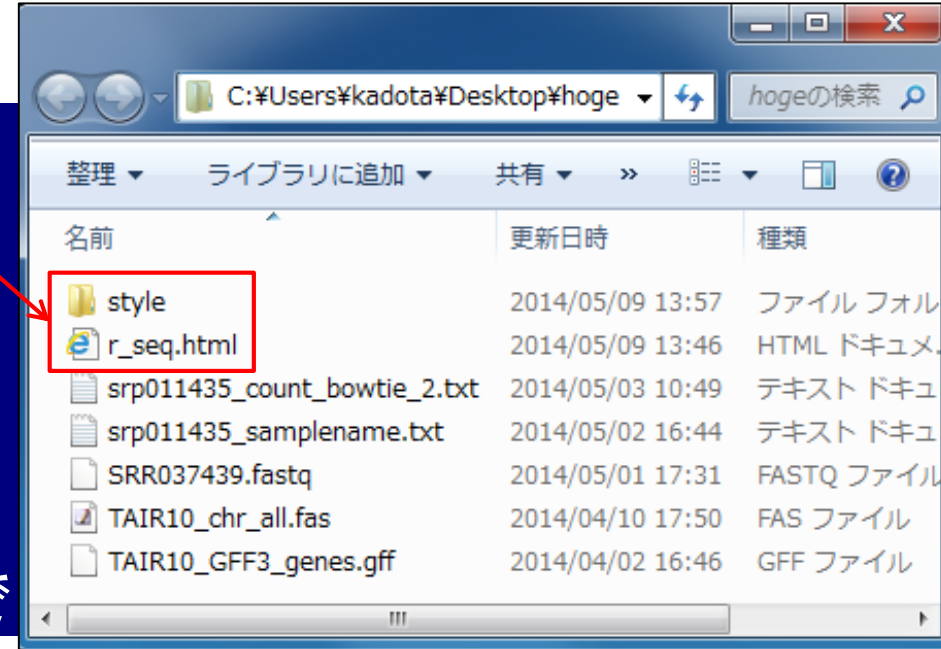


ネット接続できないヒトも、
ダブルクリックでローカルに
r_seq.htmlを起動可能です

実習は、デスクトップ上にある
hogeフォルダの中身が以下
の状態を想定して行います



(Rで)塩基配列解析の利用法： GC含量計算から発現変動解析まで

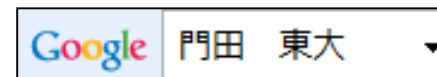
東京大学・大学院農学生命科学研究科

アグリバイオインフォマティクス教育研究プログラム

門田幸二(かどた こうじ)

kadota@iu.a.u-tokyo.ac.jp

<http://www.iu.a.u-tokyo.ac.jp/~kadota/>



参考ウェブページ

The screenshot shows a web browser window with the address bar containing the URL http://www.iu.a.u-tokyo.ac.jp/~kadota/r_seq.html. The page title is "(Rで)塩基配列解析". Below the title is a subtitle: "~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 2014/05/01, since 2010)".

The main content area has a section titled "What's new?" followed by a list of updates:

- このウェブページはフリーソフトRと利用可能なパッケージの多くをインストール済みである前提で記述していますので、[Rのインストールと起動](#)を参考に必要なパッケージのインストールを行ってください。2014年4月22日に記述内容を若干変更しています。
- 2014年9月1日～12日に「バイオインフォマティクス人材育成カリキュラム(次世代シーケンサ)速習コース」を東大農で開催します。近いうちに詳細を公開しますので興味ある方は予定を開きといてください。
- 門田幸二 著 [シリーズ Useful R 第7巻トランスクリプトーム解析](#)刊行(共立出版)。
-
- [参考資料\(講義、講習会、本など\)](#)の項目を追加しました。(2014/05/01) **NEW**
- 2014年06月12日に [NAIST植物グローバル教育プロジェクト・平成26年度ワークショップ「ImageJ+Rハンズオン実習2014」](#) が開催されます。特に門田の部分を受講したい方は2014年4月22日に作成した [より詳細なインストール手順\(Windows版\)](#) を参考にインストールしておいてください。シンプルな [Mac版のインストール手順](#) (by [孫建強氏](#)) もあります。(2014/04/27) **NEW**

At the bottom of the page, there is a list of links:

- [はじめに](#) (last modified 2014/01/30)
- [参考資料\(講義、講習会、本など\)](#) (last modified 2014/05/01) **NEW**
- [過去のお知らせ](#) (last modified 2014/04/21) **NEW**
- [Rのインストールと起動](#) (last modified 2014/04/30) **NEW**
- [サンプルデータ](#) (last modified 2014/04/01)
- [書籍 | について](#) (last modified 2014/04/20) **NEW**

A yellow callout box on the right side of the page contains the text: "(Rで)塩基配列解析の利用法の紹介".

以下の手順通りにRおよび必要なパッケージのインストールが完了しているという前提です。

前提条件

(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 2014/05/01, since 2010)

What's new?

- このウェブページはフリーソフトRと利用可能なので、[Rのインストールと起動](#)を参考にしてください。日に記述内容を若干変更しています。
- 2014年9月1日~12日に「バイオインフォマ東大農」で開催します。近いうちに詳細を公開します。
- 門田幸二 著 [シリーズ Useful R 第7巻](#) [トランプ](#)
- 参考資料(講義、講習会、本など)の項目
- 2014年06月12日に [NAIST植物グローバル実習2014](#) が開催されます。特に門田の [インストール手順\(Windows版\)](#) を参考にしてください (by [孫建強氏](#) もあります。(2014/04/27))

- [はじめに](#) (last modified 2014/01/30)
- [参考資料\(講義、講習会、本など\)](#) (last modified 2014/04/27)
- [過去のお知らせ](#) (last modified 2014/04/27)
- [Rのインストールと起動](#) (last modified 2014/04/27)
- [サンプルデータ](#) (last modified 2014/04/01)
- [書籍](#) について (last modified 2014/04/20)

Rのインストールと起動 NEW

基本的には[こちら](#)または[こちら](#)をご覧ください。

よく分からない人でWindowsユーザーの方は以下を参考にしてください。2014年4月22日に作成したより詳細なインストール手順のPDFは[こちら](#)。

1. Windows release版のインストールの場合:

- [Rのインストーラ](#)を「実行」
- 聞かれるがままに「次へ」などを押してとにかくインストールを完了させる
- Windows Vistaの人**は(パッケージのインストール中に書き込み権限に関するエラーが出るのを避けるために)「コントロールパネル」-「ユーザーアカウント」-「ユーザーアカウント制御の有効化または無効化」で、「ユーザーアカウント制御(UAC)を使ってコンピュータの保護に役立たせる」のチェックをあらかじめ外しておくことを強くお勧めします。
- インストールが無事完了したら、デスクトップに出現する「R3.X.Y(32 bitの場合; XやY中の数値はバージョンによって異なります)」または「R x64 3.X.Y(64 bitの場合)」アイコンをダブルクリックして起動
- 以下を、「R コンソール画面」でコピー&ペーストする。10GB程度のディスク容量を要しますが一番お手軽です。(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)

```
install.packages(available.packages()[,1], dependencies=TRUE)#CRAN中にある全てのパッケージをインストール
source("http://www.bioconductor.org/biocLite.R")#おまじない
biocLite(all_group())#Bioconductor中にある全てのパッケージをインストール
biocLite("BSgenome.Athaliana.TAIR.TAIR9")#Bioconductor中にあるBSgenome.Athaliana.TAIR9
```

- 「コントロールパネル」-「デスクトップのカスタマイズ」-「フォルダオプション」-「表示(タブ)」-「詳細設定」のところで、「登録されている拡張子は表示しない」のチェックを外してください。

[トップページへ](#)



自己紹介

- 2002年3月
 - 東京大学・大学院農学生命科学研究科・応用生命工学専攻 博士課程修了
 - 学位論文:「cDNAマイクロアレイを用いた遺伝子発現解析手法の開発」
(指導教官:清水謙多郎教授)
- 2002/4/1~
 - 産総研・生命情報科学研究センター(CBRC) 産総研特別研究員
 - マイクロアレイ解析手法開発
- 2003/11/1~
 - 放医研・先端遺伝子発現研究センター 研究員
 - 一次元電気泳動波形解析手法開発
- 2005/2/16~
 - 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プログラム
 - マイクロアレイ解析手法開発
 - RNA-seqデータ解析手法開発

(トランスクリプトーム解析周辺の)手法開発系のヒトです



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



ホーム > 教育プログラム > 各講義のページ

各講義のページ

(科目名をクリックすると各講義のページに移動します)

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

カテゴリー	科目名	学期・単位	実施曜日
基礎	1. 生物配列解析基礎		
	生命科学のためのデータベースの利用と基本的な解析手法について講義します。データベースの基礎、配列データベース、機能データベース、ホモロジー検索、モチーフ解析などの基本的な手法について解説します。	夏・1	火曜
	2. ゲノム情報解析基礎		

講義風景(平成26年度)



Contents

■ Rでゲノム解析

- シロイヌナズナゲノムのGC含量計算
 - multi-FASTAファイルの読み込み
 - 関数やオプションの利用法
 - パッケージの説明

■ Rでトランスクリプトーム解析

- シロイヌナズナのRNA-seqデータを一通り解析
 - 公共DBからの生データ取得
 - マッピングおよびカウントデータ取得
 - サンプル間クラスタリング
 - 発現変動遺伝子(DEG)検出



植物グローバルなので...

- 例: シロイヌナズナ (*Arabidopsis thaliana*)
 - ゲノム配列決定 (chr1-5, chrC, chrM)
 - 1番染色体: Theologis et al., *Nature*, **408**: 816-820, 2000
 - 2番染色体: Lin et al., *Nature*, **402**: 761-768, 1999
 - 3番染色体: Salanoubat et al., *Nature*, **408**: 820-822, 2000
 - ...
 - トランスクリプトーム配列 (cDNA配列) 決定
 - アノテーション: Seki et al., *Science*, **296**: 141-145, 2002
 - ...
 - まとめサイト
 - The Arabidopsis Information Resource (TAIR)
 - Lamesch et al., *Nucleic Acids Res.*, **40**: D1202-1210, 2012
 - <http://www.arabidopsis.org/>

	Length	GC contents
chr1	28.76MB	35.80%
chr2	19.60MB	35.80%
chr3	23.17MB	35.40%
chr4	17.40MB	36.02%
chr5	25.95MB	34.50%

Rでゲノム解析が可能です



(Rで)塩基配列解析

~NGS、RNA-seq、ゲノム、トランスクリプトーム、正規化、発現変動、統計、モデル、バイオインフォマティクス~
(last modified 2014/04/10, since 2010)

What's new

- 2014年9月農で開催
- 門田幸二
- 解析を係
- 向かに、
- 置いた補
- 参考資料
- 私の所属
- フォ関連
- 東大以外
- ノム情報
- 興味ある
- 機能解析
- 解析のみ

- イントロ | 一般 | [逆鎖\(reverse\)を取得](#) (last modified 2013/06/14)
- イントロ | 一般 | [2連続塩基の出現頻度情報を取得](#) (last modified 2014/02/07)
- イントロ | 一般 | [3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 | [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 2014/04/01) **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 | ゲノム配列 | 公共DBから **NEW**

- [UCSCの Sequence and Annotation Downloads](#) (Karolchik et al., Nucleic Acids Res., 2014)
 - [ヒト; Human \(H.sapiens\)](#)
 - [ラット; Rat \(R.norvegicus\)](#)
 - [ネコ; Cat \(F.catus\)](#)
 - [ウサギ; Rabbit \(O.cuniculus\)](#)
 - [ニワトリ; Chicken \(G.gallus\)](#)
 - [イヌ; Dog \(C.familiaris\)](#)
 - [ウマ; Horse \(E.caballus\)](#)
 - ...
- [Helix Systems Scientific Databases](#) (アップデートの日付順になっている。RefSeqやESTなど様々なデータベースを一度にみられる)
- イネ: [RAP-DB](#) (Sakai et al., Plant Cell Physiol., 2013)
 - 「ダウンロード」-「Genome assemblies」の [Download](#)。IRGSP-1.0_genome.fasta.gz (116MB程度)の圧縮ファイル。
- シロイヌナズナ: [The Arabidopsis Information Resource \(TAIR\)](#) (Lamesch et al., Nucleic Acids Res., 2012)
 - 「ダウンロード」-「Genes」-「TAIR10 genome release」-「TAIR10 chromosome files」の [TAIR10 chr_all.fas](#) (120MB程度)

TAIR10のゲノム配列ファイル(TAIR10_chr_all.fas)はここからダウンロードしました

02/15/2012 12:00午前	411,136	Ath miRNAs	Konika Chawla 20120215.xls
09/29/2009 12:00午前		ディレクトリ	Gene families
01/19/2013 12:00午前	528,450		Locus Primary Gene Symbol 20130117.txt
02/07/2012 12:00午前		ディレクトリ	OLD
08/23/2011 12:00午前		ディレクトリ	SmallRNAsCarrington
10/24/2013 10:52午後		ディレクトリ	TAIR10 genome release
02/24/2009 12:00午前		ディレクトリ	TAIR6 genome release
08/05/2009 12:00午前		ディレクトリ	TAIR7 genome release
11/30/2009 12:00午前		ディレクトリ	TAIR8 genome release

10/05/2011 12:00	08/22/2012 12:00午前	5,545	README TAIR10.txt
08/23/2011 12:00	08/22/2012 12:00午前	3,964,120	TAIR10-Subcellular Predictions.xlsx
08/23/2011 12:00	08/23/2011 12:00午前	ディレクトリ	TAIR10 NCBI mapping files
08/23/2011 12:00	08/23/2011 12:00午前	792,935	TAIR10 TAIRAccessionID AGI mapping.txt
02/07/2012 12:00	04/13/2012 12:00午前	1,868,951	TAIR10 TAIRlocusaccessionID AGI mapping.txt
01/30/2013 12:00	08/23/2011 12:00午前	47	TAIR10 blastsets
10/24/2013 10:5	08/23/2011 12:00午前	ディレクトリ	TAIR10 chromosome files
	08/27/2010 12:00午前	2,608,703	TAIR10 domain architectures.txt
	01/16/2013 12:00午前	25,396,877	TAIR10 functional descriptions
	11/23/2010 12:00午前	25,396,966	TAIR10 functional descriptions.bk
	10/24/2013 10:51午後	25,874,762	TAIR10 functional descriptions 20130831.txt
	08/23/2011 12:00午前	ディレクトリ	TAIR10 gene confidence ranking
	08/23/2011 12:00午前	ディレクトリ	TAIR10 gene lists
	08/23/2011 12:00午前	ディレクトリ	TAIR10 gene transcript associations
	11/18/2010 12:00午前	30	TAIR10 gff3
	11/23/2010 12:00午前	2,053,133	TAIR10 locushistory.txt
	12/07/2010 12:00午前		
	08/23/2011 12:00午前	26,977,690	NCBI Chr1.tbl
	11/23/2010 12:00午前	15,361,650	NCBI Chr2.tbl
		18,699,824	NCBI Chr3.tbl
		14,999,724	NCBI Chr4.tbl
		22,424,246	NCBI Chr5.tbl
		44	TAIR10 chr all.fas

	Length	GC contents
chr1	28.76MB	35.80%
chr2	19.60MB	35.80%
chr3	23.17MB	35.40%
chr4	17.40MB	36.02%
chr5	25.95MB	34.50%

11/10/2010 12:00午前	26,977,690	NCBI Chr1.tbl
11/10/2010 12:00午前	15,361,650	NCBI Chr2.tbl
11/10/2010 12:00午前	18,699,824	NCBI Chr3.tbl
11/10/2010 12:00午前	14,999,724	NCBI Chr4.tbl
11/10/2010 12:00午前	22,424,246	NCBI Chr5.tbl
08/23/2011 12:00午前	44	TAIR10 chr all.fas

```

TAIR10_chr_all.fas x
>1 CHROMOSOME dumped from ADB: Feb/3/09 16:9; last updated: 2009-02-02↓
CCCTAAACCCTAAACCCTAAACCCTAAACCTCTGAATCCTTAATCCCTAAATCCCTAAATCTTTAAATCCTACATCCAT
GAATCCCTAAATACCTAATTCCCTAAACCCGAAACCGGTTTCTCTGGTTGAAAATCA
ATCGTTTTTATGTAATTGCTTATTGTTGTGTGTAGATTTTTTAAAAATATCATTG
TGTGGTTTTCTTTCCTTCACTTAGCTATGGATGGTTTATCTTCATTTGTTATATTGG
CATTTGGGAATGTGAGTCTCTTATTGTAACCTTAGGGTTGGTTTATCTCAAGAATCT
TGTTTGGACATTTATTGTCATTCTTACTCCTTTGTGGAAATGTTTGTCTATCAATT
TAGTTGTAGGGATGAAGTCTTCTTCTGTTGTTGTTAGCCTTGTCACTCATCTCTCAATGATATGGGATGGTCTTTAG
  
```

115 MB (121,183,059 バイト), 1,514,793 行。 Text 2行, 80桁

TAIR10のゲノム配列ファイル(TAIR10_chr_all.fas)はこんな感じです

multi-FASTAファイル?!

FASTAフォーマット [編集]

FASTAでは、シーケンスデータの記述形式としてFASTAフォーマットという形式を使う。FASTAフォーマットはブレンテキストである。1つのシーケンスのデータは、">"で始まる1行のヘッダ行と、2行目以降の実際のシーケンス文字列で構成される。ヘッダ行では、">"の次にシーケンスデータを識別するための文字列を記述し、続けてそのシーケンスデータを説明する文字列を記述する(両方とも省略してよい)。ヘッダ行の">"と識別文字列の間にスペースを入れてはいけない。FASTAフォーマットの全ての行は、80文字未満とすることが推奨される。">"で始まる別の行が出現すると、そこでシーケンスデータが区切られ、別のシーケンスデータが始まる。

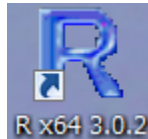
FASTAファイルフォーマットの例を示す。

```
>gi|5524211|gb|AAD44166.1| cytochrome b [Ele
LCLYTHIGRNIYYGSYLYSETWNTGIMLLLITMATAFMGYVLPW
EWIWGGFSVDKATLNRFFAFHFILPFTMVALAGVHLTFLHETGS
LLILILLLLLLLALLSPDMLGDPDNHMPADPLNTPLHIKPEWYFI
GLMPFLHTSKHRSMMLRPLSQALFWTLTMDLLTLTWIGSQPVEY
IENY
```

```
>Chr1 CHROMOSOME dumped from ADB: Jun/20/09 14:53; last updated: 2009-02-02
CCCTAAACCCTAAACCCTAAACCCTAAACCCTCTGAATCCTTAATCCCTAAATCCCTAAAT
CTTTAAATCCTACATCCATGAATCCCTAAATACCTAATTCCTAAACCCGAAACCGGTTT
CTCTGGTTGAAAATCATTGTGTATATAATGATAATTTTATCGTTTTATGTAATTGCTTA
...
>Chr2 CHROMOSOME dumped from ADB: Jun/20/09 14:54; last updated: 2009-02-02
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
...
>Chr3 CHROMOSOME dumped from ADB: Jun/20/09 14:54; last updated: 2009-02-02
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNCCCTAAACCCTAAACCCTAA
ACCCTAAACCCTAAACCCTAAACCCTAAACCCTAAATCCATAAATCCCTAAACCATAAT
...
>Chr4 CHROMOSOME dumped from ADB: Jun/20/09 14:54; last updated: 2009-02-02
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
...
>Chr5 CHROMOSOME dumped from ADB: Jun/20/09 14:54; last updated: 2009-02-02
TACCATGTACCCTCAACCTTAAAACCCTAAAACCTATACTATAAATCTTTAAACCTA
CTCTAAACCATAGGGTTTGTGAGTTTGCATAAAGTGTACAGTATAAGTGTCTTCTAACA
TGAGTTTGCATAAGAGTCTCGACTATGTGTTTGTTCAAAAGTGACGTAAGTGTTTAGA
...
```

「>のヘッダ行、塩基またはアミノ酸配列」が複数 (multi) 個からなるファイルのこと

Rの起動



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

[以前にセーブされたワークスペースを復帰します]

> |
```

デスクトップにあるhogeフォルダ中のファイルを解析

作業ディレクトリの変更

「Windows(C:)」となっている場合もあるが、気にしない

この部分はひとそれぞれ

RGui (64-bit)

ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ

R コードのソースを読み込み...
新しいスクリプト
スクリプトを開く...
ファイルの表示...
作業スペースの読み込み...
作業スペースの保存...
履歴の読み込み...
履歴の保存...
ディレクトリの変更... ①
印刷...
ファイルを保存...
終了

作業ディレクトリの変更
C:\

コンピューター
ローカル ディスク (C:) ②
SD Card (E:)

空き領域: 280 GB
合計サイズ: 453 GB

フォルダー(F): ローカル ディスク (C:)

新しいフォルダーの作成(N) OK キャンセル

'help.start()'でHTMLブラウザによるヘルプ
'q()'と入力すればRを終了します。
> |

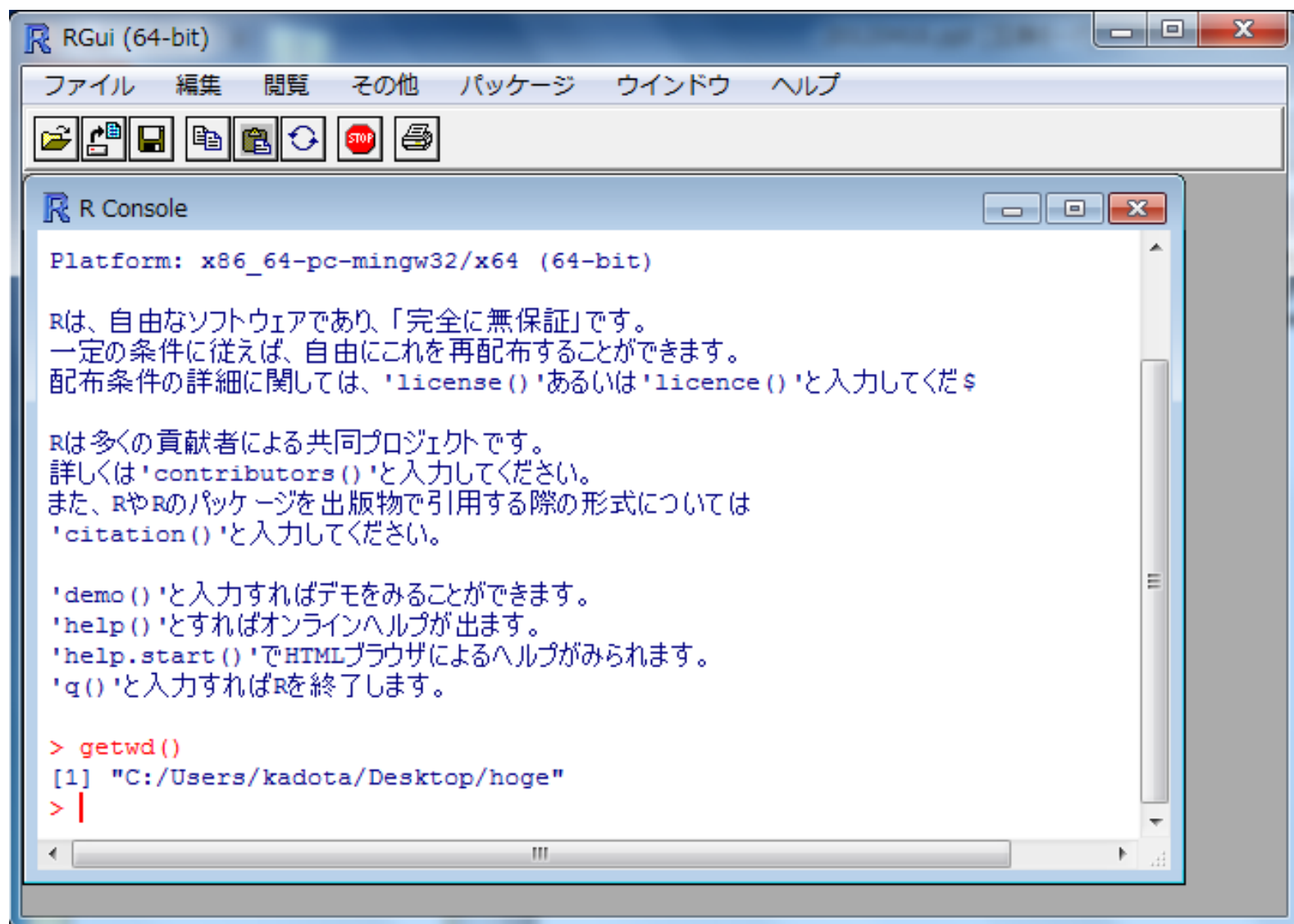
作業ディレクトリの変更
C:\Users\kadota\Desktop\hoge

Users ③
Default
kadota ④
AppData
Dropbox
Roaming
アドレス帳
お気に入り
ダウンロード
デスクトップ ⑤
hoge ⑥

フォルダー(F): hoge

新しいフォルダーの作成(N) OK ⑦ キャンセル

getwd() と打ち込んで確認



```
Platform: x86_64-pc-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$

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。

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

基本はコピー

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。

4. 120MB程度のシロイヌナズナゲノムのmulti-FASTAファイル(TAIR10_chr_all.fas)の場合:

```
in_f <- "TAIR10_chr_all.fas" #入力ファイル名を指定してin_fに格納
out_f <- "hoge4.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
library(Biostrings)
```

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

```
#本番
hoge <- alphabetFreque
CG <- rowSums(hoge[,2:
ACGT <- rowSums(hoge[,
GC_content <- CG/ACGT*
```

```
#ファイルに保存
tmp <- cbind(names(fas
colnames(tmp) <- c("de
write.table(tmp, out_f
```

- 切り取り(T)
- コピー(C) ①
- 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 (Windows I)
- すべてのアクセラレータ
- Send to OneNote

Platform: x86_64-pc-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()'と入力してください。

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を見ます。'q()'と入力すればRを終了します。

```
> getwd()
[1] "C:/Users/kadota/Desktop/hoge4"
> |
```

コピー	Ctrl+C
ペースト ②	Ctrl+V
コマンドのみペースト	
コピー&ペースト	Ctrl+X
ウインドウの消去	Ctrl+L
全て選択	
<input checked="" type="checkbox"/> バッファに出力	Ctrl+W
ウインドウを常にトップに置く	

①一連のコマンド群をコピーして
②R Console画面上でペースト

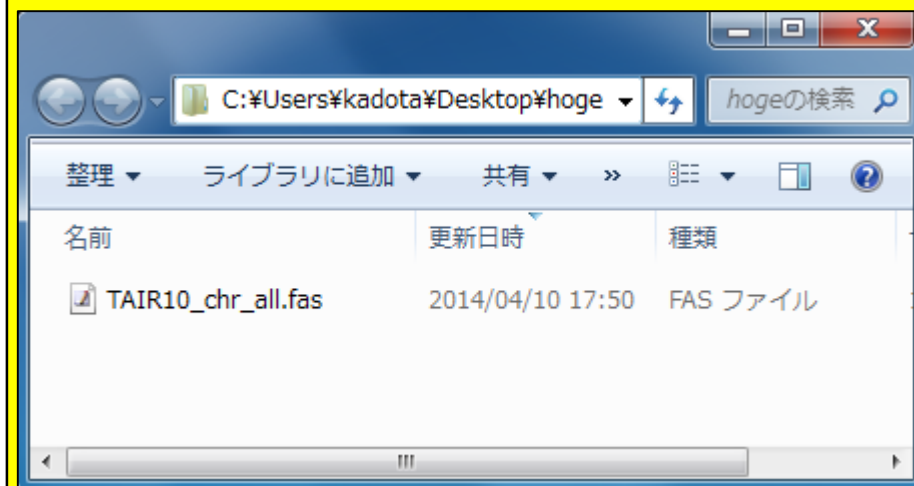
実行結果

```
R Console
> #入力ファイルの読み込み
> fasta <- readDNAStringSet(in_f, format="fasta") #in_fで指定した$
>
> #本番
> hoge <- alphabetFrequency(fasta)           #A,C,G,T,..の数を各配列$
> CG <- rowSums(hoge[,2:3])                 #C,Gの総数を計算してCGに$
> ACGT <- rowSums(hoge[,1:4])              #A,C,G,Tの総数を計算して$
> GC_content <- CG/ACGT*100                #%GC含量を計算してGC_con$
>
> #ファイルに保存
> tmp <- cbind(names(fasta), CG, ACGT, width(fasta), GC_content) # $
> colnames(tmp) <- c("description", "CG", "ACGT", "Length", "%GC_$
> write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=$
> |
```

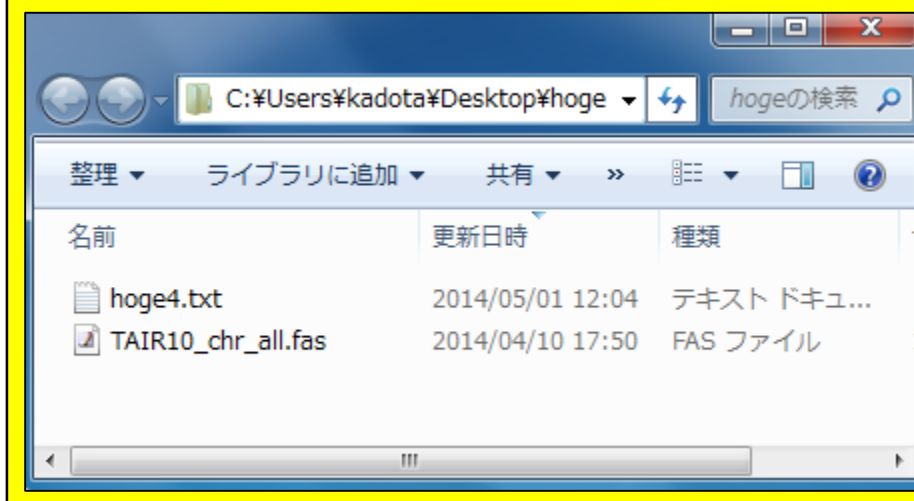
出力: hoge4.txt

	A	B	C	D	E
1	description	CG	ACGT	Length	%GC_contents
2	1 CHROMOSOME	10856525	30263312	30427671	35.874
3	2 CHROMOSOME	7063739	19695728	19698289	35.864
4	3 CHROMOSOME	8521037	23453853	23459830	36.331
5	4 CHROMOSOME	6727440	18582024	18585056	36.204
6	5 CHROMOSOME	9691012	26965224	26975502	35.939
7	mitochondria CHR	164270	366924	366924	44.769
8	chloroplast CHROM	56066	154478	154478	36.294

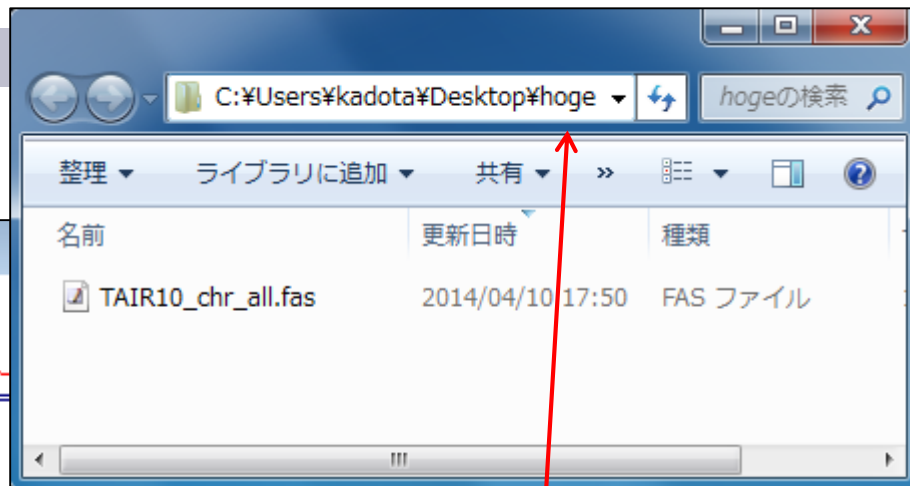
実行前のhogeフォルダ



実行後のhogeフォルダ



ありがちなミス1

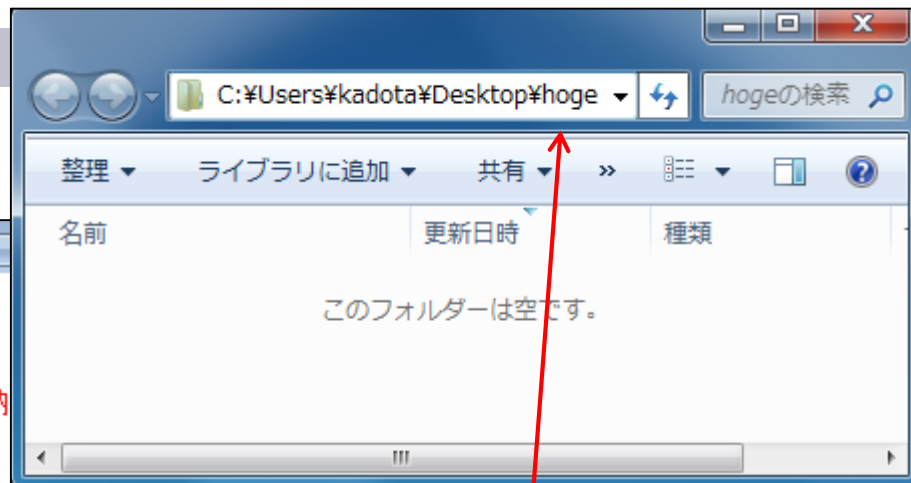


R Console

```
> #入力ファイルの読み込み
> fasta <- readDNASTringSet(in_f, format="fasta")#in_fで指定し
以下にエラー .Call2("new_input_ExternalFilePtr", fp, PACKAGE =
cannot open file 'TAIR10_chr_all.fas'
>
> #本番
> hoge <- alphabetFrequency(fasta)           #A,C,G,T,..の数を各配列ごとにカウン$
以下にエラー alphabetFrequency(fasta) :
引数 'x' の評価中にエラーが起きました (関数 'alphabetFrequency' に対する$
> CG <- rowSums(hoge[,2:3])                  #C,Gの総数を計算してCGに格納
以下にエラー is.data.frame(x) : オブジェクト 'hoge' がありません
> ACGT <- rowSums(hoge[,1:4])                #A,C,G,Tの総数を計算してACGTに格納
以下にエラー is.data.frame(x) : オブジェクト 'hoge' がありません
> GC_content <- CG/ACGT*100                  #GC含量を計算してGC_contentに格納
エラー: オブジェクト 'CG' がありません
>
> #ファイルに保存
> tmp <- cbind(names(fasta), CG, ACGT, width(fasta), GC_content)#保存したい$
以下にエラー eval(expr, envir, enclos) : オブジェクト 'fasta' がありません
> colnames(tmp) <- c("description", "CG", "ACGT", "Length", "%GC_contents")#$
以下にエラー colnames(tmp) <- c("description", "CG", "ACGT", "Length", "%GC$
オブジェクト 'tmp' がありません
> write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F, col.name$
以下にエラー is.data.frame(x) : オブジェクト 'tmp' がありません
> getwd()
[1] "C:/Users/kadota/Desktop"
> |
```

作業ディレクトリの変更を忘れて
いるため、入力ファイルの読
み込み段階でエラーとなる

ありがちなミス2



必要な入力ファイルが作業ディレクトリ中に存在しない…

```
R Console
> getwd()
[1] "C:/Users/kadota/Desktop/hoge"
> in_f <- "TAIR10_chr_all.fas"      #入力ファイル名を指定してin_fに格納
> out_f <- "hoge4.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$
以下にエラー .Call2("new_input_ExternalFilePtr", fp, PACKAGE = "Biostrings"$
cannot open file 'TAIR10_chr_all.fas'
>
> #本番
> hoge <- alphabetFrequency(fasta)  #A,C,G,T,..の数を各配列ごとにカウン$
以下にエラー alphabetFrequency(fasta) :
引数 'x' の評価中にエラーが起きました (関数 'alphabetFrequency' に対する$
>
> CG <- rowSums(hoge[,2:3])          #C,Gの総数を計算してCGに格納
以下にエラー is.data.frame(x) : オブジェクト 'hoge' がありません
> ACGT <- rowSums(hoge[,1:4])        #A,C,G,Tの総数を計算してACGTに格納
以下にエラー is.data.frame(x) : オブジェクト 'hoge' がありません
> GC_content <- CG/ACGT*100          #%GC含量を計算してGC_contentに格納
エラー: オブジェクト 'CG' がありません
>
> #ファイルに保存
> tmp <- cbind(names(fasta), CG, ACGT, width(fasta), GC_content) #保存したい$
以下にエラー eval(expr, envir, enclos) : オブジェクト 'fasta' がありません
> colnames(tmp) <- c("description", "CG", "ACGT", "Length", "%GC_contents")#$
以下にエラー colnames(tmp) <- c("description", "CG", "ACGT", "Length", "%GC$
オブジェクト 'tmp' がありません
> write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F, col.names$
以下にエラー is.data.frame(x) : オブジェクト 'tmp' がありません
> |
```

ありがちなミス3

	A	B	C	D	E
1	description	CG	ACGT	Length	%GC_contents
2	1 CHROMOSOME	10856525	30263312	30427671	35.87355
3	2 CHROMOSOME	7063739	19695728	19698289	35.86432
4	3 CHROMOSOME	8521037	23453853	23459830	36.33108
5	4 CHROMOSOME	6727440	18582024	18585056	36.20402
6	5 CHROMOSOME	9691012	26965224	26975502	35.93893
7	mitochondria CHF	164270	366924	366924	44.76949
8	chloroplast CHRC	56066	154478	154478	36.29384

R Console

```
> #入力ファイルの読み込み
> fasta <- readDNASTringSet(in_f, format="fasta")#in_fで
>
> #本番
> hoge <- alphabetFrequency(fasta)           #A,C,G,T,..の数
> CG <- rowSums(hoge[,2:3])                 #C,Gの総数を計算
> ACGT <- rowSums(hoge[,1:4])              #A,C,G,Tの総数を計算
> GC_content <- CG/ACGT*100                 #%GC含量を計算してGC_contentに格納
>
> #ファイルに保存
> tmp <- cbind(names(fasta), CG, ACGT, width(fasta), GC_content)#保存したい$
> colnames(tmp) <- c("description", "CG", "ACGT", "Length", "%GC_contents")#$
> write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F, col.names=F)
以下にエラー file(file, ifelse(append, "a", "w")) :
  コネクションを開くことができません
追加情報: 警告メッセージ:
In file(file, ifelse(append, "a", "w")) :
  ファイル 'hoge4.txt' を開くことができません: Permission denied
> |
```

出力予定のファイル名と同じものを別のプログラムで開いているため最後のwrite.table関数のところでエラーが出る

ありがちなミス4

```
R Console  
$ed, append, as.data.frame, as.vector, cbind,  
$.call, duplicated, eval, evalq, Filter, Fin  
$.sorted, lapply, Map, mapply, match, mge  
$.pmax.int, pmin, pmin.int, Position, rank,  
$.int, rownames, sapply, setdiff, sort, table  
$.e, unlist  
  
$-ジ IRanges をロード中です  
$-ジ XVector をロード中です  
  
$読み込み  
$SNAStrngSet(in_f, format="fasta")#in_fで指定したファイルの読み込み  
  
$SetFrequency(fasta) #A,C,G,T,..の数を各配列ごとにカウントした結果をhog$  
$Shoge[,2:3]) #C,Gの総数を計算してCGに格納  
$s(hoge[,1:4]) #A,C,G,Tの総数を計算してACGTに格納  
$SCG/ACGT*100 #%GC含量を計算してGC_contentに格納  
  
$  
$ames(fasta), CG, ACGT, width(fasta), GC_content)#保存したい情報をtmpに格納  
$<- c("description", "CG", "ACGT", "Length", "%GC_contents")#列名を付与  
$p, out_f, sep="\t", append=F, quote=F, row.names=F, col.names=T)#tmpの中 |
```

実行スクリプトをコピーする際、最後の行のところで改行を含まずにR Console画面上でペーストしたため、最後のコマンドが実行されない(出力ファイルが生成されない)

```
tmp <- cbind(names(fasta), CG, ACGT, width(fasta), GC_content)  
colnames(tmp) <- c("description", "CG", "ACGT", "Length", "%GC_contents")  
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F, col.names=T, tmpの中身を指定したフ
```

0 chr all.fas)の場合:

- 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 (Windows Live Hotmail)
- すべてのアクセラレータ
- Send to OneNote

をhogelに格納

に格納

与

Rで配列長とGC含量計算

出力: [hoge4.txt](#)

原著論文中の数値

description	CG	ACGT	Length	%GC_contents
1 CHROMOSOME	10856525	30263312	30427671	35.874
2 CHROMOSOME	7063739	19695728	19698289	35.864
3 CHROMOSOME	8521037	23453853	23459830	36.331
4 CHROMOSOME	6727440	18582024	18585056	36.204
5 CHROMOSOME	9691012	26965224	26975502	35.939
mitochondria CHRO	164270	366924	366924	44.769
chloroplast CHROM	56066	154478	154478	36.294

	Length	GC contents
chr1	28.76MB	35.80%
chr2	19.60MB	35.80%
chr3	23.17MB	35.40%
chr4	17.40MB	36.02%
chr5	25.95MB	34.50%

ちゃんと似た結果が得られています

詳細を説明

出力: **hoge4.txt**

description	CG	ACGT	Length	%GC_contents
1 CHROMOSOME	10856525	30263312	30427671	35.874
2 CHROMOSOME	7063731	19695728	19698289	35.864
3 CHROMOSOME	8521047	23453853	23459830	36.331
4 CHROMOSOME	6727440	18582024	18585056	36.204
5 CHROMOSOME	9671012	26965224	26975502	35.939
mitochondria CHR	64270	366924	366924	44.769
chloroplast CHRO	56066	154478	154478	36.294

4. 120MB程度のシロイヌナズナゲノムのmulti-FASTAファイル(TAIR10_chr_all.fas)の場合:

```

in_f <- "TAIR10_chr_all.fas"
out_f <- "hoge4.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, format="fasta")#in_fで指定したファイルの読み込み
fasta

#本番
hoge <- alphabetFrequency(fasta)
CG <- rowSums(hoge[,2:3])
ACGT <- rowSums(hoge[,1:4])
GC_content <- CG/ACGT*100

#ファイルに保存
tmp <- cbind(names(fasta), CG, ACGT, width(fasta), GC_content)#保存したい情報をtmpに格納
colnames(tmp) <- c("description", "CG", "ACGT", "Length", "%GC_contents")#列名を付与
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F, col.names=T)#tmpの
    
```

#入力ファイル名を指定してin_fに格納
#出力ファイル名を指定してout_fに格納

#パッケージの読み込み

入力と出力ファイル名を指定しているところ

#A,C,G,T,..の数を各配列ごとにカウントした結果をhogeに格納
#C,Gの総数を計算してCGに格納
#A,C,G,Tの総数を計算してACGTに格納
#%GC含量を計算してGC_contentに格納

詳細を説明

4. 120MB程度のシロイヌナズナゲノムのmulti-FASTAファイル(TAIR10_chr_all.fas)の場合:

```

in_f <- "TAIR10_chr_all.fas"
out_f <- "hoge4.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, format="fasta")
fasta

#本番
hoge <- alphabetFrequency(fasta)
CG <- rowSums(hoge[,2:3])
ACGT <- rowSums(hoge[,1:4])
GC_content <- CG/ACGT*100

#ファイルに保存
tmp <- cbind(names(fasta), CG, ACGT, width(fasta), GC_content)
colnames(tmp) <- c("description", "CG", "ACGT", "Length", "%GC_contents")
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F, col.names=T)

```

#入力ファイル

#出力ファイル

#パッケージ

#in_fで指定したファイルの読み込み
#確認してるだけです

#A,C,G,T,..の数を各配列ごとにカウントした結果をhogeに格納
#C,Gの総数を計算してCGに格納
#A,C,G,Tの総数を計算してACGTに格納
#%GC含量を計算してGC_contentに格納

GC含量計算をしたいときにはBiostringsというパッケージを読み込む必要があります。この作業を行っておかないと、例えば、multi-FASTAファイルを読み込むためのreadDNASTringSet関数を利用できません。

4. 120MB程度のシロイヌナズナゲノムのmulti-FASTAファイル(TAIR10 chr_all.fas)の場合:

```
in_f <- "TAIR10_chr_all.fas" #入力ファイル名を指定してin_fに格納
out_f <- #出力ファイル名を指定してout_fに格納

#必要なパッケージの読み込み
library("IRanges")
library("XVector")

#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を確認してるだけです

#本番実行
hoge <- countGC(fasta) #A,C,G,T,..の数を各配列ごとにカウントした結果をhogeに格納
CG <- roR::CG(fasta) #A,C,Gの総数を計算してCGに格納

#ファイル名を列挙
tmp <- colnames(fasta)
write.table(tmp, "tmp.txt")
```

readDNASTringSet関数を用いて…
①in_fで指定した入力ファイルを
②fasta形式で読み込んだ結果を
③fastaというオブジェクト名で格納
しています

```
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

> #③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を確認してるだけです
> fasta
A DNASTringSet instance of length 7

      width seq
[1] 30427671 CCCTAAACCCTAAACCCTAA...TTAGGGTTTAGGGTTTAGGG 1 CHROMOSOME dump...
[2] 19698289  NNNNNNNNNNNNNNNNNNNNNNN...TTAGGGTTTAGGGTTTAGGG 2 CHROMOSOME dump...
[3] 23459830  NNNNNNNNNNNNNNNNNNNNNNN...AAACCCTAAACCCTAAACCC 3 CHROMOSOME dump...
[4] 18585056  NNNNNNNNNNNNNNNNNNNNNNN...TTTAGGGTTTAGGGTTTAGG 4 CHROMOSOME dump...
[5] 26975502  TATACCATGTACCCTCAACC...GGATTTAGGGTTTTTAGATC 5 CHROMOSOME dump...
[6]   366924  GGATCCGTTTCGAAACAGGTT...GAATGGAAACAAACCGGATT mitochondria CHRO...
[7]   154478  ATGGGCGAACGACGGGAATT...ATAACTTGGTCCCGGGCATC chloroplast CHROM...

> |
```

詳細を説明

```

in_f <- "TAIR10_chr_all.fas"      #入力ファイル名を指定
out_f <- "hoge4.txt"             #出力ファイル名を指定

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

```

#入力ファイル
fasta <- readDNASTringSet(in_f,
                           format="fasta")
fasta
    
```

fastaオブジェクトは確かに multi-FASTAファイル中の情報を適切に読み込めている

```

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
>
> #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
A DNASTringSet instance of length 7
      width seq
[1] 30427671 CCCTAAACCCTAAACCCTAA...TTAGGGTTTAGGGTTTAGGG 1 CHROMOSOME dump...
[2] 19698289  NNNNNNNNNNNNNNNNNNNNNNN...TTAGGGTTTAGGGTTTAGGG 2 CHROMOSOME dump...
[3] 23459830  NNNNNNNNNNNNNNNNNNNNNNN...AAACCCTAAACCCTAAACCC 3 CHROMOSOME dump...
[4] 18585056  NNNNNNNNNNNNNNNNNNNNNNN...TTTAGGGTTTAGGGTTTAGG 4 CHROMOSOME dump...
[5] 26975502  TATACCATGTACCCTCAACC...GGATTTAGGGTTTTTAGATC 5 CHROMOSOME dump...
[6]   366924  GGATCCGTTTCGAAACAGGTT...GAATGGAAACAAACCGGATT mitochondria CHRO...
[7]   154478  ATGGGCGAACGACGGGAATT...ATAACTTGGTCCCGGGCATC chloroplast CHROM...
> |
    
```

色についての説明

(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 2014/04/30, since 2010)

What's new?

- このウェブページはフリーソフトRと利用可能なパッケージの多くをインストール済みである前提で記述していますので、[Rのインストールと起動](#)を参考に必要なパッケージのインストールを行ってください。2014年4月22日に記述内容を若干変更しています。(2014/04/22) **NEW**
- 2014年06月12日に[NAIST植物グローバル教育プロジェクト・平成26年度ワークショップ「ImageJ+Rハンズオン実習2014」](#)が開催されます。特に門田の部分を受講したい方は2014年4月22日に作成した[より詳細なインストール手順\(Windows版\)](#)を参考にインストールしておいてください。シンプルな[Mac版のインストール手順](#)(by 孫建強氏)もあります。(2014/04/27) **NEW**
- 2014年9月1日～12日に「バイオインフォマティクス人材育成カリキュラム(次世代シーケンサ)速習コース」を東大農で開催します。近いうちに詳細を公開しますので興味ある方は予定を開きといてください。(2014/04/05) **NEW**
- 門田幸二 著 [シリーズ Useful R 第7巻トランスクリプトーム解析](#)刊行(共立出版)。(2014/04/10) **NEW**
- [参考資料\(講義、講習会、本など\)](#)の項目を追加しました。(2014/04/10) **NEW**

このページ内で用いる色についての説明:

コメント

特にやらなくてもいいコマンド

プログラム実行時に目的に応じて変更すべき箇所

- [はじめに](#) (last modified 2014/01/30)
- [参考資料\(講義、講習会、本など\)](#) (last modified 2014/04/10)
- [過去のお知らせ](#) (last modified 2014/04/10)
- [Rのインストールと起動](#) (last modified 2014/04/22)
- [サンプルデータ](#) (last modified 2014/04/27)
- [書籍 11について](#) (last modified 2014/04/17)
- 書籍 | トランスクリプトーム解析 | [2.3.1 リファレンス配列](#) (last modified 2014/04/16) **NEW**
- 書籍 | トランスクリプトーム解析 | [2.3.2 アンテーション情報](#) (last modified 2014/04/17) **NEW**
- 書籍 | トランスクリプトーム解析 | [2.3.3 マッピング\(準備\)](#) (last modified 2014/04/17) **NEW**
- 書籍 | トランスクリプトーム解析 | [2.3.4 マッピング\(本番\)](#) (last modified 2014/04/17) **NEW**

[トップページへ](#)

色についての説明

このページ内で用いる色についての説明:

コメント

特にやらなくてもいいコマンド

プログラム実行時に目的に応じて変更すべき箇所

```
in_f <- "TAIR10_chr_all.fas"
out_f <- "hoge4.txt"
```

#入力ファイル
#出力ファイル

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

#パッケージの読み込み

```
#入力ファイル
fasta <- readDNASTringSet(
fasta
```

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
>
> #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
A DNASTringSet instance of length 7
      width seq
[1] 30427671 CCCTAAACCCTAAACCCTAA...TTAGGGTTTAGGGTTTAGGG 1 CHROMOSOME dump...
[2] 19698289 NNNNNNNNNNNNNNNNNNNNNNN...TTAGGGTTTAGGGTTTAGGG 2 CHROMOSOME dump...
[3] 23459830 NNNNNNNNNNNNNNNNNNNNNNN...AAACCCTAAACCCTAAACCC 3 CHROMOSOME dump...
[4] 18585056 NNNNNNNNNNNNNNNNNNNNNNN...TTTAGGGTTTAGGGTTTAGG 4 CHROMOSOME dump...
[5] 26975502 TATACCATGTACCCTCAACC...GGATTTAGGGTTTTTAGATC 5 CHROMOSOME dump...
[6] 366924 GGATCCGTTTCGAAACAGGTT...GAATGGAAACAAACCGGATT mitochondria CHRO...
[7] 154478 ATGGGCGAACGACGGGAATT...ATAACTTGGTCCCGGGCATC chloroplast CHROM...
> |
```

単に確認しているだけなので灰色になっている

浮かんでくる疑問

FASTA形式以外にどんな形式を読み込めるの
だろう？FASTQ形式は
読み込めるの？

```
in_f <- "TAIR10_chr_all.fas"
out_f <- "hoge4.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, format="fasta")#in_fで指定し
fasta
```

#入力ファイル名を指定
#出力ファイル名を指定

#パッケージの読み込み

#確認してるだけです

DNA配列以外にど
んな配列を読み込
めるのだろう？アミノ
酸配列は？



関数の使用法について

```
R Console  
> ?readDNAStringSet  
starting httpd help serv  
> |
```

```
http://127.0.0.1:24088/library/Biostrings R Documentation  
XStringSet-io {Biostrings}  
Read/write an XStringSet object from/to a file  
Description  
Functions to read/write an XStringSet object from/to a file.  
Usage  
## Read FASTA (or FASTQ) files in an XStringSet object:  
readBStringSet(filepath, format="fasta",  
               nrec=-1L, skip=0L, seek.first.rec=FALSE, use.names=TRUE)  
readDNAStringSet(filepath, format="fasta",  
                 nrec=-1L, skip=0L, seek.first.rec=FALSE, use.names=TRUE)  
readRNAStringSet(filepath, format="fasta",  
                 nrec=-1L, skip=0L, seek.first.rec=FALSE, use.names=TRUE)  
readAAStringSet(filepath, format="fasta",  
                nrec=-1L, skip=0L, seek.first.rec=FALSE, use.names=TRUE)  
## Extract basic information about FASTA (or FASTQ) files  
## without actually loading the sequence data:  
first.rec=FALSE, use.names=TRUE,  
kip=0L, seek.first.rec=FALSE)  
FASTA (or FASTQ) file:  
FALSE,  
compress=FALSE, compression_level=NA, format="fasta", ...)  
## Serialize an XStringSet object:
```

- ・「?関数名」で使用方法を記したウェブページが開く
- ・ページの下のほうに、(大抵の場合)使用例が掲載されている
- ・使用方法既知の関数のマニュアルをいくつか読んで、慣れておく

関数の使用法について

XStringSet-io {Biostrings}

R Documentation

Read/write an XStringSet object from/to a file

Description

・FASTQファイルは読み込めそうだ
・readAAStringSet関数を用いれば
アミノ酸配列を読み込めそうだ

Functions to read/write an [XStringSet](#) object from/to a file.



Usage

```
## Read FASTA (or FASTQ) files in an XStringSet object:  
readBStringSet(filepath, format="fasta",  
               nrec=-1L, skip=0L, seek.first.rec=FALSE, use.names=TRUE)  
readDNAStringSet(filepath, format="fasta",  
                 nrec=-1L, skip=0L, seek.first.rec=FALSE, use.names=TRUE)  
readRNAStringSet(filepath, format="fasta",  
                 nrec=-1L, skip=0L, seek.first.rec=FALSE, use.names=TRUE)  
readAAStringSet(filepath, format="fasta",  
                nrec=-1L, skip=0L, seek.first.rec=FALSE, use.names=TRUE)
```

関数の使用法について

Arguments

`filepath`

A character vector (of arbitrary length when reading, of length 1 when writing) containing the path(s) to the file(s) to read or write. Reading files in gzip format (which usually have the '.gz' extension) is supported.

FASTQファイルは読み込めそうだ

Note that special values like "" or "|cmd" (typically supported by other I/O functions in R) are not supported here. Also `filepath` cannot be a connection.

`format`

Either "fasta" (the default) or "fastq".

`nrec`

Single integer. The maximum of number of records to read in. Negative values are ignored.

`skip`

Single non-negative integer. The number of records of the data file(s) to skip before beginning to read in records.

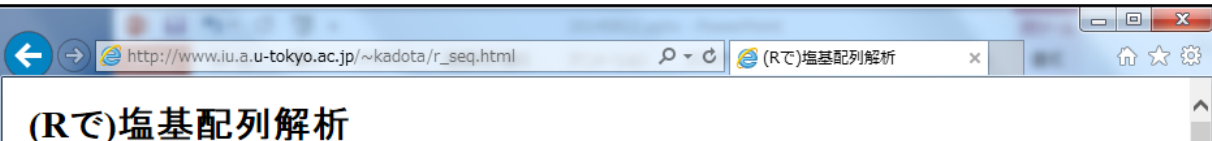
`seek.first.rec`

TRUE or FALSE (the default). If TRUE, then the reading function starts by setting the file position indicator at the beginning of the first line in



FASTQ形式ファイル読み込み

readDNASTringSet関数を用いた読み込み時に、オプションを変更することでFASTQ形式ファイルに対応している



(Rで)塩基配列解析

• イントロ NGS 配列取得 FASTQ or SRALite SRADB(Zhu 2013)(last modified 2014/04/0
• イントロ NGS アノテーション情報取得 について
• イントロ NGS アノテーション情報取得 GFF/GT
• イントロ NGS アノテーション情報取得 refFlat
• イントロ NGS アノテーション情報取得 biomaRt
• イントロ NGS アノテーション情報取得 Transcrip
• イントロ NGS アノテーション情報取得 Transcrip
• イントロ NGS アノテーション情報取得 Transcrip
• イントロ NGS アノテーション情報取得 Transcrip
• イントロ NGS 読み込み FASTA形式 基本情
• イントロ NGS 読み込み FASTA形式 script
• イントロ NGS 読み込み FASTQ形式 mod
• イントロ NGS 読み込み FASTQ形式 script
• イントロ NGS 読み込み Illuminaの * seq.txt(l
• イントロ NGS 読み込み Illuminaの * qseq.txt(
• イントロ ファイル形式の変換 について (last mo
• イントロ ファイル形式の変換 BAM -> BED(la
• イントロ ファイル形式の変換 FASTQ -> FAST
• イントロ ファイル形式の変換 Genbank -> FAS
• イントロ ファイル形式の変換 qseq -> FASTA(
• イントロ ファイル形式の変換 qseq -> Illumina

イントロ | NGS | 読み込み | FASTQ形式 NEW

Sanger FASTQ形式ファイルを読み込むやり方を示します。
「ファイル」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

1. **SRR037439.fastq**の(quality情報を除く)塩基配列情報のみ読み込みたい場合:
配列長が同じ場合のみ読み込めます。

```

in_f <- "SRR037439.fastq" #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
#入力ファイルの読み込み
fasta <- readDNASTringSet(in_f, format="fastq")#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
    
```

2. **SRR037439.fastq**のquality情報も読み込みたい場合:
配列長が異なっていても読み込めます。

```

in_f <- "SRR037439.fastq" #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(ShortRead) #パッケージの読み込み
#入力ファイルの読み込み
fastq <- readFastq(in_f) #in_fで指定したファイルの読み込み
fastq #確認してるだけです
    
```

GC含量計算の詳細を説明

#本番

```
hoge <- alphabetFrequency(fasta)
CG <- rowSums(hoge[,2:3])
ACGT <- rowSums(hoge[,1:4])
GC_content <- CG/ACGT*100
```

#A,C,G,T,..の数を各配列ごとにカウントし
 #C,Gの総数を計算し
 #A,C,G,Tの総数を計算し
 #%GC含量を計算して

alphabetFrequency関数を実行して塩基ごとの出現回数をカウントした結果をhogeに格納している

R Console

```
> hoge <- alphabetFrequency(fasta)           #A,C,G,T,..の数を各配列ごとにカウント$
> hoge
```

	A	C	G	T	M	R	W	S	Y	K	V	H	D	B	N	-	+	.
[1,]	9709674	5435374	5421151	9697113	76	36	124	30	82	53	0	0	0	0	163958	0	0	0
[2,]	6315641	3542973	3520766	6316348	5	7	18	3	12	10	0	0	0	0	2506	0	0	0
[3,]	7484757	4258333	4262704	7448059	2	4	2	1	2	0	0	0	0	0	5966	0	0	0
[4,]	5940546	3371349	3356091	5914038	1	0	0	0	0	0	0	0	1	0	3030	0	0	0
[5,]	8621974	4832253	4858759	8652238	0	0	0	0	0	0	0	0	0	0	10278	0	0	0
[6,]	102464	82661	81609	100190	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[7,]	48546	28496	27570	49866	0	0	0	0	0	0	0	0	0	0	0	0	0	0

```
> dim(hoge)
[1] 7 18
> |
```

dim関数は行列hogeの行数と列数を表示。つまりhogeは7行×18列から構成されているということ

GC含量計算の詳細を説明

```
R Console
> hoge <- alphabetFrequency(fasta)           #A, C, G, T, ...の数を各配列ごとにカウント$
> hoge
      A      C      G      T      M      R      W      S      Y      K      V      H      D      B      N      -      +      .
[1,] 9709674 5435374 5421151 9697113  76  36  124  30  82  53  0  0  0  0 163958  0  0  0
[2,] 6315641 3542973 3520766 6316348   5   7   18   3  12  10  0  0  0  0   2506  0  0  0
[3,] 7484757 4258333 4262704 7448059   2   4    2   1   2   0  0  0  0  0   5966  0  0  0
[4,] 5940546 3371349 3356091 5914038   1   0    0   0   0   0  0  0  1  0   3030  0  0  0
[5,] 8621974 4832253 4858759 8652238   0   0    0   0   0   0  0  0  0  0  10278  0  0  0
[6,]  102464   82661   81609  100190   0   0    0   0   0   0  0  0  0  0     0  0  0  0
[7,]   48546   28496   27570   49866   0   0    0   0   0   0  0  0  0  0     0  0  0  0
> dim(hoge)
[1]  7 18
> |
```

A, C, G, T, およびN(A/C/G/T)の出現回数が多いのは当たり前。それ以外は、M(A/C), R(A/G), W(A/T), S(C/G), ...といった具合です。

```
> hoge <- alphabetFrequency(fasta) #A,C,G,T,...の数を各配列ごとにカウント$
> hoge
      A      C      G      T      M      R      W      S      Y      K      V      H      D      B      N      -      +      .
[1,] 9709674 5435374 5421151 9697113 76 36 124 30 82 53 0 0 0 0 163958 0 0 0
[2,] 6315641 3542973 3520766 6316348 5 7 18 3 12 10 0 0 0 0 2506 0 0 0
[3,] 7484757 4258333 4262704 7448059 2 4 2 1 2 0 0 0 0 0 5966 0 0 0
[4,] 5940546 3371349 3356091 5914038 1 0 0 0 0 0 0 0 0 1 0 3030 0 0 0
[5,] 8621974 4832253 4858759 8652238 0 0 0 0 0 0 0 0 0 0 10278 0 0 0
[6,] 102464 82661 81609 100190 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[7,] 48546 28496
> dim(hoge)
[1] 7 18
> |
```

TAIR10_chr_all.fas

GAAGTTAACTAGTCCCAGACTCAATCACCATTGACGAGAGCTACCTAACAGGCATTACGAATCAACAAGTTAAAGCCA
AAACGCTCCCTACAACCAATACCTTGGTACAGGGC
TTTTGGGCGCGAAAATATATGGGCTCAAATTCAG
GATTTATAAAATCAAATCAACCACCTCGCATACTG
ACTCAGACATCATTGCAGAAAGCATAAACGTTGAA
CAGGATGAACAACAGTAAACGAAATCAAGAACAAA
AGGTCAAGAACACAGTGTCTACAGAATAAACTTA
TCTAAAACCCCTAATACTCAAACGACGCAGTATTG
GTGCGAGTGATAGGATGGAGTCTTCTTTTCTCCTT
CGAGTGATAGAGAT
AGTAGTGTGATCG
GTCTTGGTGGTAC
GTGCGAGTGAATGA
CCATGTTGGTAGAG
TAGAGTGATTGGTCGAGTGAATGATGATGGTGATA
TGATACTCGACCTGTTGGTAGAGTATTGCTATA
TGAGCTCGTGGTCAAGTATGTTGATTTGATCAGGT
AACACACAAAATATGCAATATGCATGCAAACTAT
ACAGTGGTGATATGATGCTAAAGTGTGACAAAATGGATGCTCAAACGTGTAATGACACTTATCAACTCCCCAAA
CTTAGTATTTGCTTGGCCCTCAAGCAAACAATTAAGAACAAGCTGGAGATGAGGTTTAAAGCGGGGACTCAGAACAAA
GCATGAGATATGACAATTAAGATCAATGTATAAGCTAACAGTCTAAAATGCAAGGTGATCGACTTCTACTTAAAACT
TTAGTTATGCTCTGTTATGATCCAAATTCACACTCAGTTGCACAATACGTCAAGATCAACCAATCCCTTTAACATTCAT

検索

検索する文字列(E):
w

前を検索(P)

次を検索(N)

すべてを検索(D)

置換(R) >>

閉じる

大文字と小文字を区別する(C)

正規表現を使用する(X)

エスケープシーケンスを使用する(E)

単語のみ検索する(W)

インクリメンタルサーチ(I)

開いているすべての文書から検索(S)

文末まで検索したら文頭に移動する(M)

一致する文字列を数える(O)

終了したら閉じる(L)

入力のシロイヌナズナゲノム配列ファイル (TAIR10_chr_all.fas) 中を検索すると、確かにWなどのACGTN以外の文字が存在します

GC含量計算の詳細を説明

```
R Console
> hoge
      A      C      G      T  M  R  W  S  Y  K  V  H  D  B      N  -  +  .
[1,] 9709674 5435374 5421151 9697113 76 36 124 30 82 53 0 0 0 0 163958 0 0 0
[2,] 6315641 3542973 3520766 6316348 5 7 18 3 12 10 0 0 0 0 2506 0 0 0
[3,] 7484757 4258333 4262704 7448059 2 4 2 1 2 0 0 0 0 0 5966 0 0 0
[4,] 5940546 3371349 3356091 5914038 1 0 0 0 0 0 0 0 1 0 3030 0 0 0
[5,] 8621974 4832253 4858759 8652238 0 0 0 0 0 0 0 0 0 0 10278 0 0 0
[6,] 102464 82661 81609 100190 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[7,] 48546 28496 27570 49866 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> rowSums(hoge)
[1] 30427671 19698289 23459830 18585056 26975502 366924 154478
> |
```

行列hogeに対して、rowSums関数を用いて行ごとのカウント数の総和を計算すると、染色体ごとの配列長と一致するのは当然です

出力: hoge4.txt

Description	CG	ACGT	Length	%GC_contents
1 CHROMOSOME	10856525	30263312	30427671	35.874
2 CHROMOSOME	7063739	19695728	19698289	35.864
3 CHROMOSOME	8521037	23453853	23459830	36.331
4 CHROMOSOME	6727440	18582024	18585056	36.204
5 CHROMOSOME	9691012	26965224	26975502	35.939
mitochondria CHRO	164270	366924	366924	44.769
chloroplast CHROM	56066	154478	154478	36.294

GC含量計算の詳細を説明

#本番

```
hoge <- alphabetFrequency(fasta)
CG <- rowSums(hoge[,2:3])
ACGT <- rowSums(hoge[,1:4])
GC_content <- CG/ACGT*100
```

#A,C,G,T,..の数を各配列ごとにカウントして
 #C,Gの総数を計算してCGに格納
 #A,C,G,Tの総数を計算してACGTに格納
 #%GC含量を計算してGC_contentに格納

```
R Console
> hoge[,2:3]
      C      G
[1,] 5435374 5421151
[2,] 3542973 3520766
[3,] 4258333 4262704
[4,] 3371349 3356091
[5,] 4832253 4858759
[6,]   82661   81609
[7,]   28496   27570
> hoge[,1:4]
      A      C      G      T
[1,] 9709674 5435374 5421151 9697113
[2,] 6315641 3542973 3520766 6316348
[3,] 7484757 4258333 4262704 7448059
[4,] 5940546 3371349 3356091 5914038
[5,] 8621974 4832253 4858759 8652238
[6,]  102464   82661   81609  100190
[7,]   48546   28496   27570   49866
> |
```

```
R Console
> rowSums(hoge[,2:3])
[1] 10856525 7063739 8521037 6727440 9691012 164270 56066
> rowSums(hoge[,1:4])
[1] 30263312 19695728 23453853 18582024 26965224 366924 154478
> |
```

CGまたはACGTのサブセットを抽出してからrowSums関数を実行

description	CG	ACGT	Length	%GC_contents
1 CHROMOSOME	10856525	30263312	30427671	35.874
2 CHROMOSOME	7063739	19695728	19698289	35.864
3 CHROMOSOME	8521037	23453853	23459830	36.331
4 CHROMOSOME	6727440	18582024	18585056	36.204
5 CHROMOSOME	9691012	26965224	26975502	35.939
mitochondria CHRO	164270	366924	366924	44.769
chloroplast CHROM	56066	154478	154478	36.294

GC含量計算の詳細を説明

#ファイルに保存

```
tmp <- cbind(names(fasta), CG, ACGT, width(fasta), GC_content) #保存したい情報を+
colnames(tmp) <- c("description", "CG", "ACGT", "Length", "%GC_contents") #列名
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F, col.names=T)
```

R Console output for `fasta`:

```
> fasta
A DNASTringSet instance of length 7
  width seq
[1] 30427671 CCCTAAACCCTAAACC...GGTTAGGGTTAGGG 1 CHROMOSOME dump...
[2] 19698289 NNNNNNNNNNNNNNNN...GGTTAGGGTTAGGG 2 CHROMOSOME dump...
[3] 23459830 NNNNNNNNNNNNNNNN...CCTAAACCCTAAACCC 3 CHROMOSOME dump...
[4] 18585056 NNNNNNNNNNNNNNNN...GGGTTAGGGTTAGG 4 CHROMOSOME dump...
[5] 26975502 TATACCATGTACCCTC...TTAGGGTTTTAGATC 5 CHROMOSOME dump...
[6] 366924 GGATCCGTTTCGAAACA...GAAACAAACCGGATT mitochondria CHRO...
[7] 154478 ATGGGCGAACGACGGG...CTTGGTCCCGGGCATC chloroplast CHROM...
```

R Console output for `names(fasta)`:

```
> names(fasta)
[1] "1 CHROMOSOME dumped from ADB: Feb/3/09 16:9; last updated: 2009$
[2] "2 CHROMOSOME dumped from ADB: Feb/3/09 16:10; last updated: 200$
[3] "3 CHROMOSOME dumped from ADB: Feb/3/09
[4] "4 CHROMOSOME dumped from ADB: Feb/3/09
[5] "5 CHROMOSOME dumped from ADB: Feb/3/09
[6] "mitochondria CHROMOSOME dumped from AD
[7] "chloroplast CHROMOSOME dumped from ADB
```

description	CG	ACGT	Length	%GC_contents
1 CHROMOSOME	10856525	30263312	30427671	35.874
2 CHROMOSOME	7063739	19695728	19698289	35.864
3 CHROMOSOME	8521037	23453853	23459830	36.331
4 CHROMOSOME	6727440	18582024	18585056	36.204
5 CHROMOSOME	9691012	26965224	26975502	35.939
mitochondria CHRO	164270	366924	366924	44.769
chlomplast CHROM	56066	154478	154478	36.294

パッケージ説明

4. 120MB程度のシロイヌナズナゲノムのmulti-FASTAファイル(T)

```

in_f <- "TAIR10_chr_all.fas"      #入力ファイル
out_f <- "hoge4.txt"             #出力ファイル

#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") #確認して
fasta

#本番
hoge <- alphabetFrequency(fasta) #A,C,G,T
CG <- rowSums(hoge[,2:3])         #C,Gの総数
ACGT <- rowSums(hoge[,1:4])      #A,C,G,T
GC_content <- CG/ACGT*100        #%GC含量を

#ファイルに保存
tmp <- cbind(names(fasta), CG, ACGT, width(fasta))
colnames(tmp) <- c("description", "CG", "ACGT", "GC")
write.table(tmp, out_f, sep="\t", append=F, quote=F)
    
```

• [BioconductorのBiostringsのwebページ](#)

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 fhrc.org>

To install this package, start R and enter:

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

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 Biostrings 2
- [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

Details

biocViews [DataImport](#), [DataRepresentation](#), [Genetics](#), [Infrastructure](#), [SequenceMatching](#), [Sequencing](#), [Software](#)

パッケージを個別にインストールする場合

使い方の解説記事はPDFのところをクリック。例えば…

Biostrings Quick Overview

Hervé Pagès
Fred Hutchinson Cancer Research Center
Seattle, WA

April 3, 2013

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

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

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

Function	Description
<code>subseq, subseq<-</code>	Extract or replace subsequences in a set of sequences.
<code>reverse</code>	Compute the reverse, complement, or reverse-complement, of a set of DNA sequences.
<code>complement</code>	
<code>reverseComplement</code>	
<code>translate</code>	Translate a set of DNA sequences into a set of Amino Acid sequences.
<code>chartr</code>	Translate the letters in a set of sequences.
<code>replaceLetterAt</code>	Replace the letters specified by a set of positions by new letters.

Table 2: Basic transformations of sequences.

Function	Description
<code>alphabetFrequency</code> <code>letterFrequency</code>	Tabulate the letters (all the letters in the alphabet for <code>alphabetFrequency</code> , only the specified letters for <code>letterFrequency</code>) of a sequence or set of sequences.
<code>letterFrequencyInSlidingView</code>	Specialized version of <code>letterFrequency</code> that tallies the requested letter frequencies for a fixed-width view that is conceptually slid along the input sequence.
<code>consensusMatrix</code>	Computes the consensus matrix of a set of sequences.
<code>dinucleotideFrequency</code> <code>trinucleotideFrequency</code> <code>oligonucleotideFrequency</code>	Fast 2-mer, 3-mer, and k-mer counting for DNA or RNA.
<code>nucleotideFrequencyAt</code>	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
<code>matchPattern</code> <code>countPattern</code>	Find/count all the occurrences of a given pattern (typically short) in a reference sequence (typically long). Support mismatches and indels.
<code>vmatchPattern</code> <code>vcountPattern</code>	Find/count all the occurrences of a given pattern (typically short) in a set of reference sequences. Support mismatches and indels.
<code>matchPDict</code> <code>countPDict</code> <code>whichPDict</code>	Find/count all the occurrences of a set of patterns in a reference sequence. (<code>whichPDict</code> only identifies which patterns in the set have at least one match.) Support a small number of mismatches.
<code>vmatchPDict</code> <code>vcountPDict</code> <code>vwhichPDict</code>	[Note: <code>vmatchPDict</code> not implemented yet.] Find/count all the occurrences of a set of patterns in a set of reference sequences. (<code>whichPDict</code> only identifies for each reference sequence which patterns in the set have at least one match.) Support a small number of mismatches.
<code>pairwiseAlignment</code>	Solve (Needleman-Wunsch) global alignment, (Smith-Waterman) local alignment, and (ends-free) overlap alignment problems.
<code>matchPWM</code> <code>countPWM</code>	Find/count all the occurrences of a Position Weight Matrix in a reference sequence.
<code>trimLeft</code> <code>trimRight</code>	Trim left and/or right flanking patterns from sequences.
<code>findPairs</code>	Find all paired matches in a reference sequence i.e. matches specified by a left and a right pattern, and a maximum distance between them.
<code>findAmplicons</code>	Find all the amplicons that match a pair of probes in a reference sequence.

Biostringsパッケージ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多くを実行可能

原著論文引用をお願いします

- 解析 | 一般 | [アラインメント\(ベアワイズ;基本編2\)](#) (last modified 2010/6/8)
- 解析 | 一般 | [アラインメント\(ベアワイズ;応用編\)](#) (last modified 2010/6/8)
- 解析 | 一般 | [パターンマッチング](#) (last modified 2013/06/19)
- 解析 | 一般 | [GC含量 \(GC contents\)](#) (last modified 2014/05/01) **NEW**
- 解析 | 一般 | [Sequence logos\(Schneider 1990\)](#) (last modified 2014/04/27) **NEW**
- 解析 | 一般 | 上流配列解析 | [LDSS\(Yamamoto 2007\)](#) (last modified 2012/07/17)
- 解析 | 一般 | 上流配列解析 | [Relative Appearance Ratio\(Yamamoto 2011\)](#) (last m

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 bpl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」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

1. 入力

in_+

```
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
fasta <- fasta[obj] #objがTRUEとなるもののみ抽出した結果をf
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fa:
fasta #確認してるだけです

#本番(sequence logoを実行)
```

◦ [seqLogo: Schneider and Stephens, Nucleic Acids Res., 1990](#)

「(Rで)塩基配列解析」を利用した証拠もないしアクセスログもとっていないので引用や謝辞は必要なし

(Rで)塩基配列解析のことは見なかったことにしても、用いたRパッケージや元となるプログラムの原著論文は引用してください



by KDT39

Contents

■ Rでゲノム解析

- シロイヌナズナゲノムのGC含量計算
 - multi-FASTAファイルの読み込み
 - 関数やオプションの利用法
 - パッケージの説明

■ Rでトランスクリプトーム解析

- シロイヌナズナのRNA-seqデータを一通り解析
 - 公共DBからの生データ取得
 - マッピングおよびカウントデータ取得
 - サンプル間クラスタリング
 - 発現変動遺伝子(DEG)検出



トランスクリプトーム解析

- シロイヌナズナのRNA-seqデータを一通りRで解析
 - 2群間比較用: 4 DEX-treated vs. 4 mock-treated
 - 生データ(FASTQファイル)のID: GSE36469

[Development](#), 2012 Jun;139(12):2161-9. doi: 10.1242/dev.075069. Epub 2012 May 9.

RBE controls microRNA164 expression to effect floral organogenesis.

[Huang T¹](#), [López-Giráldez F](#), [Townsend JP](#), [Irish VF](#).

⊕ Author information

Abstract

The establishment and maintenance of organ boundaries are vital for animal and plant development. In the *Arabidopsis* flower, three microRNA164 genes (MIR164a, b and c) regulate the expression of CUP-SHAPED COTYLEDON1 (CUC1) and CUC2, which encode key transcriptional regulators involved in organ boundary specification. These three miR164 genes are expressed in distinct spatial and temporal domains that are crucial for their function. Here, we show that the C2H2 zinc finger transcriptional repressor encoded by RABBIT EARS (RBE) regulates the expression of all three miR164 genes. Furthermore, we demonstrate that RBE directly interacts with the promoter of MIR164c and negatively regulates its expression. We also show that the role of RBE in sepal and petal development is mediated in part through the concomitant regulation of the CUC1 and CUC2 gene products. These results indicate that one role of RBE is to fine-tune miR164 expression to regulate the CUC1 and CUC2 effector genes, which, in turn, regulate developmental events required for sepal and petal organogenesis.

PMID: 22573623 [PubMed - indexed for MEDLINE] [Free full text](#)

生データ取得から発現変動解析までをRのみで実行

トランスクリプトーム解析

- シロイヌナズナのRNA-seqデータを一通りRで解析
 - 2群間比較用: 4 DEX-treated vs. 4 mock-treated
 - 生データ(FASTQファイル)のID: GSE36469

Step1: 生データをダウンロード(するために必要なID情報を取得)

- 作図 | [M-A plot\(基本編\)](#)(last modified 2012/10/01)
- 作図 | [M-A plot\(ggplot2編\)](#)(last modified 2013/07/30)
- 作図 | [ROC曲線](#)(last modified 2012/10/01)
- 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\(Neyret-Kahn_2013\)](#)
- [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\(Neyret-Kahn_2013\)](#)
- [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\(Huang_2012\)](#)
- [パイプライン](#) | [ゲノム](#) | [small RNA](#) | [SRP016842\(Nie_2013\)](#)(last modified 2013/11/12)
- [リンク集](#)(last modified 2012/03/29)

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(Huang_2012)

[Huang et al., Development, 2012](#)の2群間比較用シロイヌナズナRNA-seqデータ(4 DEX-treated vs. 4 mock-treated)が [GSE36469](#)に登録されています。ここでは、[SRADB](#)パッケージを用いたそのFASTQ形式ファイルのダウンロードから、[QuasR](#)パッケージを用いたマッピングおよびカウントデータ取得、そして[TCC](#)パッケージを用いた発現変動遺伝子(DEG)検出までを行う一連の手順を示します。多数のファイルが作成されるので、ここでは「デスクトップ」上に「SRP011435」というフォルダを作成しておき、そこで作業を行うことにします。

Step1. RNA-seqデータのgzip圧縮済みのFASTQファイルをダウンロード

論文中の記述から[GSE36469](#)を頼りに、RNA-seqデータが[GSE36469](#)に収められていることを見出し、その情報から[SRP011435](#)にたどり着いています。したがって、ここで指定するのは「SRP011435」となります。

計8ファイル、合計10Gb程度の容量のファイルがダウンロードされます。東大の有線LANで2時間程度かかります。早く終わらせたい場合は、最後のgetFASTQfile関数のオプションを'ftp'から'fasp'に変更すると時間短縮可能です。

[イントロ](#) | [NGS](#) | [配列取得](#) | [FASTQ or SRALite](#) | [SRADB\(Zhu_2013\)](#)の記述内容と基本的に同じです。

```
param <- "SRP011435" #取得したいSRA IDを指定
```

```
#必要なパッケージをロード
```

トランスクリプトーム解析

- シロイヌナズナのRNA-seqデータを一通りRで解析
 - 2群間比較用: 4 DEX-treated vs. 4 mock-treated
 - 生データ(FASTQファイル)のID: GSE36469

生データをダウンロードするために必要なIDはSRP011435だということを知る

NCBI GEO Accession Display for GSE36469

Scope: Self | Format: HTML | Amount: Quick | GEO accession: GSE36469

Series GSE36469

Status: Public on Jul 12, 2012
 Title: High-throughput Illumina RNA-seq of RABBIT EARS (RBE) in the
 Organism: Arabidopsis thaliana
 Experiment type: Expression profiling by high throughput sequencing
 Summary: In order to identify putative target mRNAs from dexamethasone (DEX)-treated Arabidopsis thaliana line 35S:GR-RBE (RBE coding region driven by the constitutive 35S promoter), we performed RNA-seq. Results from DEX and mock-treated plants were compared to mock-treated plants (EEP1) as a candidate target for molecular and genetic analysis of normal floral organ formation.

Overall design: We used two biological replicates for each of the 4-hour DEX or mock treated floral buds.

Contributor(s): Huang T, López-Giráldez F, Torres
 Citation(s): Huang T, López-Giráldez F, Torres
 Submission date: Mar 13, 2012

Query DataSets for GSE36469

GSM894356	mRNA from DEX-treated 35S:GR-RBE floral buds, biological replicate 1, technical replicate 2
GSM894357	mRNA from DEX-treated 35S:GR-RBE floral buds, biological replicate 2, technical replicate 1

Relations

SRA: SRP011435
 BioProject: PRJNA153493

Download family

Format
SOFT formatted family file(s)
MINiML formatted family file(s)
Series Matrix File(s)

Supplementary file	Size	Download	File type/resource
GSE36469_LOX_output_combined_final.txt.gz	1.3 Mb	(ftp)(http)	TXT
SRP/SRP011/SRP011435		(ftp)	SRA Study

Raw data provided as supplementary file
 Processed data is available on Series record

Step1. RNA-seqデータのgzip圧縮済みのFASTQファイルをダウンロード:

論文中の記述から[GSE36469](#)を頼りに、RNA-seqデータが[GSE36469](#)として収められていることを見出し、その情報から[SRP011435](#)にたどり着いています。したがって、ここで指定するのは"[SRP011435](#)"となります。

計8ファイル、合計10Gb程度の容量のファイルがダウンロードされます。東大の有線LANで2時間程度かかります。早く終わらせたい場合は、最後のgetFASTQfile関数のオプションを'ftp'から'fasp'に変更すると時間短縮可能です。

[イントロ](#) | [NGS](#) | [配列取得](#) | [FASTQ or SRALite](#) | [SRADB\(Zhu 2013\)](#)の記述内容と基本的に同じです。

```
param <- "SRP011435" #取得したいSRA IDを指定

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

#前処理
#sqlfile <- "SRAmetadb.sqlite" #最新でなくてもよく、手元に予めダウン
sqlfile <- getSRADBFile() #最新のSRAmetadb SQLiteファイルをダウンロードして解凍(圧縮状態で300
sra_con <- dbConnect(SQLite(), sqlfile)#おまじない

#前処理(実験デザインの全体像を表示)
hoge <- sraConvert(param, sra_con=sra_con)#paramで指定したSRA IDに付随するstudy (SRP...), sample(SRS...
hoge #hogeの中身を表示
apply(hoge, 2, unique) #hoge行列の列ごとにユニークな文字列を表示させている。

#前処理(FASTQファイルサイズを表示)
k <- getFASTQinfo(in_acc=hoge$run) #「hoge$run」で指定したSRRから始まるIDのFASTQファイルサイズ情報など
k #kの中身を表示
hoge2 <- cbind(k$library.name, #ライブラリ名と、
              k$run.read.count, #総リード数と、
              k$file.name, #ファイル名と、
              k$file.size) #ファイルサイズ、の順番で列方向で結合した結果をhoge2に格納
```

SRP011435を入力として、R上でFASTQファイルをダウンロード可能(東大有線LANで数時間) 実習ではやらないで!!

無事ダウンロードが終了すると、作業ディレクトリ(「デスクトップ」上の「SRP011435」フォルダ)中に9つのファイルが存在するはずですが、4Gb程度ある"SRAmetadb.sqlite"ファイルは無視して構いません。残りの"SRR"からはじまる8つのファイルがダウンロードしたRNA-seqデータです。オリジナルのサンプル名(の略称)で対応関係を表すと[srp011435_samplename.txt](#)のようになっていることがわかります。尚このファイルはマッピング時の入力ファイルとしても用います。

Step1: 生データのダウンロード中...

The screenshot shows the RGui (64-bit) interface on the left and a Windows Explorer window on the right. The R Console displays the execution of the `getFASTQfile` function, which downloads FASTQ files from the SRA database. A progress bar indicates that 34% of the file has been downloaded. The Windows Explorer window shows the local directory `C:/Users/kadota/Desktop/SRP011435` containing the downloaded files.

R Console Output:

```
[6,] "36683370" "SRR444602.fastq.gz" "1Gb"
[7,] "39741115" "SRR444599.fastq.gz" "1Gb"
[8,] "32125368" "SRR444596.fastq.gz" "1Gb"
>
> #本番 (FASTQファイルのダウンロード)
> getFASTQfile(hoge$run, srcType='ftp') #「hoge$run」で指定したS
Files are saved to:
'C:/Users/kadota/Desktop/SRP011435'

URL 'ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR444/SRR444597/SRR
ftp d
開か
downl

URL
ftp d
開か
downl

URL 'ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR444/SRR444598/SRR444598.fastq.gz$
ftp data connection made, file length 1086139920 bytes
開かれた URL
```

Windows Explorer File List:

名前	更新日時	サイズ	種類
SRAmetadb.sqlite	2014/05/02 13:15	5,873,861 KB	SQLITE ファ
SRR444595.fastq.gz	2014/05/02 13:40	1,485,278 KB	GZ ファイル
SRR444597.fastq.gz	2014/05/02 13:25	1,145,983 KB	GZ ファイル
SRR444598.fastq.gz	2014/05/02 13:40	0 KB	GZ ファイル

ここでは作業ディレクトリとして、デスクトップ上のSRP011435を指定している

Step1: 生データのダウンロード終了後

- シロイヌナズナのRNA-seqデータを一通りRで解析
 - 2群間比較用: 4 DEX-treated vs. 4 mock-treated

IDとサンプル属性(ラベル)との対応関係を知りたい

```
R Console
file.name  file.size  md5
1 SRR444597.fastq.gz  1Gb  0d3afb726664be6c8a6a72bc17047433
2 SRR444595.fastq.gz  1Gb  1135536edabffe7a4189e3ead941f27b
3 SRR444598.fastq.gz  1Gb  c36633fa250c99dff9d7
4 SRR444600.fastq.gz  1Gb  03ca6fad778d52c881d8
5 SRR444601.fastq.gz  1Gb  559e199b96548f737832
6 SRR444602.fastq.gz  1Gb  ce0ef78f2e5dda74df76
7 SRR444599.fastq.gz  1Gb  3404499034b24872534c
8 SRR444596.fastq.gz  1Gb  3e28013ba7948f020388

1 ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR444/SRR4445
2 ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR444/SRR4445
3 ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR444/SRR4445
4 ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR444/SRR4446
5 ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR444/SRR4446
6 ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR444/SRR4446
7 ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR444/SRR4445
8 ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR444/SRR4445
> |
```

C:\Users\kadota\Desktop\SRP011435

名前	更新日時	サイズ	種類
SRAmetadb.sqlite	2014/05/02 13:15	5,873,861 KB	SQLITE ファイル
SRR444595.fastq.gz	2014/05/02 13:40	1,485,278 KB	GZ ファイル
SRR444596.fastq.gz	2014/05/02 15:07	1,239,187 KB	GZ ファイル
SRR444597.fastq.gz	2014/05/02 13:25	1,145,983 KB	GZ ファイル
SRR444598.fastq.gz	2014/05/02 13:55	1,060,684 KB	GZ ファイル
SRR444599.fastq.gz	2014/05/02 14:56	1,489,645 KB	GZ ファイル
SRR444600.fastq.gz	2014/05/02 14:09	1,382,869 KB	GZ ファイル
SRR444601.fastq.gz	2014/05/02 14:29	1,444,211 KB	GZ ファイル
SRR444602.fastq.gz	2014/05/02 14:42	1,236,288 KB	GZ ファイル

Step1. RNA-seqデータのgzip圧縮済みのFASTQファイルをダウンロード:

論文中の記述からGSE36469を頼りに、RNA-seqデータがGSE36469として収められていることを見出し、その情報からSRP011435にたどり着いています。したがって、ここで指定するのは"SRP011435"となります。

計8ファイル、合計10Gb程度の容量のファイルがダウンロードされます。東大の有線LANで2時間程度かかります。早く終わらせたい場合は、最後のgetFASTQfile関数のオプションを'ftp'から'fasp'に変更すると時間短縮可能です。

[イントロ](#) | [NGS](#) | [配列取得](#) | [FASTQ or SRALite](#) | [SRADB\(Zhu 2013\)](#)の記述内容と基本的に同じです。

```
#前処理
#sqlfile <- "SRAMetadb.sqlite" #最新でなくてもよく、手元に予めダウンロードしてある"SRAMetadb.sqlite"
sqlfile <- getSRADBFile() #最新のSRAMetadb SQLiteファイルをダウンロードして解凍(圧縮状態で300
sra_con <- dbConnect(SQLite(), sqlfile) #おまじない
```

```
#前処理(実行)
hoge <- sra <- getSRADBFile()
hoge
apply(hoge, 1, function(x) {
  k <- getFASTQfile(x)
  k
})
#前処理(FASTQファイルのダウンロード)
hoge2 <- cbind(k$library.name, k$run.read.count, k$file.name, k$file.size)
#hoge2の中身を表示(表示される情報を限定しているだけです)
```

```
R Console
> hoge2 <- cbind(k$library.name, k$run.read.count, k$file.name, k$file.size)
#ライブラリ名と、総リード数と、ファイル名と、ファイルサイズ、の順番で列方向で結合した結果をhoge2に格納
#hoge2の中身を表示(表示される情報を限定しているだけです)
hoge2
[1,] "GSM894357: mRNA from DEX-treated 35S:GR-RBE floral buds, biological replicate 2, technical replicate 1"
[2,] "GSM894355: mRNA from DEX-treated 35S:GR-RBE floral buds, biological replicate 1, technical replicate 1"
[3,] "GSM894358: mRNA from DEX-treated 35S:GR-RBE floral buds, biological replicate 2, technical replicate 2"
[4,] "GSM894360: mRNA from mock-treated 35S:GR-RBE floral buds, biological replicate 1, technical replicate 2"
[5,] "GSM894361: mRNA from mock-treated 35S:GR-RBE floral buds, biological replicate 2, technical replicate 1"
[6,] "GSM894362: mRNA from mock-treated 35S:GR-RBE floral buds, biological replicate 2, technical replicate 2"
[7,] "GSM894359: mRNA from mock-treated 35S:GR-RBE floral buds, biological replicate 1, technical replicate 1"
[8,] "GSM894356: mRNA from DEX-treated 35S:GR-RBE floral buds, biological replicate 1, technical replicate 2"
#本番(FASTQファイルのダウンロード)
getFASTQfile(hoge2$run, srcType='ftp') #「hoge2$run」で指定したSRRから始まるIDのFASTQファイルのダウンロード
```

hoge2実行結果を眺めることで対応付けが可能

Step1. RNA-seqデータのgzip圧縮済みのFASTQファイルをダウンロード:

論文中の記述から [GSE36469](#) を頼りに、RNA-seqデータが [GSE36469](#) として収められていることを見出し、その情報から [SRP011435](#) にたどり着いています。したがって、ここで指定するのは "SRP011435" となります。

計8ファイル、合計10Gb程度の容量のファイルがダウンロードされます。東大の有線LANで2時間程度かかります。早く終わらせたい場合は、最後の `getFASTQfile` 関数のオプションを 'ftp' から 'fasp' に変更すると時間短縮可能です。

[イントロ](#) | [NGS](#) | [配列取得](#) | [FASTQ or SRALite](#) | [SRadb\(Zhu 2013\)](#) の記述内容と基本的に同じです。

ここまでで、Step1生データのダウンロードが完了

```
param <- "SRP011435" #取得したいSRA IDを指定

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

#前処理
#sqlfile <- "SRametadb.sqlite" #最新でなくてもよく、手元に予めダウンロードしてある"SRametadb.sqlit
sqlfile <- getSRadbFile() #最新のSRametadb SQLiteファイルをダウンロード
sra_con <- dbConnect(SQLite(), sqlfile) #おまじない

#前処理(実験デザインの全体像を表示)
hoge <- sraConvert(param, sra_con=sra_con) #paramで指定したSRA IDに付随するstudy
hoge #hogeの中身を表示
apply(hoge, 2, unique) #hoge行列の列ごとにユニークな文字列を表示

#前処理(FASTQファイルサイズを表示)
k <- getFASTQinfo(in_acc=hoge$run) #「hoge$run」で指定したSRRから始まるIDのF
k #kの中身を表示
hoge2 <- cbind(k$library.name, #ライブラリ名と、
               k$run.read.count, #総リード数と、
               k$file.name, #ファイル名と、
               k$file.size) #ファイルサイズ、の順番で列方向で結合した結果をhoge2に格納
```

FileName	SampleName
SRR444595.fastq.gz	DEX_bio1 tec1
SRR444596.fastq.gz	DEX_bio1 tec2
SRR444597.fastq.gz	DEX_bio2 tec1
SRR444598.fastq.gz	DEX_bio2 tec2
SRR444599.fastq.gz	mock_bio1 tec1
SRR444600.fastq.gz	mock_bio1 tec2
SRR444601.fastq.gz	mock_bio2 tec1
SRR444602.fastq.gz	mock_bio2 tec2

無事ダウンロードが終了すると、作業ディレクトリ(「デスクトップ」上の「SRP011435」フォルダ)中に9つのファイルが存在するはずですが、4Gb程度ある "SRametadb.sqlite" ファイルは無視して構いません。残りの "SRR" から始まる8つのファイルがダウンロードしたRNA-seqデータです。オリジナルのサンプル名(の略称)で対応関係を表すと [srp011435_samplename.txt](#) のようになっていることがわかります。尚このファイルはマッピング時の入力ファイルとしても用います。

Step2: マッピングおよびカウントデータ取得

■ マッピングに必要な情報

- FASTQファイル: 8個の*.fastq.gz
- リストファイル: [srp011435_samplename.txt](#)
- リファレンスゲノム: [TAIR10_chr_all.fas](#)

FileName	SampleName
SRR444595.fastq.gz	DEX_bio1 tec1
SRR444596.fastq.gz	DEX_bio1 tec2
SRR444597.fastq.gz	DEX_bio2 tec1
SRR444598.fastq.gz	DEX_bio2 tec2
SRR444599.fastq.gz	mock_bio1 tec1
SRR444600.fastq.gz	mock_bio1 tec2
SRR444601.fastq.gz	mock_bio2 tec1
SRR444602.fastq.gz	mock_bio2 tec2

■ カウントデータ取得に必要な情報

- 遺伝子アノテーションファイル: [TAIR10_GFF3_genes.gff](#)

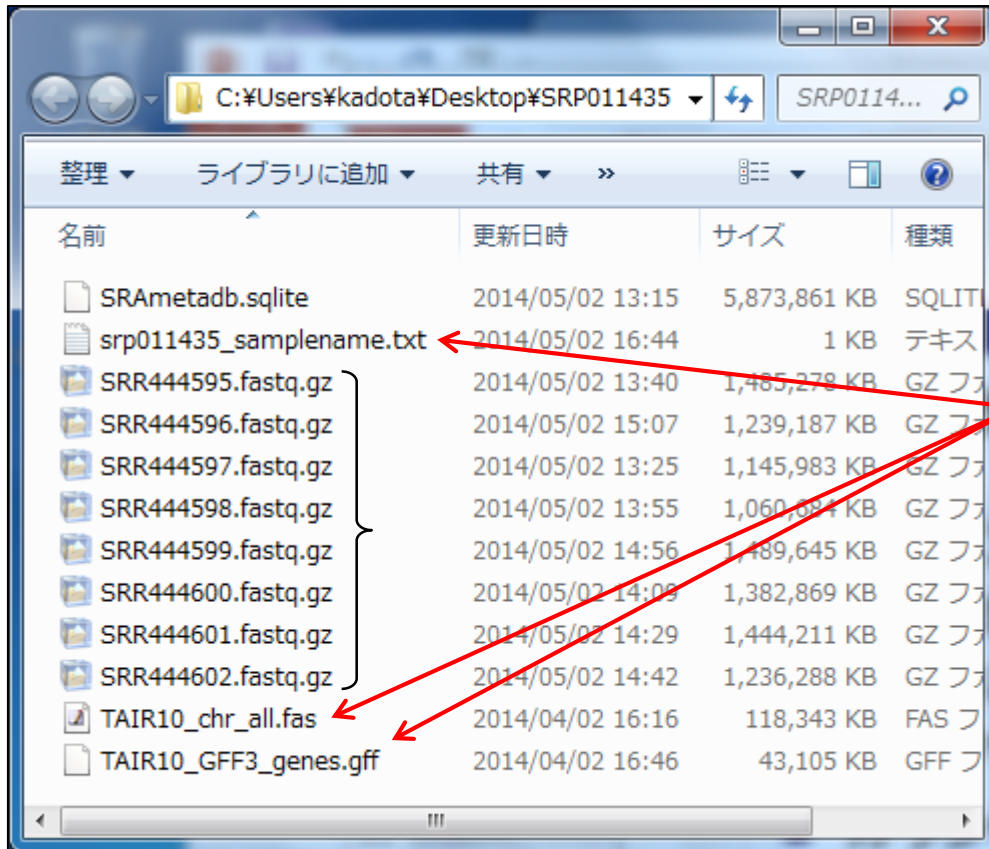


▲	A	B	C	D	E	F	G	H	I
1	Chr1	TAIR10	chromosome	1	30427671	.	.	.	ID=Chr1;Name=Chr1
2	Chr1	TAIR10	gene	3631	5899	.	+	.	ID=AT1G01010;Note=protein_coding_gene;Name=AT1G01010
3	Chr1	TAIR10	mRNA	3631	5899	.	+	.	ID=AT1G01010.1;Parent=AT1G01010;Name=AT1G01010.1;Index=1
4	Chr1	TAIR10	protein	3760	5630	.	+	.	ID=AT1G01010.1-Protein;Name=AT1G01010.1;Derives_from=AT1G01010.1
5	Chr1	TAIR10	exon	3631	3913	.	+	.	Parent=AT1G01010.1
6	Chr1	TAIR10	five_prime_UTR	3631	3759	.	+	.	Parent=AT1G01010.1
7	Chr1	TAIR10	CDS	3760	3913	.	+	0	Parent=AT1G01010.1,AT1G01010.1
8	Chr1	TAIR10	exon	3996	4276	.	+	.	Parent=AT1G01010.1
9	Chr1	TAIR10	CDS	3996	4276	.	+	2	Parent=AT1G01010.1,AT1G01010.1
10	Chr1	TAIR10	exon	4486	4605	.	+	.	Parent=AT1G01010.1
11	Chr1	TAIR10	CDS	4486	4605	.	+	0	Parent=AT1G01010.1,AT1G01010.1
12	Chr1	TAIR10	exon	4706	5095	.	+	.	Parent=AT1G01010.1

遺伝子ごとに、どの染色体のどの座標上に存在するのかななどの情報を含むタブ区切りテキストファイル

Step2: マッピングおよびカウントデータ取得

- マッピングに必要な情報
 - リストファイル: `srp011435_samplename.txt` (通常はテキストエディタで自作)
 - リファレンスゲノム: `TAIR10_chr_all.fas` (TAIRからダウンロード)
- カウントデータ取得に必要な情報
 - 遺伝子アノテーションファイル: `TAIR10_GFF3_genes.gff` (TAIRからダウンロード)



必要なファイルを作業ディレクトリに保存

(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 2014/04/10, since 2010)

What's

- 2014 農で
- 門田 解析
- 向ナ 置い
- 参考
- 私の フォ
- 東大 ノム
- 興味
- 機能 解析

- 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 2014/03/26) **NEW**
- 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 modified 2014/04/10) **NEW**
- イントロ | NGS | [アノテーション情報取得](#) | [refFlat形式ファイル](#) (last modified 2013/09/25)
- 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 | GFF/GTF形式ファイル **NEW**

多くの生物種について [Ensembl \(Flicek et al., Nucleic Acids Res., 2013\)](#) の [FTPサイト](#) から GTF形式 (GFF ver. 2) の遺伝子アノテーションファイルを得ることができます。refFlat形式同様、どの領域にどの遺伝子があるのかという座標 (Coordinates) 情報を含みます。ゲノム配列のバージョンと同じであることを確認した上で用いましょう。

• [Ensembl \(Flicek et al., Nucleic Acids Res., 2013\)](#)

圧縮 (gzip) ファイル形式です。基本は [FTPサイト](#) です。代表的なものを以下にリストアップしています。

- [ヒト; Human \(H.sapiens\)](#)
- [ラット; Rat \(R.norvegicus\)](#)
- [ネコ; Cat \(F.catus\)](#)
- [ウサギ; Rabbit \(O.cuniculus\)](#)
- [ニワトリ; Chicken \(G.gallus\)](#)
- [イヌ; Dog \(C.familiaris\)](#)
- [ウマ; Horse \(E.caballus\)](#)
- [ゼブラフィッシュ; Zebrafish \(D. rerio\)](#)
- ...
- イネ: [RAP-DB \(Sakai et al., Plant Cell Physiol., 2013\)](#)
 - 「[ダウンロード](#)」-「[Gene set](#)」-「[Gene structure and function in IRGSP-1.0_representative_2014-03-05.tar.gz](#) (12.4MB程度)」の圧縮ファイルが待っています
- シロイヌナズナ: [The Arabidopsis Information Resource \(TAIR\) \(Lamesch et al., Nucleic Acids Res., 2012\)](#)
 - 「[ダウンロード](#)」-「[Genes](#)」-「[TAIR10 genome release](#)」-「[TAIR10 gff3](#)」の [TAIR10 GFF3 genes.gff](#) (42MB程度)

TAIR10のアノテーションファイル (TAIR10_GFF3_genes.gff) はここからダウンロードしました

Home Help Contact About Us Login/Register

Search Browse Tools Portals Download Submit News ABRC Stocks

The Arabidopsis Information Resource

The Arabidopsis Information Resource (TAIR) maintains a database of biology data for the model higher plant *Arabidopsis thaliana*. Data includes the complete genome sequence along with gene structure, gene expression, DNA microarray data, and information on protein domains. The database is updated every week with new gene annotations and submissions. TAIR resources include:

- Genes
- GO and PO Annotations
- Maps

Download Overview
 ABRC Documents
 Genes
 GO and PO Annotations
 Maps

Subscribe to news feed
 Follow our Twitter feed
 Join our Facebook group

Breaking News

02/15/2012	12:00午前	411,136	Ath miRNAs	Konika Chawla	20120215.xls
09/29/2009	12:00午前	ディレクトリ	Gene families		
01/19/2013	12:00午前	528,450	Locus Primary Gene Symbol	20120117.txt	
02/07/2012	12:00午前	ディレクトリ	OLD		
08/23/2011	12:00午前	ディレクトリ	SmallRNAs	Carrington	
10/24/2013	10:52午後	ディレクトリ	TAIR10 genome release		
02/24/2009	12:00午前	ディレクトリ	TAIR6 genome release		
08/05/2009	12:00午前	ディレクトリ	TAIR7 genome release		
11/30/2009	12:00午前	ディレクトリ	TAIR8 genome release		
10/05/2011	12:00午前	ディレクトリ	TAIR9 genome release		
08/22/2012	12:00午前	5,545	README	TAIR10.txt	
08/23/2011	12:00午前	3,964,120	TAIR10-Subcellular Predictions	.xls	
08/23/2011	12:00午前	ディレクトリ	TAIR10 NCBI mapping files		
08/23/2011	12:00午前	792,935	TAIR10 TAIRAccessionID AGI mapping	.txt	
02/07/2011	04/13/2012	1,868,951	TAIR10 TAIRlocusaccessionID AGI mapping	.txt	
01/30/2011	08/23/2011	47	TAIR10 blastsets		
10/24/2011	08/23/2011	ディレクトリ	TAIR10 chromosome files		
08/27/2010	12:00午前	2,608,703	TAIR10 domain architectures	.tab.t10	
01/16/2013	12:00午前	25,396,877	TAIR10 functional descriptions		
11/23/2010	12:00午前	25,396,966	TAIR10 functional descriptions	.bk	
10/24/2013	10:51午後	25,874,762	TAIR10 functional descriptions	20130831.txt	
08/23/2011	12:00午前	ディレクトリ	TAIR10 gene confidence ranking		
08/23/2011	12:00午前	ディレクトリ	TAIR10 gene lists		
08/23/2011	12:00午前	ディレクトリ	TAIR10 gene transcript associations		
11/18/2010	12:00午前	30	TAIR10 eff3		
11/23/2010	12:00午前	2,053,133	TAIR10 locus history	.txt	
12/07/2010	12:00午前	904	TAIR10 sequence edits	.txt	
08/23/2011	12:00午前	ディレクトリ	TAIR10 transposable elements		
11/23/2010	12:00午前	ディレクトリ	TAIR10 YAC		

01/04/2011	12:00午前	2,662,626	Blist	TAIR10.gff
02/15/2012	12:00午前	ディレクトリ	Community annotation	GFF
03/31/2011	12:00午前	ディレクトリ	DNA replication origin	
04/06/2011	12:00午前	885	README	TAIR10 GFF3.txt
01/04/2011	12:00午前	266	README	gbrowse data.txt
01/04/2011	12:00午前	43,226,098	Spliced Junctions	clustered.gff
12/14/2010	12:00午前	44,139,005	TAIR10 GFF3 genes	.gff
12/14/2010	12:00午前	49,811,410	TAIR10 GFF3 genes	transcript.gff
01/04/2011	12:00午前	121,054	TAIR10 Models	obsolete.gff
01/04/2011	12:00午前	3,115,098	TAIR10 unconfirmed exons	.gff
10/18/2011	12:00午前	39,825,304	TAIR GFF3 ssrs	.gff
07/13/2011	12:00午前	134,836	arabidopsis	.conf

TAIRウェブインターフェースからアノテーションファイル (TAIR10_GFF3_genes.gff) を取得する際のイメージ

Step2: マッピングおよびカウントデータ取得

Step2. シロイヌナズナ(Arabidopsis thaliana)ゲノムへのマッピングおよびカウントデータ取得:

マップしたいFASTQファイルリストおよびそのサンプル名を記述した [srp011435_samplename.txt](#) を作業ディレクトリに保存したうえで、下記を実行します。

[BSgenome](#) パッケージで利用可能な [BSgenome.Athaliana.TAIR.TAIR9](#) へマッピングしています。名前から推測できるように "TAIR" の "TAIR9" にマッピングしているのと同じです。BSgenome.Athaliana.TAIR.TAIR9 パッケージがインストールされていない場合は自動でインストールしてあげると便利です。

Step2. シロイヌナズナ(A. thaliana)ゲノムへのマッピングおよびカウントデータ取得(リファレンスがmulti-FASTAファイルの場合):

マップしたいFASTQファイルリストおよびそのサンプル名を記述した [srp011435_samplename.txt](#) を作業ディレクトリに保存したうえで、下記を実行します。シロイヌナズナのゲノム配列ファイル([TAIR10_chr_all.fas](#))へマッピングしています。但し、マッピングに用いる [QuasR](#) パッケージ中の [qAlign](#) 関数がリファレンス配列ファイルの拡張子として `*.fasta`, `*.fa`, `*.fna` しか認識してくれませんが、[TAIR10_chr_all.fas](#) のダウンロード後に拡張子を変更して [TAIR10_chr_all.fasta](#) にしています。また、`description` 行の染色体名が以下の `gff` ファイルと対応がとれませんので、`description` 行の記述を `param1` で置換しています。カウントデータを取得するために遺伝子アノテーションファイルを利用する必要があります。[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。マシンパワーにもよりますが、ノートPCでも10時間程度で終わると思います。[マップ後 | カウント情報取得 | ゲノム | アノテーション有 | QuasR \(Lerch XXX\)](#) の記述内容と基本的に同じです。

```

in_f <- "TAIR10_chr_all.fas" #入力ファイル名を指定してin_fに格納
out_f <- "tmp_genome.fasta" #出力ファイル名を指定してout_fに格納
param <- 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(Biostings) #パッケージの読み込み

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

#本番
names(fasta) <- param #names(fasta)の中身をparamで置換
fasta #確認してるだけです

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

Step2が二つ存在するが、リファレンスとしてRパッケージ [BSgenome.Athaliana.TAIR.TAIR9](#) ではなく [TAIR10_chr_all.fas](#) を利用するほうで説明します。

```

in_f1 <-
in_f2 <-
in_f3 <-
out_f1 <-
out_f2 <-
out_f3 <-
out_f4 <-
out_f5 <-
out_f6 <-
param_map
param3 <-
    
```

#必要なパッケージ

Step2. シロイヌナズナ (*A. thaliana*) ゲノムへのマッピングおよびカウントデータ取得(リファレンスがmulti-FASTAファイルの場合):

マップしたいFASTQファイルリストおよびそのサンプル名を記述した `srp011435_samplename.txt` を作業ディレクトリに保存したうえで、下記を実行します。シロイヌナズナのゲノム配列ファイル (`TAIR10_chr_all.fas`) へマッピングしています。但し、マッピングに用いる `QuasR` パッケージ中の `qAlign` 関数がリファレンス配列ファイルの拡張子として `*.fasta`, `*.fa`, `*.fna` しか認識してくれませんが、`TAIR10_chr_all.fas` のダウンロード後に拡張子を変更して `TAIR10_chr_all.fasta` にしています。また、`description` 行の染色体名が以下の `gff` ファイルと対応がとれませんが、`description` 行の記述を `param1` で置換しています。カウントデータを取得するために遺伝子アノテーションファイルを利用する必要があります。`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。マシンパワーにもよりますが、ノートPCでも10時間程度で終わると思います。マップ後 | カウント情報取得 | ゲノム | アノテーション有 | `QuasR` (`Lerch XXX`) の記述内容と基本的に同じです。

```
in_f <- "TAIR10_chr_all.fas"
out_f <- "tmp_genome.fasta"
param <- c("Chr1", "Chr2", "Chr3", "Chr4")

#必要なパッケージをロード
library(Biostrings)

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

#本番
names(fasta) <- param

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

最初に、`description` 行の記述を `Chr1` や `Chr2` に変更した `tmp_genome.fasta` を作成

```
R Console
> #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
A DNAStringSet instance of length 7
  width seq
[1] 30427671 CCCTAAACCCTAAACCCTAA...TTAGGGTTTAGGGTTAGGG
[2] 19698289 NNNNNNNNNNNNNNNNNNN...TTAGGGTTTAGGGTTAGGG
[3] 23459830 NNNNNNNNNNNNNNNNNNN...AAACCCTAAACCCTAAACC
[4] 18585056 NNNNNNNNNNNNNNNNNNN...TTTAGGGTTTAGGGTTAGG
[5] 26975502 TATACCATGTACCCTCAACC...GGATTTAGGGTTTTAGATC
[6] 366924 GGATCCGTTTCAAACAGGTT...GAATGGAAACAAACCGGATT
[7] 154478 ATGGGCGAACGACGGGAATT...ATAACTTGGTCCCAGGCATC
names
1 CHROMOSOME dump...
2 CHROMOSOME dump...
3 CHROMOSOME dump...
4 CHROMOSOME dump...
5 CHROMOSOME dump...
mitochondria CHRO...
chloroplast CHROM...

> #本番
> names(fasta) <- param #names(fasta)の中身をparamで置換
> fasta #確認してるだけです
A DNAStringSet instance of length 7
  width seq
[1] 30427671 CCCTAAACCCTAAACCCTAA...TTAGGGTTTAGGGTTAGGG Chr1
[2] 19698289 NNNNNNNNNNNNNNNNNNN...TTAGGGTTTAGGGTTAGGG Chr2
[3] 23459830 NNNNNNNNNNNNNNNNNNN...AAACCCTAAACCCTAAACC Chr3
[4] 18585056 NNNNNNNNNNNNNNNNNNN...TTTAGGGTTTAGGGTTAGG Chr4
[5] 26975502 TATACCATGTACCCTCAACC...GGATTTAGGGTTTTAGATC Chr5
[6] 366924 GGATCCGTTTCAAACAGGTT...GAATGGAAACAAACCGGATT ChrM
[7] 154478 ATGGGCGAACGACGGGAATT...ATAACTTGGTCCCAGGCATC ChrC

> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", width=50) #fastaの中身を$
> |
```

description行の記述を揃えるのは基本

□ 遺伝子アノテーションファイル: [TAIR10_GFF3_genes.gff](#)

	A	B	C	D	E	F	G	H	I
1	Chr1	TAIR10	chromosome	1	30427671	.	.	.	ID=Chr1;Name=Chr1
2	Chr1	TAIR10	gene	3631	5899	.	+	.	ID=AT1G01010;Note=protein_coding_gene;Name=AT1G01010
3	Chr1	TAIR10	mRNA	3631	5899	.	+	.	ID=AT1G01010.1;Parent=AT1G01010;Name=AT1G01010.1;Index=1
4	Chr1	TAIR10	protein	3760	5630	.	+	.	ID=AT1G01010.1-Protein;Name=AT1G01010.1;Derives_from=AT1G01010.1
5	Chr1	TAIR10	exon	3631	3913	.	+	.	Parent=AT1G01010.1
6	Chr1	TAIR10	five_prime_UTR	3631	3759	.	+	.	Parent=AT1G01010.1
7	Chr1	TAIR10	CDS	3760	3913	.	+	0	Parent=AT1G01010.1,AT1G01010.1-Protein;
8	Chr1	TAIR10	exon	3996	4276	.	+	.	Parent=AT1G01010.1
9	Chr1	TAIR10	CDS	3996	4276	.	+	2	Parent=AT1G01010.1,AT1G01010.1-Protein;
10	Chr1	TAIR10	exon	4486	4605	.	+	.	Parent=AT1G01010.1
11	Chr1	TAIR10	CDS	4486	4605	.	+	0	Parent=AT1G01010.1,AT1G01010.1-Protein;
12	Chr1	TAIR10	exon	4706	5095	.	+	.	Parent=AT1G01010.1

遺伝子アノテーションファイル中の1列目の表記法と同じにするのが基本

```

> #本番
> names(fasta) <- param #names(fasta)の中身をparamで置換
> fasta #確認してるだけです
A DNASTringSet instance of length 7
width seq
[1] 30427671 CCCTAAACCCTAAACCCTAA...TTAGGGTTTAGGGTTAGGG Chr1
[2] 19698289 NNNNNNNNNNNNNNNNNNNNN...TTAGGGTTTAGGGTTAGGG Chr2
[3] 23459830 NNNNNNNNNNNNNNNNNNNNN...AAACCCTAAACCCTAAACC Chr3
[4] 18585056 NNNNNNNNNNNNNNNNNNNNN...TTTAGGGTTTAGGGTTAGG Chr4
[5] 26975502 TATACCATGTACCCTCAACC...GGATTTAGGGTTTTAGATC Chr5
[6] 366924 GGATCCGTTTCGAAACAGGT...GAATGGAAACAAACCGGATT ChrM
[7] 154478 ATGGCGAACGACGGGAATT...ATAACTTGGTCCCGGGCATC ChrC
>
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, format="fasta", width=50)#fastaの中身を$
> |
    
```

Step2: マッピングおよびカウントデータ取得

```
in_f <- "TAIR10_chr_all.fas"      #入力ファイル名を指定してin_fに格納
out_f <- "tmp_genome.fasta"      #出力ファイル名を指定してout_fに格納
param <- c("Chr1","Chr2","Chr3","Chr4","Chr5","ChrM","ChrC")#置換したい文字列を指定
```

名前	更新日時	サイズ	種類
SRAmetadb.sqlite	2014/05/02 13:15	5,873,861 KB	SQLITE
srp011435_samplename.txt	2014/05/02 16:44	1 KB	テキスト
SRR444595.fastq.gz	2014/05/02 13:40	1,485,278 KB	GZ ファイル
SRR444596.fastq.gz	2014/05/02 15:07	1,239,187 KB	GZ ファイル
SRR444597.fastq.gz	2014/05/02 13:25	1,145,983 KB	GZ ファイル
SRR444598.fastq.gz	2014/05/02 13:55	1,060,684 KB	GZ ファイル
SRR444599.fastq.gz	2014/05/02 14:56	1,489,645 KB	GZ ファイル
SRR444600.fastq.gz	2014/05/02 14:09	1,382,869 KB	GZ ファイル
SRR444601.fastq.gz	2014/05/02 14:29	1,444,211 KB	GZ ファイル
SRR444602.fastq.gz	2014/05/02 14:42	1,236,288 KB	GZ ファイル
TAIR10_chr_all.fas	2014/04/02 16:16	118,343 KB	FAS ファイル
TAIR10_GFF3_genes.gff	2014/04/02 16:46	43,105 KB	GFF ファイル



名前	更新日時	サイズ	種類
SRAmetadb.sqlite	2014/05/02 13:15	5,873,861 KB	SQLITE
srp011435_samplename.txt	2014/05/02 16:44	1 KB	テキスト
SRR444595.fastq.gz	2014/05/02 13:40	1,485,278 KB	GZ ファイル
SRR444596.fastq.gz	2014/05/02 15:07	1,239,187 KB	GZ ファイル
SRR444597.fastq.gz	2014/05/02 13:25	1,145,983 KB	GZ ファイル
SRR444598.fastq.gz	2014/05/02 13:55	1,060,684 KB	GZ ファイル
SRR444599.fastq.gz	2014/05/02 14:56	1,489,645 KB	GZ ファイル
SRR444600.fastq.gz	2014/05/02 14:09	1,382,869 KB	GZ ファイル
SRR444601.fastq.gz	2014/05/02 14:29	1,444,211 KB	GZ ファイル
SRR444602.fastq.gz	2014/05/02 14:42	1,236,288 KB	GZ ファイル
TAIR10_chr_all.fas	2014/04/02 16:16	118,343 KB	FAS ファイル
TAIR10_GFF3_genes.gff	2014/04/02 16:46	43,105 KB	GFF ファイル
tmp_genome.fasta	2014/05/02 23:08	121,538 KB	FASTA

コード実行後、確かに
tmp_genome.fastaが作成されている

Step2: マッピングおよびカウントデータ取得

```

in_f1 <- "srp011435_samplename.txt" #入力ファイル名を指定してin_f1に格納(RNA-seqファイル)
in_f2 <- "tmp_genome.fasta" #入力ファイル名を指定してin_f2に格納(リファレンス配列)
in_f3 <- "TAIR10_GFF3_genes.gff" #入力ファイル名を指定してin_f3に格納(GFF3またはGTF形式のアノテー
out_f1 <- "srp011435_QC_bowtie_2.pdf" #出力ファイル名を指定してout_f1に格納
out_f2 <- "srp011435_count_bowtie_2.txt" #出力ファイル名を指定してout_f2に格納
out_f3 <- "srp011435_genelength_2.txt" #出力ファイル名を指定してout_f3に格納
out_f4 <- "srp011435_RPKM_bowtie_2.txt" #出力ファイル名を指定してout_f4に格納
#out_f5 <- "srp011435_transcript_seq_2.fa" #出力ファイル名を指定してout_f5に格納
out_f6 <- "srp011435_other_info1_2.txt" #出力ファイル名を指定してout_f6に格納
param_mapping <- "-m 1 --best --strata -v 2" #マッピング時のオプションを指定
param3 <- "gene" #カウントデータ取得時のレベルを指定: "gene", "exon", "promoter"

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

#マッピングおよびQCレポート用ファイル作成
time_s <- proc.time() #計算時間を計測するため
out <- qAlign(in_f1, in_f2, alignmentParameter=param_mapping, #マッピングを行うqAlign関数を実行した結果
             splicedAlignment=F) #マッピングを行うqAlign関数を実行した結果をoutに格納
time_e <- proc.time() #計算時間

```

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るので積極的に活用。7時間程度かかるので実行しないで!!

```

R Console
> #ファイルに保存
> writeXStringSet(fasta, file=out_f, for$
> getwd()
[1] "C:/Users/kadota/Desktop/SRP011435"
> |

```

Step2: マッピングおよびカウントデータ取得

```
out <- qAlign(in_f1, in_f2, alignmentParameter=param_mapping, #マッピングを行うqAlign関数を実行した結果をoutに格納
             splicedAlignment=F) #マッピングを行うqAlign関数を実行した結果をoutに格納
time_e <- proc.time() #計算時間を計測するため
```

無事マッピングが終了すると、指定した6つのファイルが生成されているはずです。

1. QCLレポートファイル([srp011435_QC_bowtie_2.pdf](#)): Quality Controlレポートです。よく利用されるFastQCのようなものです。
2. カウントデータファイル([srp011435_count_bowtie_2.txt](#)): グループ(サンプル)間での発現変動遺伝子同定に用います。
3. 遺伝子配列長情報ファイル([srp011435_genelength_2.txt](#)): 配列長とカウント数の関係を調べたいときなどに用います。これはおまけです。
4. RPKM補正後のファイル([srp011435_RPKM_bowtie_2.txt](#)): 同一サンプル内での発現レベルの大小関係を知りたいときなどに用います。
5. その他の各種情報ファイル([srp011435_other_info1_2.txt](#)): 論文作成時に必要な、マッピング時に用いたオプション情報、マップされたリード数、Rおよび用いたパッケージのバージョン情報などを含まれます。

	DEX_bio1 tec1	DEX_bio1 tec2	DEX_bio2tec1	DEX_bio2tec2	mock_bio1 tec1	mock_bio1 tec2	mock_bio2tec1	mock_bio2tec2
AT1G01010	257	206	253	249	245	240	254	257
AT1G01020	383	344	276	269	320	322	386	308
AT1G01030	290	229	228	198	325	274	310	304
AT1G01040	2969	2397	2416	2054	2634	2334	2508	2322
AT1G01050	2139	1902	1448	1281	2188	2011	2169	1834
...								

私はカウントデータを入力として
その後の各種解析を行います

Step2: マッピングおよびカウントデータ取得

■ マッピングに必要な情報

- FASTQファイル: 8個の*.fastq.gz
- リストファイル: `srp011435_samplename.txt` →
- リファレンスゲノム: `TAIR10_chr_all.fas`

FileName	SampleName
SRR444595.fastq.gz	DEX_bio1 tec1
SRR444596.fastq.gz	DEX_bio1 tec2
SRR444597.fastq.gz	DEX_bio2 tec1
SRR444598.fastq.gz	DEX_bio2 tec2
SRR444599.fastq.gz	mock_bio1 tec1
SRR444600.fastq.gz	mock_bio1 tec2
SRR444601.fastq.gz	mock_bio2 tec1
SRR444602.fastq.gz	mock_bio2 tec2

■ カウントデータ取得に必要な情報

- 遺伝子アノテーションファイル: `TAIR10_GFF3_genes.gff`

ゲノム上の遺伝子座標情報ファイルを読み込んでいるから遺伝子ごとのカウントデータを取得可能なんです

リストファイル中に記載した任意のサンプル名がカウントデータファイルのヘッダー行となる

カウントデータファイル: `srp011435_count_bowtie_2.txt`

	DEX_bio1 tec1	DEX_bio1 tec2	DEX_bio2 tec1	DEX_bio2 tec2	mock_bio1 tec1	mock_bio1 tec2	mock_bio2 tec1	mock_bio2 tec2
AT1G01010	257	206	253	249	245	240	254	257
AT1G01020	383	344	276	269	320	322	386	308
AT1G01030	290	229	228	198	325	274	310	304
AT1G01040	2969	2397	2416	2054	2634	2334	2508	2322
AT1G01050	2139	1902	1448	1281	2188	2011	2169	1834
...								

トランスクリプトーム解析

- シロイヌナズナのRNA-seqデータを一通りRで解析
 - 2群間比較用: 4 DEX-treated vs. 4 mock-treated
 - 生データ(FASTQファイル)のID: GSE36469

[Development](#), 2012 Jun;139(12):2161-9. doi: 10.1242/dev.075069. Epub 2012 May 9.

RBE controls microRNA164 expression to effect floral organogenesis.

[Huang T¹](#), [López-Giráldez F](#), [Townsend JP](#), [Irish VF](#).

⊕ Author information

Abstract

The establishment and maintenance of organ boundaries are vital for animal and plant development. In the *Arabidopsis* flower, three microRNA164 genes (MIR164a, b and c) regulate the expression of CUP-SHAPED COTYLEDON1 (CUC1) and CUC2, which encode key transcriptional regulators involved in organ boundary specification. These three miR164 genes are expressed in distinct spatial and temporal domains that are crucial for their function. Here, we show that the C2H2 zinc finger transcriptional repressor encoded by RABBIT EARS (RBE) regulates the expression of all three miR164 genes. Furthermore, we demonstrate that RBE directly interacts with the promoter of MIR164c and negatively regulates its expression. We also show that the role of RBE in sepal and petal development is mediated in part through the concomitant regulation of the CUC1 and CUC2 gene products. These results indicate that one role of RBE is to fine-tune miR164 expression to regulate the CUC1 and CUC2 effector genes, which, in turn, regulate developmental events required for sepal and petal organogenesis.

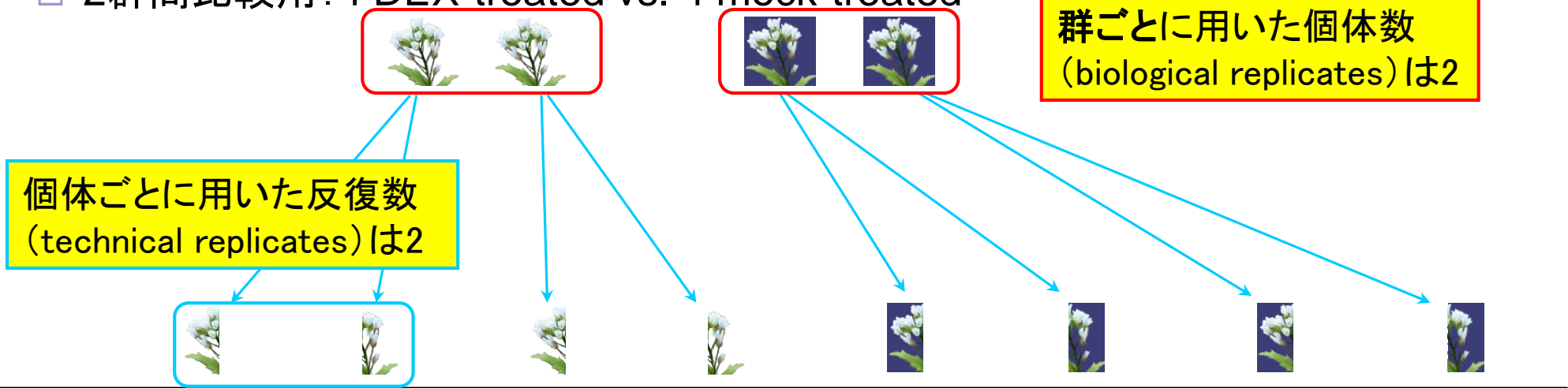
PMID: 22573623 [PubMed - indexed for MEDLINE] [Free full text](#)

ここまでで、生データ取得から
カウントデータ生成まで終了

トランスクリプトーム解析

■ 実験デザイン再確認

□ 2群間比較用: 4 DEX-treated vs. 4 mock-treated



	DEX_bio1 tec1	DEX_bio1 tec2	DEX_bio2tec1	DEX_bio2tec2	mock_bio1 tec1	mock_bio1 tec2	mock_bio2tec1	mock_bio2tec2
AT1G01010	257	206	253	249	245	240	254	257
AT1G01020	383	344	276	269	320	322	386	308
AT1G01030	290	229	228	198	325	274	310	304
AT1G01040	2969	2397	2416	2054	2634	2334	2508	2322
AT1G01050	2139	1902	1448	1281	2188	2011	2169	1834
...								

個体数は2群合わせて4個体

Step3: サンプル間クラスタリング

Step3. サンプル間クラスタリング:

カウントデータ([srp011435_count_bowtie_2.txt](#))を用いてサンプル間の全体的な類似度を眺めることを目的として、サンプル間クラスタリングを行います。類似度は「1-Spearman相関係数」、方法は平均連結法で行っています。TCC論文(Sun et al., 2013)のFig.3でも同じ枠組みでクラスタリングを行った結果を示していますので、英語論文執筆時の参考にどうぞ。PearsonではなくSpearmanで行っているのは、ダイナミックレンジが広いので、順序尺度程度にしておいたほうが良いだろうという思想が一番大きいです。log2変換してダイナミックレンジを圧縮してPearsonにするのも一般的には「アリ」だとは思いますが、マップされたリード数が100万以上あるにも関わらずRPKMデータを用いると、RPKM補正後の値が1未満のものがかなり存在すること、そしてlogをとれるようにゼロカウントデータの処理が必要ですがやりかた次第で結果がこころこわりうるという状況が嫌なので、RNA-seqデータの場合には私はSpearman相関係数にしています。また、ベクトルの要素間の差を基本とするdistance metrics (例: ユークリッド距離やマンハッタン距離など)は、比較的最近のRNA-seqデータ正規化法 (TMM: Robinson and Oshlack, 2010, Tbt: Kadota et al., 2012, TCC; Sun et al., 2013)論文の重要性が理解できれば、その類似度は少なくともfirst choiceにならないと思われます。つまり、サンプルごとに転写物の組成比が異なるため、RPMやCPMのような総リード数を補正しただけのデータを用いて「サンプル間の数値の差」に基づいて距離を定めるのはいかになものか? という思想です。逆に、ユークリッド距離などを用いてクラスタリングを行った結果と比較することで、転写物の組成比に関する知見が得られるのかもしれませんが、さらに、全体的な発現レベルが低いものを予めフィルタリングしておく必要もあるのだろうとは思いますが。このあたりは、真の回答はありませんので、(手持ちのデータにこの類似度を適用したときの理論上の短所をきちんと理解したうえで)いろいろ試すというのは重要だとは思いますが。

ここではカウントデータでクラスタリングをしています。おそらく配列長補正後のRPKMデータ([srp011435_RPKM_bowtie_2.txt](#))でも得られる樹形図のトポロジー(相対的な位置関係)はほぼ同じになるのではないかと思います。配列長補正の有無で、サンプル間の相関係数の値自体は変わりますが、同じグループに属するサンプルであれば反復実験間でそれほど変わらないので、多少順位に変動があっても全体としては相殺されるはずですが、確認はありません。

```
in_f3 <- "srp011435_count_bowtie_2.txt" #入力ファイル名を指定してin_f3に格納
out_f6 <- "srp011435_count_cluster.png" #出力ファイル名を指定してout_f6に格納
param_fig <- c(500, 400) #ファイル出力時の横幅と縦幅を指定(単位はピクセル)
```

#入力ファイルの読み込み

```
data <- read.table(in_f3, header=TRUE, row.names=1, sep="\t", quote="") #指定したファイルを読み込み
dim(data) #オブジェクトdataの行数と列数を表示
```

#前処理(フィルタリング)

```
obj <- as.logical(rowSums(data) > 0) #条件を満たすかどうかを判定した結果をobjに格納
data <- unique(data[obj,]) #objがTRUEとなる行のみ抽出し、ユニークパターンのみにした結果をdata1に格納
dim(data) #オブジェクトdataの行数と列数を表示
```

```
#条件を満たすかどうかを判定した結果をobjに格納
#objがTRUEとなる行のみ抽出し、ユニークパターンのみにした結果をdata1に格納
#オブジェクトdataの行数と列数を表示
```

**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
全選択できるので積極的に活用。**

Step3: サンプル間クラスタリング実行結果

```

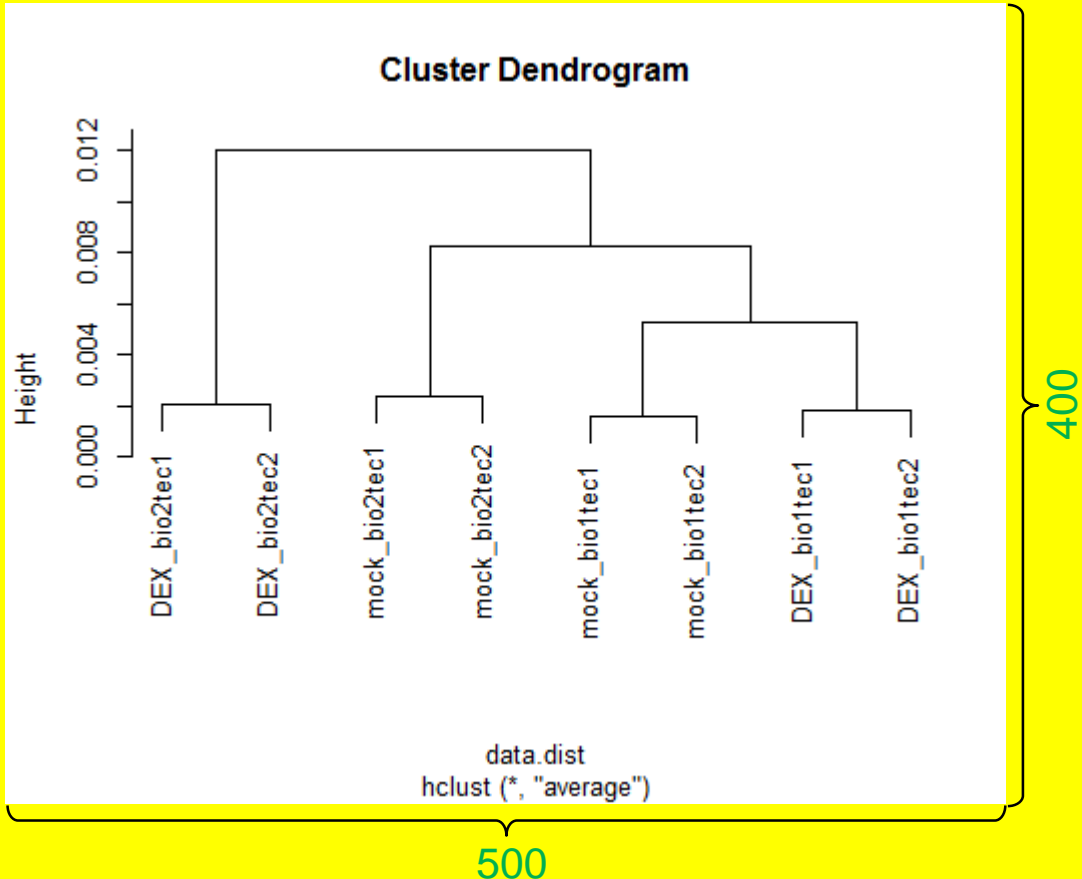
in_f3 <- "srp011435_count_bowtie_2.txt"#入力ファイル名を指定してin_f3に格納
out_f6 <- "srp011435_count_cluster.png"#出力ファイル名を指定してout_f6に格納
param_fig <- c(500, 400) #ファイル出力時の横幅と縦幅を指定(単位はピクセル)

#入力ファイルの読み込み
data <- read.table(in_f3, header=TRUE, row.names=1, sep="\t", quote="")#指定したファイルの読み込み
dim(data) #オブジェクトdata

#前処理(フィルタリング)
obj <- as.logical(rowSums(data) > 0) #条件を満たすかどうか
data <- unique(data[obj,]) #objがTRUEとなるデータのみを抽出
dim(data) #オブジェクトdata

#クラスタリングおよび結果の保存
data.dist <- as.dist(1 - cor(data, method="spearman"))
out <- hclust(data.dist, method="average")#階層的クラスタリング
png(out_f6, pointsize=13, width=param_fig[1], height=param_fig[2])
plot(out) #樹形図(デンドログラム)
dev.off() #おまじない
    
```

出力: srp011435_count_cluster.png



Step4: 発現変動遺伝子(DEG)同定の前に

Step4. 発現変動遺伝子(DEG)同定:

カウントデータファイル([srp011435_count_bowtie_2.txt](#))を入力として2群間で発現の異なる遺伝子の検出を行います。このデータはtechnical replicatesを含むので、それをマージしたのちbiological replicatesのデータにしてからTCCパッケージ(Sun et al., 2013)の推奨ガイドラインに従って、iDEGES/edgeR正規化(Sun et al., 2013; Robinson et al., 2010; Robinson and Oshlack, 2010; Robinson and Smyth, 2008)を行ったのち、edgeRパッケージ中のan exact test (Robinson and Smyth, 2008)を行って、DEG検出を行っています。[解析 | 発現変動 | 2群間 | 対応なし | 複製あり | iDEGES/edgeR-edgeR\(Sun_2013\)](#)の記述内容と基本的に同じです。technical replicatesデータのマージ。ここでは、アドホックに2列分ごとのサブセットを抽出し、行の総和を計算したのち、結合しています。

```
in_f <- "srp011435_count_bowtie_2.txt" #入力ファイル名を指定してin_fに格納
out_f <- "srp011435_count_bowtie_3.txt" #出力ファイル名を指定してout_fに格納
```

#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で指定したファイルの読み込み
head(data) #確認してるだけです
```

#本番(technical replicatesをマージ)

```
DEX_bio1 <- rowSums(data[,1:2]) #サブセットを抽出し、行の総和を計算
DEX_bio2 <- rowSums(data[,3:4]) #サブセットを抽出し、行の総和を計算
mock_bio1 <- rowSums(data[,5:6]) #サブセットを抽出し、行の総和を計算
mock_bio2 <- rowSums(data[,7:8]) #サブセットを抽出し、行の総和を計算
out <- cbind(DEX_bio1, DEX_bio2, mock_bio1, mock_bio2) #列方向で結合した結果をoutに格納
head(out) #確認してるだけです
```

#ファイルに保存

```
tmp <- cbind(row.names(out), 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out_fで指定したファイル名で保存
```

Technical replicatesデータのマージを行います

Step4: 発現変動遺伝子(DEG)同定の前に

入力: `srp011435_count_bowtie_2.txt`

	DEX_bio1 tec1	DEX_bio1 tec2	DEX_bio2tec1	DEX_bio2tec2	mock_bio1 tec1	mock_bio1 tec2	mock_bio2tec1	mock_bio2tec2
AT1G01010	257	206	253	249	245	240	254	257
AT1G01020	383	344	276	269	320	322	386	308
AT1G01030	290	229	228	198	325	274	310	304
AT1G01040	2969	2397	2416	2054	2634	2334	2508	2322
AT1G01050	2139	1902	1448	1281	2188	2011	2169	1834
...								

Technical replicatesデータのマージを行います

出力: `srp011435_count_bowtie_3.txt`

	DEX_bio1	DEX_bio2	mock_bio1	mock_bio2
AT1G01010	463	502	485	511
AT1G01020	727	545	642	694
AT1G01030	519	426	599	614
AT1G01040	5366	4470	4968	4830
AT1G01050	4041	2729	4199	4003
...				

Step4: 発現変動遺伝子(DEG)同定

入力: srp011435_count_bowtie_3.txt

G1群

	DEX_bio1	DEX_bio2	mock_bio1	mock_bio2
AT1G01010	463	502	485	511
AT1G01020	727	545	642	694
AT1G01030	519	426	599	614
AT1G01040	5366	4470	4968	4830
AT1G01050	4041	2729	4199	4003
...				

G2群

```

in_f4 <- "srp011435_count_bowtie_3.txt" #入力ファイル
out_f7 <- "srp011435_DEG_bowtie.txt" #出力ファイル
out_f8 <- "srp011435_MAplot_bowtie.png" #出力ファイル
out_f9 <- "srp011435_other_info2.txt" #出力ファイル
param_G1 <- 2 #G1群のサンプル数
param_G2 <- 2 #G2群のサンプル数
param_FDR <- 0.05 #DEG検出時のFDR
param_fig <- c(430, 390) #MA-plotのx軸とy軸の範囲
    
```

```

#必要なパッケージをロード
library(TCC) #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_f4, header=TRUE, row.names=1, sep="\t", quote="") #in_f4で指定したファイルの読み込み

#前処理(TCCクラスオブジェクトの作成)
data.c1 <- c(rep(1, param_G1), rep(2, param_G2)) #G1群を1、G2群を2としたベクトルdata.c1を作成
tcc <- new("TCC", data, data.c1) #TCCクラスオブジェクトtccを作成

#本番(正規化)
tcc <- calcNormFactors(tcc, norm.method="tmm", test.method="edger", #正規化を実行した結果をtccに格納
                      iteration=3, FDR=0.1, floorPDEG=0.05) #正規化を実行した結果をtccに格納
    
```

```

in_f4 <- "srp011435_count_bowtie_3.txt" #入力ファイル名を指定してin_f4に格納
out_f7 <- "srp011435_DEG_bowtie.txt"   #出力ファイル名を指定してout_f7に格納
out_f8 <- "srp011435_MAplot_bowtie.png" #出力ファイル名を指定してout_f8に格納
out_f9 <- "srp011435_other_info2.txt"  #出力ファイル名を指定してout_f9に格納
param_G1 <- 2                          #G1群のサンプル数を指定
param_G2 <- 2                          #G2群のサンプル数を指定
param_FDR <- 0.05                      #DEG検出時のfalse discovery rate (FDR)閾値を指定
param_fig <- c(430, 390)               #MA-plot描画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(TCC)

```

```

#入力ファイルの読み込み

```

```

data <- read.table(in_f4, header=TRUE, row.names=1, sep="\t")

```

```

#前処理(TCCクラスオブジェクトの作成)

```

```

data.cl <- c(rep(1, param_G1), rep(2, param_G2)) #G1群とG2群のサンプル数を指定
tcc <- new("TCC", data, data.cl)                #TCCクラスオブジェクトの作成

```

```

#本番(正規化)

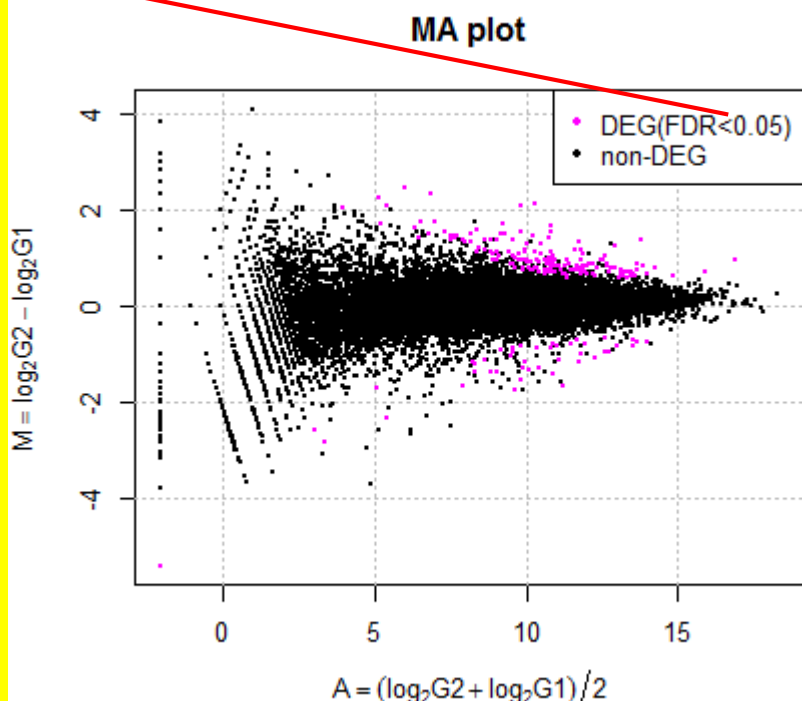
```

```

tcc <- calcNormFactors(tcc, norm.method="tmm", test.method="edgeR",
                       iteration=3, FDR=0.1, floorPDE=1)

```

出力: [srp011435_MAplot_bowtie.png](#)



390

430

無事計算が終了すると、指定した3つのファイルが生成されているはずで

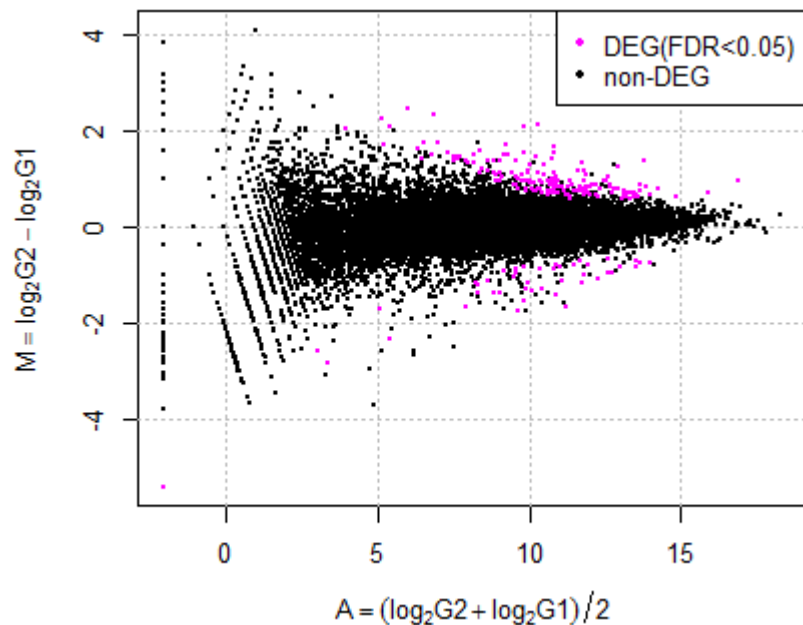
1. 発現変動解析結果ファイル([srp011435 DEG bowtie.txt](#)): iDEGESソートすると発現変動の度合い順になります。"q.value"列の情報(左側の実数の数値データはiDEGES/edgeR正規化後のデータです)は列長補正は掛かっておりませんのでご注意ください。
2. M-A plotファイル([srp011435 MAplot bowtie.png](#)): M versus A plot。軸が $\log(G2/G1)$ で、0より下がG1群で高発現、0より上がG2群で高発現を示します。
3. その他の各種情報ファイル([srp011435 other info2.txt](#)): FDR < 0.05のDEGを用いたパッケージのバージョン情報(特にTCC)などを含みます。

出力ファイルの説明

正規化後のデータ

p-valueとその順位

rownames(tc)	DEX_bio1	DEX_bio2	mock_bio1	mock_bio2	gene_id	a.value	m.value	p.value	q.value	rank	estimatedDEG
AT1G72600	9082.6	8177.4	21137.5	23583.5	AT1G72600	13.76	1.37	2.10E-17	3.42E-13	1	1
AT1G72610	9262.3	8276.6	21490.9	23970.6	AT1G72610	13.79	1.37	2.50E-17	3.42E-13	2	1
AT3G29030	1239.7	796.2	2771.7	3625.3	AT3G29030	10.82	1.65	1.01E-12	9.25E-09	3	1
AT2G01520	5414.6	4799.1	2348.0	2125.2	AT2G01520	11.72	-1.19	2.38E-12	1.63E-08	4	1
AT1G57750	5789.0	4168.3	12115.8	11477.4	AT1G57750	12.90	1.24	1.51E-11	8.27E-08	5	1
AT5G65730	2541.8	1806.7	4768.2	6148.9	AT5G65730	11.75	1.33	1.32E-10	6.03E-07	6	1
AT4G16370	2696.9	2056.6	5462.7	4953.7	AT4G16370	11.78	1.13	7.67E-10	3.00E-06	7	1
AT1G09750	4299.7	4440.2	8047.8	9880.0	AT1G09750	12.61	1.04	9.55E-10	3.27E-06	8	1
AT4G13410	24.6	29.4	144.3	147.3	AT4G13410	5.97	2.43	2.20E-09	6.69E-06	9	1
AT2G42200	2799.0	2223.2	5803.0	4800.7	AT2G42200	11.83	1.08	5.45E-09	1.49E-05	10	1



M-A plotのA値とM値

q-value

FDR閾値判定結果。q-value < 0.05
を満たすDEGが1、non-DEGが0。

まとめ

■ Rでゲノム解析

- シロイヌナズナゲノムのGC含量計算
 - multi-FASTAファイルの読み込み
 - 関数やオプションの利用法
 - パッケージの説明

■ Rでトランスクリプトーム解析

- シロイヌナズナのRNA-seqデータを一通り解析
 - 公共DBからの生データ取得
 - マッピングおよびカウントデータ取得
 - サンプル間クラスタリング
 - 発現変動遺伝子 (DEG) 検出



Rでいろいろできます



スライドPDFはウェブから取得可能です

- [はじめに](#) (last modified 2014/01/30)
- [参考資料\(講義、講習会、本など\)](#) (last modified 2014/05/01) **NEW**
- [過去のお知らせ](#) (last modified 2014/04/21) **NEW**

参考資料(講義、講習会、本など) **NEW**

- [Rのインストールと起動](#) (last modified 2014/01/30)
- [サンプルデータ](#) (last modified 2014/01/30)
- [書籍 II について](#) (last modified 2014/01/30)
- [書籍 | トランスクリプトーム解析](#)
- [書籍 | トランスクリプトーム解析](#)
- [書籍 | トランスクリプトーム解析](#)
- [書籍 | トランスクリプトーム解析](#)
- [書籍 | トランスクリプトーム解析](#)
- [書籍 | トランスクリプトーム解析](#)
- [書籍 | トランスクリプトーム解析](#)

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

書籍

- 門田幸二著(金明哲 編), シリーズ Useful R 第7巻トランスクリプトーム解析, 共立出版, 2014. [ISBN: 978-4-320-12370-0](#)
内容: マイクロアレイとRNA-seq解析を例としてRを用いてトランスクリプトーム解析を行うための体系的な本としてまとめました。数式が苦手なヒト向けに、重みつき平均の具体的な計算例などを挙げてオプションの意味などがわかるような自身の理解に重点を置いた構成にしております。書籍中のRコードは「[書籍 | トランスクリプトーム解析 | ...](#)」をご覧ください。
- 門田幸二, 「トランスクリプトミクスの推奨データ解析ガイドライン」, ニュートリゲノミクスを基盤としたバイオマーカーの開発, シーエムシー出版, 45-52, 2013. [ISBN: 978-4-7813-0820-3](#)
内容: マイクロアレイ解析の話がメインです。実験デザインの重要性を述べています。Affymetrix GeneChipデータの数値化と発現変動遺伝子(DEG)検出法の組合せの重要性の話や、サンプル間クラスターリングである程度DEGに関する情報がわかることを述べています。MAS5データを用いる場合は特に倍率変化で議論することも無意味であること、RMAのようなマルチアレイ正規化法を用いて得られたマイクロアレイデータの場合にはなぜ倍率変化でうまくいく傾向にあるかなどの理由をM-A plotを用いて説明しています。

講習会、講義、講演資料

- 門田幸二, 「[講義資料](#)」, [アグリバイオインフォマティクス教育研究プログラム](#)の大学院講義科目: [ゲノム情報解析基礎](#), 東京大学(東京), 2014.04.30
内容: Rで塩基配列解析を行うための基本的なところ。例題としてシロイヌナズナゲノムのCpG出現頻度を解析し考察。Rパッケージのインストール、エラーメッセージへの対処法、利用可能な関数の概観。sequence logosを主な講義内容とし、エントロピー計算や、なぜエントロピーをそのまま利用せずに情報量に変換するかの意義。subseq関数のオプションをうまく利用して効率的に目的のプロモーター配列領域を切り出して計算するやり方など。課題4はプログラムの一部を任意に変更する基礎的な能力を問うもの。他の例題の中に回答が存在するので、それを効率的に見つける能力を見ている。講義自体はスライド39までで、スライド40以降はうまくいかないこともあるという事例やRのバージョンの違いに気を付ける的な話。「[農学生命情報科学特論](#)」で改めて話す予定。1コマ(90 min)分。

今後の予定

門田 幸二のホームページ

講演など(上記講義以外) (last modified: 2014.05.09)

32. 門田幸二,「Rでゲノム・トランスクリプトーム解析: CpG解析から機能解析まで」, [HPCI講習会・バイオインフォマティクス実習コース](#), 産業技術総合研究所ゲノム情報研究センター(東京), 2015.03.05-06
31. 門田幸二,「フリーソフト Rを用いたデータ解析:塩基配列解析を中心に」, [生命医薬情報学連合大会2014, 中級者向けバイオインフォマティクス入門講習会](#), 東北大学(宮城), 10:50-12:20, 2014.10.04
30. 門田幸二,「ビッグデータ解析とR」, [生命医薬情報学連合大会2014, HPCIワークショップ「医療とビッグデータ解析」](#), 東北大学(宮城), 9:00-10:30, 2014.10.04
29. 門田幸二,「3. データ解析基礎」, [バイオインフォマティクス人材育成カリキュラム\(次世代シーケンサ\) 速習コース](#), 東京大学(東京), 2014.09.08-09
28. 門田幸二,「演題未定」..., 2014.06.26-27
27. 門田幸二,「[比較トランスクリプトーム解析とその周辺:モデル, 正規化, 発現変動検出など](#)」, [よく分かる次世代シーケンサー解析ワークショップ](#), 九州大学(福岡), 2014.03.19
26. 門田幸二,「[Rでゲノム・トランスクリプトーム解析](#)」, [HPCIチュートリアル・バイオインフォマティクス実習コース](#), 産業技術総合研究所生命情報工学研究センター(東京), 2014.03.07
25. 門田幸二,「[トランスクリプトーム解析の現況2013\(詳細版\)](#)」, 東京大学大学院農学生命科学研究科第123回アグリバイオインフォマティクスセミナー, 東京大学(東京), 2013.11.01
24. 門田幸二,「[トランスクリプトーム解析の現況:マイクロアレイ vs. RNA-seq](#)」, [生命医薬情報学連合大会「オミックス・計算そして創薬」](#)・オミックス解析における実務者意見交換会, タワーホール船堀(東京), 2013.10.30

研究
トランス
クリプト
ームによ
る応用を
めまです
ます
配列解

NGS速習コース開催(9/1~12@東大農)

実施日	実施時間	大項目	項目番号および項目	習得技術	レベル	形式	担当講師(敬称略)					
9月1日	10:40-12:00	1. コンピュータリテラシーとサーバー設計	1-1. OS、ハード構成	コンピューターの基本の理解	初級	講義	中村保一(DBBJ)					
	13:15-14:45		1-2. ネットワーク基礎	インターネット、セキュリティの基本の理解	初級	講義	中村保一(DBBJ)					
	15:00-16:30		1-3. UNIX I	UNIXの基礎の理解 Linux導入		初級	実習	仲里猛留(DBCLS)				
	16:45-18:15							仲里猛留(DBCLS)				
9月2日	10:30-12:00							仲里猛留(DBCLS)				
	13:15-14:45							仲里猛留(DBCLS)				
9月3日	15:00-16:30		1-4. スクリプト言語	Perl シェルスクリプト	中級	実習	山口昌雄(アメリエフ)					
	9月4日						10:30-12:00	山口昌雄(アメリエフ)				
							13:15-14:45	山口昌雄(アメリエフ)				
	15:00-16:30						山口昌雄(アメリエフ)					
9月5日	16:45-18:15		2. 配列インフォマティクス	2-1. 配列解析基礎	配列、ゲノムデータ記述のフォーマット、アラインメント(DP)、データベース検索(BLAST, BLAT)等の基礎的な配列比較解析の原理と実習	初級	実習	坊慶秀雅(DBCLS)				
	9月6日							10:30-12:00	2-2. バイオ系データベース概論	基本的な各種バイオ系データベースの理解、統合DBの利用法	初級	実習
		13:15-14:45						小野浩雅(DBCLS)				
	9月8日	15:00-16:30						3. データ解析基礎	3-1. R基礎1	R言語の基礎(インストールから利用まで)	初級	実習
16:45-18:15		3-2. R基礎2	ファイルの読み込み、行列演算の基本	初級	実習	門田幸二(東京大学)						
9月9日		10:30-12:00	3-3. R各種パッケージ	Rの各種パッケージのインストール法と代表的なパッケージの利用法	中級	実習	門田幸二(東京大学)					
		13:15-14:45	3-4. R bioconductor I	Bioconductorの利用法	中級	実習	門田幸二(東京大学)					
9月10日	15:00-16:30	3-5. R bioconductor II	FASTAandFASTQ形式ファイルの読み込み ファイル形式の変換(FASTQ→FASTA)、クオリティチェック、リード配列長分布、フィルタリングやトリミング、GC含量計算など	中級	実習	門田幸二(東京大学)						
	16:45-18:15					門田幸二(東京大学)						
9月10日	10:30-12:00	4. 次世代シーケンサ	4-1. 次世代シーケンサ基礎I	原理の理解	初級	講義	倉田哲也(NAIST)					
	13:15-14:45		4-2. 次世代シーケンサ基礎II	応用分野とそのための計測技術の理解 (RNA-seq, ChIP-seq, がんゲノム, 個人ゲノム, 環境ゲノム, Hi-C)	初級	講義	倉田哲也(NAIST)					
	15:00-16:30		4-3. 次世代シーケンサ実習I	ファイル形式、可視化、quality check、マッピング、アセンブル	初級	実習	山口昌雄(アメリエフ)					
	16:45-18:15						山口昌雄(アメリエフ)					
9月11日	10:30-12:00		4-4. 次世代シーケンサ実習II	代表的なパイプラインについての実習: 多型解析(IGV)	初級	実習	山口昌雄(アメリエフ)					
	13:15-14:45						山口昌雄(アメリエフ)					
	15:00-16:30						山口昌雄(アメリエフ)					
	16:45-18:15						山口昌雄(アメリエフ)					
9月12日	10:30-12:00	6. 分子生命科学	6-1. 分子生命科学概論	複製、転写、翻訳、代謝、シグナル伝達などの基礎知識	初級	講義	河岡慎平(ATR)					
	9月12日						13:15-14:45	6-2. オミクス概論	ゲノム以外のオミクスデータの基礎知識	河岡慎平(ATR)		
							15:00-16:30				6-3. 遺伝/進化概論	ゲノムデータを扱う上での遺伝学、進化学の基礎知識
	16:45-18:15						5. ゲノム関連の倫理・法律	5-1. ゲノム情報倫理概論	ゲノム情報を扱う上で、プライバシー保護などの必要な倫理的問題、法的問題の国内外の状況を理解し、ゲノム情報を適切に利用できるようにする。匿名化、暗号化、情報セキュリティ概要	眞輪真理(NBDC) 川崎美苗(NBDC)		

謝辞

共同研究者

清水 謙多郎 先生(東京大学・大学院農学生命科学研究科)

西山 智明 先生(金沢大学・学際科学実験センター)

孫 建強 氏(東京大学・大学院農学生命科学研究科・大学院生)

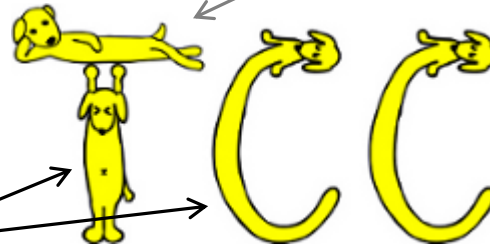
グラント

- 基盤研究(C)(H24-26年度):「シーケンスに基づく比較トランスクリプトーム解析のためのガイドライン構築」(代表)
- 新学術領域研究(研究領域提案型)(H22年度-):「非モデル生物におけるゲノム解析法の確立」(分担;研究代表者:西山智明)



挿絵やTCCのロゴなど

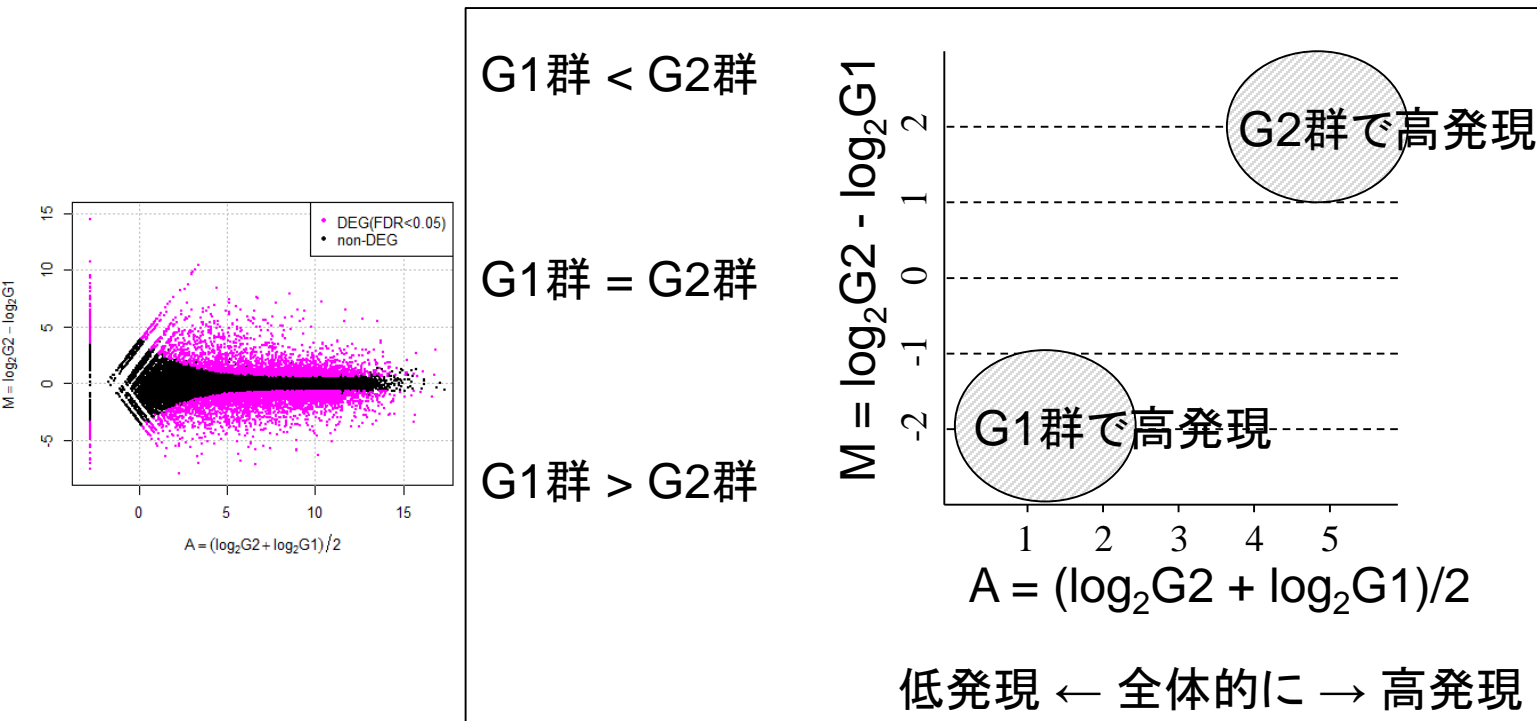
(妻の)門田 雅世さま作



(有能な秘書の)三浦 文さま作

M-A plot

- 2群間比較用
- 横軸が全体的な発現レベル、縦軸がlog比からなるプロット
- 名前の由来は、おそらく対数の世界での縦軸が引き算 (Minus)、横軸が平均 (Average)



DEGが存在しないデータのM-A plotを眺めることで、縦軸の閾値のみに相当する倍率変化を用いたDEG同定の危険性が分かります

多重比較問題：FDRって何？

■ p -value (false positive rate; FPR)

- 本当はDEGではないにもかかわらずDEGと判定してしまう確率
- 全遺伝子に占めるnon-DEGの割合 (分母は遺伝子総数)
- 例：10,000個のnon-DEGからなる遺伝子を p -value < 0.05 で検定すると、 $10,000 \times 0.05 = 500$ 個程度のnon-DEGを間違ってDEGと判定することに相当
 - 実際のDEG検出結果が900個だった場合：500個は偽物で400個は本物と判断
 - 実際のDEG検出結果が510個だった場合：500個は偽物で10個は本物と判断
 - 実際のDEG検出結果が500個以下の場合：全て偽物と判断

■ q -value (false discovery rate: FDR)

- DEGと判定した中に含まれるnon-DEGの割合
- DEG中に占めるnon-DEGの割合 (分母はDEGと判定された数)
- non-DEGの期待値を計算できれば、 p 値でも上位 x 個でもDEGと判定する手段はなんでもよい。以下は10,000遺伝子の検定結果でのFDR計算例
 - $p < 0.001$ を満たすDEG数が100個の場合：FDR = $10,000 \times 0.001 / 100 = 0.1$
 - $p < 0.01$ を満たすDEG数が400個の場合：FDR = $10,000 \times 0.01 / 400 = 0.25$
 - $p < 0.05$ を満たすDEG数が926個の場合：FDR = $10,000 \times 0.05 / 926 = 0.54$

