用する際の形式につ
'citation()' と入力してください。

'demo()' と入力すればデモをみることができます。
'help()' とすればオンラインヘルプが出ます。
'help.start()' で HTML ブラウザによるヘルプがみられます
'q()' と入力すれば R を終了します。

 [以前にセーブされたワークスペースを復帰します]

> getwd()
[1] "C:/Users/kadota/Desktop/hoge"
```

R Console

```
|   chrUn_gl000220        chrUn_gl000221        chrUn_gl000222
|   chrUn_gl000223        chrUn_gl000224        chrUn_gl000225
|   chrUn_gl000226        chrUn_gl000227        chrUn_gl000228
|   chrUn_gl000229        chrUn_gl000230        chrUn_gl000231
|   chrUn_gl000232        chrUn_gl000233        chrUn_gl000234
|   chrUn_gl000235        chrUn_gl000236        chrUn_gl000237
|   chrUn_gl000238        chrUn_gl000239        chrUn_gl000240
|   chrUn_gl000241        chrUn_gl000242        chrUn_gl000243
|   chrUn_gl000244        chrUn_gl000245        chrUn_gl000246
|   chrUn_gl000247        chrUn_gl000248        chrUn_gl000249
|
| multiple sequences (see '?mseqnames'):
|   upstream1000  upstream2000  upstream5000
|
| (use the '$' or '[[' operator to access a given sequence)
>
> #本番
> tmp <- paste("hoge$", param2, sep="")   #param2で指定した文字列を含むオブジ$
> fasta <- eval(parse(text=tmp))          #文字列tmpをRオブジェクトとしてfast$
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", width=50)#fastaの中身を$
>
```

5. インストール済みのヒト("BSgenome.Hsapiens.UCSC.hg19")の転写開始点上流配列(1000bp)をmulti-FASTAファイルで保存したい場合:

上流1000bp以外に2000bp, 5000bpもあります...

```
out_f <- "hoge5.txt"                        #出力ファイル名を指定してout_fに格納
param1 <- "BSgenome.Hsapiens.UCSC.hg19"#パッケージ名を指定
param2 <- "upstream1000"                    #上流2000bpを指定

#必要なパッケージをロード
library(param1, character.only=T)           #param1で指定したパッケージの読

#前処理(param1で指定したパッケージ中のオブジェクト名をhogeに統一)
#tmp <- unlist(strsplit(param, ".", fixed=TRUE))[2]#param1で指定した文
tmp <- ls(paste("package", param1, sep=":"))#param1で指定したパッケー
hoge <- eval(parse(text=tmp))               #文字列tmpをRオブジェクトとしてhogeに格納(パッケー
hoge                                        #確認

#本番
tmp <- paste("hoge$", param2, sep="")   #param
fasta <- eval(parse(text=tmp))          #文字列

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

R ver. 3.0.3では、特に何の問題もなく上流配列を取得できます。理由は、param2で指定したupstream1000情報が存在するからです。

R Console

```
> #本番
> tmp <- paste("hoge$", param2, sep="")   #param2で指定した文字列を含むオブジ$
> fasta <- eval(parse(text=tmp))          #文字列tmpをRオブジェクトとしてfast$
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", width=50)#fastaの中身を$
> fasta
  A DNAStringSet instance of length 28020
        width seq                                               names
    [1]  1000 CCACCTGGGGAAGCGAGGCC...AGCTTGCCGTTCCTCTCCC NM_032291_up_1000...
    [2]  1000 TGGACAACGACTTGGAAGTC...CCCCCGTGCTCCTGCCGCC NM_013943_up_1000...
    [3]  1000 GAGGCAGAGGTTGCAGTGAG...CGGCACTATGGGCGGGGCC NM_052998_up_1000...
    [4]  1000 ATGTGAGAGAGTTCAAGCTG...AGGCGTCCCTCCCGCCCTC NM_032785_up_1000...
    [5]  1000 ATCAGAAGTTTGGGATCAGC...GCGAGCTGCCGCTCTAGCC NM_001145277_up_1...
    ...   ... ...
[28016]  1000 AGCGACGCGGGGACTGGGGG...GCCCTCCCCCACCACCCCC NM_001127389_up_1...
[28017]  1000 AAAGACAGAGCGACGCGGGG...GCCACCACGCCCTCCCCCA NM_033178_up_1000...
[28018]  1000 AAAGACAGAGCGACGCGGGG...GCCACCACGCCCTCCCCCA NM_033178_up_1000...
[28019]  1000 TTGTATTTTTAGTAGAGATG...GAGCCCTCTAGCTGTGTGT NM_006625_up_1000...
[28020]  1000 TTGTATTTTTAGTAGAGATG...GAGCCCTCTAGCTGTGTGT NM_054016_up_1000...
> tmp
[1] "hoge$upstream1000"
>
```

2013年7月以降のリニューアルで、コードのコピーがやりずらくなっています。**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。

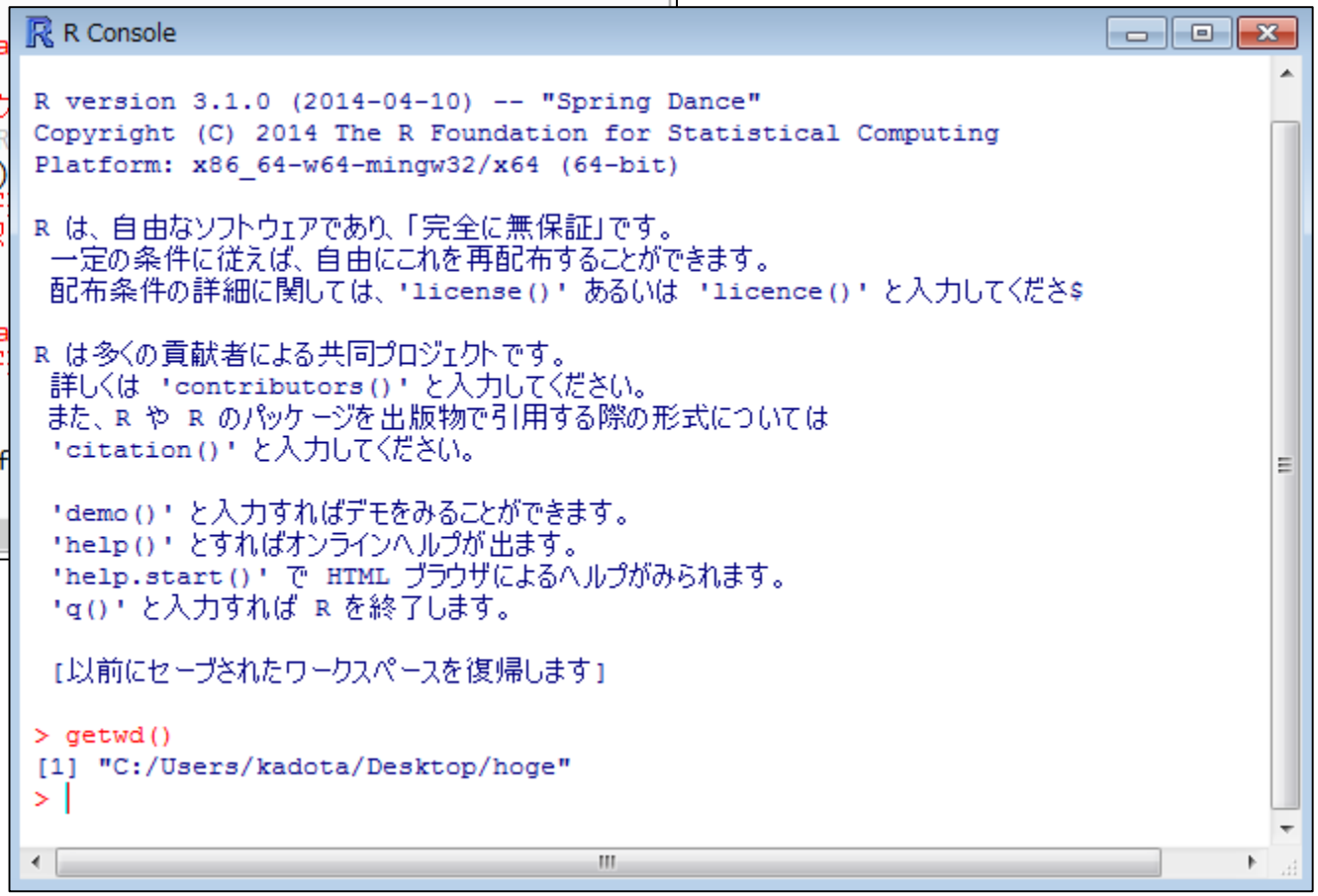
プロモーター解析とRのバージョン周辺

5. インストール済みのヒト("BSgenome.Hsapiens.UCSC.hg19")の転写開始点上流配列(1000bp)をmulti-FASTAファイルで保存したい場合:

上流1000bp以外に2000bp, 5000bpもあります...

```
out_f <- "hoge5.txt"                    #出力ファイル名を指定してout_fに格納
param1 <- "BSgenome.Hsapiens.UCSC.hg19"#パッケージ名を指定
param2 <- "upstream1000"                #上流2000bpを指定

#必要なパッケージをロード
library(param1, character.only=T)       #para

#前処理(param1で指定したパッケージ中のオブジェク
#tmp <- unlist(strsplit(param, ".", fixed=TR
tmp <- ls(paste("package", param1, sep=":"))
hoge <- eval(parse(text=tmp))           #文字
hoge                                    #確認

#本番
tmp <- paste("hoge$", param2, sep="")   #para
fasta <- eval(parse(text=tmp))          #文字

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

R ver. 3.1.0でやってみます。



プロモーター解析とRのバージョン周辺

5. インストール済みのヒト("BSgenome.Hsapiens.UCSC.hg19")の転写開始点上流配列(1000bp)をmulti-FASTAファイルで保存したい場合:

上流1000bp以外に2000bp, 5000bpもあります...

```
out_f <- "hoge5.txt"                          #出
param1 <- "BSgenome.Hsapiens.UCSC.hg19"#パ
param2 <- "upstream1000"                      #上

#必要なパッケージをロード
library(param1, character.only=T)             #pa

#前処理(param1で指定したパッケージ中のオブジ
#tmp <- unlist(strsplit(param, ".", fixed=
tmp <- ls(paste("package", param1, sep=":"
hoge <- eval(parse(text=tmp))                 #文
hoge                                          #確

#本番
tmp <- paste("hoge$", param2, sep="")         #pa
fasta <- eval(parse(text=tmp))                #文

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

R ver. 3.1.0でやってみると、警告メッセージが出ます…。

R Console

```
| multiple sequences (see '?mseqnames'):
|   upstream1000  upstream2000  upstream5000
|
| (use the '$' or '[[' operator to access a given sequence)
>
> #本番
> tmp <- paste("hoge$", param2, sep="")   #param2で指定した文字列を含むオブジ$
> fasta <- eval(parse(text=tmp))          #文字列tmpをRオブジェクトとしてfast$
 警告メッセージ:
  Starting with BioC 2.14, upstream sequences are deprecated.
  However they can easily be extracted from the full genome
  sequences with something like (for example for hg19):

      library(TxDb.Hsapiens.UCSC.hg19.knownGene)
      txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
      gn <- sort(genes(txdb))
      up1000 <- flank(gn, width=1000)
      library(BSgenome.Hsapiens.UCSC.hg19)
      genome <- BSgenome.Hsapiens.UCSC.hg19
      up1000seqs <- getSeq(genome, up1000)

  IMPORTANT: Make sure you use a TxDb package (or TranscriptDb object)
  that contains a gene model based on the exact same reference genome
  as the BSgenome object you pass to getSeq(). Note that you can make
  your own custom TranscriptDb object from various annotation resources.
  See the makeTranscriptDbFromUCSC(), makeTranscriptDbFromBiomart(), and
  makeTranscriptDbFromGFF() functions in the GenomicFeatures package.
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", width=50)#fastaの中身を$
>
```

プロモーター解析とRのバージョン周辺

5. インストール済みのヒト("BSgenome.Hsapiens.UCSC.hg19")の転写開始点上流配列(1000bp)をmulti-FASTAファイルで保存したい場合:

上流1000bp以外に2000bp, 5000bpもあります...

```
out_f <- "hoge5.txt"                          #出
param1 <- "BSgenome.Hsapiens.UCSC.hg19"#パ
param2 <- "upstream1000"                      #上

#必要なパッケージをロード
library(param1, character.only=T)             #pa

#前処理(param1で指定したパッケージ中のオブジ
#tmp <- unlist(strsplit(param, ".", fixed=
tmp <- ls(paste("package", param1, sep=":"
hoge <- eval(parse(text=tmp))                 #文
hoge                                          #確

#本番
tmp <- paste("hoge$", param2, sep="")         #pa
fasta <- eval(parse(text=tmp))                #文

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

R Console

```
      txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
      gn <- sort(genes(txdb))
      up1000 <- flank(gn, width=1000)
      library(BSgenome.Hsapiens.UCSC.hg19)
      genome <- BSgenome.Hsapiens.UCSC.hg19
      up1000seqs <- getSeq(genome, up1000)

  IMPORTANT: Make sure you use a TxDb package (or TranscriptDb object)
  that contains a gene model based on the exact same reference genome
  as the BSgenome object you pass to getSeq(). Note that you can make
  your own custom TranscriptDb object from various annotation resources.
  See the makeTranscriptDbFromUCSC(), makeTranscriptDbFromBiomart(), and
  makeTranscriptDbFromGFF() functions in the GenomicFeatures package.
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", width=50)#fastaの中身を$
> fasta
  A DNAStringSet instance of length 28020
        width seq                                               names
    [1]  1000 CCACCTGGGGAAGCGAGGCC...AGCTTGCCGTTCCTCTCCC NM_032291_up_1000...
    [2]  1000 TGGACAACGACTTGGAAGTC...CCCCCGTGCTCCTGCCGCC NM_013943_up_1000...
    [3]  1000 GAGGCAGAGGTTGCAGTGAG...CGGCACTATGGGCGGGGCC NM_052998_up_1000...
    [4]  1000 ATGTGAGAGAGTTCAAGCTG...AGGCGTCCCTCCCGCCCTC NM_032785_up_1000...
    [5]  1000 ATCAGAAGTTTGGGATCAGC...GCGAGCTGCCGCTCTAGCC NM_001145277_up_1...
    ...   ... ...
[28016]  1000 AGCGACGCGGGGACTGGGGG...GCCCTCCCCCACCACCCCC NM_001127389_up_1...
[28017]  1000 AAAGACAGAGCGACGCGGGG...GCCACCACGCCCTCCCCCA NM_033178_up_1000...
[28018]  1000 AAAGACAGAGCGACGCGGGG...GCCACCACGCCCTCCCCCA NM_033178_up_1000...
[28019]  1000 TTGTATTTTTAGTAGAGATG...GAGCCCTCTAGCTGTGTGT NM_006625_up_1000...
[28020]  1000 TTGTATTTTTAGTAGAGATG...GAGCCCTCTAGCTGTGTGT NM_054016_up_1000...
>
```

R ver. 3.1.0でやってみると、警告メッセージが出ます…が一応上流1000bpの配列は取得できているようです。

6. インストール済みのヒト("BSgenome.Hsapiens.NCBI.GRCh38")の転写開始点上流配列(1000bp)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のRパッケージです。上流配列情報を含まないのでエラーがでます。

```
out_f <- "hoge6.txt"                          #出力ファイル名を指定してout_fに格納
param1 <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定
param2 <- "upstream1000"                      #上流1000bpを指定

#必要なパッケージをロ
library(param1, cha

#前処理(param1で指定
#tmp <- unlist(strs
tmp <- ls(paste("pa
hoge <- eval(parse(
hoge

#本番
tmp <- paste("hoge$
fasta <- eval(parse

#ファイルに保存
writeXStringSet(fas
```

R Console

```
 要求されたパッケージ XVector をロード中です
>
> #前処理(param1で指定したパッケージ中のオブジェクト名をhogeに統一)
> #tmp <- unlist(strsplit(param, ".", fixed=TRUE))[2]#param1で指定した文字列$
> tmp <- ls(paste("package", param1, sep=":"))#param1で指定したパッケージで$
> hoge <- eval(parse(text=tmp))              #文字列tmpをRオブジェクトとしてhoge$
> hoge                                       #確認してるだけです(ここで、multipl$
Human genome
|
| organism: Homo sapiens (Human)
| provider: NCBI
| provider version: GRCh38
| release date: 2013-12-17
| release name: Genome Reference Consortium Human Build 38
|
```

ヒトゲノム最新版パッケージには、上流配列情報はない(R ver. 3.1.0)

R Console

```
|   HSCHR19KIR_FH08_BAX_HAP_CTG3_1      HSCHR19KIR_FH13_A_HAP_CTG3_1
|   HSCHR19KIR_FH13_BA2_HAP_CTG3_1      HSCHR19KIR_FH15_A_HAP_CTG3_1
|   HSCHR19KIR_RP5_B_HAP_CTG3_1
|
| (use the '$' or '[[' operator to access a given sequence)
>
> #本番
> tmp <- paste("hoge$", param2, sep="")   #param2で指定した文字列を含むオブジ$
> fasta <- eval(parse(text=tmp))          #文字列tmpをRオブジェクトとしてfast$
 以下にエラー x[[name]] : no such sequence
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", width=50)#fastaの中身を$
 以下にエラー is(x, "XStringSet") :  オブジェクト 'fasta' がありません
>
```

バージョンの違いに気をつけよう

- R本体、パッケージ、データ取得先のデータベース…

5. インストール済みのヒト("BSgenome.Hsapiens.UCSC.hg19")の転写開始点
FASTAファイルで保存したい場合:

上流1000bp以外に2000bp, 5000bpもあります…

```
out_f <- "hoge5.txt"                          #出力ファイル名を指定
param1 <- "BSgenome.Hsapiens.UCSC.hg19"#パッケージ名を指定
```

R Console

```
> hoge                                        #確認して$
Human genome
|
| organism: Homo sapiens (Human)
| provider: UCSC
| provider version: hg19
| release date: Feb. 2009
| release name: Genome Reference Consortium GRCh37
|
| single sequences (see '?seqnames'):
|   chr1                    chr2                    $
```

6. インストール済みのヒト("BSgenome.Hsapiens.NCBI.GRCh38")の転写開始
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のRパッケ
ないのでエラーがでます。

```
out_f <- "hoge6.txt"                          #出力ファイル名を指定
param1 <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定
```

R Console

```
> hoge                                        #確認してるだけで$
Human genome
|
| organism: Homo sapiens (Human)
| provider: NCBI
| provider version: GRCh38
| release date: 2013-12-17
| release name: Genome Reference Consortium Human Build 38
|
| sequences (see '?seqnames'):
|   1                                                 $
```

基本的には半年ごとにRの最新版をインストールして利用

R Console

```
> sessionInfo()
R version 3.0.3 (2014-03-06)
Platform: x86_64-w64-mingw32/x64 (64-bit)

locale:
[1] LC_COLLATE=Japanese_Japan.932  LC_CTYPE=Japanese_Japan.932
[3] LC_MONETARY=Japanese_Japan.932 LC_NUMERIC=C
[5] LC_TIME=Japanese_Japan.932

attached base packages:
[1] parallel  stats     graphics  grDevices utils     datasets  methods
[8] base

other attached packages:
[1] BSgenome.Hsapiens.UCSC.hg19_1.3.19 BSgenome_1
[3] Biostrings_2.30.1                  GenomicRan
[5] XVector_0.2.0                      IRanges_1.
[7] BiocGenerics_0.8.0

loaded via a namespace (and not attached):
[1] stats4_3.0.3
>
```

バージョン情報取得手段

R Console

```
> sessionInfo()
R version 3.1.0 (2014-04-10)
Platform: x86_64-w64-mingw32/x64 (64-bit)

locale:
[1] LC_COLLATE=Japanese_Japan.932  LC_CTYPE=Japanese_Japan.932
[3] LC_MONETARY=Japanese_Japan.932 LC_NUMERIC=C
[5] LC_TIME=Japanese_Japan.932

attached base packages:
[1] parallel  stats     graphics  grDevices utils     datasets  methods
[8] base

other attached packages:
[1] BSgenome.Hsapiens.NCBI.GRCh38_1.3.999
[2] BSgenome_1.32.0
[3] Biostrings_2.32.0
[4] XVector_0.4.0
[5] GenomicRanges_1.16.1
[6] GenomeInfoDb_1.0.2
[7] IRanges_1.22.3
[8] BiocGenerics_0.10.0

loaded via a namespace (and not attached):
[1] bitops_1.0-6     Rsamtools_1.16.0 stats4_3.1.0     zlibbioc_1.10.0
>
```

プロモーター解析とRのバージョン周辺

5. インストール済みのヒト("BSgenome.Hsapiens.UCSC.hg19")の転写開始点上流配列(1000bp)をmulti-FASTAファイルで保存したい場合:

上流1000bp以外に2000bp, 5000bpもあります…

```
out_f <- "hoge5.txt"
param1 <- "BSgenome.Hsapiens.UCSC.hg
param2 <- "upstream1000"

#必要なパッケージをロード
library(param1, character.only=T)

#前処理(param1で指定したパッケージ中の
#tmp <- unlist(strsplit(param, "."
tmp <- ls(paste("package", param1, s
hoge <- eval(parse(text=tmp))
hoge

#本番
tmp <- paste("hoge$", param2, sep=""
fasta <- eval(parse(text=tmp))

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

R Console

```
| multiple sequences (see '?mseqnames'):
|   upstream1000  upstream2000  upstream5000
|
| (use the '$' or '[[' operator to access a
>
> #本番
> tmp <- paste("hoge$", param2, sep="")   #param2で指定した文字列を含むオブジ$
> fasta <- eval(parse(text=tmp))          #文字列tmpをRオブジェクトとしてfast$
 警告メッセージ:
  Starting with BioC 2.14, upstream sequences are deprecated.
  However they can easily be extracted from the full genome
  sequences with something like (for example for hg19):

      library(TxDb.Hsapiens.UCSC.hg19.knownGene)
      txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
      gn <- sort(genes(txdb))
      up1000 <- flank(gn, width=1000)
      library(BSgenome.Hsapiens.UCSC.hg19)
      genome <- BSgenome.Hsapiens.UCSC.hg19
      up1000seqs <- getSeq(genome, up1000)

  IMPORTANT: Make sure you use a TxDb package (or TranscriptDb object)
  that contains a gene model based on the exact same reference genome
  as the BSgenome object you pass to getSeq(). Note that you can make
  your own custom TranscriptDb object from various annotation resources.
  See the makeTranscriptDbFromUCSC(), makeTranscriptDbFromBiomart(), and
  makeTranscriptDbFromGFF() functions in the GenomicFeatures package.
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", width=50)#fastaの中身を$
>
```

R ver. 3.1.0で出た警告メッセージを要約すると…
・upstreamの情報は使うな
・TranscriptDbオブジェクトを使え

```
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
gn <- sort(genes(txdb))
up1000 <- flank(gn, width=1000)
library(BSgenome.Hsapiens.UCSC.hg19)
genome <- BSgenome.Hsapiens.UCSC.hg19
up1000seqs <- getSeq(genome, up1000)
```


推奨手順に従った記述法。確かに上流配列の塩基数を自在に設定できるので便利ではある。

8. インストール済みのヒト("BSgenome.Hsapiens.UCSC.hg19")の転写開始点上流配列(1000bp)をmulti-FASTAファイルで保存したい場合:

2014年4月リリースのBioconductor 2.14での推奨手順です。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)を読み込んで実行しています。

```
out_f <- "hoge8.txt"                          #出力ファイル名を指定してout_fに格納
param1 <- "BSgenome.Hsapiens.UCSC.hg19"#パッケージ名を指定(BSgenome系のゲノムパッケージ)
param2 <- "TxDb.Hsapiens.UCSC.hg19.knownGene"#パッケージ名を指定(Transcript DBオブジェクト系のアノテーションパ
param3 <- 1000                                #上流 x bp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
library(param1, character.only=T)             #指定したパ
tmp <- ls(paste("package", param1, sep=":"))#指定
genome <- eval(parse(text=tmp))               #文字列tmp

library(param2, character.only=T)             #指定したパ
tmp <- ls(paste("package", param2, sep=":"))#指定
txdb <- eval(parse(text=tmp))                 #文字列tmp

#本番
gn <- sort(genes(txdb))                       #遺伝子の座
hoge <- flank(gn, width=param3)               #指定した範
fasta <- getSeq(genome, hoge)                 #指定した範

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

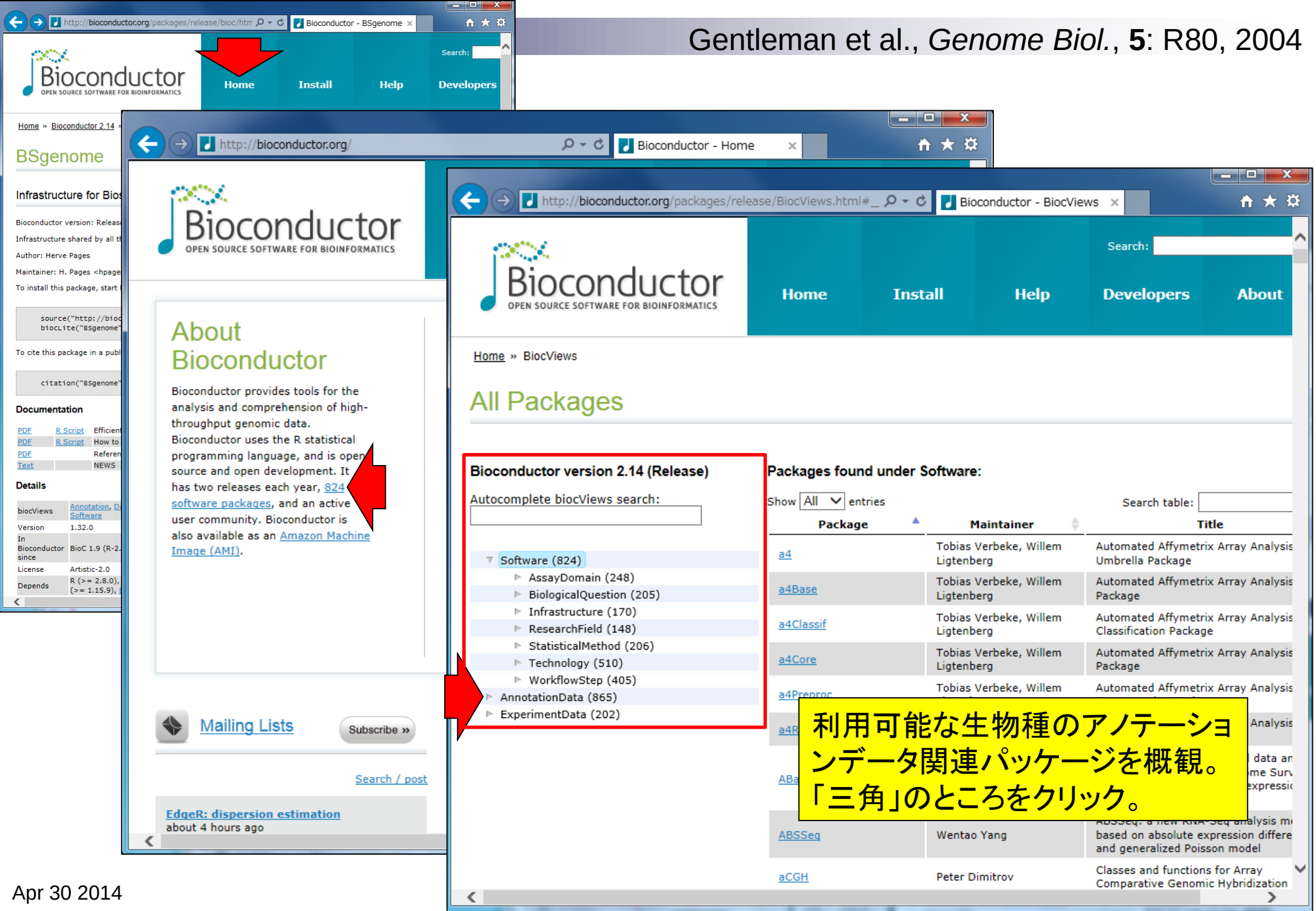
- BioconductorのBSgenomeのwebページ

R Console

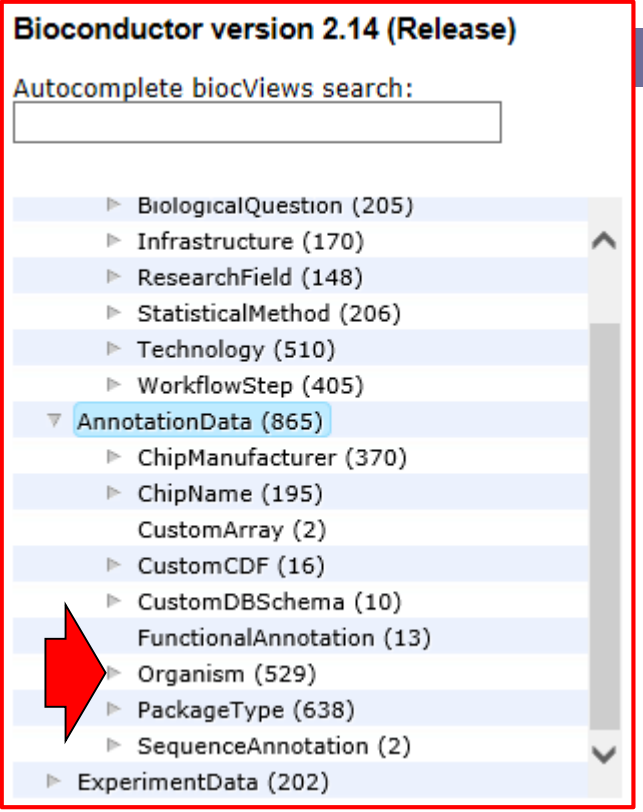
```
> fasta <- getSeq(genome, hoge)          #指定した範囲の塩基配列情報$
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", width=50)#fasta$
> fasta
  A DNAStringSet instance of length 23056
        width seq                                                names       $
    [1]  1000 ACACATGCTACCGCGTCCAG...CCATTTTTCTTTTCGTTAA 100287102
    [2]  1000 GCTATTATCACCTATATTTT...GAGTGAAACGAATAACTCT 79501
    [3]  1000 GAATTAGGCTTCTGCTGCCC...GGCAGAGAAAAAGGCGGGG 643837
    [4]  1000 CGGGGAGCCCCGAGGCCCTG...CTCCCCCCAGCTTGGGCCA 148398
    [5]  1000 CGGCGGGGCTCCTATGCAAA...TGCGGGCGGGAGCGGCGGG 339451
    ...   ... ...
[23052]  1000 GGTGAGCCAATCCTGACTCC...GGAAGTGGAAATCTCAGCC 283788
[23053]  1000 AGCCCTCCACACAAGGGGCT...TTGTTTCTTCCTCTCCAAC 100507412
[23054]  1000 CGGGGCCCAGGGAGTGGGCG...GGGCAGGCCTCCTGGCTGC 728410
[23055]  1000 CGGGGCCCAGGGAGTGGGCG...GGGCAGGCCTCCTGGCTGC 100653046
[23056]  1000 CAGGCTGAGCCCTGCAACGC...CCGGCCGGGGCTCACCGCG 100288687
>
```

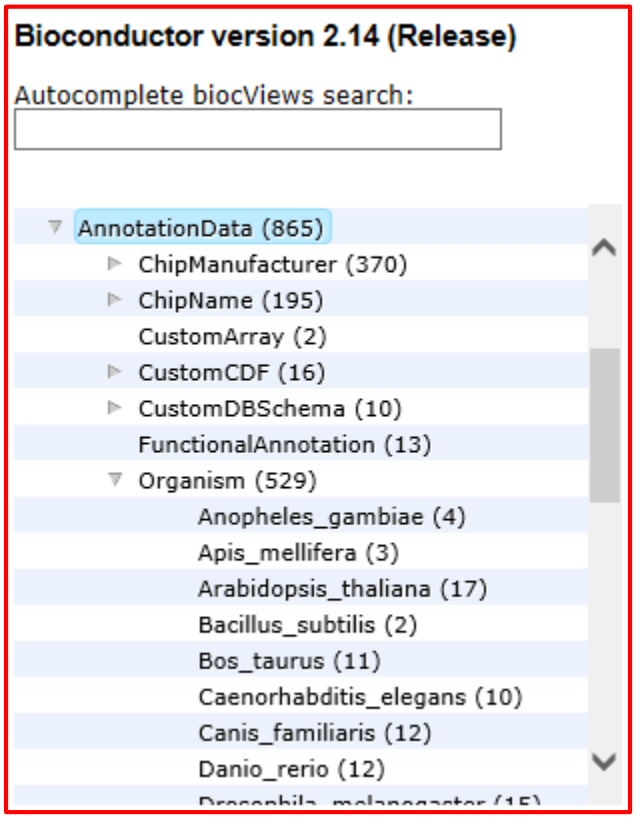
ヒトゲノムの最新版(hg38)や他の生物種はどう取り扱うのか？

Gentleman et al., *Genome Biol.*, **5**: R80, 2004



Apr 30 2014





様々な生物種のアノテーションパッケージがあることがわかります。

Bioconductor version 2.14 (Release)

Autocomplete biocViews search:

- Anopheles_gambiae (4)
- Apis_mellifera (3)
- Arabidopsis_thaliana (17)
- Bacillus_subtilis (2)
- Bos_taurus (11)
- Caenorhabditis_elegans (10)
- Canis_familiaris (12)
- Danio_rerio (12)
- Drosophila_melanogaster (15)
- Escherichia_coli (12)
- Gallus_gallus (9)
- Gasterosteus_aculeatus (2)
- Homo_sapiens (196)
- Hordeum_vulgare (2)
- Macaca_mulatta (7)
- Mus_musculus (101)
- Oryza_sativa (1)

Packages found under Homo_sapiens:

Show All entries Search table:

Package	Maintainer	Title
BSgenome.Hsapiens.NCBI.GRCh38	Bioconductor Package Maintainer	Full genome sequences for Homo sapiens (GRCh38)
BSgenome.Hsapiens.UCSC.hg17	Bioconductor Package Maintainer	Full ge… sapiens (UCSC version hg17)
BSgenome.Hsapiens.UCSC.hg17.masked	Bioconductor Package Maintainer	Full masked genome sequences for Homo sapiens (UCSC version hg17)
BSgenom…		
BSgenom…		
BSgenom…		
BSgenom…		
test3probe	Bioconductor Package Maintainer	Probe sequence data for microarrays of type test3
TxDb.Hsapiens.UCSC.hg18.knownGene	Bioconductor Package Maintainer	Annotation package for TranscriptDb object(s)
TxDb.Hsapiens.UCSC.hg19.knownGene	Bioconductor Package Maintainer	Annotation package for TranscriptDb object(s)
TxDb.Hsapiens.UCSC.hg19.lincRNAsTranscripts	Bioconductor Package Maintainer	Annotation package for TranscriptDb object(s)
u133aaofav2cdf	Bioconductor Package Maintainer	u133aaofav2cdf
u133x3p.db	Bioconductor Package	Affymetrix Human X3P Array … u133x3p)
u133x3pcdf		
u133x3pprobe	Package Maintainer	… for microarrays of type u133x3p

Showing 1 to 196 of 196 entries Previous Next

ヒトゲノムの最新版(hg38)

ヒトゲノムの最新版(hg38)に対応したアノテーションパッケージはまだ提供されていない…

Apr 30 2014

プロモーター解析とRのバージョン周辺

イントロ | 一般 | 配列取得 | プロモーター配列 | BSgenome NEW

BSgenome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。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(
(R.norvegicus)、
わかりますが、生
「ファイル」-「ディレク

1. 利用可能な生物種

```
#必要なパッケー
library(BSgenc

#本番（利用可能
available.genc

#本番（インスト
installed.genc

#後処理（パッケ
installed.genc
```

2. ゼブラフィッシュ("

400MB程度あり

6. インストール済みのヒト("BSgenome.Hsapiens.NCBI.GRCh38")の転写開始点上流配列(1000bp)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のRパッケージです。上流配列情報を含まないのでエラーがでます。

```
out_f <- "hoge6.txt"                    #出力ファイル名を指定してout_fに格納
param1 <- "BSgenome.Hsapiens.NCBI.GRCh38"#パッケージ名を指定
param2 <- "upstream1000"                #上流1000bpを指定

#必要なパッケージをロード
library(param1, character.only=T)       #param1で指定したパッケージの読み込み

#前処理(param1で指定したパッケージ中のオブジェクト名をhogeに統一)
#tmp <- unlist(strsplit(param,
tmp <- ls(paste("package", para
hoge <- eval(parse(text=tmp))
hoge

#本番
tmp <- paste("hoge$", param2, s
fasta <- eval(parse(text=tmp))

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

R Console

```
|   HSCHR19KIR_FH06_BA1_HAP_CTG3_1      HSCHR19KIR_FH08_A_HAP_CTG3_1  $
|   HSCHR19KIR_FH08_BAX_HAP_CTG3_1      HSCHR19KIR_FH13_A_HAP_CTG3_1  $
|   HSCHR19KIR_FH13_BA2_HAP_CTG3_1      HSCHR19KIR_FH15_A_HAP_CTG3_1  $
|   HSCHR19KIR_RP5_B_HAP_CTG3_1
|
| (use the '$' or '[[' operator to access a given sequence)
>
> #本番
> tmp <- paste("hoge$", param2, sep="")   #param2で指定した文字列を含む$
> fasta <- eval(parse(text=tmp))          #文字列tmpをRオブジェクトとし$
 以下にエラー x[[name]] : no such sequence
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", width=50)#fastaの$
 以下にエラー is(x, "XStringSet") :  オブジェクト 'fasta' がありません
>
```

ヒトゲノムの最新版(hg38)パッケージ中には上流配列データが含まれていない。

プロモーター解析とRのバージョン周辺

(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 m

What'

- この
- 201
- 門田
- 参考

- はじ
- 参考
- 過去
- Rの
- サン

- イントロ | 一般 | 配列取得 | ゲノム配列 | 公共DBから (last modified 2014/04/10) NEW
- イントロ | 一般 | 配列取得 | ゲノム配列 | BSgenome (last modified 2014/04/22) 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) NEW
- イントロ | 一般 | 配列取得 | プロモーター配列 | BSgenome (last modified 2014/04/23) NEW
- 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) NEW
- 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 2014/04/02) NEW
- イントロ | 一般 | 配列取得 | トランスクリプトーム配列 | biomaRt(Durinck_2009) (last modified 2013/09/25)

イントロ | 一般 | 配列取得 | プロモーター配列 | GenomicFeatures(Lawrence_2013) NEW

GenomicFeaturesパッケージを主に用いてプロモーター配列(転写開始点近傍配列)を得るやり方を示します。

ここで

Trans

用いて

示してい

1. ヒト("BS

ヒトゲノム

ブジェクト

out_f
param1
param2
param3
param4
param5

#必要な

5. シロイヌナズナ(TAIR10_chr_all.fas)の[上流500塩基, 下流0塩基]のプロモーター配列を取得する場合:

UCSCからはArabidopsisの遺伝子アノテーション情報が提供されていないため、TAIR10_GFF3_genes.gffを予めダウンロードしておき、makeTranscriptDbFromGFF関数を用いてTranscriptDbオブジェクトを作成しています。

イントロ | NGS | 読み込み | FASTA形式 | description行の記述を整形を参考にして、description行の文字列をgffファイルと対応がとれるように変更しています。それでもまだ「以下にエラー value[[3L]](cond) : record 16760 (Chr3:23459805-23460304) was truncated」というエラーに遭遇します。Chr3は23459830 bpしかないこと、gffファイルでも23460304のような存在しない領域を指定しているわけでもないので、理解不能です。

```
in_f1 <- "TAIR10_chr_all.fas"          #入力ファイル名を指定してin_f1に格納(リファレンス配列)
in_f2 <- "TAIR10_GFF3_genes.gff"       #入力ファイル名を指定してin_f2に格納(GFF3またはGTF形式のアノテ
out_f <- "hoge5.txt"                   #出力ファイル名を指定してout_fに格納
param1 <- 500                          #転写開始点上流の塩基配列数を指定
param2 <- 0                            #転写開始点下流の塩基配列数を指定
param3 <- c("Chr1","Chr2","Chr3","Chr4","Chr5","ChrM","ChrC")#置換したい文字列を指定

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

3, 4, 5のいずれもうまくいかない例です