

基礎生物学研究所
ゲノムインフォマティクス・トレーニングコース2015秋

RNA-seq 入門

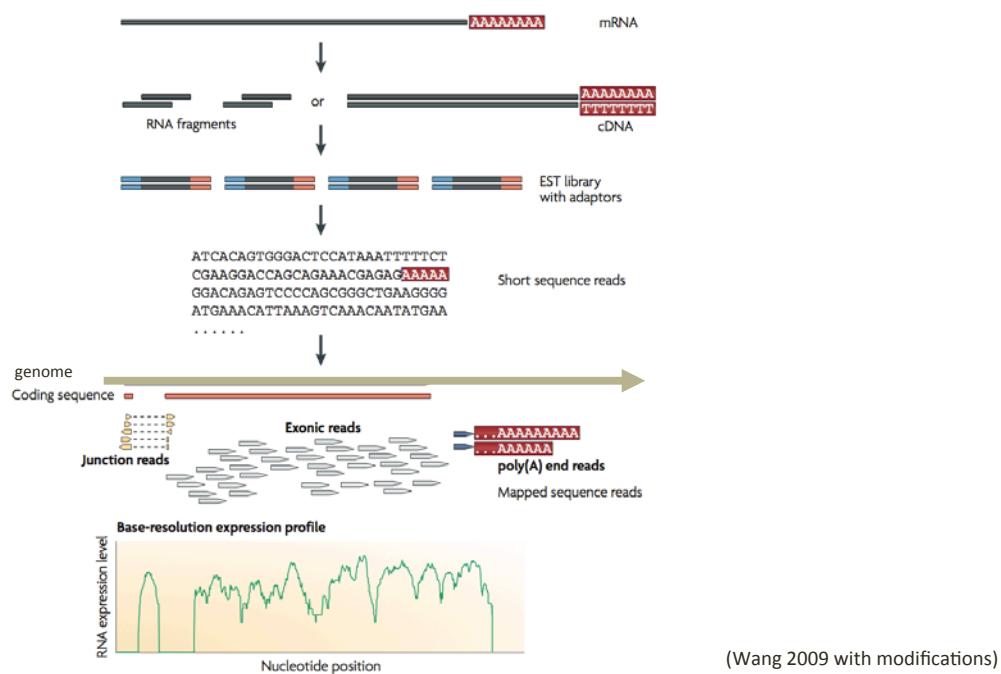
～概論～

September 9-11, 2015 in NIBB (Okazaki)

重信秀治 / Shuji Shigenobu

RNA-seq

RNA-seq is a revolutionary tool for transcriptomics using deep-sequencing technologies.

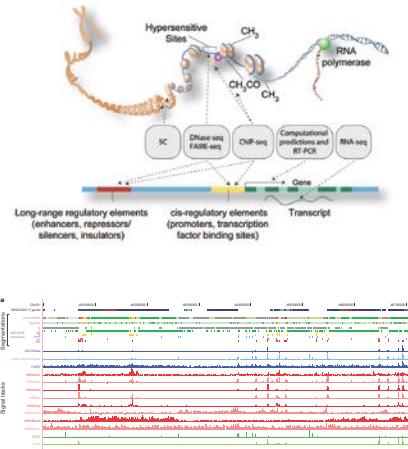


RNA-seq is unraveling complexities of eukaryotic transcriptomes in model and non-model organisms

- Gene expression analysis
- Novel gene discovery (model org.)
 - Coding and non-coding genes
- Gene cataloguing (non-model org.)
- Anti-sense transcripts
- RNA editing
- Novel splicing variants & fusion genes
- Allele-specific expression

Beyond transcriptome

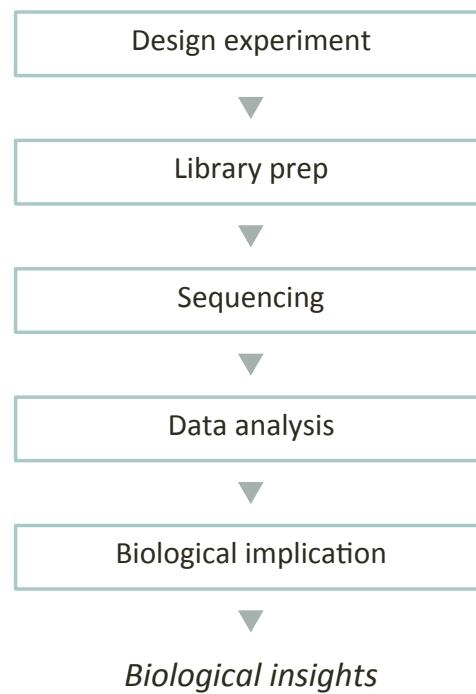
- DB for proteome analysis
- SNP finding
- *and more ...*



Two major goals of RNA-seq

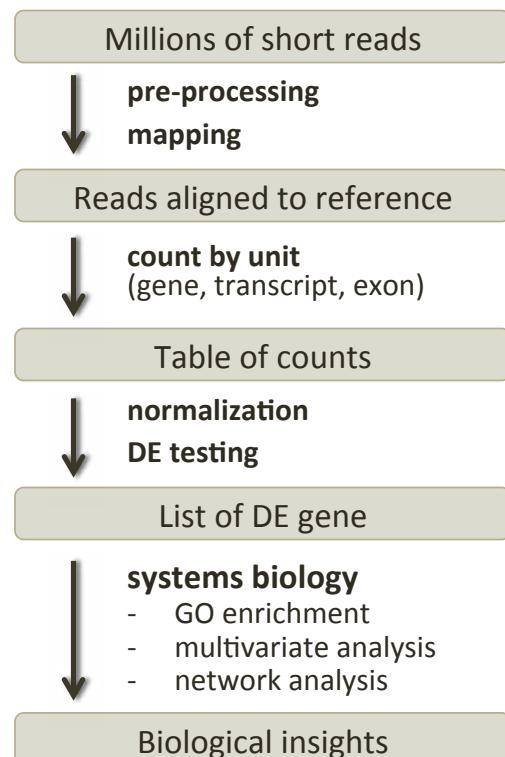
- Gene cataloguing
- Gene expression analysis

General workflow of NGS studies



RNA-seq analysis pipeline for DE

Differential Expression analysis



解析ツールの現状: RNA-seq

- 全てのプロセスをこなせる万能ツールはない。
- それぞれのステップに特化したツール群が次々に登場している。

基本戦略

- 各ステップに最適なツールをチョイス、組み合わせた、解析パイプラインの構築。

Pipeline

- 本コースで学ぶオススメの2つのパイプライン
 - Genome-based: TopHat/Cufflinks
 - Transcriptome-based: Trinity/Bowtie/eXpress/edgeR

Biologist が身に付けるべき 4つの informatics skills for RNA-seq

- 業界標準のツールとフォーマット
- UNIXの基礎
- 統計的な考え方と技術
- 初級のprogramming

勉強法

- コマンドやプログラムは自分で試してみる。
- スクリプトやコマンドはcopy & pasteでなくタイピングすること。
(熊楠メソッド)
- 気軽に質問する。
- その一方で、ヘルプやマニュアルドキュメントを活用すること。
- わからないことがあったら、隣や前後の受講生にもきいてみましょう。
- この3日間のトレーニングコースはあくまでも出発点であると認識し、学習を継続する。
- 良いツールを選んだら、とことんまで使い倒し、すみずみまで熟知する。
- 良い教科書を選ぶ。そしてその1冊を読みこむ。
- 新しい情報やネットにあふれる情報に翻弄されない。
- 学んだことを同僚に教える。
- 自分の研究との接点を常に意識する。自分の研究に応用する。データをあらゆる角度から解析し味わい尽くす。
- ゴールは何か？問うべきBiological questionを忘れない。

Support

- サポートWiki

<https://github.com/nibb-gitc/gitc2015a-rnaseq/wiki>

講義や演習の補足情報を提供します。

コース終了後もフォローアップページとして存続します。

- USB disk

講義資料および、講義や演習に使ったデータが保存してあります。持ち帰って、復習に活用下さい。



shige@nibb.ac.jp

UNIX 入門

基礎生物学研究所
ゲノムインフォマティクストレーニングコース
2015

西出 浩世

hiroyo@nibb.ac.jp

UNIX を使う理由

- UNIX でしか使えないアプリケーション
 - 最新の研究用ソフト、並列化や巨大メモリに対応したソフト
- たくさんの処理を一度に行なう
 - スクリプトを用いたコマンドの連続実行
- 独自のプログラムを作成
 - Perl, Ruby等のスクリプト言語、豊富な開発ユーティリティ
- WWWサーバやデータベースサーバを立ち上げたい
 - サーバとしての高い安定性、apache や postgres などのフリーウェア

PCでUNIXを使うには

MacOS X	OS自体がUNIX #	アプリケーション→ユーティリティ→ターミナルでUNIX端末が使用できる
	リモートログイン	UNIXサーバへリモートログイン ターミナルからsshを使用する
Windows	Cygwin	Windows上で動作するUNIXライクな環境
	VMware + Linux	仮想マシンを構築してLinuxそのものをインストールする
	リモートログイン	UNIXサーバへリモートログイン TeraTermからsshを使用する

#|) フリーウェアなどのインストールが必要な場合は「MacOSXでのUNIX環境構築方法」
を参照してください

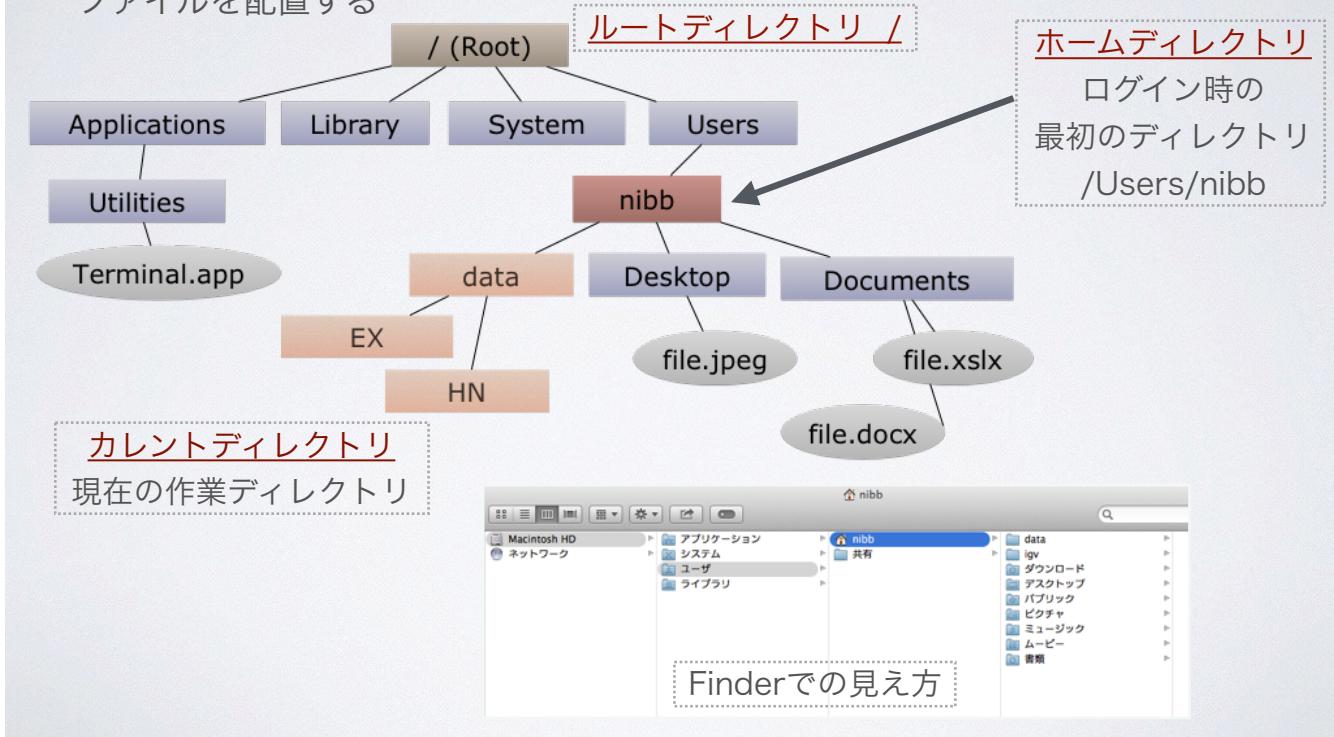
実習 1

キーボード配置とUNIX環境の確認



UNIXのファイルシステム：階層型ディレクトリ

トップのルートディレクトリ下に、子ディレクトリ、孫(...)ディレクトリがあり、
ファイルを配置する



UNIXにおけるディレクトリの表現

UNIXではディレクトリを上へ下へ移動しつつ作業する

名前	表現方法	説明
ルートディレクトリ	/ (スラッシュ)	ファイルシステムの頂点
カレントディレクトリ	. (ドット)	起点とするディレクトリ 今いる場所
親ディレクトリ	.. (ドット2つ)	カレントディレクトリの上の ディレクトリ
ホームディレクトリ	~ (チルダ)	ユーザ用のディレクトリ ログイン直後にいる場所

プログラム、コマンドが作成するファイルは、
使った時のカレントディレクトリに作成される

ファイル/ディレクトリ名の指定方法

• パス (Path)

- ファイルやディレクトリを指定する記述
- UNIXではディレクトリ名をスラッシュ (/) で区切る
- 絶対パスと相対パスの2つの記述法

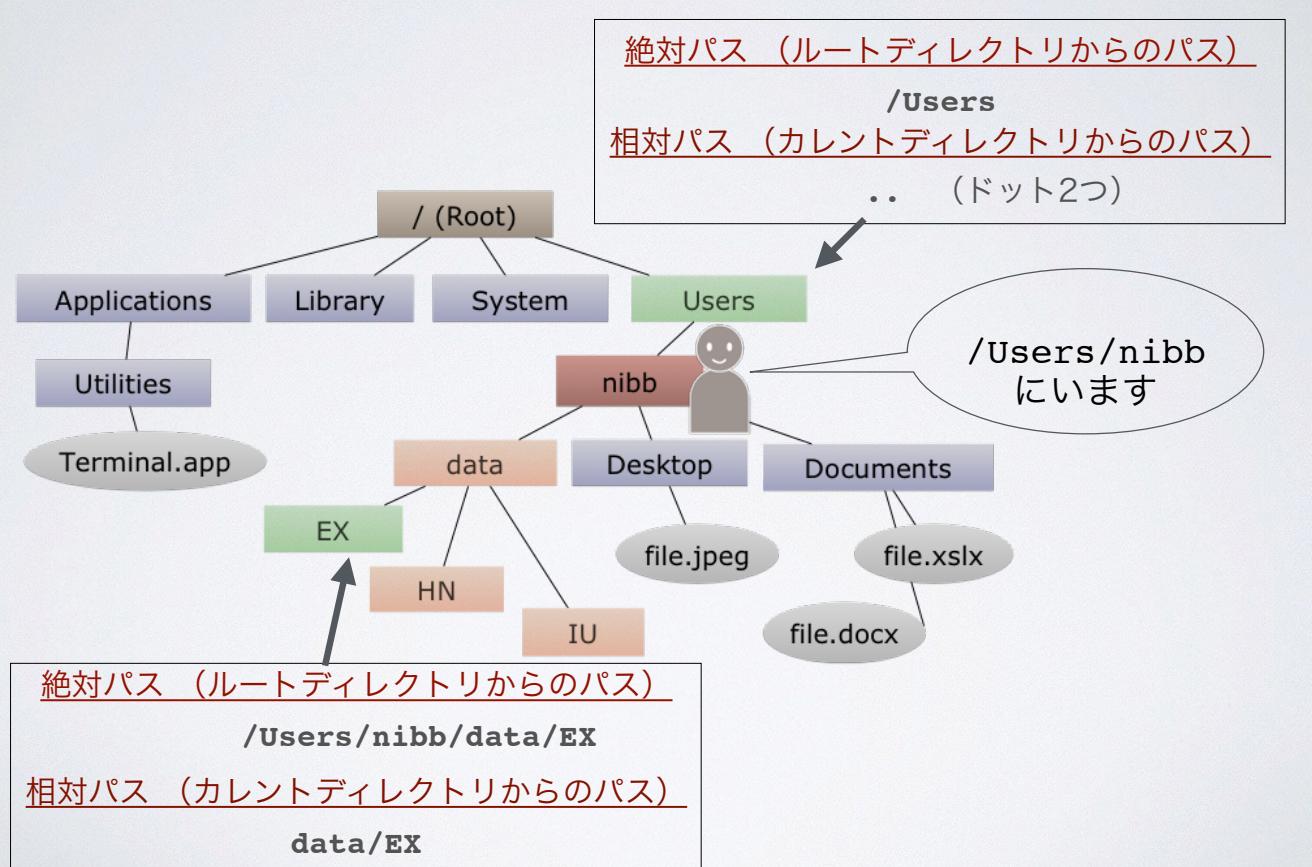
• 絶対パス

- ルートディレクトリから目的のファイル・ディレクトリへの道筋の記述
- 行頭は必ずルートディレクトリ (/) となる
- 例 : /Users/nibb/data/HN/readme.txt

• 相対パス

- 現在位置から目的のファイル・ディレクトリへの道筋の記述
- 例 : data/HN/readme.txt

ファイル/ディレクトリ名の指定方法

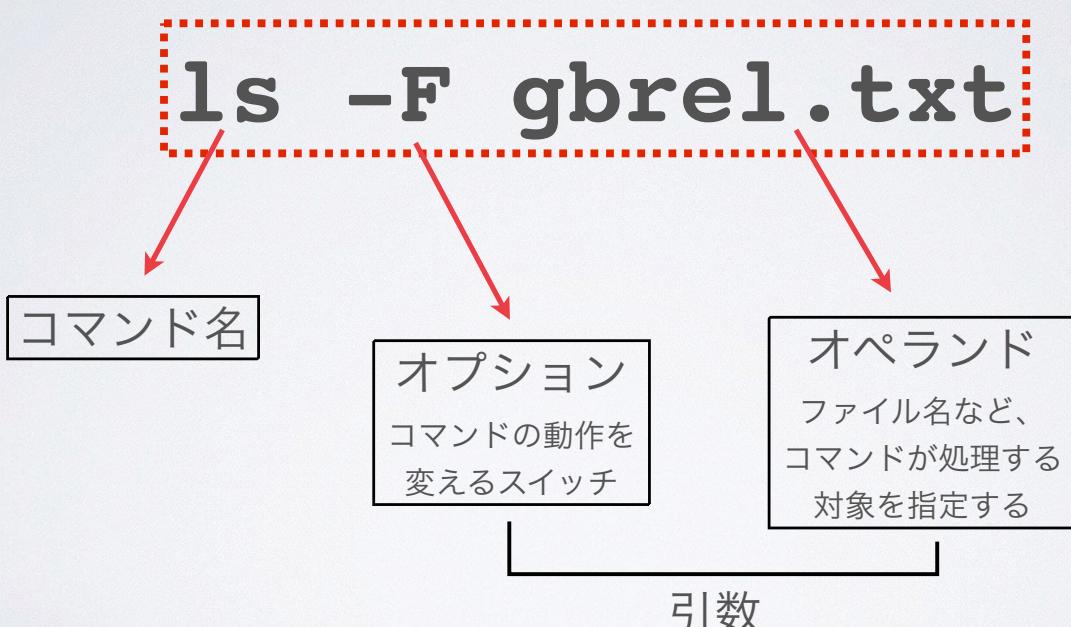


ディレクトリの中身を見る (`ls`)

- **`ls`**
 - カレントディレクトリの内容（ファイル名のリスト）を表示する
 - **`ls ディレクトリ名`**
 - 指定したディレクトリの内容を表示
- `$ ls data` ディレクトリdataの内容を表示
`$ ls /` ルートディレクトリ / の内容を表示
`$ ls ..` 一つ上のディレクトリの内容表示
`$ ls .` ただの `ls` と同じ
- **`ls -F`**
 - ファイル名の末尾にファイルの種類に応じた記号をつけて表示
 /ディレクトリ、@シンボリックリンク、*実行権付きファイル
 - **`ls -a`**
 - ファイル名の先頭がドット（.）で始まる隠しファイルも表示する

実習 2

UNIX コマンドの基本形



補完機能

- コマンド名やファイル/ディレクトリ名の補完

- ファイル名を途中まで入力して、タブ (Tab) キーを押す

\$ ls d と入力してタブキーを押す

- 一意に決まらない場合は、一意に展開できるところまでを展開する

\$ ls data/HN/sprot/143 と入力してタブキーを押す。

- それ以上展開できない場合は、再度（つまり2回）タブキーを押すと、候補ファイルのリストを表示する

\$ ls data/HN/sprot/1433 の状態で2回続けてタブキーを押す

- 入力を著しく効率的にするので、積極的な活用を！
 - ・複雑なファイル名の入力ミスを防ぐ
 - ・うろ覚えのファイル名でも入力できる
 - ・ファイル名だけでなく、コマンド名でも同様の補完機能が使える

実習 4

コマンドヒストリと編集

- 過去に実行したコマンドの呼び出しとコマンドの編集

- Control キーと同時に p を押す…

キー操作（十字キー代替）	動き
Control + p (↑キー)	1つ前のコマンド
Control + n (↓キー)	1つ後のコマンド
Control + b (←キー)	カーソルを左に移動
Control + f (→キー)	カーソルを右に移動
Control + a	カーソルを行頭に移動
Control + e	カーソルを行末に移動

- 引数の影響で長くなってしまったコマンドを再実行する際に便利

実習 5

ディレクトリを移動する (cd)

- **cd** ディレクトリ名

- 指定したディレクトリに移動（カレントディレクトリの変更）

`$ cd data` dataディレクトリに移動

`$ cd ..` 一つ上のディレクトリに移動

`$ cd ~/data/HN` ホームディレクトリ以下の /data/HN に移動

- **cd**

- ディレクトリ名を省略すると、ホームディレクトリに移動する

- **pwd**

- 現在のディレクトリ名の確認

実習 6

ワイルドカード

- ファイル名やディレクトリ名を指定するときに使う「任意の文字」
- パターンマッチによって複数のファイル名を一度に指定できる。
 - ***** 任意の文字列（0文字以上）とマッチ
 - 例) `ls *.fasta` 例) `ls *_HUMAN*`
 - **?** 任意の1文字とマッチ
 - 例) `ls 1A2?_HUMAN.fasta`
 - **[1-5]** 1から5までの数字とマッチ
 - 例) `ls 1A2[1-5]*.fasta`
 - **{文字列1, 文字列2}** "文字列1"または"文字列2"とマッチ
 - 例) `ls 1A25_HUMAN.{fasta,phylip}`

実習 7

ファイルの内容を一括表示する(cat)

- **cat** ファイル名

- ファイルの内容を表示する。

```
$ cat 1A25_HUMAN.fasta
```

- ファイル名を複数指定すると内容を連結して表示する。

```
$ cat *.fasta
```

実習 8

ファイルの部分表示 (head, tail)

- **head** [-行数] ファイル名

- ファイルの先頭から指定した行数（指定しないと10行）だけ出力する

```
$ head 1A25_HUMAN.sprot      (最初の10行を出力)
```

- 大きなファイルの内容をとりあえず確認したい場合などに便利

- **tail** [-行数] ファイル名

- ファイルの最後から指定した行数を出力する

```
$ tail -20 1A25_HUMAN.sprot  (最後の20行を出力)
```

- -n +行数 とすると、先頭から数えて指定された行以降を出力する

```
$ tail -n +2 1A25_HUMAN.fasta
```

(先頭行を除去して2行目以降を出力)

実習 9

ファイルの内容を見る (less)

- **less** ファイル名

- ファイルの内容をページビューで見る
- ページの移動には独自のキー操作が必要

キー操作	動き
スペースキー, f	1ページ先へ進む
b	1ページ前に戻る
j	1行先へ進む
k	1行前に戻る
g	先頭行へ移動
G	最終行へ移動
/[パターン]	現在位置から後にパターン検索をして移動
?[パターン]	現在位置から前にパターン検索をして移動
n	検索を再実行して次のヒットに移動
N	逆方向に検索を再実行して次のヒットに移動
q, zz	lessを終了
h, ?	ヘルプを表示する

実習10

ディレクトリの作成と削除 (mkdir, rmdir)

- **mkdir** ディレクトリ名

- 新規ディレクトリの作成
 - \$ **mkdir unixtest**

- **rmdir** ディレクトリ名

- ディレクトリの削除。
- ただし、ディレクトリ内にファイルがあると削除できない

実習11

ファイルのコピー (cp)

- **cp file1 file2**
 - file1 のコピーを file2 として作成
- **cp file1 [file2...] dir**
 - file1 (, file2, ...) のコピーを、dir 内に同一名のファイルとして作成
 - 同一名ファイルが既に存在しているときは上書きされる
- **cp -r dir1 dir2**
 - dir1 を dir2 としてディレクトリごとコピーを作成
 - dir2 が存在する場合は、dir2 以下に dir1 として作成

実習 1 2

ファイル/ディレクトリ名の変更 ファイル/ディレクトリの移動 (mv)

- **mv file1 file2**
 - file1 の名前を file2 に変更する
- **mv file1 [file2...] dir**
 - file1 (, file2, ...) を、dir 内に移動する。
 - 同一名ファイルが既に存在しているときは上書きされる



実習 1 3

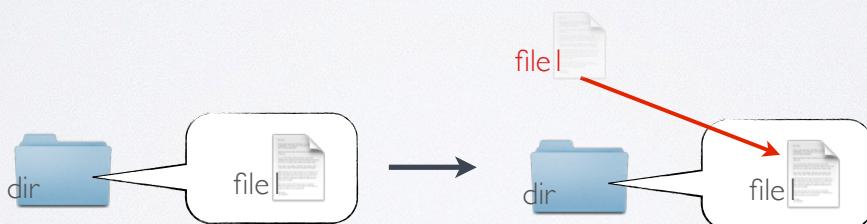
シンボリックリンクの作成 (ln -s)

- **ln -s ファイル/ディレクトリ名 シンボリックリンク名**

- ファイル/ディレクトリの「別名」を作成する。
- Macのエイリアス、Windowsのショートカットに相当

```
$ ln -s dir/file1 .
```

- dir内のfile1のシンボリックリンクをカレントディレクトリ(.)に作成



- オリジナルファイルが移動・消去されると、シンボリックリンクを通じた参照はできなくなる。

実習 14

ファイル情報の表示 (ls -l)

- **ls -l ファイル/ディレクトリ名**

- ファイルの大きさや最終更新日などの情報のリスト表示（名前順）

- **ls -lt ファイル/ディレクトリ名**

- 上記と同様だが日付順に表示

```
$ ls -l 1433B_HUMAN.sprot
```

```
-rw-r--r-- 1 nibb staff 21675 2011-03-09 05:23 1433B_HUMAN.sprot
```

ファイルの種類	アクセス権
-: 通常のファイル	
d: ディレクトリ	
l: シンボリックリンク	

所有者とグループ

サイズ

更新日付

ファイル名

実習 15

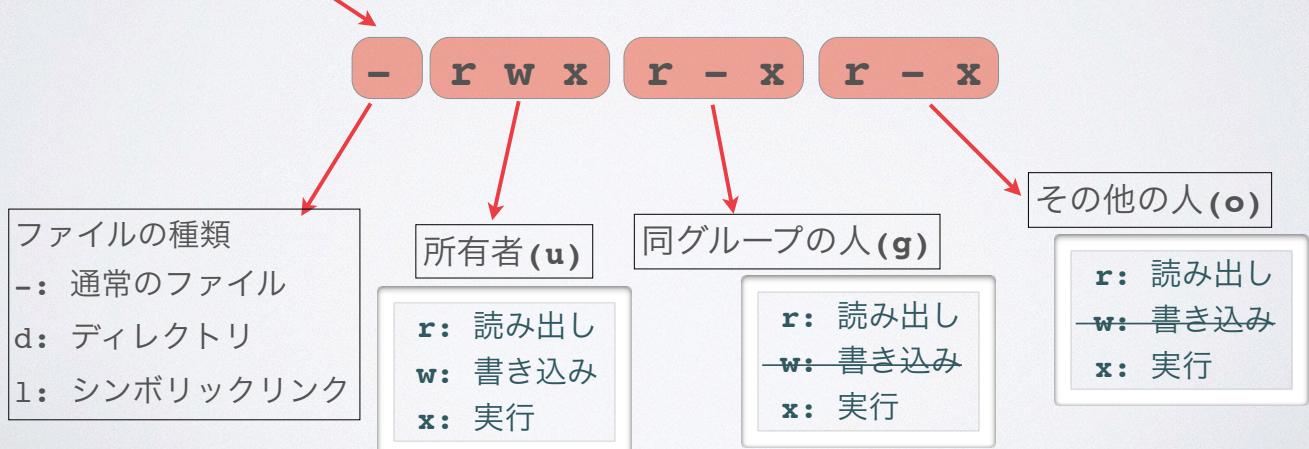
ファイルのアクセス権 (permission)

- ファイル (ディレクトリ) について、どんな人が、何ができるか
 - どんな人?
 - ファイルの所有者 (u)
 - 同一グループの利用者 (g)
 - その他の利用者 (o)
 - それぞれ何ができる?
 - 読み出し (r)
 - 書き込み (w)
 - 実行 (x)
 - ファイルをプログラムとして実行
 - ディレクトリの場合は、そこに移動できる

アクセス権の確認

- ls -l ファイル/ディレクトリ名

```
$ ls -l myfile
-rwxr-xr-x 1 nibb cproom 21675 2011-03-09 05:23 myfile
```



アクセス権の変更 (chmod)

- **chmod [アクセス権] ファイル名**

- ファイルのアクセス権の変更
- アクセス権の書式：「どの人に」に続けて「+」で付与、「-」で削除

\$ chmod u+x file 所有者に実行権を与える

\$ chmod go-rwx file 所有者以外から全てのアクセス権を除く

- 8進数3桁で各レベルを列挙する

\$ chmod 600 file 所有者のみ読み書き可能

	所有者(u)			グループ(g)			その他(o)		
パーミッション	r	w	x	r	w	x	r	w	x
8進数	4	2	1	4	2	1	4	2	1
設定値	合計値			合計値			合計値		

ファイルの削除 (rm)

- **rm ファイル名**

- 指定したファイルを削除する

- **rm -rf ディレクトリ名**

- ディレクトリの中身を確認なしですべて消去した上でディレクトリを削除

- **rm -ri ディレクトリ名**

- ディレクトリの中身を確認しながら消去

• 「ごみ箱へ移動」ではない

• 完全に削除してしまう

• 削除したファイルを復活することは不可能！

コマンドマニュアルの参照 (man)

- **man** コマンド名

```
$ man ls
```

[]で囲まれている引数は

省略可能

ls - list contents of directory

SYNOPSIS

ls [RadLCxmlnogrtucpFbqisf1AMSDP] [names]

DESCRIPTION

For 指定可能なオプション the conte 下線付きは変数
 each file argument, ls repeats its name and any other information requested. The output is sorted alphabetically by default. When arguments are not given, the current directory is listed. When several arguments are given, the arguments are first sorted appropriately, but file arguments appear before directories and their contents. ls processes supplementary code set characters according to the locale specified in the LC_CTYPE and LC_COLLATE environment variables [see LANG on environ(5)], except as noted under the -b and -q options below

実習 17

行数・単語数のカウント (wc)

- **wc** ファイル名

- ファイルの行数、単語数、文字数を出力する

```
$ wc 1433B_HUMAN.fasta
```

6 25 481 1433B_HUMAN.fasta

- ファイルは、6行、25単語、481文字からなる。

- ファイル名を省略すると、端末から入力待ちの状態になる。適当な文章を入力して Controlキー+d を押すと、入力した文章に対して wc が実行される。

```
$ wc
```

This is a pen.

(Control+d)

1 4 15

(結果、1行、4単語、15文字)

実習 18

パターン検索 (grep)

- **grep パターン ファイル名**

- ファイル中でパターンを含む行を出力する

```
$ grep GO 1433B_HUMAN.sprot
```

- 1433B_HUMAN.sprot から GO を含む行を検索する。

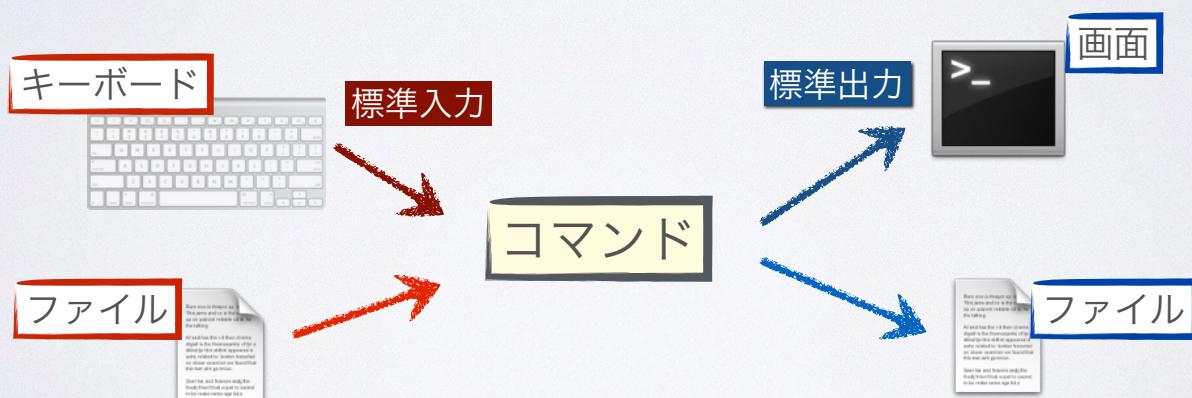
```
$ grep ^FT 1433B_HUMAN.sprot
```

- 1433B_HUMAN.sprot から FT で始まる行を検索する
- ("^" は行の先頭を意味する)。
- **grep -v** とすると (オプション `-v`) パターンを含まない行を出力する
- ファイル名を省略すると、やはり端末から文字列を読み込んでパターンを検索する

実習 19

入出力のリダイレクション

- UNIXは、コマンドへの入力元 および 実行結果の出力先を切り替えられる「リダイレクション機能」を持つ
- 通常、標準入力は端末（キーボード）、標準出力は端末（画面）



エラー内容を出力する「標準エラー出力」もある

リダイレクションの例 (出力)

- 結果を画面に出力

```
$ grep GO 1433B_HUMAN.sprot
```

- 結果をファイルに保存：「>」

- コマンドの右に `> filename` を加える
- 結果が `GO_count` というファイルに格納
- 同名ファイルがある場合は上書きされる

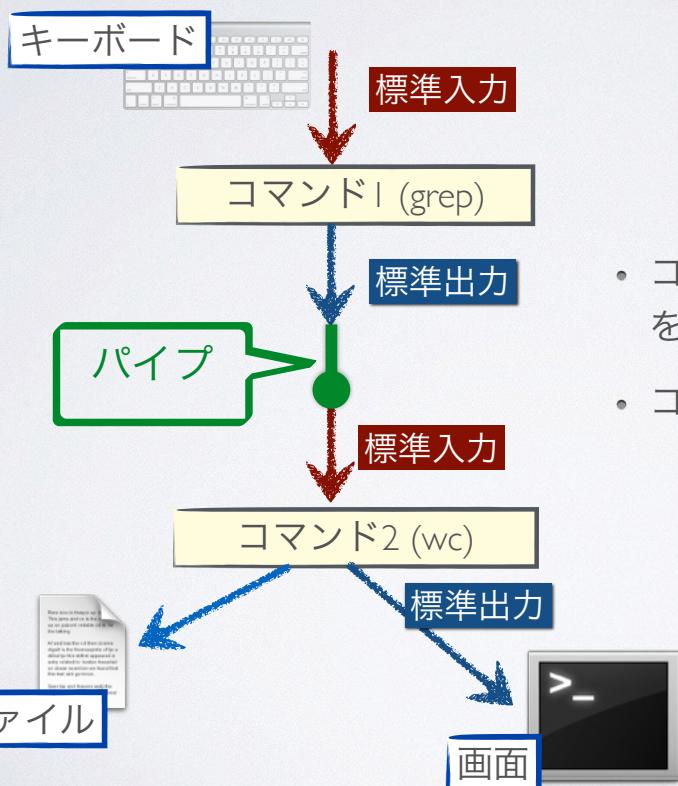
```
$ grep GO 1433B_HUMAN.sprot > GO_count
```

- すでに存在するファイルに追加書き：「>>」

```
$ grep GO 1433B_HUMAN.sprot >> GO_count
```

実習 20

パイプによるコマンドの結合



- コマンド間の標準出力→標準入力を直接つなげる方法
- コマンドでは「|」で表す

パイプ(|)の例

- grep の出力行数をwcで数える

```
$ grep ^FT 1433B_HUMAN.sprot | wc
```

- grep の結果を less で見る

```
$ grep ^FT 1433B_HUMAN.sprot | less
```

- grepの結果をさらにgrepして絞り込む。

```
$ grep ^FT 1433B_HUMAN.sprot | grep HELIX | less
```

- パイプによって、コマンドはいくつでも結合できる

実習 21

シェルスクリプト

- 一連のコマンドを記述したファイル（スクリプトファイル）

- 例) ~/data/HN/sprot/testpg の内容

```
#!/bin/sh
pwd
ls -la
```

- 実行権を与え、コマンドとして実行できる。

```
$ chmod +x testpg
```

- 実行

```
$ ./testpg
```

- コマンドパスの通ったディレクトリに保存すれば、コマンド名のみで実行できる。

実習 22

コマンドパス

- 打ったコマンドは「コマンドパス」と呼ばれるパスのリストの中から探して実行される
 - 例) **ls** コマンドの場所は **/bin/ls**
- コマンドパスは、それぞれのユーザが持つ設定「PATH変数」に格納されている
 - **echo \$PATH** で確認できる。
 - コマンドパスはコロン(:)区切りで複数のパスをつなげて指定する。
 - 同じコマンドが複数のパスに存在するときは、最初のパスのものが優先して実行される
- シェルが、実行プログラムを見つけられない場合、"Command not found" のメッセージが返る。この場合、以下の可能性がある。
 - プログラムが（正しく）インストールされていない
 - コマンドパスが正しく設定されていない

実習 2 3

コマンドパスの設定

- 一時的にパスを設定する場合

```
$ PATH=$PATH:~/bin
```

- ホームディレクトリ配下の bin ディレクトリをパスに追加

- 恒常にパスを設定する場合

- `~/.bash_profile` に下記 2 行を追記

```
PATH=$PATH:~/bin
export PATH
```

- ログイン時に読み込まれる初期設定ファイルに、環境変数（シェルから起動されるプログラムにも引き継がれる変数）として PATH 変数を設定する

実習 2 4

エイリアス

- **alias 別名=コマンド名**

- エイリアスとはコマンドの別名のこと

- エイリアスを新たに作成する

```
$ alias 別名='コマンド名'
```

- エイリアスを取り消す

```
$ unalias 別名
```

```
$ \別名
```

- 現在設定されているエイリアスの一覧を確認

```
$ alias
```

- コマンドの初期設定を変更する際に使用

- 例えば ls の表示結果をカラー対応にする場合

```
$ alias ls='ls -G'
```

メタキャラクタ

- コマンド内で特別な意味を持つ記号

- 例) * ? [] < > | ! \$; & など

- コマンドの引数（ファイル名など）にスペースや記号があるときは要注意

- 引数全体を '' で囲むことにより、スペースやメタキャラクタの特別な意味を抑止できる

```
$ grep '^>' 1A01_HUMAN.fasta
```

- 「>」で始まる行を検索。 '' をつけないと、リダイレクトと解釈され、 1A01_HUMAN.fasta が上書きされる

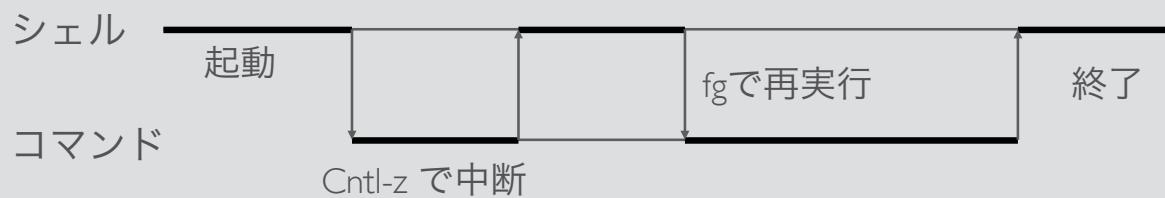
- 基本的に、ファイル名には英数字と「.」「_」だけを用い、それ以外の記号は使わないこと

ジョブの中止と中断

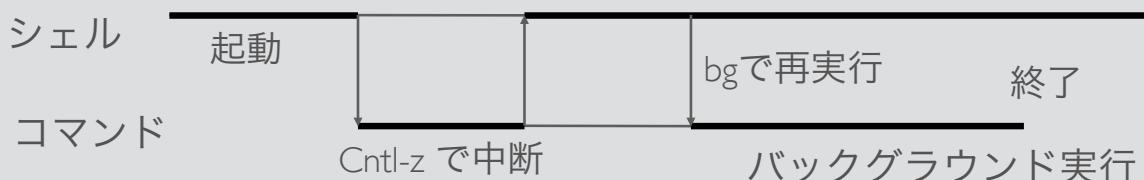
- コマンドを起動すると、端末は待ち状態となり、コマンドを受け付けなくなる
 - 実行中のコマンド（ジョブ）を途中で中止
 - コントロールキーと「c」を同時に押す
\$ Control + c
 - ジョブを一時的に中断
\$ Control + z
 - 一時中断したジョブを再開する
 - fg** フォアグラウンドで再開（端末は結果待ちの状態）
 - bg** バックグラウンドで再開（端末から入力可能）
- はじめからバックグラウンドでジョブを走らせるにはコマンドの最後に & をつける

フォアグラウンドとバックグラウンド

フォアグラウンド実行（実行中はコマンドを受け付けない）



バックグラウンド実行（実行中でもコマンド実行が可能）



マシンの稼働状況を把握する (top)

```
$top
Processes: 262 total, 2 running, 7 stuck, 253 sleeping, 1643 threads
18:51:35
Load Avg: 2.98, 1.81, 1.53 CPU usage: 1.40% user, 1.14% sys, 97.44% idle SharedLibs: 103M resident, 0B data, 17M linkedit.
MemRegions: 125942 total, 9971M resident, 202M private, 2001M shared. PhysMem: 15G used (1918M wired), 8918M unused.
VM: 765G vsize, 1312M framework vsize, 4619(0) swapins, 10377(0) swapouts. Networks: packets: 14584362/19G in, 3809549/380M out. Disks: 3655689/628G read, 3609878/822G
written.

PID  COMMAND %CPU TIME #TH #WQ #PORTS #MREGS MEM RPRVT PURG CMPRS VPRVT VSIZE PGRP PPIID STATE UID FAULTS COW MSGSENT MSGRECV
99935 Calendar 0.0 00:29.88 4 1 196 1198 88M 79M 0B 116K 518M 3227M 99935 180 sleeping 501 140706 1242 276213 106575
94491 ssh 0.0 00:00.34 1 0 19 48 1032K 660K 0B 0B 63M 2429M 94491 1742 sleeping 501 786 125 36 12
86585- Microsoft Wo 0.0 00:17.71 4 1 151 806 99M 68M+ 4096B 0B 362M+ 1263M 86585 180 sleeping 501 63647 2262 23468+ 22257
85695 GitHub Condu 0.0 00:00.36 4 1 130 133 5996K 4672K 0B 0B 266M 2677M 85695 180 sleeping 501 5499 375 4738 1130
83080 com.apple.hi 0.0 00:00.31 2 0 44 84 2864K 2296K 0B 0B 261M 2627M 83080 1 sleeping 501 4881 197 3375 2251
83009 Keynote 0.0 28:29.73 6 1 492 11182 804M 653M 91M 1552K 1209M 6050M 83009 180 sleeping 501 9963602 46302 5801889 1631017
82764 com.apple.Me 0.0 00:10.42 3 1 82 277 110M 109M 0B 0B 326M 2719M 82764 1 sleeping 501 110668 672 5533 2833
82734 Google Chrom 0.0 00:48.58 10 0 101 515 63M 62M 568K 364K 396M 3544M 72524 72524 sleeping 501 70728 1650 301744 83094
81883 com.apple.ap 0.0 00:16.37 4 1 204 334 43M 37M 28K 0B 278M 2941M 81883 1 sleeping 501 126725 599 92332 44584
81366- mi 0.0 00:09.05 4 1 134 428 35M 26M+ 0B 528K 252M+ 975M 81366 180 sleeping 501 45749 640 119888 48991
79429 Google Chrom 1.5 24:49.84 12 0 103 1450 132M 135M 4672K 628K 468M 3628M 72524 72524 sleeping 501 466539 1705 506245+ 308343+
77397 com.apple.hi 0.0 00:00.01 2 0 34 58 1408K 856K 0B 32K 139M 2505M 77397 1 sleeping 501 1688 209 135 63
77257 com.apple.hi 0.0 00:00.01 2 0 34 53 1316K 764K 0B 24K 106M 2472M 77257 1 sleeping 501 1634 208 117 51
77188 Evernote 0.0 00:51.01 16 2 290 1663 104M 82M 0B 1676K 540M 4255M 77188 180 stuck 501 245407 2002 391446 171586
75296 com.apple.au 0.0 00:00.07 2 1 47 83 2168K 1468K 0B 0B 241M 2609M 75296 1 sleeping 501 2462 269 415 183
75295 com.apple.au 0.0 00:00.01 2 1 28 55 1376K 820K 0B 0B 112M 2479M 75295 1 sleeping 501 1668 190 114 41
75293- com.apple.qt 0.0 00:00.14 2 0 72 133 4592K 3032K 0B 0B 217M 822M 75293 1 sleeping 501 2645 406 1603 759
75158 com.apple.We 0.0 00:05.97 17 1 290 1131 75M 71M 132K 0B 668M 3972M 75158 1 sleeping 501 39398 1628 30708 17195
```

- リアルタイムで稼働状況を表示
- q キーで終了

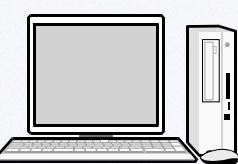
ネットワークを介したサービス

Webサーバ、FTPサーバ



アカウントが不要なサービス

- Web閲覧
 - ファイル転送
- (アノニマスftp, curl, wget)



計算サーバ等

- ### アカウントが必要なサービス
- リモートログイン (ssh)
 - ファイル転送 (sftp, scp)

ファイル転送 (ftp / sftp)

- **ftp ホスト名** または **sftp ホスト名**

- ftpログインしてからのコマンド

コマンド	用途
ls	ホストのファイルの一覧
lls	ローカルのファイルの一覧
cd	ホストのディレクトリ移動
lcd	ローカルのディレクトリ移動
get	ファイルをホストから手元に取得
mget	複数のファイルを取得、ワイルドカード利用
put	ファイルをローカルからホストに送信
mput	複数のファイルを送信、ワイルドカード利用
help	使用できるコマンドの簡易ヘルプ

データ転送 (curl)

- **curl URL**

- HTTPやFTP経由のファイル取得

```
$ curl http://www.nibb.ac.jp
```

- http://www.nibb.ac.jp のトップページを標準出力に書き出し

オプション	用途
-o filename url	urlのデータをfilenameに保存
-O url	url上のファイル名で保存 (http://hoge.com/fig.jpg のように「ファイル名がある」urlのみ)
-O url[start-end]	http://www.hoge.com/[01-10].jpg 等とすることで、01.jpg, 02.jpg, 03.jpg ... 10.jpg のファイルを全て取得
-L url	urlのリダイレクトを追う
-L -Z number url	リダイレクトを追う回数
-h	ヘルプを表示

ファイルの圧縮と解凍

- データ圧縮
 - 一定のアルゴリズムを使い、情報を失わずにファイルサイズを小さくする
 - 圧縮ファイルは使用する前に戻す（解凍）必要がある
- **gzip** :
 - 拡張子 `.gz` または `.Z`
 - 圧縮：**gzip ファイル名** （ファイル名.gzというファイルができる）
 - 解凍：**gunzip ファイル名.gz** または **gzip -d ファイル名.gz**
- **bzip2** :
 - 拡張子 `.bz2`
 - 圧縮：**bzip2 ファイル名** （ファイル名.bz2というファイルができる）
 - 解凍：**bunzip2 ファイル名.bz2** または **bzip2 -d ファイル名.bz2**

アーカイブの作成と展開 (tar)

- アーカイブ
 - 複数のファイルやディレクトリを1つのファイルにまとめること
 - 圧縮と組み合わせて、データ配布やバックアップの保存などに用いる
 - 通常アーカイブファイルには拡張子 `.tar` をつける（自動ではつかない！）
- アーカイブの作成・追加


```
$ tar cvf archive.tar directory
```

 - `directory` を `archive.tar` としてアーカイブ
- アーカイブの展開


```
$ tar xvf archive.tar
```

 - `archive.tar` をカレントディレクトリに展開
- 圧縮・解凍の同時実行
 - オプションに `z` (`gzip`) または `j` (`bzip2`) をつけることで圧縮・解凍を同時に行うこともできる

```
$ tar xzvf archive.tar.gz
```

(`gzip`で解凍しつつ展開)

プログラムのインストール

- アノニマスFTPまたはWebからファイルをダウンロード
- 圧縮ファイルを解凍してアーカイブファイルを展開
- README, INSTALLなどのファイルを見てインストール方法や注意点を確認
- ソースコードをダウンロードした場合はコンパイルする。
一般的には以下のコマンドを順に実行する

\$ configure	計算機環境に合った設定を自動的に行う
\$ make	プログラムをコンパイル
\$ sudo make install	プログラムを適当な場所にコピー

SAMtools 1.2 のインストール

- SAMtoolsのダウンロードと解凍
 - URL : <http://samtools.sourceforge.net/>
 - File : samtools-1.2.tar.bz2
 - 解凍 : tar xvfj samtools-1.2.tar.bz2
- SAMtoolsのコンパイル、インストール、テスト

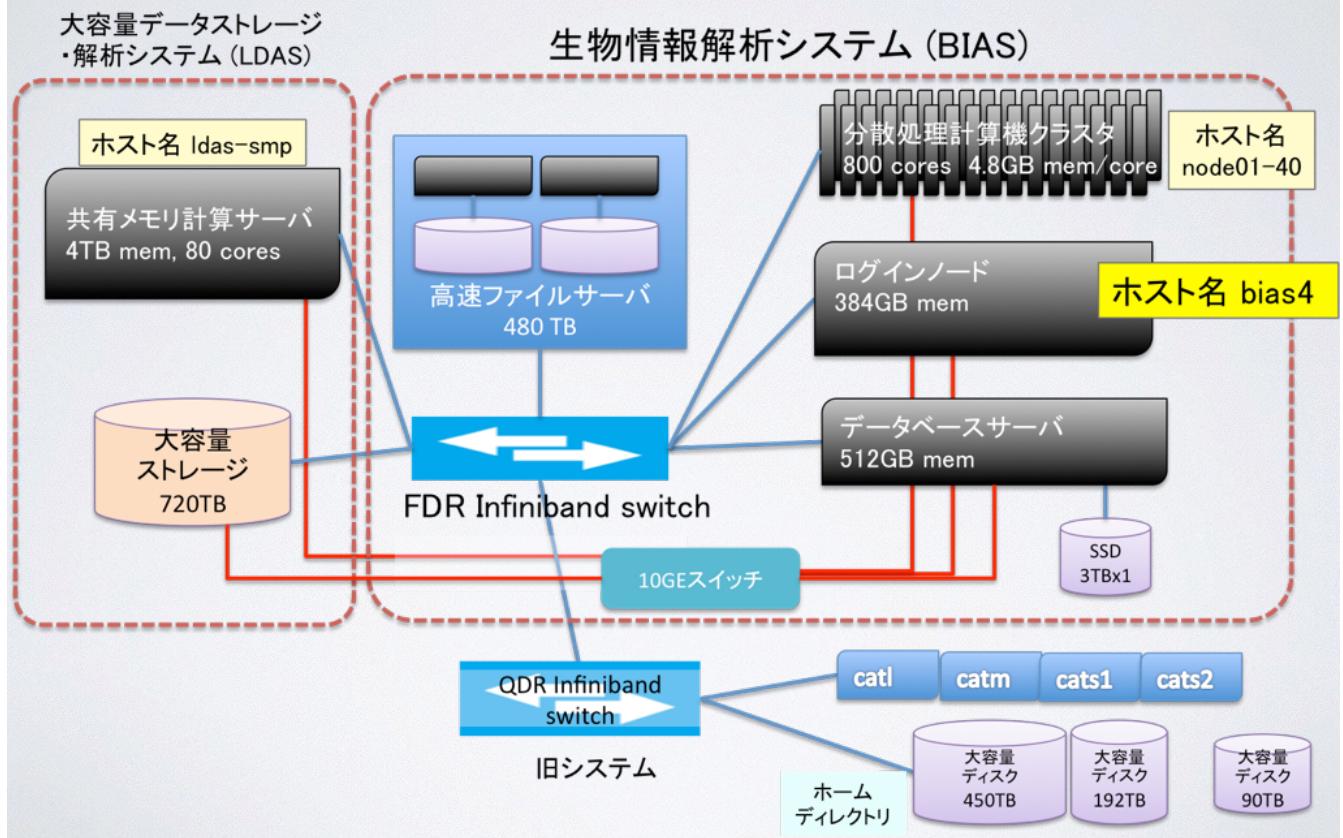
```

$ cd
$ mv Downloads/samtools-1.2.tar.bz2 unixtest
$ cd unixtest
$ tar xvfj samtools-1.2.tar.bz2
$ cd samtools-1.2
$ less INSTALL
$ make prefix=~/install
$ samtools

```

実習 25

基礎生物学研究所 生物情報解析システム



基生研ゲノムインフォマティックストレーニングコース 2015秋
2015.09.09–2015.09.11

NGS基本データフォーマット と基本ツール

基礎生物学研究所
生物機能解析センター
山口勝司

概要

序論

- データフォーマットとは？
- フォーマットを学ぶ理由
- 効率よい学習のポイント

NGS基本データフォーマット

- FASTA FASTQ SRA
- BED GFF/GTF WIG
- SAM/BAM

NGS基本ツール

- SRAtoolkit
- SAMtools
- IGV

概要

序論

- データフォーマットとは？
- フォーマットを学ぶ理由
- 効率よい学習のポイント

NGS基本データフォーマット

- FASTA FASTQ SRA
- BED GFF/GTF WIG
- SAM/BAM

NGS基本ツール

- SRAtoolkit
- SAMtools
- IGV

データフォーマットとは？

データを記録するルール

ルールがあれば情報を効率良く正確に共有できる

例: Webページ → HTMLフォーマット

を使用することで

ハード(PC/スマートフォン)

OS (Windows/Mac)

ソフト(IE/Chrome/Safari)

が違っても、どんな環境でも同じページを閲覧可能

次世代シーケンサー解析では
様々なフォーマットが使われる
これらの把握が解析に必須

フォーマットを学ぶ理由

NGS解析の基礎知識だから

研究者間のコミュニケーションや解析方法の理解に必須

例1) 同僚A : A遺伝子の塩基配列データ見せて
あなた : 了解です。fastaで送りますね

fasta形式が塩基配列情報を含むことを理解していれば、やりとりがスムーズ

例2) マニュアル : このソフトはfastaからtree/phylipファイルを生成します
あなた : 系統解析をするソフトなんだな

入力と出力の形式から行った解析がわかる

研究目的にあわせた解析に必要だから

フォーマットを知ると、そこから自力で必要な情報を獲得できる
これにより、独自性の高い研究が可能になります

例3) 1. 巨大なfastaファイルから配列名だけ取り出したい
2. fasta形式では、配列名の頭に常に">"がつく
3. ">"がある行だけ集めれば、配列名のリストができる！
(エクセルの"並べ替え"機能でできそうだ！)

専用のプログラムがなくても自分がほしい結果を得られる

効率良い学習のポイント

Wet 研究者がつまずく点

1: たくさん形式があって区別がつかない！

- 実態はなじみ深い生物学的情報です
- 各フォーマットが含む生物学的情報や解析で使われる場面に注目しましょう

2: 意味不明な文字がでてくる！

- \$ や#など“意味不明文字”が頻出しますが、実は重要な情報が含まれています
- 「ヒトとコンピュータ、両方に扱いやすい表記」を考えた開発者の努力の結晶です
- 使い方を理解すれば強力な武器になります。がんばって理解しましょう

以上を踏まえて、各フォーマットを見ていきましょう

概要

序論

- データフォーマットとは？
- フォーマットを学ぶ理由
- 効率よい学習のポイント

NGS基本データフォーマット

- FASTA FASTQ SRA
- BED GFF/GTF WIG
- SAM/BAM

NGS基本ツール

- SRAtoolkit
- SAMtools
- IGV

NGS基本データフォーマット

数十以上のフォーマットがあります
頻出フォーマットだけを紹介します

▪ 配列用

FASTA, FASTQ, SRA

▪ アノテーション用

BED, GFF/GTF, WIG

▪ マッピング(アライメント)用

SAM/BAM

FASTA

概要	配列情報の標準フォーマット
内容	塩基配列 アミノ酸配列
例	公共DBからの配列情報ダウンロード

○規則

“>”で始まる行がタイトル行、改行後に配列
タイトル行は改行不可 配列中では改行可能

○ファイル例

```
>gi|31342400 Bos taurus crystallin, gamma S (CRYGS), mRNA ←タイトル行
TGCACCAAAACATGTCTAAAGCTGGAACCAAAATTACTTTCTTGAAAGACAAAA
ACTTTCAAGGCCGCCACTATGACAGCGATTGCGACTGTGCAGATTCCACATG
TACCTGAGCCGCTGCAACTCCATCAGAGTGGAAAGGAGGCACCTGGGCTGTGTA
TGAAAGGCCCA
>gi|31342400 Bos taurus crystallin, gamma S (CRYGS), mRNA
TGCACCAAAACATGTCTAAAGCTGGAACCAAAATTACTTTCTTGAAAGACAAAA
ACTTTCAAGGCCGCCACTATGACAGCGATTGCGACTGTGCAGATTCCACATG
TACCTGAGCCGCTGCAACTCCATCAGAGTGGAAAGGAGGCACCTGGGCTGTGTA
TGAAAGGCCCA
```

FASTQ

概要	NGS結果データの実質的な標準形式
内容	塩基配列、一塩基ごとの品質情報 (Quality value)
例	マッピング、アセンブルでの入力データ形式

○規則

- 1行目：“@”の後にタイトル(配列IDや説明)
- 2行目：塩基配列
- 3行目：“+”の後にタイトル(省略可)
- 4行目：配列のクオリティ
* 配列とクオリティには基本的に改行を入れない

○ファイル例

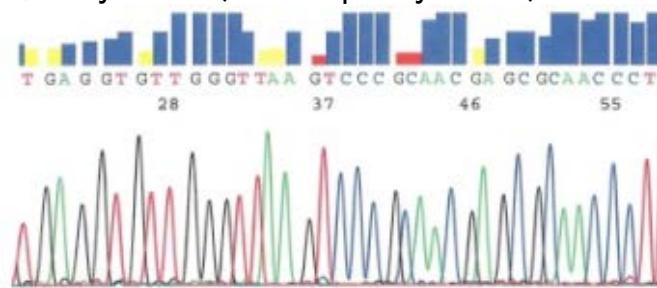
```
@SEQ_ID ←配列ID
GATTTGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTGTTCAACTCACAGTTT ←塩基配列
+ ←配列ID(省略)
!!!!*((((****))%%%++)(%%%%.1***-+*''))**55CCF>>>>>CCCCCCCC65 ←クオリティ
```

実習1-1 lessコマンドでEx1_1.fqの中身を見て、fastq形式を確認しよう

FASTQのポイント

塩基配列の信頼性も示せる

Quality value (Phred quality score)



+
! ' * (((***+) % % + +) (% % %) . 1 ***-
+ * ')) **55C

ABI キャピラリーシーケンサーで
この部分で表されていた値

$QV = -10 \log_{10} p$ (p : 間違った 塩基決定である確率)

$QV = 30 \rightarrow p = 0.001$

$QV = 20 \rightarrow p = 0.01$

数値でなく謎の文字が書かれている！

実際のFASTQデータをみると、

```
@SEQ_ID
GATTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTGTTCAACTCACAGTTT
+
! ' * ( ( ( ***+ ) % % + + ) ( % % % ) . 1 ***-+* ' ) ) **55CCF>>>>CCCCCCCC65
```

謎の文字の正体 → “ASCIIコード”を使ってQVを1文字で表したもの

ASCII: American Standard Code for Information Interchange

コンピュータでは文字を数値で表す

通信のため文字と数値の対応関係を規定（1965年）

0～126の数値に文字を割り当て

A → 65

Apple → 65;112;112;108;101;

FASTQ → ASCIIコードを逆に使って、QV(数値)を文字で表す

65 → A

利点: 10進数表記よりもファイルサイズを減らせる
(字数が半分、区切り文字も不要)

塩基: G A T T G G T A A C G A C G C T T	文字: ! ? @ > = ; 9 7 4 0 ,	文字が各塩基 のQVを表現
---------------------------------------	---------------------------	------------------

QVから文字への変換規則

問題点: ASCIIコードでは0-32はコンピューター用の特殊文字に割り当てられている

ASCIIコード表

数値	文字
0	null文字
1	SOH (ヘッダ開始)
2	STX (テキスト開始)
3	ETX (テキスト終了)
4	EOT (転送終了)
.....
30	RS (レコード区切り)
31	US (ユニット区切り)
32	(スペース)
33	!
34	"

- NGSでは10-30を頻用
 $p = 0.001 \rightarrow QV=30$
- 妥協案として特定値を加算してから文字に変換
Phred(QV)値 + X = ASCII値とする
- X値は現在 X=33 でほぼ統一
- 例) QV 30を表す場合
 $30 + 33 = 63$
→ ASCIIコードで63に該当する文字を当てる ("?"が該当)
- 変換にはコード表と簡単な計算が必要

実習1-2 Ex1_2.fqのQV値を求め、すべての配列のp値(エラー確率)が 0.01以下となるように3'側をトリミングしよう

Ex1_2.fq

```
@SEQ_ID
GATTGGTGAATT
+
??@A>;9740,
```

QV値 + 33 = ASCII値

ASCIIコード表

文 字 進 進	10	16	文 字 進 進	10	16															
NUL	0	00	DLE	16	10	SP	32	20	0	48	30	@	64	40	P	80	50	'	96	60
SOH	1	01	DC1	17	11	!	33	21	1	49	31	A	65	41	Q	81	51	a	97	61
STX	2	02	DC2	18	12	"	34	22	2	50	32	B	66	42	R	82	52	b	98	62
ETX	3	03	DC3	19	13	#	35	23	3	51	33	C	67	43	S	83	53	c	99	63
EOT	4	04	DC4	20	14	\$	36	24	4	52	34	D	68	44	T	84	54	d	100	64
ENQ	5	05	NAK	21	15	%	37	25	5	53	35	E	69	45	U	85	55	e	101	65
ACK	6	06	SYN	22	16	&	38	26	6	54	36	F	70	46	V	86	56	f	102	66
BEL	7	07	ETB	23	17	'	39	27	7	55	37	G	71	47	W	87	57	g	103	67
BS	8	08	CAN	24	18	(40	28	8	56	38	H	72	48	X	88	58	h	104	68
HT	9	09	EM	25	19)	41	29	9	57	39	I	73	49	Y	89	59	i	105	69
LF*	10	0a	SUB	26	1a	*	42	2a	:	58	3a	J	74	4a	Z	90	5a	j	106	6a
VT	11	0b	ESC	27	1b	+	43	2b	;	59	3b	K	75	4b	[91	5b	k	107	6b
FF*	12	0c	FS	28	1c	,	44	2c	<	60	3c	L	76	4c	¥	92	5c	l	108	6c
CR	13	0d	GS	29	1d	-	45	2d	=	61	3d	M	77	4d]	93	5d	m	109	6d
SO	14	0e	RS	30	1e	.	46	2e	>	62	3e	N	78	4e	^	94	5e	n	110	6e
SI	15	0f	US	31	1f	/	47	2f	?	63	3f	O	79	4f	_	95	5f	o	111	6f

* LFはNL、FFはNPと呼ばれることもある。

<http://e-words.jp/p/r-ascii.html>

* 赤字は制御文字、SPは空白文字(スペース)、黒字と緑字は图形文字。

* 緑字はISO 646で割り当てる変更が認められており、例えば日本ではバックスラッシュが円記号になっている

解説

```
@SEQ_ID
GATTGGTGAATT
+
??@A>=; 9740 ,
```

①p値が0.01の時のQV値を求める

$$\begin{aligned} QV &= -10 \log_{10} p \\ &= -10 \log_{10} 0.01 \\ &= -10 (-2) \\ &= 20 \end{aligned}$$

$QV < 20$ 部分をトリムすればよい

文 字	10	16	文 字	10	16	文 字	10	16
SP	32	20	0	48	30	@	64	40
!	33	21	1	49	31	A	65	41
"	34	22	2	50	32	B	66	42
#	35	23	3	51	33	C	67	43
\$	36	24	4	52	34	D	68	44
%	37	25	5	53	35	E	69	45
&	38	26	6	54	36	F	70	46
'	39	27	7	55	37	G	71	47
(40	28	8	56	38	H	72	48
)	41	29	9	57	39	I	73	49
*	42	2a	:	58	3a	J	74	4a
+	43	2b	;	59	3b	K	75	4b
,	44	2c	<	60	3c	L	76	4c
-	45	2d	=	61	3d	M	77	4d
.	46	2e	>	62	3e	N	78	4e
/	47	2f	?	63	3f	O	79	4f

②各文字をコード表からASCII値になおし、33を引いてQV値にする

塩基: G A T T G G T G A A T T

文字: ? ? @ A > = ; 9 7 4 0 ,

ASCII値: 63; 63; 64; 65; 62; 58; 59; 57; 55; 52; 48; 44;

QV値: 30; 30; 31; 32; 29; 25; 26; 24; 22; 19; 15; 11;

QV値 + 33 = ASCII値

ASCII値 - 33 = QV値

fastqファイルを見る上での注意点

1. QV値はあくまでシーケンサーによる推定値 目安として利用

2. 古いSolexa/Illuminaデータでは規格が乱立！！ ←重要

解析ソフト ver. (CASAVA)	~1.3	1.3~1.5	1.5~1.8	1.8~
参考使用時期	~2009	2009~2010	2010~2012	2012~
QV値算出法	Solexa	Phred	Phred	Phred
X値	64	64	64	33
QV range	-5~40	0~40	3~40 (2=end of read)	0~40

Phred(QV)値 + X = ASCII値

自分のデータがどのバージョン由来か確認し
解析ソフトの設定を補正する必要がある

FASTQのまとめ

概要: 塩基配列情報と各塩基の信頼性を表現する

規則:

- 1行目：“@” 配列名
- 2行目：塩基配列
- 3行目：“+”(配列名)
- 4行目:配列のクオリティ

ポイント: クオリティは ASCII文字で表現されている

$$QV\text{値} + 33 = \text{ASCII値}$$

fastqの仲間 [SRA \(Sequence Read Archive\)](#)

公共DBへの登録とダウンロードに使用。
バイナリ化(機械語化)された生シーケンスデータ
fastqに変換可能

後ほど詳しく説明

NGS基本データフォーマット

数十以上のフォーマットがあります
頻出フォーマットだけを紹介します

■ 配列用

[FASTA](#), [FASTQ](#), [SRA](#)

■ アノテーション用

[BED](#), [GFF/GTF](#), [WIG](#)

■ マッピング(アライメント)用

[SAM](#), [BAM](#)

BED, GFF/GTF

概要	ゲノム上の特徴配列を表現する (アノテーション情報)
内容	遺伝子名 染色体上の位置 向き エクソン構造
例	公共DBからアノテーション情報をダウンロード 解析したい領域の指定 アノテーション作業 遺伝子構造予測ソフトの結果出力

<3形式の違い>

BED	ブラウザでの描画情報 (色など) を記録可能
GFF	拡張性が高く様々な特徴情報を記録可能
GTF	GFFの厳格化版 一貫した規則で特徴情報を記録可能

BED (Browser Extensible Data format)

ブラウザでの描画情報(色など)を記録可能

○規則

項目数 3-12 タブ区切り

省略する場合は何も書かない(タブを2個連続させる)

染色体/ Scaffold 名	指定領域		領域名	スコ ア/表 記の 濃淡	ストラ ンド	太線表示		表示色 赤, 緑, 青 の強度 (0-255)	プロック(exon等)の情報 コンマ区切りで表記		
	開始 位置	終止 位置				開始 位置	終了 位置		個数	サイズ	開始 位置
chr22	1000	5000	cloneA	960	+	1000	5000	255,0,0	2	567,488,	0,3512
chr22	2000	6000	cloneB	900	-	2000	6000	0,0,255	2	433,399,	0,3601

1-3項目は
必須

4-12項目は省略可

領域開始位置=0
とした位置

実習1-3 Ex1_3.bedはヒトゲノム(GRCh37)の一部をbed形式にしたものである
lessコマンドで開いてbed形式を確認しよう

GFF (General Feature Format / Gene Finding Format)

拡張性が高く様々な特徴情報を記録可能

ゲノムアノテーションの標準的形式

○規則

項目数 5-9 タブ区切り

セミコロンで区切られた タグ
値の対

省略する場合は “-” や “.” を入れる

				指定領域							
染色体/ Scaffold 名	予測ソフト 名	領域の 種類	開始 位置	終止 位置	スコア	ストランド	読 絆			属性	
chr22	Manual	exon	1001	5000	960	+	0	.	.		
chr22	Manual	exon	2001	6000	900	-	0	NAME	“pol1”;		

必須

省略可

属性カラムに様々な情報を追加できる → 拡張性高

GTF (General Transfer Format)

基本的にGFFと同じだが、仕様をより細かく規定

○規則

				指定領域							
染色体/ Scaffold 名	予測ソフト 名等	領域の 種類	開始 位置	終止 位置	スコア	ストランド	読 絆			属性	
chr22	Twinscan	CDS	380	401	.	+	0	gene_id	“001”; transcript_id	“001.1”;	
chr22	Twinscan	CDS	501	650	.	+	2	gene_id	“001”; transcript_id	“001.1”;	
chr22	Twinscan	CDS	700	707	.	+	2	gene_id	“001”; transcript_id	“001.1”;	
chr22	Twinscan	start_codon	380	382	.	+	0	gene_id	“001”; transcript_id	“001.1”;	
chr22	Twinscan	stop_codon	708	710	.	+	0	gene_id	“001”; transcript_id	“001.1”;	

必須: CDS, start_codon, stop_codon

遺伝子と転写産物のIDを

任意: 5UTR, 3UTR, inter, inter CNS, intron_CNS, exon

表記する

それ以外は無効

実習1-4

Ex1_4.gtfは 1_3と同じ領域をgtf形式にしたものである。
lessコマンドで開いてgtf形式を確認しよう

注意 GFF/GTFとBEDでは座標の表現が異なる

GFF/GTF: 開始、終了ともに 1-based (1 から始まる) 座標

BED : 開始は 0-based, 終了は 1-based 座標

具体例

GFF/GTF	1	2	3	4	5	6	7	8	
	A	G	T	A	C	T	C	G	
BED	0	1	2	3	4	5	6	7	8

黄色部分を示す時

GFF/GTF format: 開始 3, 終了 6 (長さは $6-3+1=4$)

BED format : 開始 2, 終了 6 (長さは $6-2=4$)

実習1-5

[Ex1_3.bed](#)と[Ex1_4.gtf](#)を開き、実際に座標がずれていることを確認しよう

WIG (Wiggle Format)

概要	ゲノム上の量的特徴を表現するための形式
内容	ゲノム上の座標に対する”数値”情報
例	GC含量、発現量などを表す

○規則 2形式から選べる

1) VariableStep 柔軟な指定が可能

```
variableStep chrom=chr2
300601      22.5
300701      30.5
300751      28.2
```

位置と値の組で領域を指定するため
間隔は位置ごとに変更可能

2) FixedStep コンパクトな表現が可能

```
fixedStep chrom=chr3 start=300601 step=100
22.5
30.5
25.8
```

定開始位置と間隔は先頭
行で指定し、後は値のみ
を示していく

NGS基本データフォーマット

数十以上のフォーマットがあります
頻出フォーマットだけを紹介します

- 配列用

FASTA, FASTQ, SRA

- アノテーション用

BED, GFF/GTF, WIG

- マッピング(アライメント)用

SAM, BAM

SAM (Sequence Alignment/Map format)

概要	マッピング(アライメント)結果を表現
内容	マッピング情報 (位置, インデル, ミスマッチ) ペアフラグメントの状況, 塩基配列
例	SNP、発現量解析への入力データ形式

○ファイル例

ヘッダ一部												マッピング結果	
@HD VN:1.5 SO:coordinate													
@SQ SN:ref LN:45													
r001	163	ref	7	30	8M2I4M1D3M	=	37	39	TTAGATAAAGGATACTG	*			
r002	0	ref	9	30	3S6M1P1i4M	*	0	0	AAAGATAAGGATAT	*			
r003	0	ref	9	30	5S6M	*	0	0	GCCTAACGCTAA	*	SA:Z:ref,29,-,6H5M		
r004	0	ref	16	30	6M14N5M	*	0	0	ATAGCTTCAGC	*			
r003	2064	ref	29	17	6H5M	*	0	0	TAGGC	*	SA:Z:ref,9,+,5S		
r001	83	ref	37	30	9M	=	7	-39	CAGCGGCAT	*	NM:i:1		

実習1-6

Ex1_7.samを開きsam形式を確認しよう

○規則

ヘッダー部

@HD VN:1.5 SO:coordinate
@SQ SN:ref LN:45

“@”で開始

@HD VN: (バージョン) SO: (ソート状況)

@SQ SN: (リファレンス名) LN: (リファレンスの長さ)

マッピング結果部分

項目間はタブで区切る

フラグメント名	FLAG	リファレンス配列名	アライメント開始位置	マッピングQV	CIGAR	ペアフラグメントの場所			配列	配列QV	オプション
						Ref名	開始	長さ			
r001	163	ref	7	30	8M2I4M1D3M	=	37	39	TTAGATAAAGGATAC TG	*	
r002	0	ref	9	30	3S6M1P1i4M	*	0	0	AAAGATAAGGATAT	*	
r003	0	ref	9	30	5S6M	*	0	0	GCCTAACGCTAA	*	SA:Z:ref,29
r004	0	ref	16	30	6M14N5M	*	0	0	ATAGCTTCAGC	*	
r003	2064	ref	29	17	6H5M	*	0	0	TAGGC	*	SA:Z:ref,9,
r001	83	ref	37	30	9M	=	7	-39	CAGCGGCAT	*	NM:i:1

ポイント！ “CIGAR” “FLAG”

SAMのポイント1 : CIGAR

数字と文字を組み合わせアライメント状況を示す

フラグメント名	FLAG	リファレンス配列名	アライメント開始位置	マッピングQV	CIGAR	ペアフラグメントの場所			配列	配列QV	オプション
						Ref名	開始	長さ			
r001	163	ref	5	30	3M2D2M	=	37	39	GCAAG	44>>>	

3M2D2M

塩基数 状況

3塩基一致、2個挿入、2塩基一致

ref : ATGCGCATTAGCCTAA
read : GCA--AG

記号	状況
M	一致
I	挿入
D	欠失
N	インtron(RNAvsDNAのみ)
S	クリップ(塩基情報残す)
H	クリップ(塩基情報削除)
P	他リードが挿入を入れている

SAMのポイント2: FLAG リードの状態を示す数値

理解すると「マップされなかったリードだけ選ぶ」などの操作が可能になる

数値 (10進数)	意味
1	ペアリードがある
2	両方適切にマップされている
4	自分がマップされていない
8	ペア相手がマップされていない
16	逆鎖にマップされた (配列も逆鎖で表記)
32	ペア相手は逆鎖にマップされた
64	Read1の配列である
128	Read2の配列である
256	Multiple hitでトップヒットでないアライメント
512	マッピングQVが低い

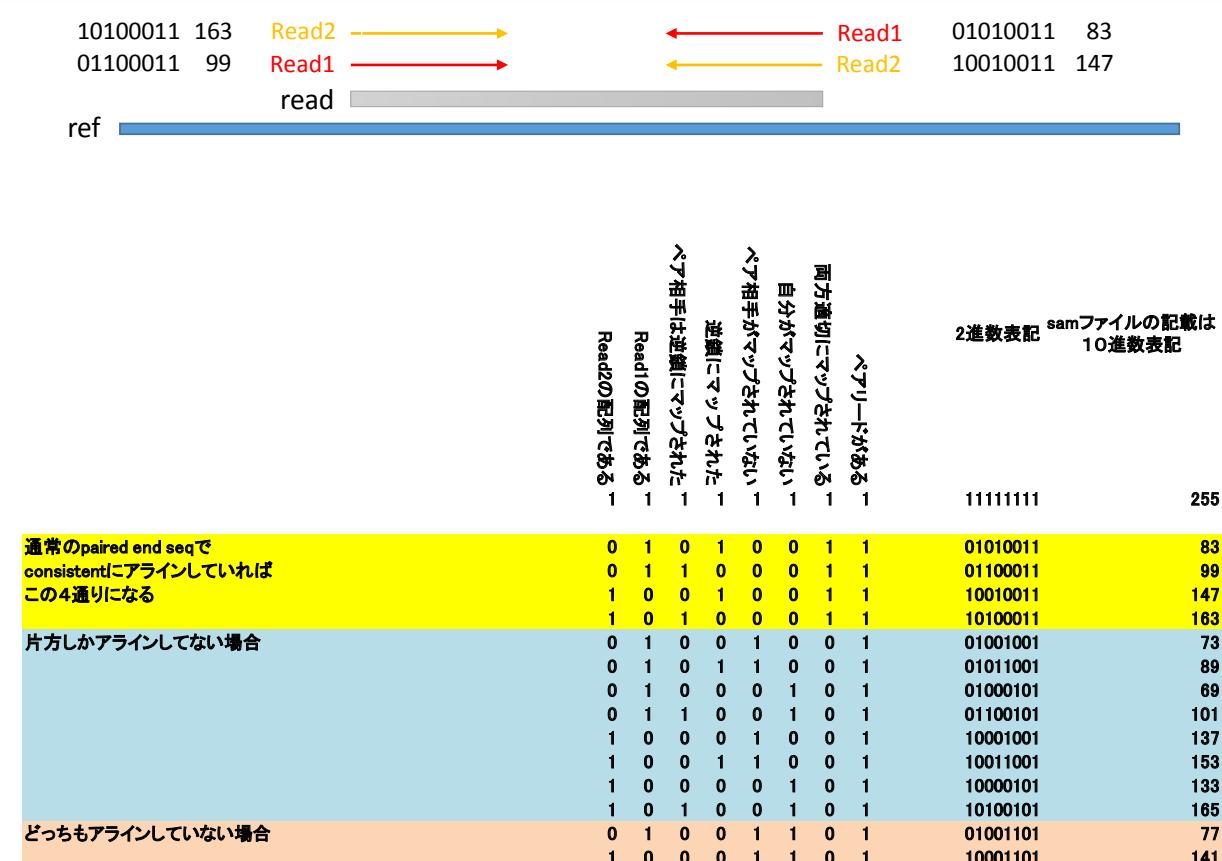
複数の状況に合致する場合は数値を加算

ペアリード、両方マップされた → $1+2=3$

2進数の個々の有無で評価されている

加算した結果が、ほかの状況と一致しないようになっている

Paired end readでFLAG値の組み合わせを見てみる



自動でflagを計算してくれるサイトがある

<http://broadinstitute.github.io/picard/explain-flags.html>

This utility explains SAM flags in plain English.
It also allows switching easily from a read to its mate.

Flag: Explain

[Switch to mate](#)

Explanation:

- read paired
- read mapped in proper pair
- read unmapped
- mate unmapped
- read reverse strand
- mate reverse strand
- first in pair
- second in pair
- not primary alignment
- read fails platform/vendor quality checks
- read is PCR or optical duplicate
- supplementary alignment

Summary:

SAMのまとめ

概要: 各リードがマップされた場所と状態を表す

規則: ヘッダ部とアライメント部からなる タブ区切り

ポイント:	FLAG値	→ リードのマップ状況
	CIGAR値	→ リードのアライメント状況

触れなかった重点点

ペアフラグメント部分の“長さ”列 → フラグメント間距離 + 両リード長

SAM formatの詳細な仕様書

<http://samtools.github.io/hts-specs/SAMv1.pdf>

BAM

■ BAM

SAMをバイナリ(機械語)化したもの

容量が小さくなるが、人には理解できない

SAMに戻すことも可能なので必要に応じて変換

■ BAM indexing file

BAMファイルに対して作られる検索用ファイル

高速検索や可視化ソフトなどに必要

後ほど詳しく説明

フォーマット各論まとめ

	FASTA	FASTQ	SAM
概要	配列情報の標準形式	NGS結果の標準形式	マッピング結果を示す
内容	塩基配列 アミノ酸配列	塩基配列と 一塩基to毎の品質情報	マッピング情報 ペアの状況, 塩基配列
例	公共DBからの配列情報 ダウンロード	マッピング、アセンブル解析で の入力データ形式	マップ結果の閲覧、集計 SNP、発現量解析への入力
特徴		QV値はASCII文字で表現 SRAから変換可能	CIGAR, FLAG値を利用 バイナリ化したのがBAM
	BED	GFF	GTF
概要	ゲノム上の特徴配列を表現する		
内容	遺伝子名 染色体上の位置 向き エクソン構造		
例	公共DBからアノテーション情報をダウンロード 解析したい領域の指定 アノテーション作業 遺伝子構造予測ソフトの結果出力		
特徴	ブラウザでの描画 情報を記録	拡張性高	GFFの厳格化版 一貫した規則
			2つの形式 VariableStep/FixedStep

概要

序論

- データフォーマットとは？
- フォーマットを学ぶ理由
- 効率よい学習のポイント

NGS基本データフォーマット

- FASTA FASTQ SRA
- BED GFF/GTF WIG
- SAM/BAM

NGS基本ツール

- SRAToolkit
- SAMtools
- IGV

NGS基本ツール

実践データ用の専用ソフトの使い方を紹介

実践データはコマンドラインを使った専用ソフトでの操作となる
データ量が多すぎて、マニュアルの編集は不可能 普通のソフトでも困難

各NGSフォーマットを利用したNGS基本ツール

FASTQ : SRAToolkit, cutadapt, fastQC

BED GFF/GTF : BEDtools

SAM/BAM : SAMtools, Picard

可視化ツール : IGV, JBrowser

NGS解析に特に有用な
SRAToolkit, SAMtools, IGVに注目して解説します

SRAToolkit

The screenshot shows the NCBI SRAToolkit website. At the top, there's a blue header bar with links for NCBI, Site map, All databases, Search, Sequence Read Archive, Main, Browse, Search, Download, Submit, Documentation, Software, Trace Archive, Trace Assembly, Trace Home, Trace BLAST. Below this is a secondary blue bar with links for Download, Toolkit Documentation, and XML Schema. The main content area has a green header "SRAToolkit". Below it, a note says "For Toolkit documentation click here." followed by a link. A numbered list 1. NCBI SRA Toolkit latest release (February 3 2015, version 2.4.4 release) compiled binaries and [md5 checksums*](#): lists operating system architectures: CentOS Linux 64 bit architecture, Ubuntu Linux 64 bit architecture, MacOS 64 bit architecture, MS Windows 64 bit architecture, and vdb-view Windows Installer.

機能

SRA → fastq の変換

SRA形式の利点

fastqに比べて1/10のファイルサイズ

→保存が楽だし、ダウンロードも早くなる

fastq以外のデータ形式を使うシーケンサーのデータもSRA形式に変換可能

→共通のDB構造を持たせることができ管理が楽

基本事項 toolの呼び出し(SRAToolkitを例として)

実習2-1 ./sratoolkit/fastq-dump と打ち
SRAToolkit の fastq-dump を呼び出そう

→fastq-dumpが呼び出され、使い方が表示された

Usage:

```
./sratoolkit/fastq-dump [options] <path> [<path>...]
./sratoolkit/fastq-dump [options] <accession>
```

Use option --help for more information

./sratoolkit/fastq-dump : 2.5.2

基本事項 Linuxのマニュアルの見方

[] は、省略可能なオプション（とその引数）
| は選択肢で、列挙されたいずれかを排他的に選択する

Usage:

./sratoolkit/fastq-dump [options] <path> [<path>...]

↑

省略可能な
オプション

↑

変換したいSRAファイル
へのパスを記述

実習2-2

1) Ex2_1.sraをfastq形式に変換しよう。

例) ./sratoolkit/fastq-dump Ex2_1.sra

2) lessでfastqに変換された事を確認しよう。

例) less Ex2_1.fastq

3) lsコマンドでファイルサイズを確認しよう。

例) ls -l

SAMtools

Samtools

Home

Download ▾

Workflows ▾

Documentation ▾

Support ▾

Samtools

Samtools is a suite of programs for interacting with high-throughput sequencing data. It consists of three separate repositories:

<http://www.htslib.org>

機能

1. 形式の変換・整理

- 1-1, SAM (人間用) ⇔ BAM (コンピュータ用) の変換
- 1-2, 検索や可視化ソフト用の索引ファイル作成

2. データ抽出

- 2-1, 特定のリードの選出
- 2-2, 統計情報収集(発現量解析)

SAMtoolsの呼び出し

実習2-3 ./samtools と打ち SAMtools を呼び出そう

→SAMtoolsが呼び出され、使い方が表示された
samtools の後にさらに各command名を打って使う

```
Program: samtools (Tools for alignments in the SAM format)
Version: 1.0 (using htllib 1.0)

Usage: samtools <command> [options]

Commands:
  -- indexing
    faidx      index/extract FASTA
    index      index alignment
  -- editing
    calmd      recalculate MD/NM tags and '=' bases
    fixmate   fix mate information
    reheader  replace BAM header
    rmdup    remove PCR duplicates
    targetcut cut fosmid regions (for fosmid pool only)
  -- file operations
    bamshuf   shuffle and group alignments by name
    cat       concatenate BAMs
    merge    merge sorted alignments
    mpileup  multi-way pileup
    sort     sort alignment file
    split    splits a file by read group
    bam2fq   converts a BAM to a FASTQ
  -- stats
    bedcov   read depth per BED region
    depth    compute the depth
    flagstat simple stats
    idxstats BAM index stats
    phase   phase heterozygotes
    stats   generate stats (former bamcheck)
  -- viewing
    flags   explain BAM flags
    tview   text alignment viewer
    view    SAM<->BAM<->CRAM conversion
```

検索高速化

統計情報

**SAM ⇄ BAM の変換
リードの選出**

SAMtools SAM/BAM変換

実習2-4 ./samtools view と打ちview機能の使い方情報を呼び出そう

```
Usage: samtools view [options] <in.bam>|<in.sam>|<in.cram>
[region ...]

Options:
  -b      output BAM
  -C      output CRAM (requires -T)
  -1      use fast BAM compression (implies -b)
  -u      uncompressed BAM output (implies -b)
  -h      include header in SAM output
  -H      print SAM header only (no alignments)
  -c      print only the count of matching records
  -o FILE output file name [stdout]
  -U FILE output reads not selected by filters to FILE
  [null]
  -t FILE FILE listing reference names and lengths (see
long help) [null]
  -T FILE reference sequence FASTA FILE [null]
  -L FILE only include reads overlapping this BED FILE
  [null]
  -r STR only include reads in read group STR [null]
```

bam → sam 変換のしかた

bamファイルを指定して実行(リターン) 出力ファイルは > で指定

例) samtools view ./in.bam > out.sam

(ヘッダーも出力するときは -h オプションをつける)

sam → bam 変換のしかた

-b で出力をbamにしてほしいことを示す

例) samtools view -b ./in.sam > ./out.bam

SAMtools SAM/BAM変換

実習2-5 sam/bamの違いを実感しよう

- 1) lessコマンドを使ってEx1_7.bamの内容を見てみよう
 - “Ex1_7.bam” may be a binary file. See it anyway? と表示される
 - y と打つ (yes、読みます)
 - 読めない文字が表示される(バイナリ化されている)
- 2) samtools viewを使ってEx1_7.bamの内容をsamに変えて再度中身を確認しよう
例) samtools view Ex1_7.bam > Ex1_7_new.sam
- 3) ls コマンドでEx1_7.sam, Ex1_7.bamのサイズを比較しよう
 - ls -l (ls = ディレクトリ中のファイル情報を表示、-l = long 長い説明で)
 - bamの方がサイズが小さいハズ(情報を圧縮できている)

BAM化のメリット:ストレージ領域の節約 データ送付の高速化 処理の高速化
 SAM化のメリット:人が見て理解できる

SAMtools 検索・indexファイル作成1

データをあらかじめ整理しておき検索や可視化を容易にする
 以下の2ステップを続けて行う

- Step 1 samtools sort をつかってリードの順番を整理する
- Step 2 samtools index で索引を作成し効率よく検索できるようにする

実習2-6 samtools sort を呼び出してみよう

```
Usage: samtools sort [options...] [in.bam] ← input はbam形式
Options:
  -l INT      Set compression level, from 0 to 9 [-1]
  -m INT      Set maximum memory per thread; suffix K/M/G recognized [768M]
  -n          Sort by read name
  -o FILE     Write final output to FILE rather than standard output
  -O FORMAT   Write output as FORMAT ('sam'/'bam'/'cram') (either -O or
  -T PREFIX   Write temporary files to PREFIX.nnnn.bam           -T is required)
```

出力フォーマットを指定

実習2-7 samtools index を呼び出してみよう

```
Usage: samtools index [-bc] [-m INT] <in.bam> [out.index]
Options:
  -b          Generate BAI-format index for BAM files [default]
  -c          Generate CSI-format index for BAM files
  -m INT     Set minimum interval size for CSI indices to 2^INT [14]
```

SAMtools 検索・indexファイル作成2

Step 1 samtools sort をつかってリードの順番を整理する

実習2-8 samtools sort を使ってEx1_7.bamの内容をソートしよう

(出力ファイル名はEx1_7_s.bam)

例) samtools sort -O bam -T test Ex1_7.bam > Ex1_7_s.bam

実習2-9 sortの効果を実感しよう

1) samtools viewを使ってEx1_7_s.bamをsamに変えて保存しよう

例) samtools view Ex1_7_s.bam > Ex1_7_s.sam

 samtools view -h Ex1_7_s.bam > Ex1_7_s_h.sam

2) テキストの頭や末尾を表示するhead や tail コマンドで結果を見てみよう

例) head -n 10 Ex1_7.sam

 head -n 10 Ex1_7_s.sam

 head -n 10 Ex1_7_s_h.sam

tail -n 10 Ex1_7.sam

 tail -n 10 Ex1_7_s.sam

Ex1_7_s.sam ではリファレンス配列上の位置順にリードの順番が整理されている。

SAMtools 検索・indexファイル作成3

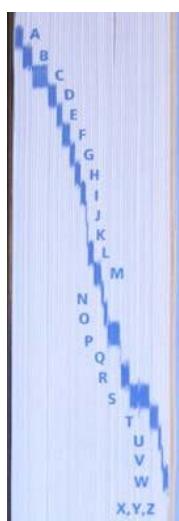
Step 2 samtools index で索引を作成し、効率よく検索できるようにする

実習2-10 samtools index を使ってEx1_7_s.bamのインデックスファイル

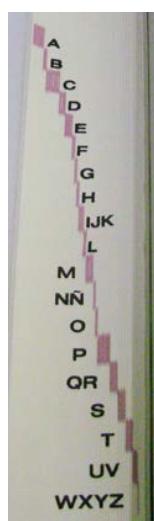
(Ex1_7_s.bam.bai)を作ろう

例) samtools index Ex1_7_s.bam

<インデックスファイルとは?>



英語辞書



スペイン語辞書

辞書の小口印刷のような物

索引をつけることで直接見たい場所(の近く)から
検索を始められる

データの偏りによって適切なインデックス区分は変わる
(英語→Wで始まる語は多いので独自の区分を与える。 西語→WとXYZは同じ区分)

↓
各ファイルにあわせて個別に
インデックスを作る必要がある

SAMtools

Samtools

Home

Download ▾

Workflows ▾

Documentation ▾

Support ▾

Samtools

Samtools is a suite of programs for interacting with high-throughput sequencing data. It consists of three separate repositories:

<http://www.htslib.org>

機能

1. 形式の変換・整理

1-1, SAM (人間用) ⇔ BAM (コンピュータ用) の変換

1-2, 検索や可視化ソフト用の索引ファイル作成

2. データ抽出

2-1, 特定のリードの選出

2-2, 統計情報収集(発現量解析！)

SAMtools 2, データ抽出

2-1 特定のリードの選出

方法1) samtools view を使ったマップ位置による選出

indexファイルが作成されれば、viewをつかって
特定のリファレンス部分にマップされたリードを選び出せる

例) ./samtools view Ex1_7.s.bam cp1:1000-2000

リファレンス名 位置

方法2) samtools view -f/-F を使ったマップ状況 (flag値) による選出
-f/-F オプションを使うことで、特定のマップ状況にあるリードを
選び出すことができる。

-f 該当するflag値をもつリードを抽出

-F 該当するflag値をもつリード”以外”を抽出

例) ./samtools view -f 83 Ex1_7.s.bam

SAMtools 2, データ抽出

2-1 特定のリードの選出

実習2-11

Ex1_7_s.bamから以下の遺伝子にマップされたリードを取り出し、数を比較しよう

染色体名	遺伝子名	位置
cp1	16SrRNA	27762-29094
cp1	rbcL	60184-61611
cp1	rpoC	68739-75880

例) ./samtools view Ex1_7_s.bam cp1:1000-2000 |wc -l

↑
行数を数えるコマンド

実習2-12

前頁の例(-f 83) はどんなペアリード関係を指定しているのだろうか？

例) ./samtools flags 83

実習2-13

ペアが両方マップされていないリードを抽出し、数を調べよう

例) ./samtools view -f 12 Ex1_7_s.bam |wc -l

マッピングソフトによって
区分が異なる場合もあるので注意

アライメント結果を目視しながら
確認してから使用すること

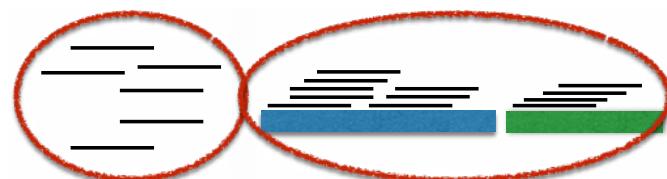
数値 (16進数)	数値 (10進数)	意味
1	1	ペアリードがある
2	2	両方適切にマップされている
4	4	自分がマップされていない
8	8	ペア相手がマップされていない
10	16	逆鎖にマップされた（配列も逆鎖で表記）
20	32	ペア相手は逆鎖にマップされた
40	64	ペアリードの1番目である
80	128	ペアリードの2番目である
100	256	Multiple hitでトップヒットでないアライメント
200	512	マッピングQVが低い

SAMtools 2, データ抽出

2-2 統計情報収集

方法 1 samtools flagstat を使ってマッピング結果全体の簡単な情報を得る

n本マップされm本マップされなかった



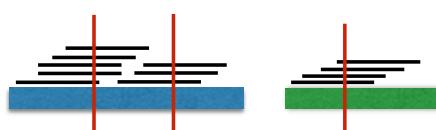
方法 2 samtools idxstats でRef配列毎にマップされたリード数を得る

RefAにM本 RefBにN本マップされた



方法 3 samtools depth で塩基毎に深度（読まれた回数）を得る

RefAのX塩基目はY回読まれた

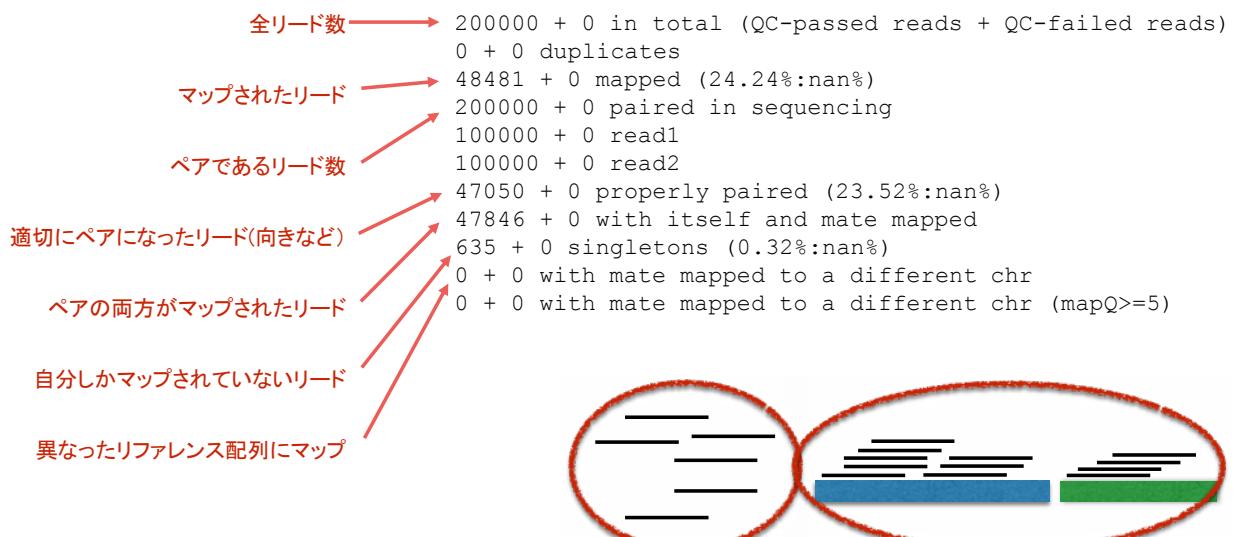


SAMtools 2, データ抽出 2-2 統計情報収集

方法1 samtools flagstat を使ってマッピング結果の簡単な統計情報を得る

Usage: samtools flagstat <in.bam>

例) samtools flagstat Ex1_7_s.bam



各行の詳細な意味は使ったマッピングソフトの出力形式に影響される

SAMtools 2, データ抽出 2-2 統計情報収集

方法2 samtools idxstats でRef配列毎にマップされたリード数を得る

Usage: samtools idxstats <in.bam>

例) samtools idxstats Ex1_7_s.bam

Ref.名	Ref.配列長	マップされたリード数	マップされなかったリード数
cp1	86483	48481	635
*	0	0	150884

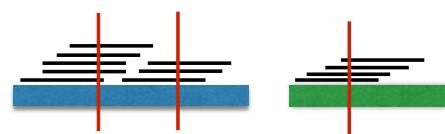
ペアなのにマップされなかったリード数

方法3 samtools depth を使って深度(読まれた回数)の統計情報を得る

Usage: samtools depth [options] in1.bam [in2.bam [...]]

例) samtools depth Ex1_7_s.bam

Ref.名	位置	深度 読まれた回数
Cp1	3313	83
Cp1	3314	120
Cp1	3315	144
Cp1	3316	148



SAMtoolsのまとめ

マッピングデータの整理から、実験結果解析まで
様々な場面で活躍する必須ツール

samtools	
view	← SAM/BAM変換 特定リード抽出
sort	
index] ←検索高速化
flagstat	
idxstats] ←統計情報抽出
depth]

他のSAMtoolsの機能

mpileup	SNP検出などで活躍
tview	簡便なアライメントビューワー
merge	複数のBAMファイルを結合する

まとめ

最初は意味がわからないフォーマットでも、読み解けば
日々の実験で接する生物学的情報を表しているにすぎない

解説しなかったフォーマット、ツールも類似の構造をもつ
仕様書をよめば恐るるにたらず！

“Practice makes perfect！”

ぜひ自分のデータも読み解いてみてください。

データ可視化ツール・IGVの紹介・実習

The screenshot shows the official website for IGV. On the left is a sidebar with links to Home, Downloads, Documents, Hosted Genomes, FAQ, IGV User Guide, File Formats, Release Notes, IGV for iPad, Credits, and Contact. It also includes a search bar and links to the Broad Home and Cancer Program. The main content area features a large banner for 'Integrative Genomics Viewer' showing a complex genomic visualization interface. Below the banner are sections for 'What's New' (with a news item about the iPad app), 'Citing IGV' (with citation information), 'Overview' (describing IGV as a high-performance visualization tool for genomic datasets), 'Downloads' (with a download link), and 'Funding' (mentioning funding from the National Cancer Institute, National Institute of General Medical Sciences, and the Starr Cancer Consortium). Logos for the National Human Genome Research Institute and GENOME SPACE are at the bottom.

<https://www.broadinstitute.org/igv/>

なぜIGVを取り上げるか

データ可視化ツール

- ・自分のパソコン(ローカル環境)にインストールして使うタイプ
- ・サーバーに構築して、ネットワークで使うタイプ

コミュニティーに広く利用、あるいはウェブ公開を目的とするには良いが、ネットワーク・情報セキュリティの高度な知識も要求される。

より大容量なデータに対応できる。

管理者的な人がいて、その人がやってくれるなら、これも良いが。

もっとお手軽なものとしてIGVを紹介

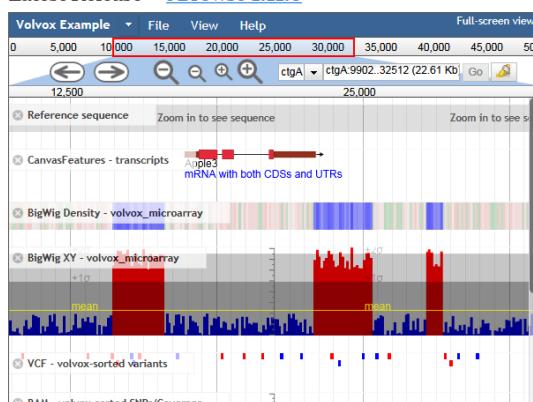
The JBrowse Genome Browser

JBrowse is a fast, embeddable genome browser built completely with JavaScript and HTML5, with optional run-once data formatting tools written in Perl.

Featured Post

[Exploring structural variation using JBrowse](#) by Richard Finkers

Latest Release – [JBrowse 1.11.6](#)



可視化ツールに求められるものは何か

膨大なデータを如何に直感的に理解できるようにするか
sortや絞り込みができる表データと対比双璧

- ・配列、GC ratio、遺伝子情報
- ・遺伝子発現情報
- ・SNPの位置情報・頻度情報
- ・様々なデータの精度情報

レファレンス配列 / gene model / gene annotationとNGSデータを並べて比較
複数のデータセットを並べて比較

色々なデータ(variant, 発現, ChIP, BSseq等々)を、様々なスケールで
比較・統合的に解釈できるようにしたい

ゲノムviewerに自分のデータを乗せ、
統合的直感的に解釈できること

可視化ツールをどう選ぶか

選択の基準

genome data viewing に求められるもの
取捨選択の基準

1. 無料 / 有料 / 基本無料
2. 個人的レベルの使用 / コミュニティーレベルの使用
3. 見るだけ/自分から色々工夫
4. アクセスのしやすさ・使いやすさ
 - 導入に必要なコンピュータスペック
 - マニュアルは分かりやすいか
 - 情報の多さ
 - 利用の簡便さ
 - 使っている人が近くにいるか

Integrative Genomics Viewer(IGV)

お手軽ツール

- ・アカデミックウェアで無料
- ・コミュニティーでの利用者が多いから、情報も多い
- ・javaのプログラムなので、オールプラットフォーム対応
- ・マニュアルは親切、サンプルデータのある
- ・WEBサーバーではなく、PCレベルができる
- ・データ閲覧環境の共有が可能

誰もが簡便に使えるものが良い。

Home

Integrative Genomics Viewer

What's New

September 2014. The IGV iPad app can now be installed from the Apple App Store. *IGV for iPad* is a lightweight genomic data viewer that provides some of the functionality available in our regular desktop IGV. See the [IGV for iPad documentation](#) for details.

Overview

The Integrative Genomics Viewer (IGV) is a high-performance visualization tool for interactive exploration of large, integrated genomic datasets. It supports a wide variety of data types, including array-based and next-generation sequence data, and genomic annotations.

Downloads

Please [register](#) to download IGV. After registering, you can log in at any time using your email address. Permission to use IGV is granted under the [GNU LGPL license](#).

Citing IGV

To cite your use of IGV in your publication:

James T. Robinson, Helga Thorvaldsdóttir, Wendy Winckler, Mitchell Guttman, Eric S. Lander, Gad Getz, Jill P. Mesirov, *Integrative Genomics Viewer*. *Nature Biotechnology* 29, 24–26 (2011)

Helga Thorvaldsdóttir, James T. Robinson, Jill P. Mesirov. *Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration*. *Briefings in Bioinformatics* 14, 178–192 (2013).

Funding

Development of IGV is made possible by funding from the National Cancer Institute, the National Institute of General Medical Sciences of the National Institutes of Health, and the Starr Cancer Consortium.

IGV participates in the [GenomeSpace](#) initiative, which is funded by the [National Human Genome Research Institute](#).

NATIONAL INSTITUTE OF GENERAL MEDICAL SCIENCES, NATIONAL CANCER INSTITUTE, NATIONAL HUMAN GENOME RESEARCH INSTITUTE, GENOMESPACE



nature.com ▶ journal home ▶ archive ▶ issue ▶ opinion and comment ▶ correspondence ▶ abstract

◀ previous abstract next abstract ▶

NATURE BIOTECHNOLOGY | OPINION AND COMMENT | CORRESPONDENCE

Integrative genomics viewer

James T Robinson, Helga Thorvaldsdóttir, Wendy Winckler, Mitchell Guttman, Eric S Lander, Gad Getz & Jill P Mesirov

Affiliations | Corresponding authors

Nature Biotechnology 29, 24–26 (2011) | doi:10.1038/nbt.1754

Published online 10 January 2011

To the Editor:

Rapid improvements in sequencing and array-based platforms are resulting in a flood of diverse genome-wide data, including data from exome and whole-genome sequencing, epigenetic surveys, expression profiling of coding and noncoding RNAs, single nucleotide polymorphism (SNP) and copy number profiling, and functional assays. Analysis of these large, diverse data sets holds the promise of a more comprehensive understanding of the genome and its relation to human disease. Experienced and knowledgeable human review is an essential component of this process, complementing computational approaches. This calls for efficient and intuitive visualization tools able to scale to very large data sets and to flexibly integrate multiple data types, including clinical data. However, the sheer volume and scope of data pose a significant challenge to the development of such tools.

Journal home

Current issue

For authors

Subscribe

E-alert sign up

RSS feed



Subscribe today, save 50% and receive 51 weekly issues of *Nature* in print, online and mobile.

Citations to this article

Crossref (10) Scopus (12) Web of Science (0)

Science jobs from naturejobs

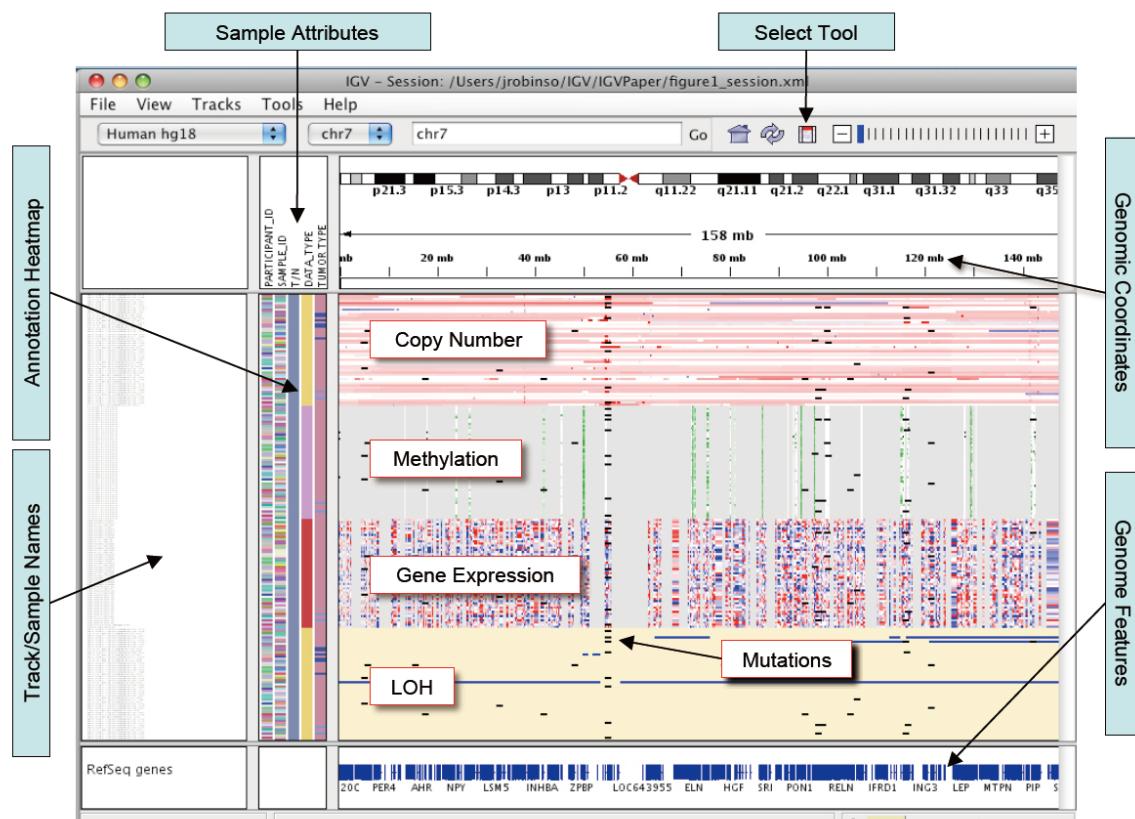
Faculty Position

Harvard Medical School

Ramalingaswami Re-Entry Fellowship

Ministry of Science & Technology, Government of India

The screenshot shows the IGV website interface. On the left, there's a sidebar with a navigation menu including Home, Downloads, Documents, Hosted Genomes, FAQ, IGV User Guide (which is highlighted with a red box), File Formats, Release Notes, IGV for iPad, and Credits. Below this is a search bar and a link to the Broad Institute. The main content area features a large banner for 'Integrative Genomics Viewer' with a complex genomic visualization. Below the banner are sections for 'What's New' (mentioning the IGV iPad app), 'Citing IGV' (with a citation to a Nature Biotechnology paper), 'Overview' (describing IGV as a high-performance visualization tool), 'Downloads' (with a download icon), and 'Funding' (mentioning National Cancer Institute and National Institutes of Health funding). Logos for the National Human Genome Research Institute and GenomeSpace are at the bottom right.



Nature Biotech. 29:24–26 (2011) Supplement figureからの抜粋

Downloads

Integrative Genomics Viewer (IGV) (Version 2.3)

Install IGV

Options for installing and running IGV:

1. (Mac only) Download and run the Mac application; or
2. (All systems) Use the Java Web Start buttons (Mac users: see below for limitations); or
3. (All systems) Download the binary distribution and run IGV from the command line.

1. Mac Application

Download and unzip the Mac App archive, then double-click the IGV application to run it. The application can be moved to the "Applications" folder, or anywhere else. **Note: This requires Java 7.** **Mac users with Java 6 (JRE 1.6) should use the binary distribution archive or the Java Web Start buttons below.**

[Download Mac App](#)

2. Java Web Start

The buttons below use Java Web Start (JWS) to install and launch IGV directly from our web site.

***Mac Users:** The Java Web Start option is not recommended for Mac OSX Mountain Lion or higher. Using it requires that you set Gatekeeper security to its lowest level, and it is possible that even this will not be enough.

Chrome: Chrome does not automatically launch the Java Webstart files by default. Instead, the launch buttons below will download a ".jnlp" file. This should appear in the lower left corner of the browser. Double-click the downloaded file to run.

Windows users: To run with more than 1.2 GB of memory you must install 64-bit Java. **Most Windows installs do not include 64-bit Java by default, even if the operating system is 64-bit.** Attempting to use the 2GB or greater launch options with 32-bit Java will result in the error "could not create virtual machine".

Launch Launch with 750 MB	Launch Launch with 1.2 GB Maximum usable memory for Windows OS with 32-bit Java.	Launch Launch with 2 GB Maximum usable memory for 32-bit Mac OS.	Launch Launch with 10 GB For large memory machines with 64-bit Java.
------------------------------	--	--	--

3. Binary Distribution

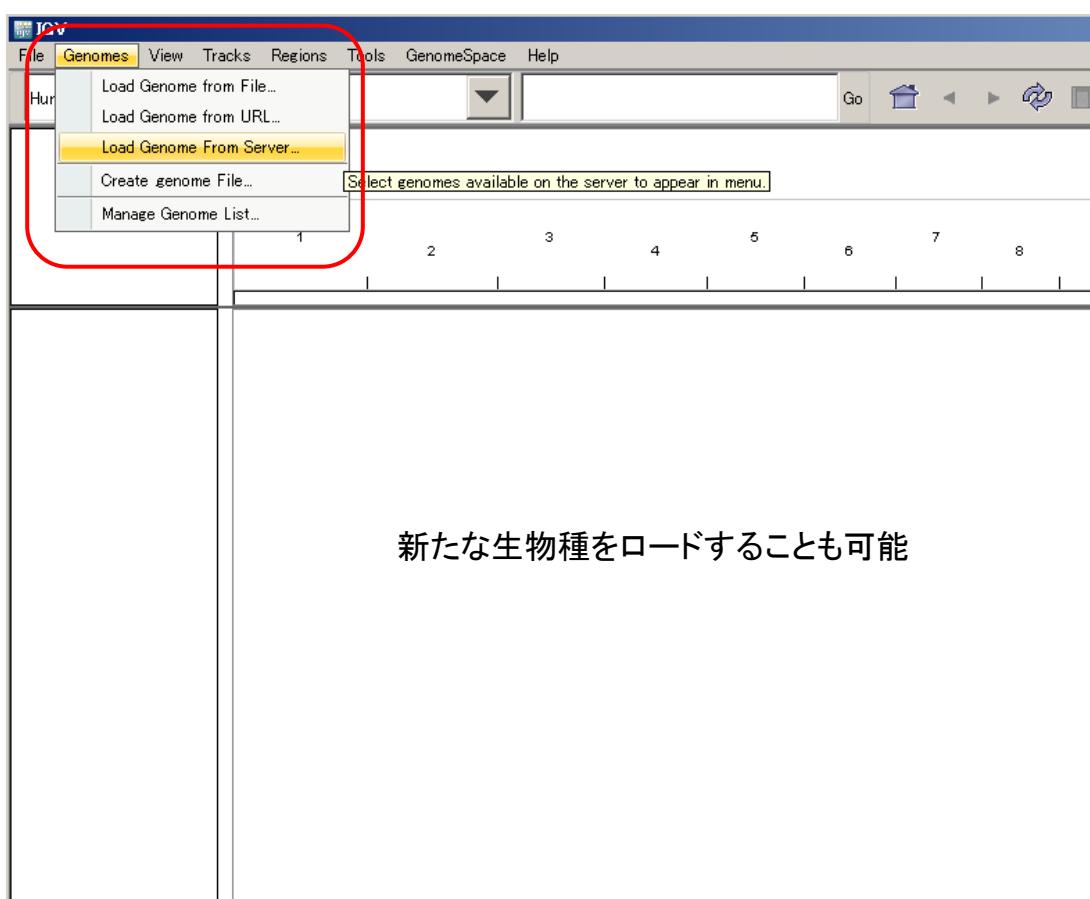
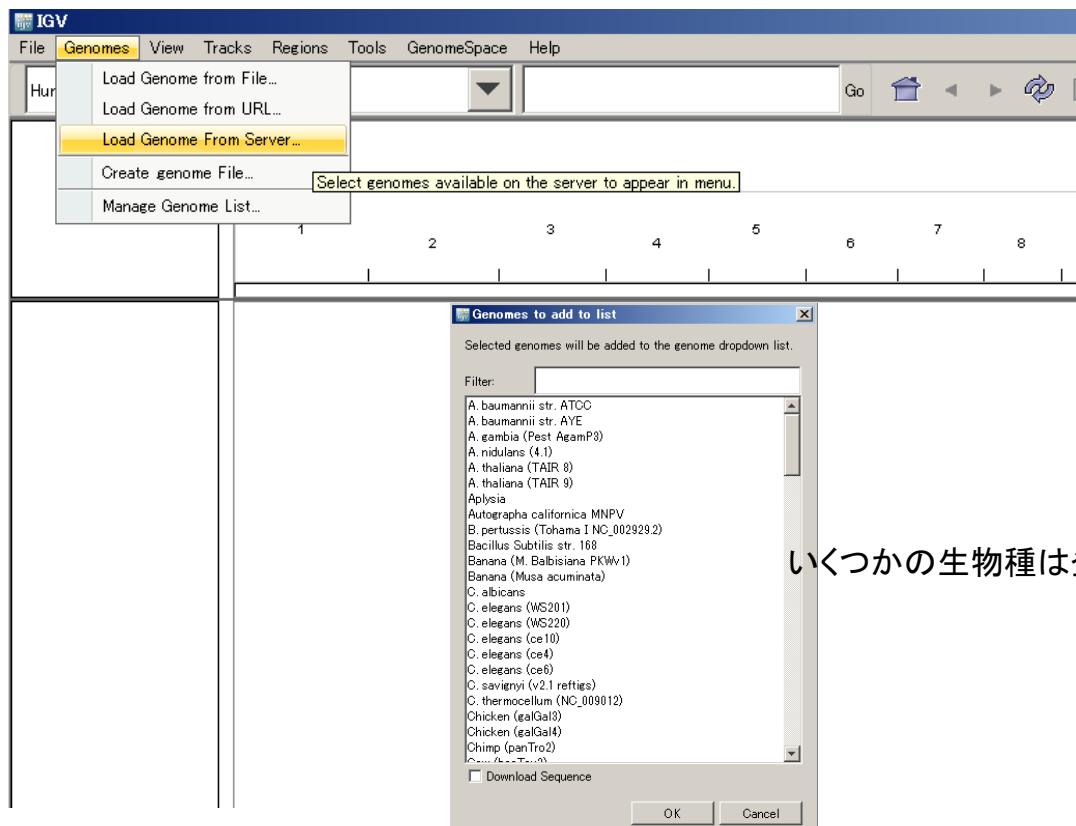
Download and unzip the binary distribution archive in a folder of your choosing. IGV is launched from a command prompt -- follow instructions in the "readme" file. To launch IGV on Mac or Linux platforms use the shell script "igv.sh". On Windows use "igv.bat".

[Download Binary Distribution](#)

igvtools

Utilities for preprocessing data files.

- [igvtools_2.3.40.zip](#)



ゲノムViewerなので次世代DNAシーケンサーのデータに限定されない。
マイクロアレイの結果や、ゲノムアノテーションの情報も随時表示できる。

対応するファイル形式に応じて、表示方法が決まる。

File Formats

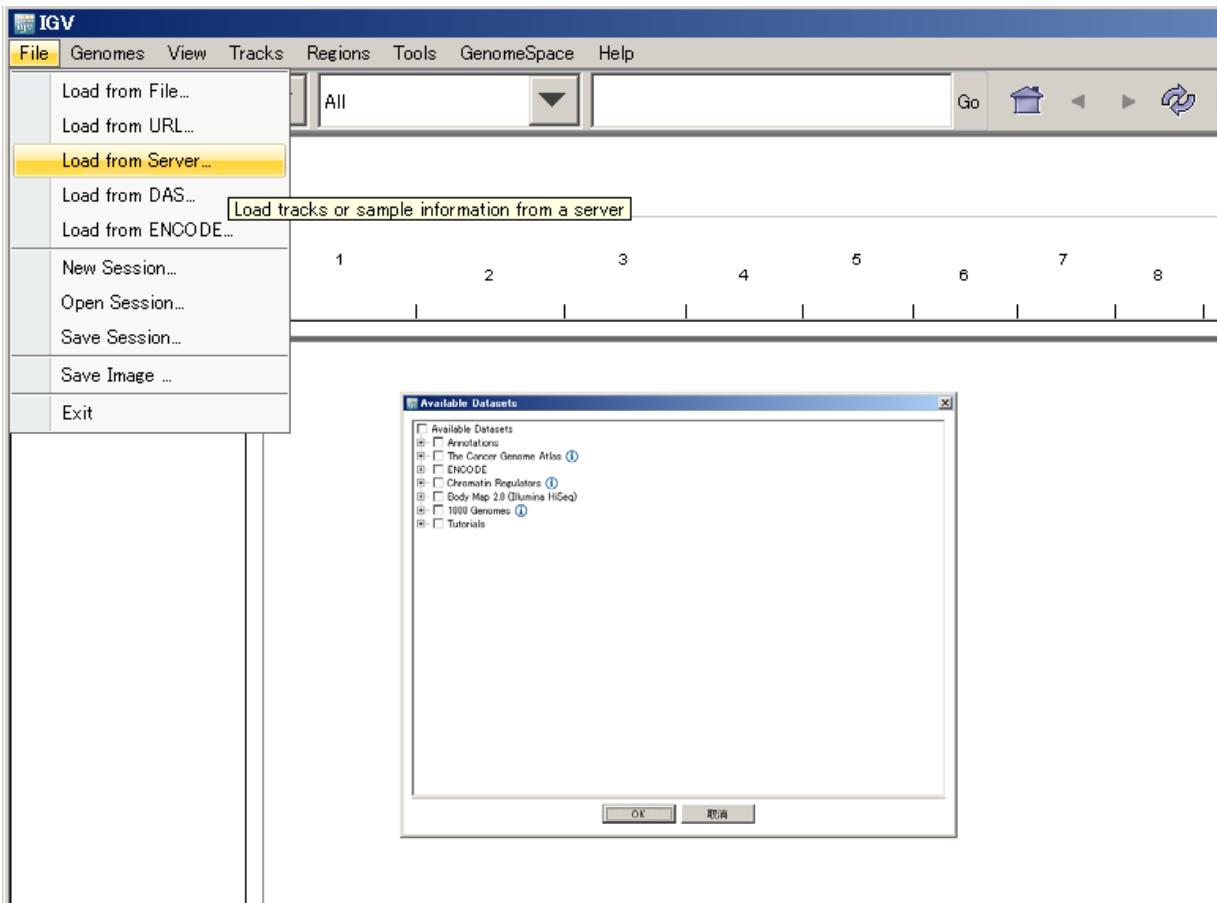
- [File Extension Identifies Format](#)
- [Recommended File Formats](#)
- [BAM](#)
- [BED](#)
- [BedGraph](#)
- [bigBed](#)
- [bigWig](#)
- [Birdsuite Files](#)
- [broadPeak](#)
- [CBS](#)
- [CN](#)
- [Custom File Formats](#)
- [CytoBand](#)
- [FASTA](#)
- [GCT](#)
- [genePred](#)
- [GFF/GTF](#)
- [GISTIC](#)
- [Gob](#)
- [GWAS](#)
- [IGV](#)
- [LOH](#)
- [MAF \(Multiple Alignment Format\)](#)
- [MAF \(Mutation Annotation Format\)](#)
- [Merged BAM File](#)
- [MUT](#)
- [narrowPeak](#)
- [PSL](#)
- [RES](#)
- [SAM](#)
- [Sample Information](#)
- [SEG](#)
- [SNP](#)
- [TAB](#)
- [TDF](#)
- [Track Line](#)
- [Type Line](#)
- [VCF](#)
- [WIG](#)
- [chrom.sizes](#)

File Formats

IGV supports a number of different file formats for experimental data and genome annotations. For a complete list of supported formats see <http://www.broadinstitute.org/igv/FileFormats>. The following table shows the recommended file formats for a number of common data types.

Source Data	Recommended File Formats
ChIP-Seq, RNA-Seq	WIG, TDF
Copy number	CN, SNP, TDF, canary_calls (Birdsuite)
Gene expression data	GCT, RES, TDF
Genome annotations	GFF, BED, GTF, PSL, UCSC table format
GISTIC data	GISTIC
LOH data	LOH, TDF
Mutation data	MUT, MAF
Variant calls	VCF
RNAi data	GCT
Segmented data	SEG, CBS
Sequence alignment data	BAM, SAM, PSL
Any numeric data	IGV, WIG, TDF
Sample metadatadata	Tab-delimited sample info file

公開情報のviewerとして



その他の便利機能

セッションの保存

表示しているデータの読み込み状況を、それごと保存。

セッションをロードすることで、意図した画面を表示できる。

データセットが揃っていること、フォルダー構造が同一である必要がある。

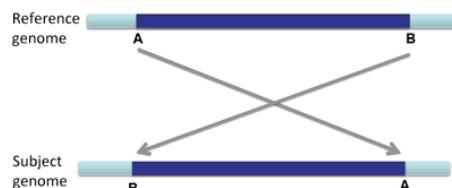
バッチ処理

重要領域の画面スナップショットを自動で取ったりできる。

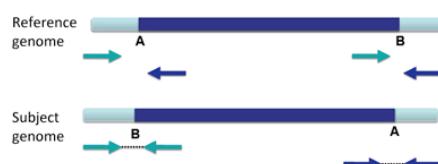
```
new
load myfile.bam
snapshotDirectory mySnapshotDirectory
genome hg18
goto chr1:65,289,335-65,309,335
sort position
collapse
snapshot
goto chr1:113,144,120-113,164,120
sort base
collapse
snapshot
```

Inversions

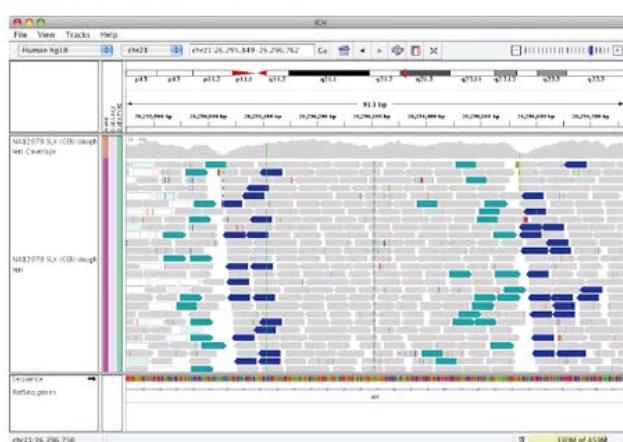
An inversion is a large section of DNA that is reversed in the subject genome compared to the reference genome.



When an inversion shows up in paired-end reads, the reads are distinctively variant from the reference genome.



This appears in IGV as shown below.



Interpreting Color by Insert Size

The inferred insert size can be used to detect structural variants, such as:

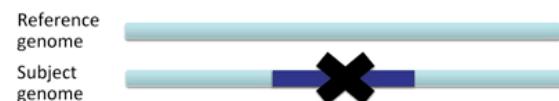
- deletions
- insertions
- inter-chromosomal rearrangements

IGV uses color coding to flag anomalous insert sizes. When you select Color alignments>by insert size in the popup menu, the default coloring scheme is:

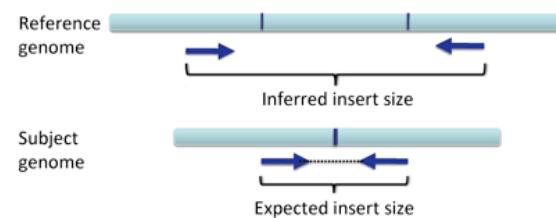
- for an insert that is larger than expected
- for an insert that is smaller than expected
- for paired end reads that are coded by the chromosome on which their mates can be found

Deletions

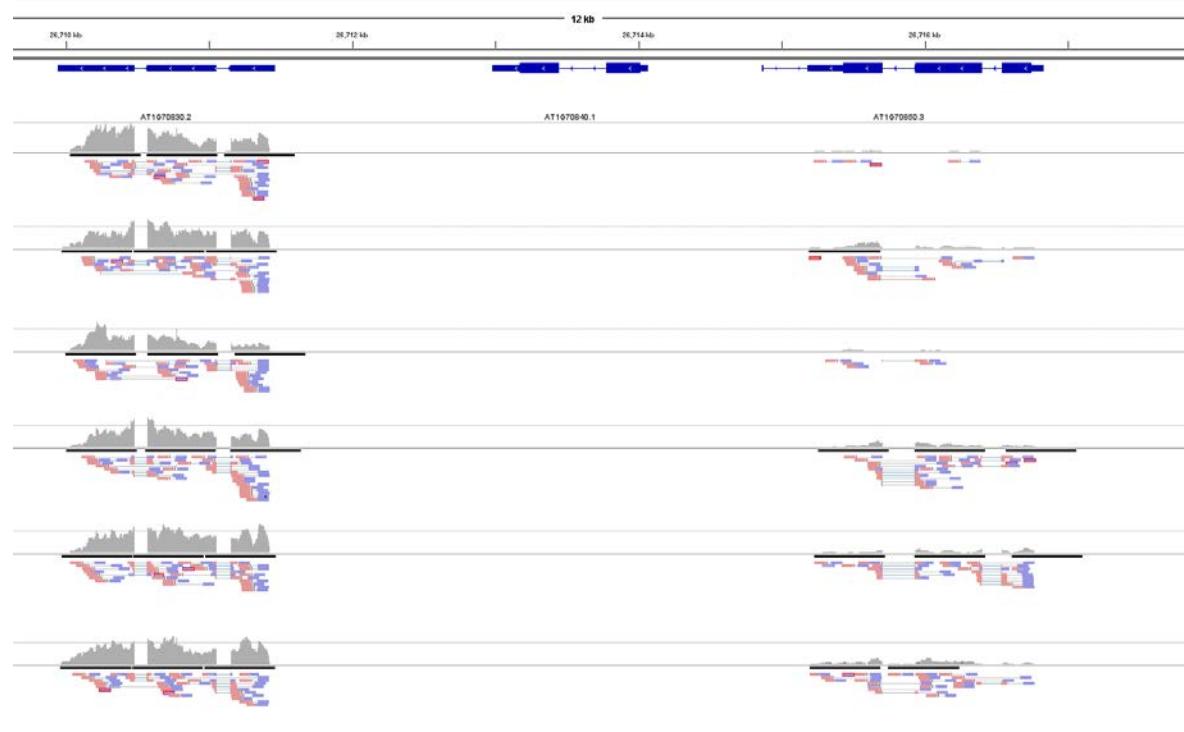
A deletion is a large section of DNA that is absent in the subject genome compared to the reference genome.



The "expected" insert size is the insert size obtained in sequencing the subject genome. The "inferred" insert size is the insert size that would result in the reference genome, assuming the same pair of reads.



RNA-Seqのデータ表示させる

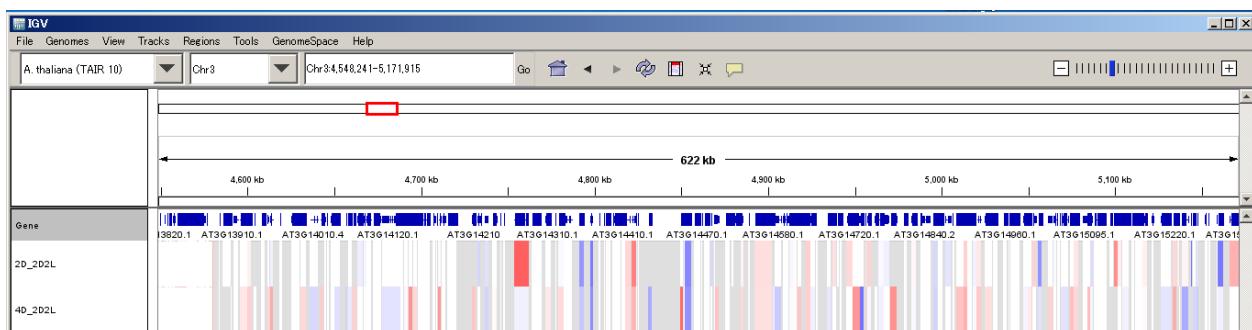


GCTファイルでgene ローカスの発現情報を図示

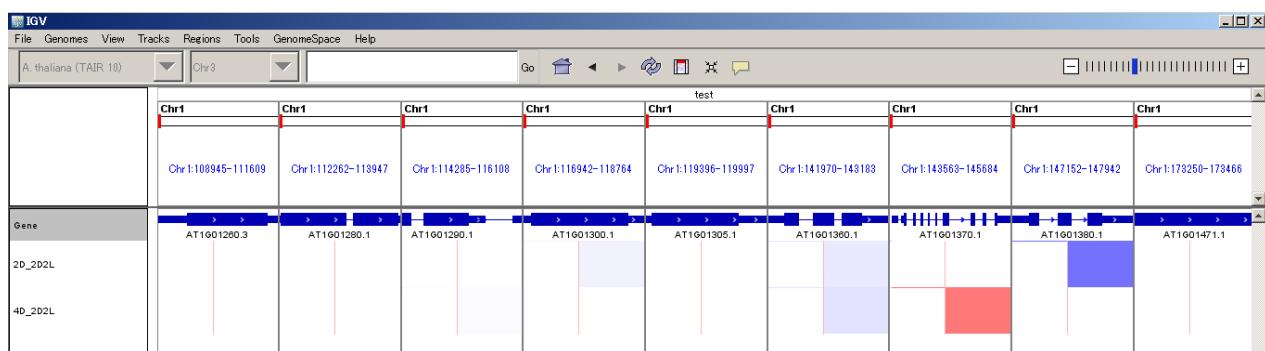
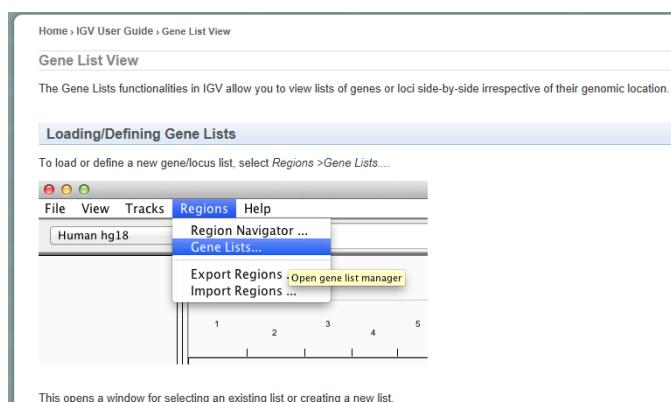
```

#
#
Name      Description      2D_2D2L      4D_2D2L
ANAC001   |@Chr1:3630-5899  -2.60184    -2.60956
DCL1      |@Chr1:23145-33153  -0.742675   -1.5642
MIR838A   |@Chr1:23145-33153  0           0
AT1G01073  |@Chr1:44676-44787  0           0
IQD18     |@Chr1:52238-54692  -1.93871    -1.13128
AT1G01115  |@Chr1:56623-56740  0           0
GIF2      |@Chr1:72338-74737  -0.251287   -0.616679
AT1G01180  |@Chr1:75582-76758  0.45929     -0.809567
AT1G01210  |@Chr1:88897-89745  1.6964      0.857196
FKGP      |@Chr1:91375-95651  -0.174589   0.725947
AT1G01240  |@Chr1:99893-101834  -0.226384   -0.936641
AT1G01260  |@Chr1:108945-111609  -0.161848   0.315699
CYP703A2  |@Chr1:112262-113947  0           0
CNX3      |@Chr1:114285-116108  0.111249    -0.551359
AT1G01300  |@Chr1:116942-118764  -0.68348    0.108578

```



Gene listを定義して
サンプルごと
条件ごと
の発現・発現変動を
カラーマップできる



IGV実習

The screenshot shows the IGV website's download page. It includes a sidebar with links like Home, Downloads, Documents, Hosted Genomes, and Contact. The main content area is titled "Downloads" and "Integrative Genomics Viewer (Version 2.3)". It provides instructions for Mac users to download and unzip the archive, then double-click the application to run it. It also provides instructions for Windows and Linux users to download and unzip the archive in a folder of their choosing. Below this, there's a section for "Java Webstart" with four launch buttons: "Launch with 750 MB", "Launch with 1.2 GB", "Launch with 2 GB", and "Launch with 10 GB". A note states that Chrome does not automatically launch Java Webstart files by default. At the bottom, there are sections for "Development Snapshot Build", "Archived Versions", "igvtools", and "Source Code".

IGVの使用法を学ぶと共に
先のファイルフォーマットも
確認しよう

以下のファイルを確認

buc.genome.fasta
buc.gtf
buc_cg.wig
illumina_ex_B2_Read_bowtie2.mate.sort.bam
illumina_ex_B2_Read_bowtie2.mate.sort.bam.bai
illumina_ex_B4_Read_bowtie2.mate.sort.bam
illumina_ex_B4_Read_bowtie2.mate.sort.bam.bai

The screenshot shows the IGV software interface. The menu bar includes File, Genomes, View, Tracks, Regions, Tools, GenomeSpace, and Help. The "Genomes" tab is selected, and a dropdown menu is open with options: "Load Genome from File...", "Load Genome from URL...", "Load Genome From Server...", "Create .genome File...", and "Manage Genome List...". The main window displays a genome browser with chromosomes 3 through Y. On the left, there is a panel labeled "RefSeq genes". The bottom status bar shows "buc.genome.fasta genome rem..." and "478M of 635M". The text below the screenshot reads:

登録されていない生物種・配列でも、自分でimportすればOK
Buchneraゲノムを使う
Genome -> Load Genome from File
buc.genome.fasta
を読み込む

ゲノム構造を記述したgtfファイルを読み込む

File-> Load from File
buc.gtf ファイルを読み込む

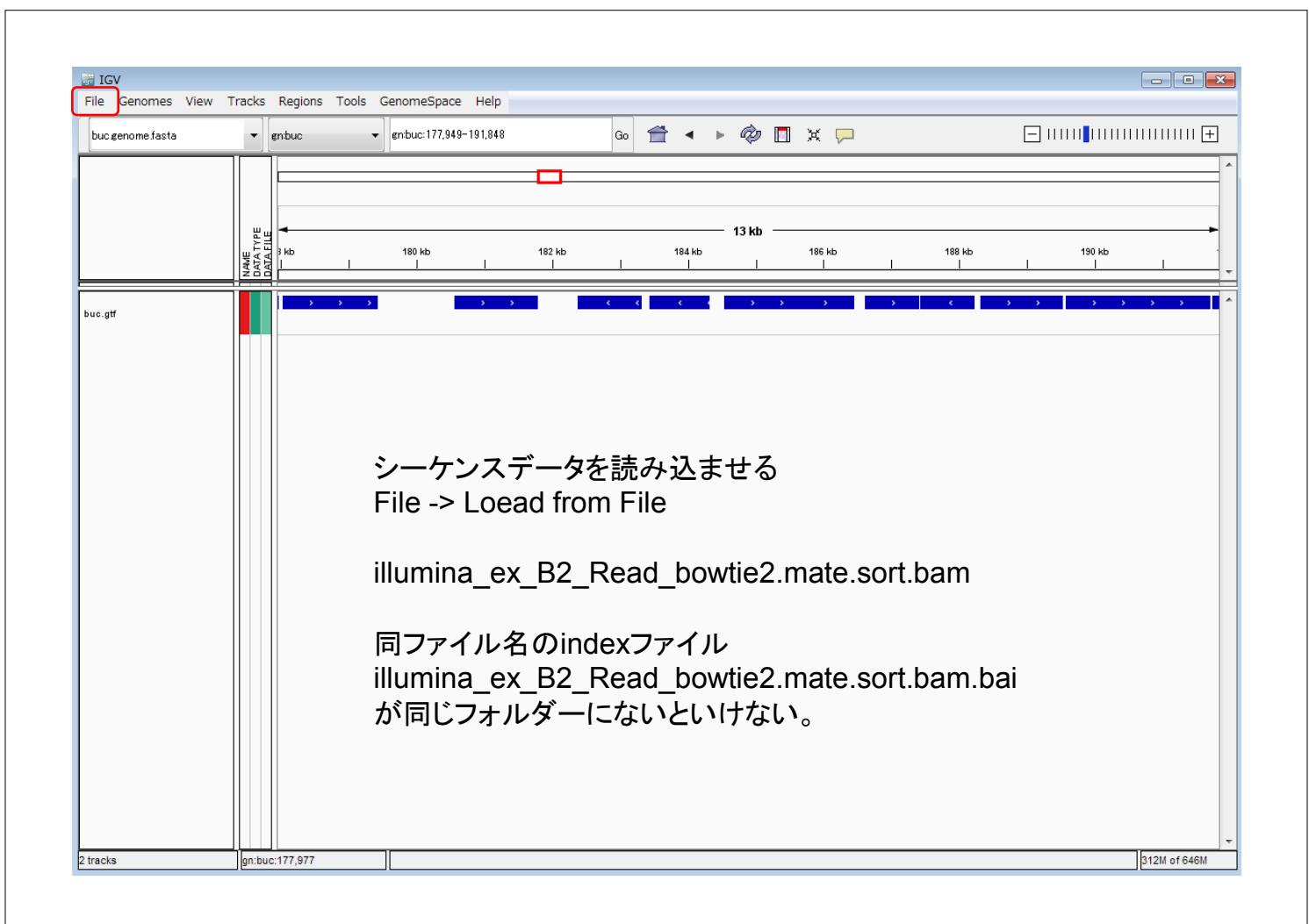
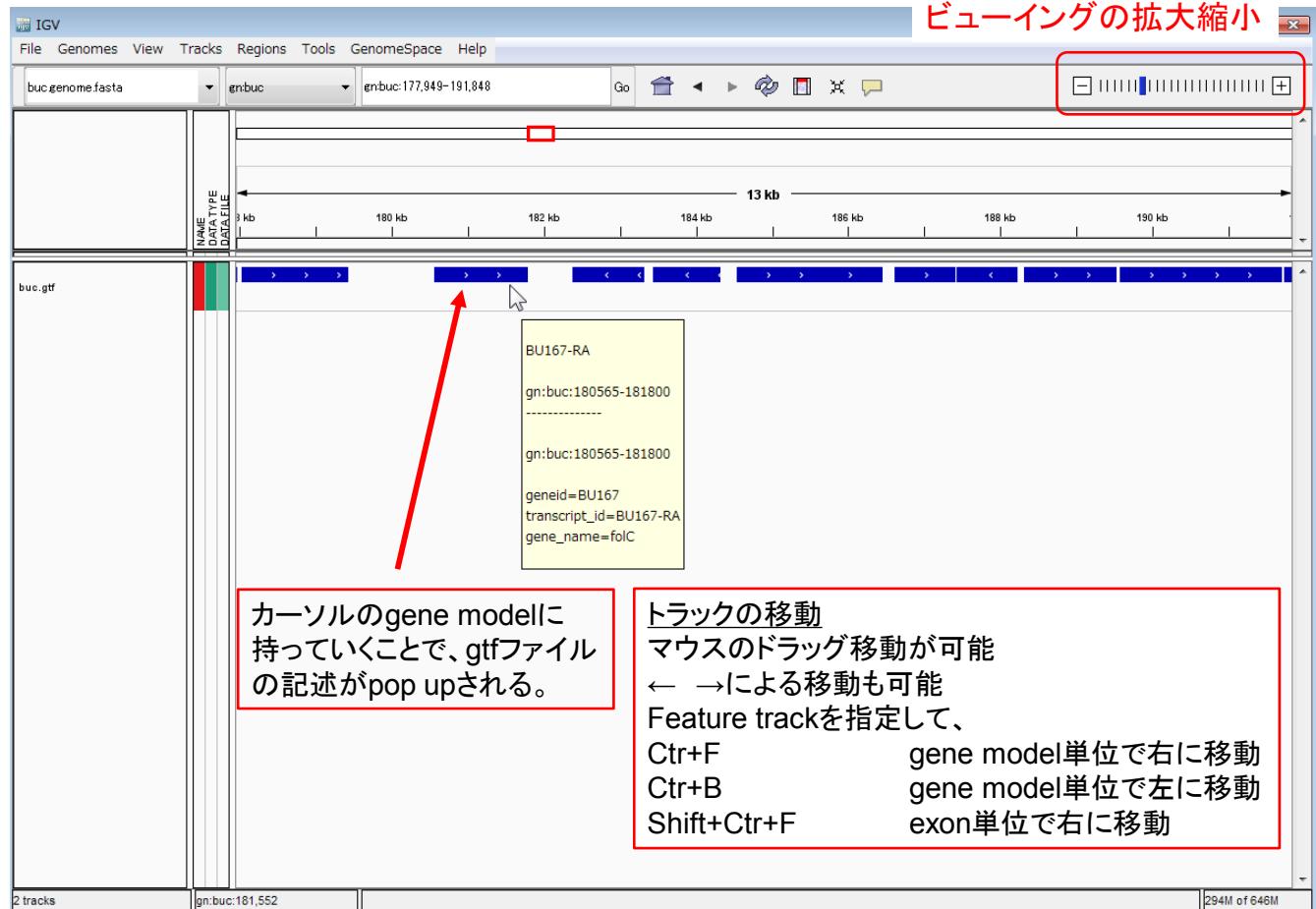
buc.gtf						
gn:buc	KEGG:GENES	CDS	197	2083	.	geneid "BU001"; transcript_id "BU001-RA"; gene_name "gidA";
gn:buc	KEGG:GENES	CDS	2278	3102	.	geneid "BU002"; transcript_id "BU002-RA"; gene_name "atpB";
gn:buc	KEGG:GENES	CDS	3139	3378	.	geneid "BU003"; transcript_id "BU003-RA"; gene_name "atpE";
gn:buc	KEGG:GENES	CDS	3497	3982	.	geneid "BU004"; transcript_id "BU004-RA"; gene_name "atpF";
gn:buc	KEGG:GENES	CDS	3982	4515	.	geneid "BU005"; transcript_id "BU005-RA"; gene_name "atpH";
gn:buc	KEGG:GENES	CDS	4530	6068	.	geneid "BU006"; transcript_id "BU006-RA"; gene_name "atpA";
gn:buc	KEGG:GENES	CDS	6101	6973	.	geneid "BU007"; transcript_id "BU007-RA"; gene_name "atpC";
gn:buc	KEGG:GENES	CDS	6997	8394	.	geneid "BU008"; transcript_id "BU008-RA"; gene_name "atpD";
gn:buc	KEGG:GENES	CDS	8421	8837	.	geneid "BU009"; transcript_id "BU009-RA"; gene_name "atpC";
gn:buc	KEGG:GENES	CDS	8911	11322	.	geneid "BU010"; transcript_id "BU010-RA"; gene_name "gyrB";
gn:buc	KEGG:GENES	CDS	11449	12549	.	geneid "BU011"; transcript_id "BU011-RA"; gene_name "dnahN";
gn:buc	KEGG:GENES	CDS	12554	13918	.	geneid "BU012"; transcript_id "BU012-RA"; gene_name "dnahA";
gn:buc	KEGG:GENES	CDS	14369	14512	.	geneid "BU013"; transcript_id "BU013-RA"; gene_name "rmpH";
gn:buc	KEGG:GENES	CDS	14525	14872	.	geneid "BU014"; transcript_id "BU014-RA"; gene_name "rnpA";
gn:buc	KEGG:GENES	CDS	15011	16609	.	geneid "BU015"; transcript_id "BU015-RA"; gene_name "yidC";

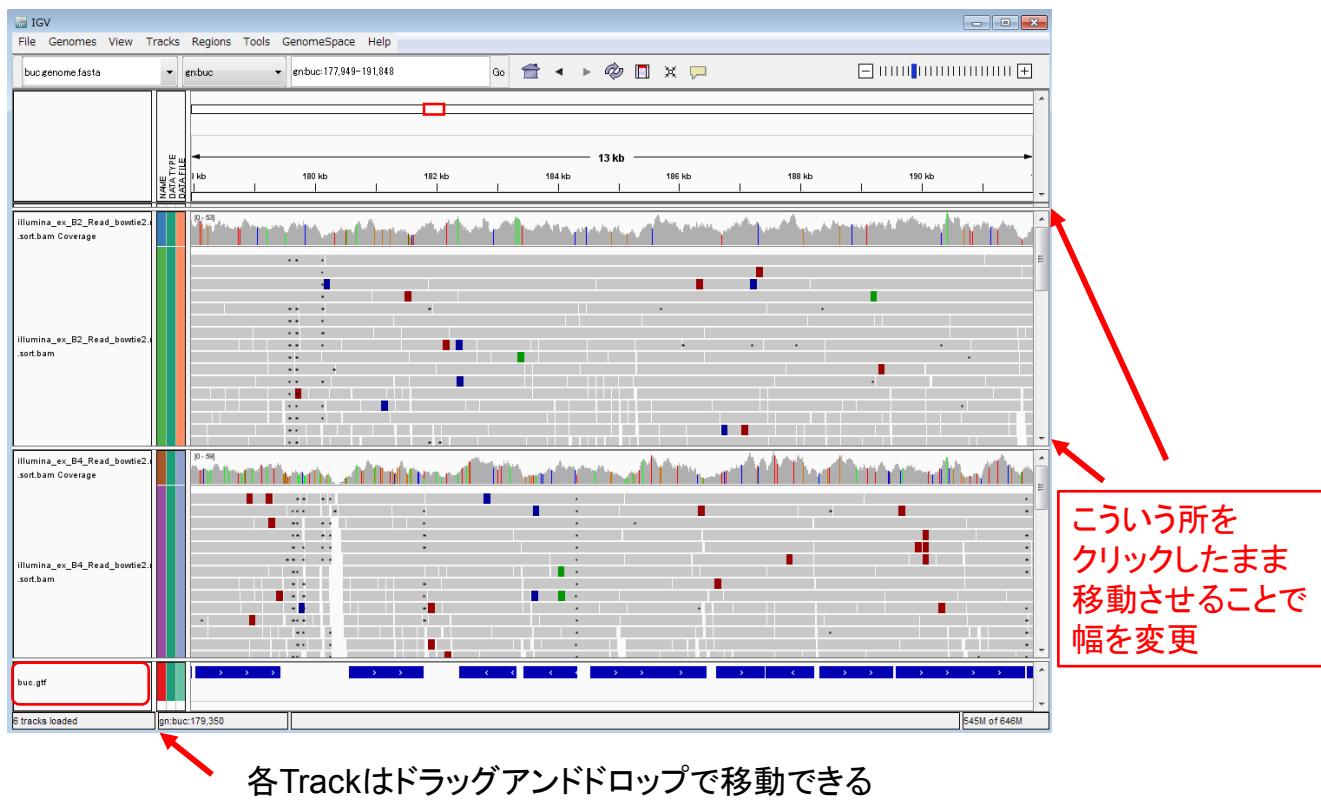
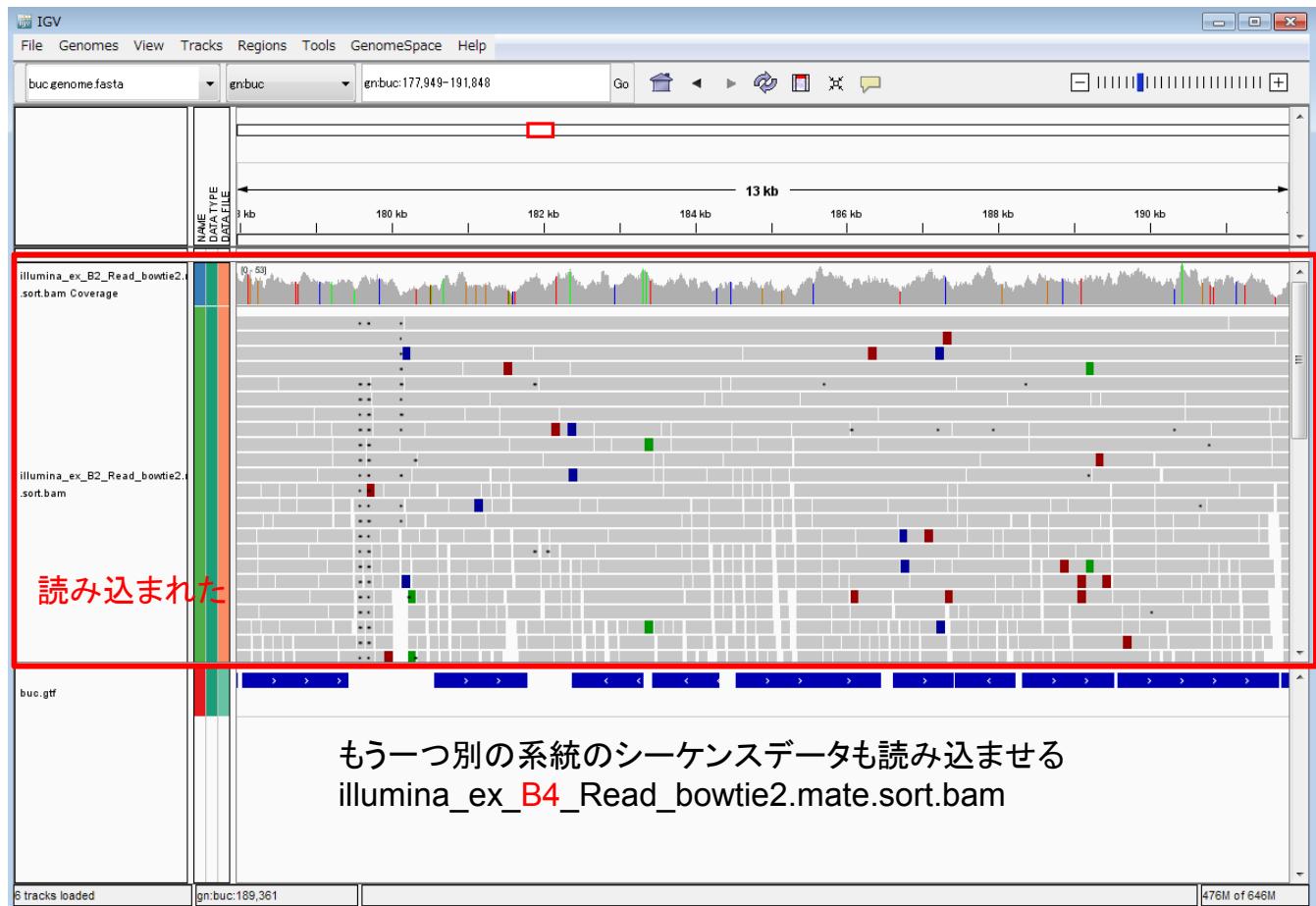
buc.genome.fasta genome rem... 511M of 635M

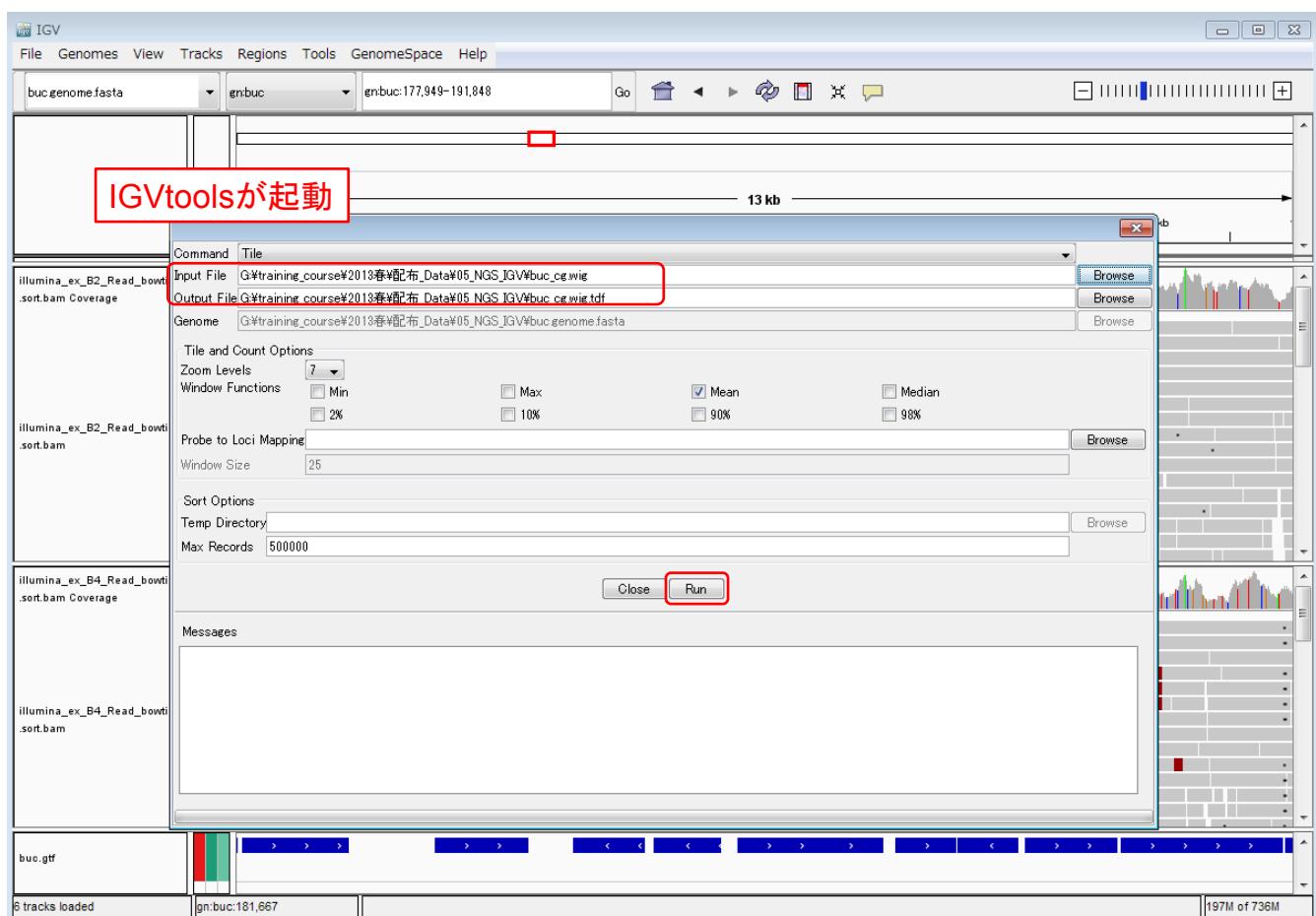
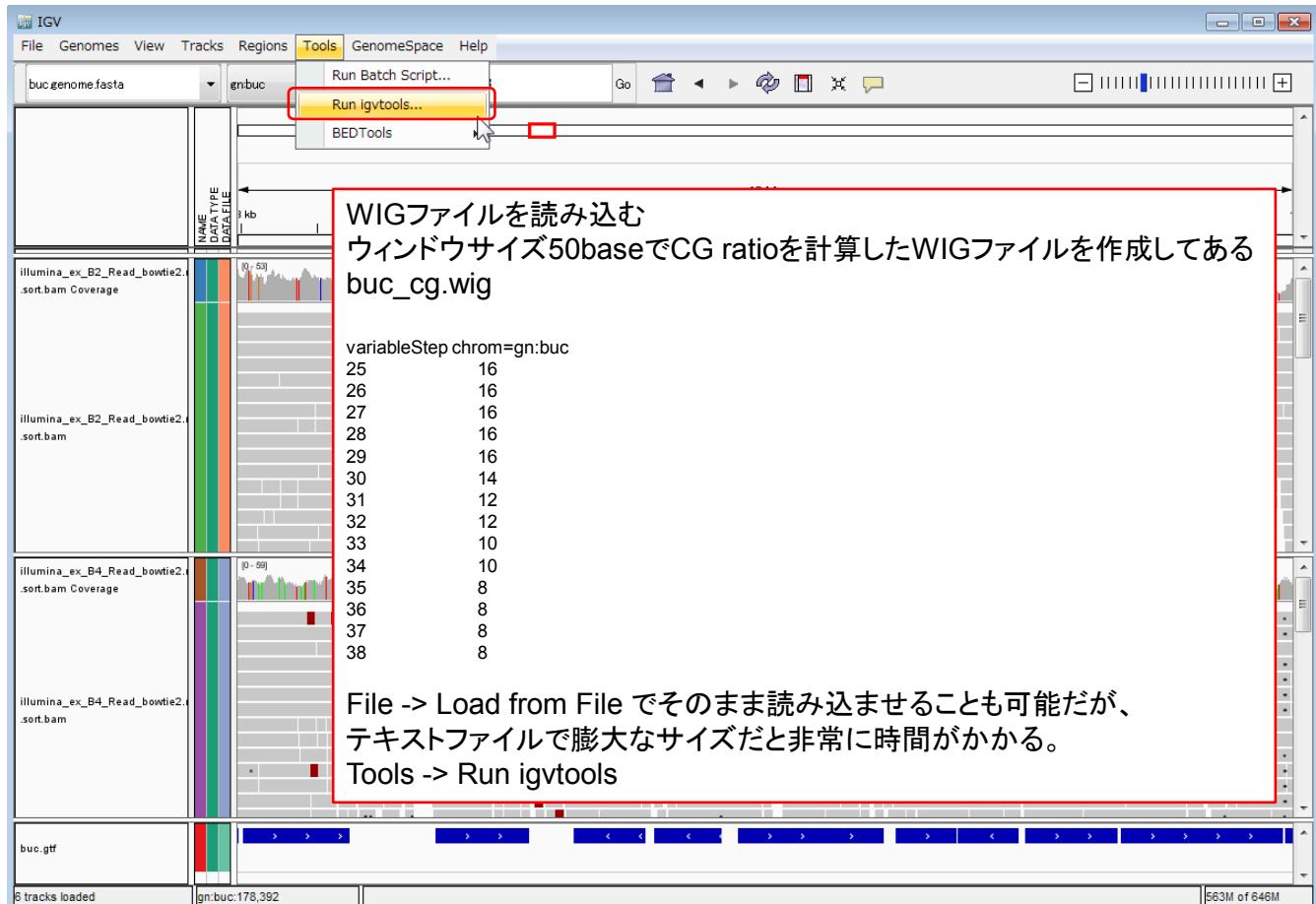
どれか1つの染色体 scaffoldを選ぶ

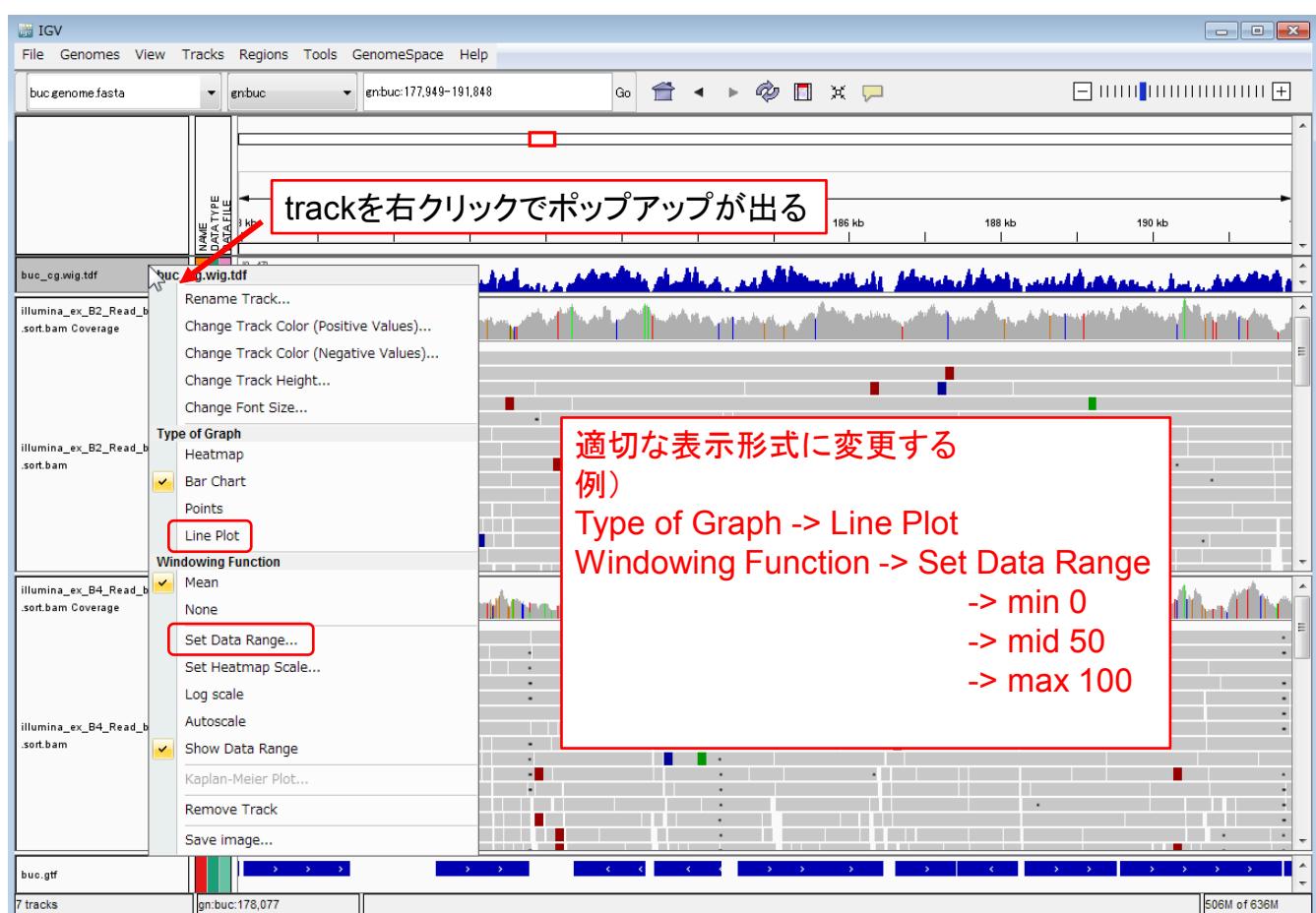
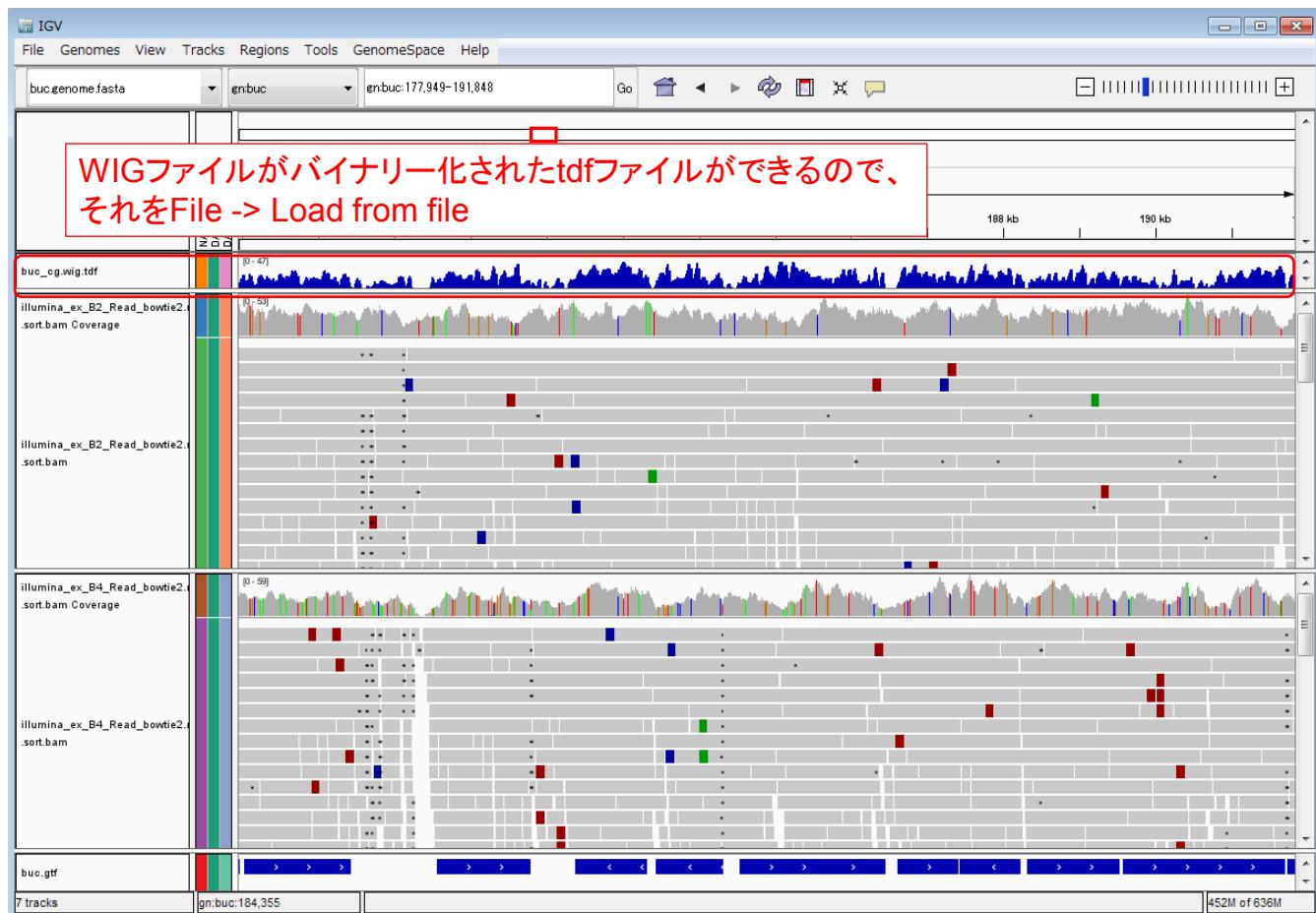
NAME DATA TYPE DATA FILE

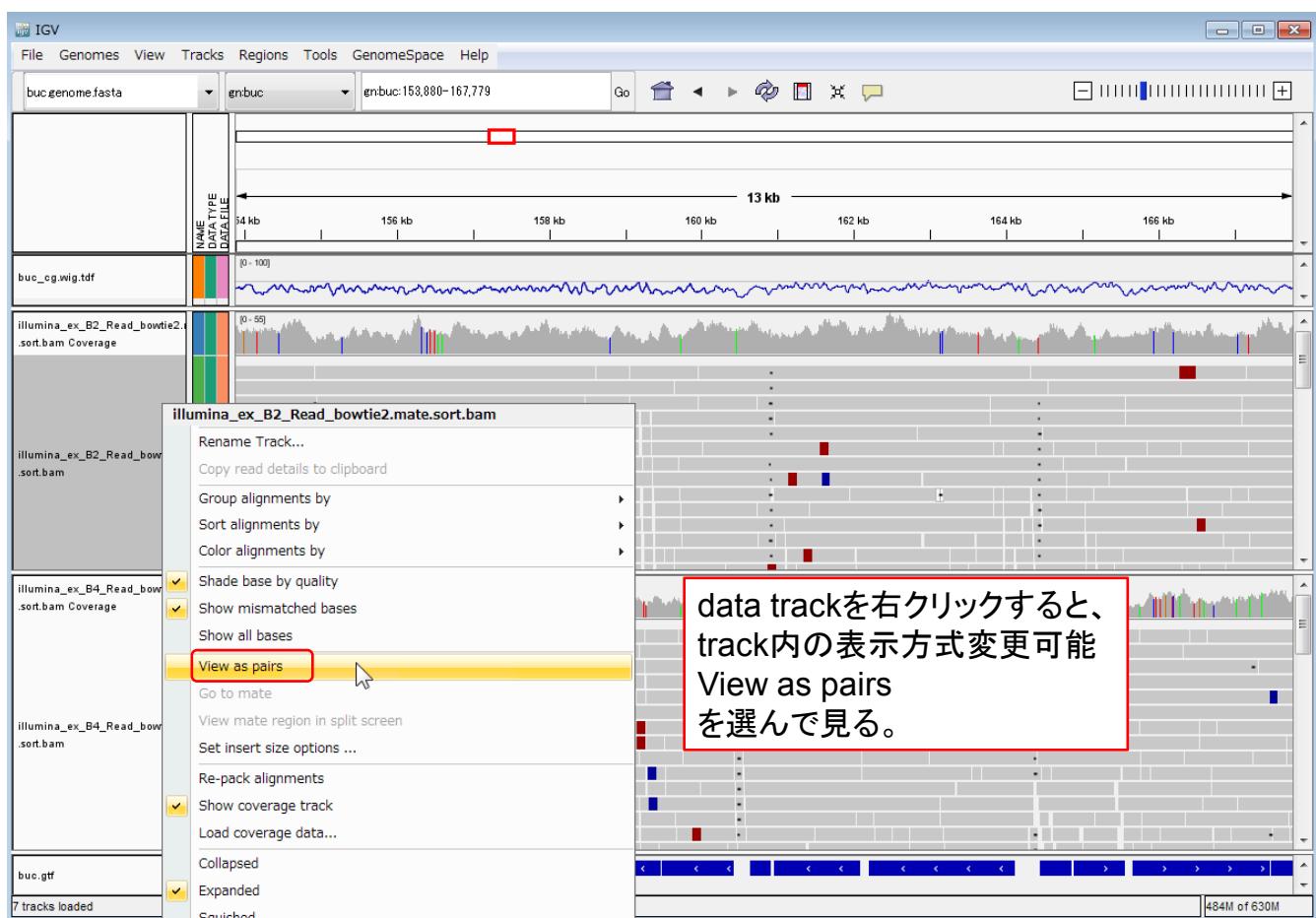
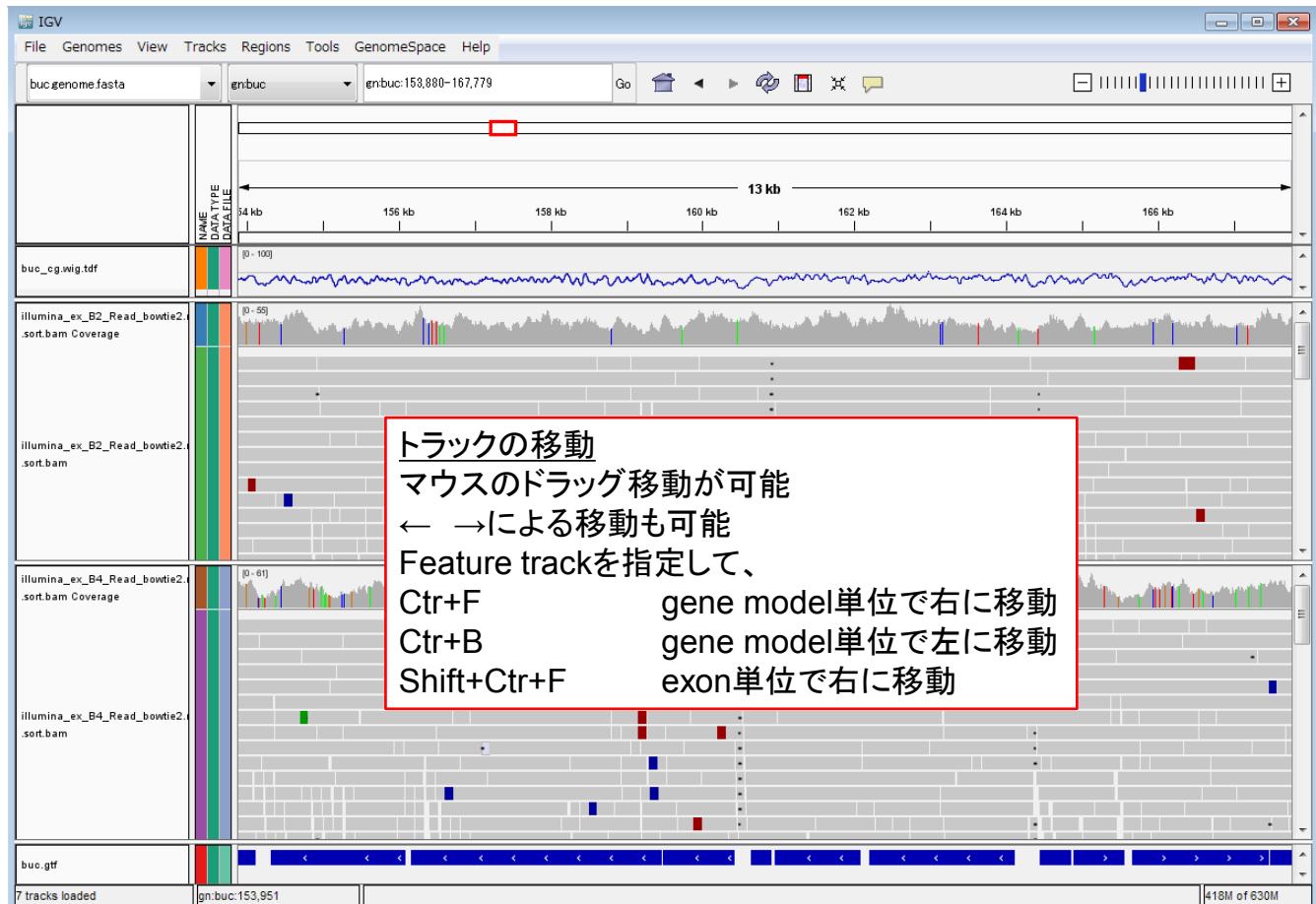
buc.gtf 2 tracks 537M of 635M

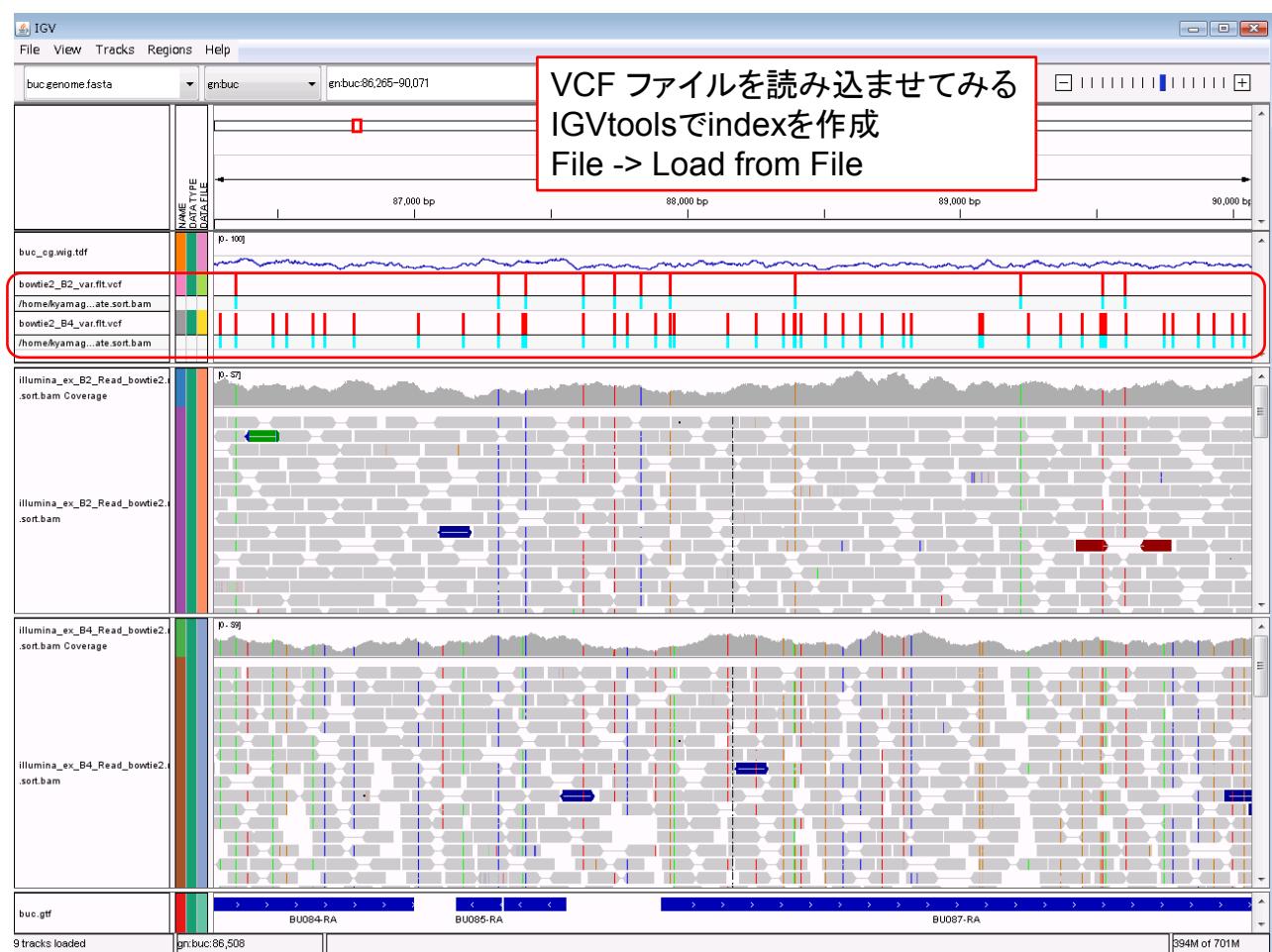
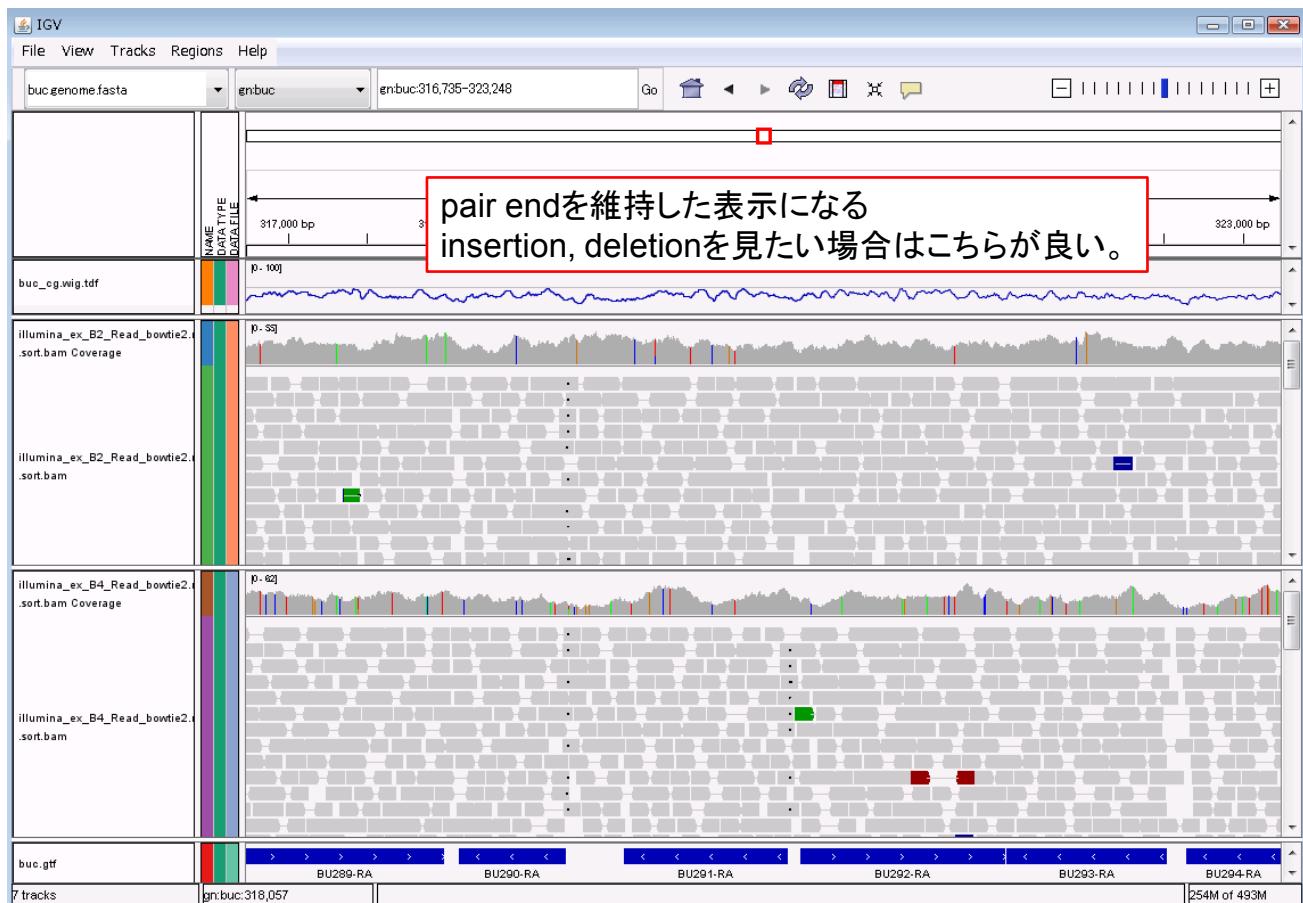


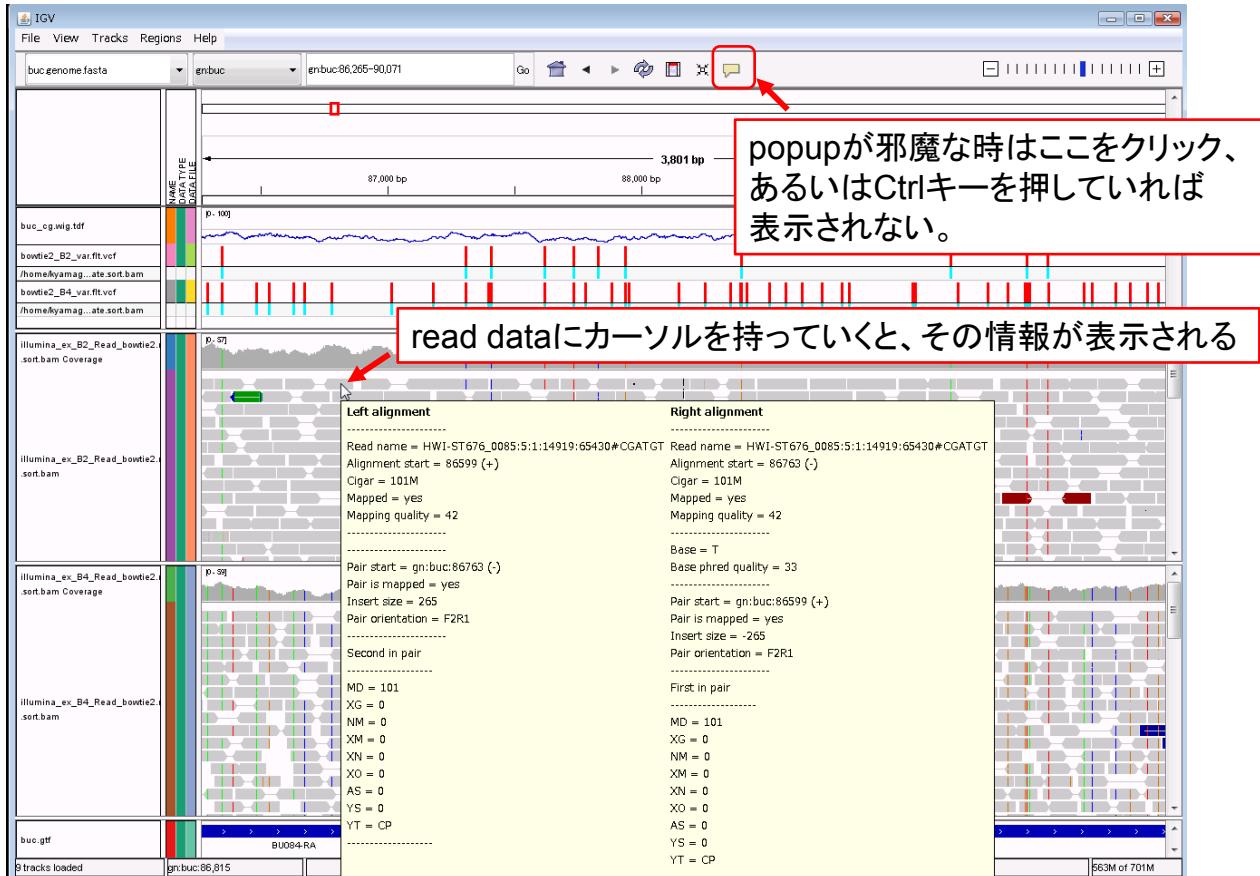












IGV紹介のまとめ

可視化ツールとして十分な機能を持つ

- ・無料
- ・比較的簡単・お手軽
- ・自分で見るためにも良し、人に見せるためにも良し
- ・利用範囲は次世代DNAシーケンサーに限定しない
広くゲノミクスの解析に有用

ごく一部のみの機能を紹介しました。
ウェブサイトを見ながら復習をお勧めします。

統計学入門

慶應義塾大学 先端生命科学研究所
佐藤昌直

統計の役割（一般論）

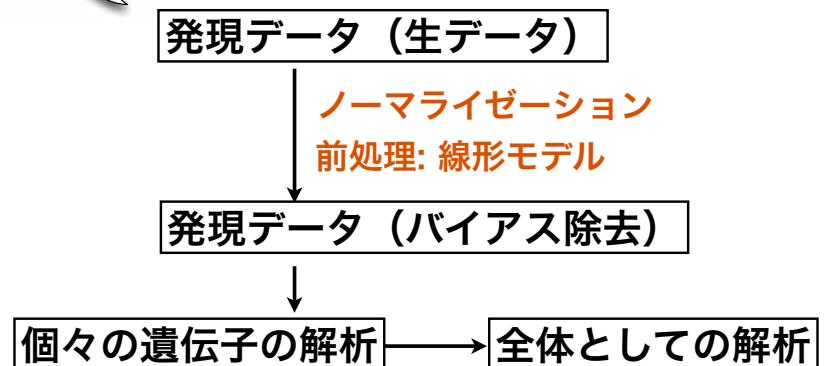
- 仮説検定
- 推定
- 予測（モデル構築）

多くの遺伝子発現解析

遺伝子発現解析における統計の役割

まずはそれ以前の
心得から

解析の流れ



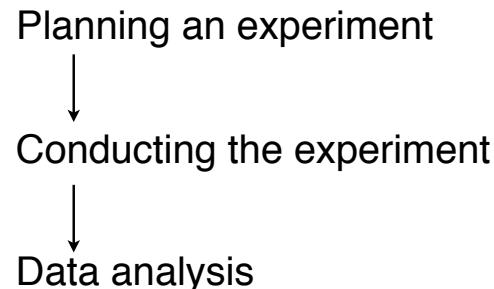
私の担当（統計解析）で 重視しているポイント

- 研究全体における統計の役割、
実験と統計との連携を意識する
- 遺伝子発現解析の基礎的な概念を
解説する

これらをこれから学習していくためには

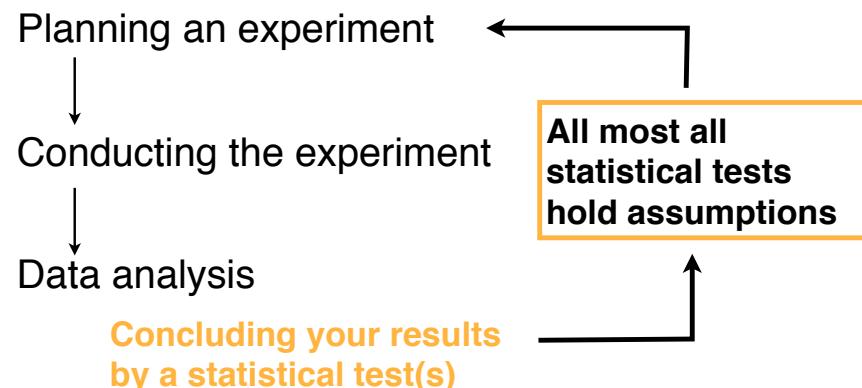
- 測定、統計とは何かを見直す
- 汎用される統計の仕組みを知る
- 教科書を読めるように統計用語・
表記に慣れる
- 道具を準備する - R

A workflow (that you might imagine)



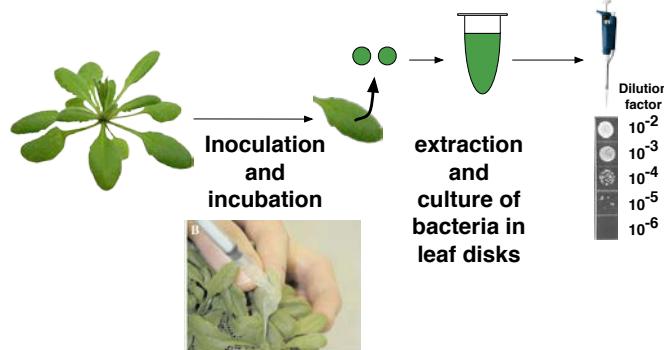
**Concluding your results
by a statistical test(s)**

**Reality: You have to design your
experiments **BEFORE** you obtain
results**



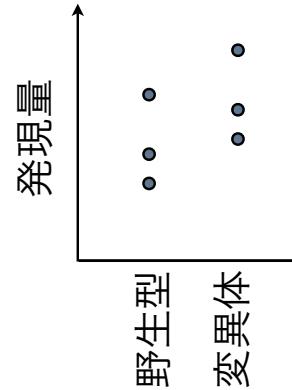
データのばらつきと 実験デザイン・統計学的観点

例: バクテリア増殖定量



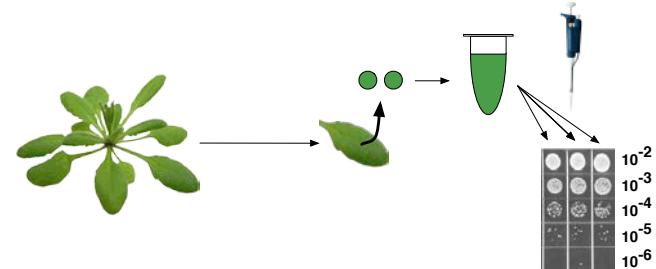
Katagiri, Thilmony R, and He S (2002) The *Arabidopsis Thaliana-Pseudomonas Syringae* Interaction. The *Arabidopsis* Book.

測定データはバラつく

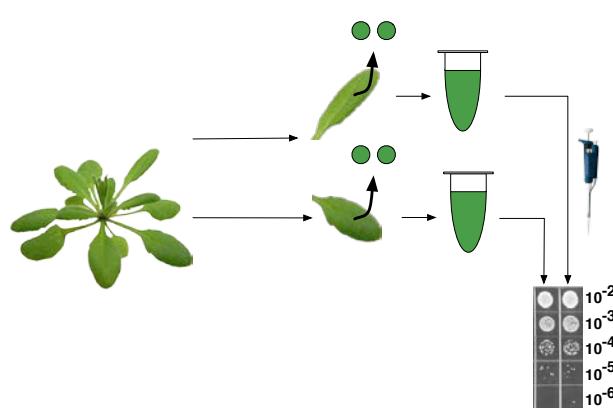


- 実験（測定）を反復する
- 何を「真」と考えるか
- 論文として発表できるデータには**再現性**が必要

反復？



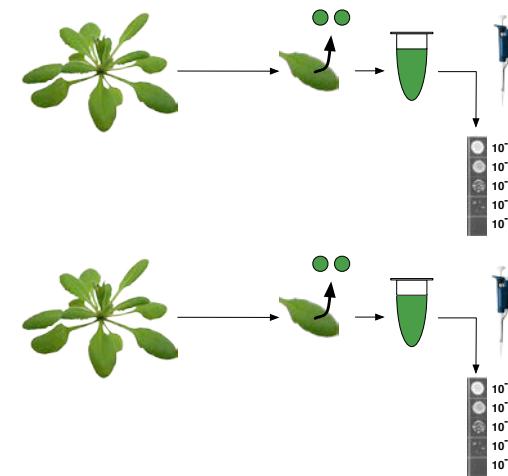
反復？



我々が1データポイントから
得ているもの

- ・ 生物学的にはらつきの中のある1点
- ・ 測定技術のはらつきの中のある1点

反復？



我々が1データポイントの
測定で得ているもの

- ・ 生物学的にはらつきの中のある1点
- ・ 測定技術のはらつきの中のある1点

測定における2要因

- Precision - 精度
- Accuracy - 正確度

測定における2要因

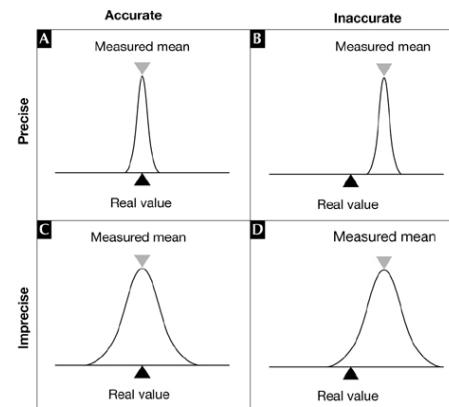
- Precision - 精度
ある1測定を繰り返した際のばらつきの
尺度

測定における2要因

- Accuracy - 正確度

ある測定値が「真の値」にどれだけ近い
かの尺度

測定における2要因



Real value: 真の値

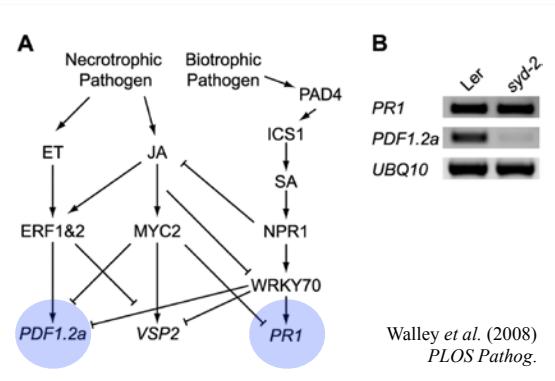
Measured mean:
測定値から
得られた平均

我々が1データポイントから得ているもの

- 生物学的にはらつきの中のある1点
- 測定技術のはらつきの中のある1点

“マーカー遺伝子”測定

- 何が再現されうるか？再現されたとするか？



明瞭な違いを示す遺伝子:
明瞭な再現性

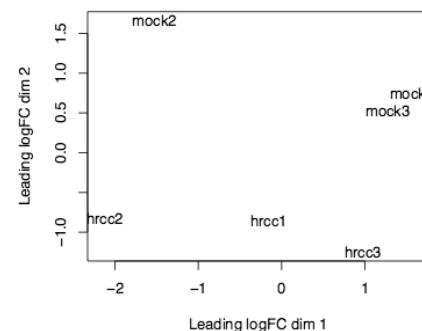
Walley et al. (2008)
PLOS Pathog.

定量的測定が可能且つ要求される時代の再現性のあるデータとは何か？

- 何が再現されうるか？再現されたとするか？
- いつ行っても再現できる？
- どこで行っても再現できる？
- 誰が行っても再現できる？

“トランскルiptーム”測定

- 何が再現されうるか？再現されたとするか？



網羅的測定:
再現性の
再定義

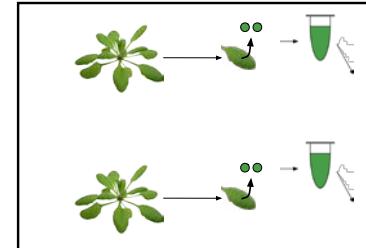
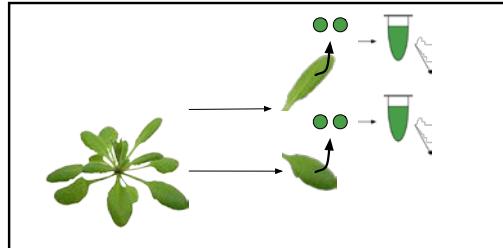
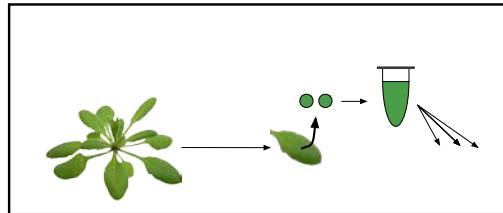
定量的測定が可能且つ要求される時代の 再現性のあるデータとは何か？

- ・何が再現されうるか？再現されたとするか？

- ・いつ行っても再現できる？
- ・どこで行っても再現できる？
- ・誰が行っても再現できる？

バラつきの
定量と割当て

何を知るための実験か？
再現性のあるデータとは何か？
どのように反復を行うのが適切か？



統計学を使って我々ができる事を
考えてみましょう

我々にできる事

少数の測定値から
「母集団」を推定すること

我々の実験対象

- ある遺伝子型の生物の
- ある環境での + 制御不能な実験要因
- ある遺伝子の発現量 + 生化学反応のノイズ

我々にできる事

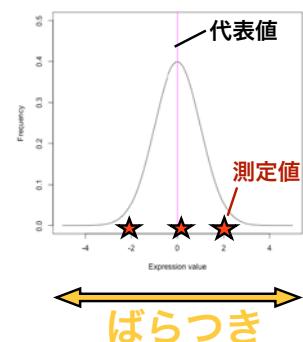
少数の測定値から
「母集団」を推定すること

生体サンプルを繰り返し取る:
biological replicates

同一サンプルを繰り返し測る:
technical replicates

母集団を推定する統計量

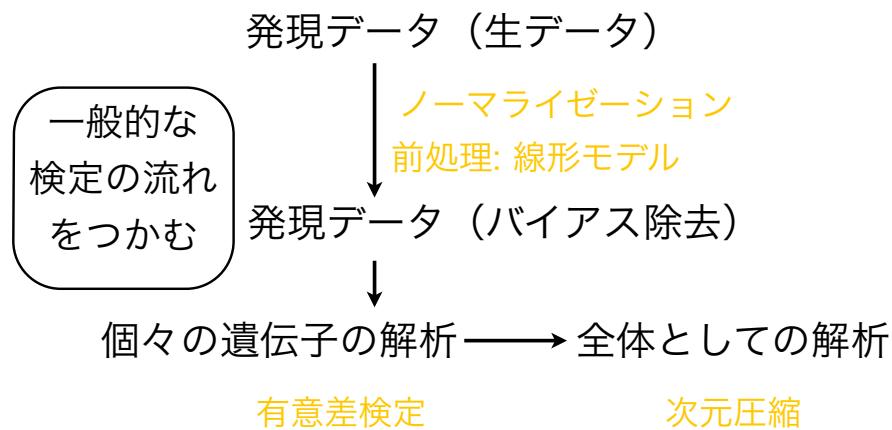
1. (真の値に近い)代表値



2. ばらつきの範囲

仮説検定 - t検定を例に

解析の流れ



この講義の目標

- t検定の背景知識を得る - 勉強のとっかかりを作る

- | | |
|--------|-------|
| • t統計量 | • t分布 |
| • 自由度 | • p値 |

統計における検定の手続き

1. 仮説を立てる
2. 統計量を求める
3. 求めた統計量を確率分布に照らし合わせる
4. 判定: 求めた確率が棄却限界値より大きいか、小さいか

統計における検定の **t検定** 手続き

1. 仮説を立てる
2. 統計量を求める
3. 求めた統計量を確率分布に照らし合わせる
4. 判定: 求めた確率が棄却限界値より大きいか、小さいか

- 2つのサンプル間で遺伝子発現量（平均値）の違いがある？
- 平均、標準誤差、自由度からt統計量を求める
- t分布からp値を求める
- 有意差の判定

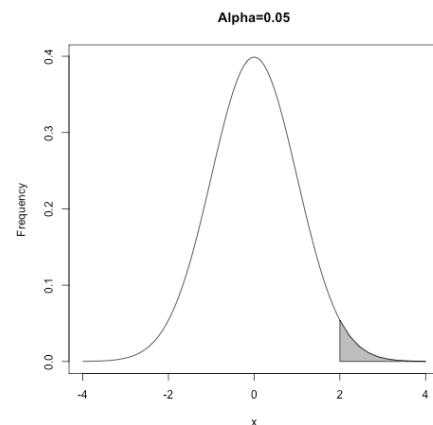
1. 仮説を立てる: 帰無仮説

- 最終的に棄却される仮定:
「AとBに差がある」かを検定する場合は「AとBには差がない」と仮定する

2. 統計量を求める:

- 統計量: データから導いた具体的な数値
↔ 母数: 未知の数値

3. 確率分布を求める:



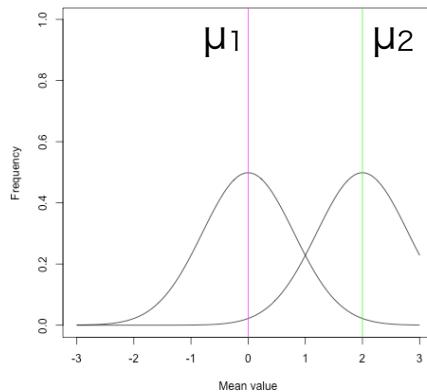
4. 判定: 帰無仮説が棄却されるか?

- 最終的に棄却される仮定:
「AとBに差がある」かを検定する場合は「AとBには差がない」という仮定

t検定:

2サンプルの平均の検定

- 平均値 = μ_1, μ_2
- データは正規分布

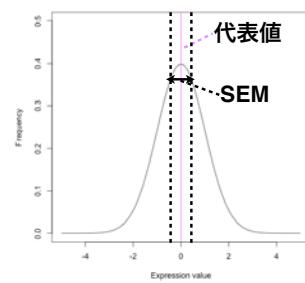


統計量その2:

平均値も推定値

(平均) 標準誤差

$$SEM = \frac{s}{\sqrt{n}}$$



統計量その1

平均値: 相加平均。すべてのデータを足して、データ数で割って得られる値

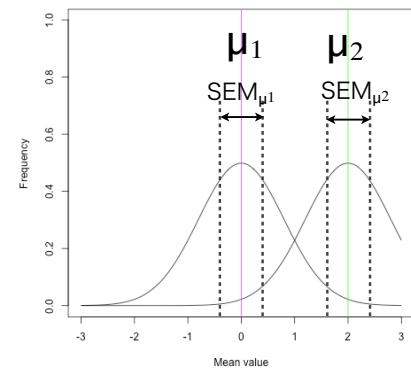
$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

統計量その3:

平均の差とその誤差

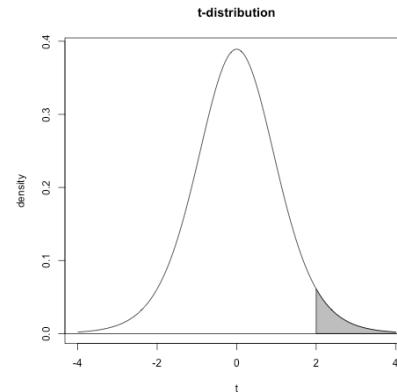
- t統計量

$$t = \frac{|\hat{\mu}_1 - \hat{\mu}_2|}{SEM_{|\hat{\mu}_1 - \hat{\mu}_2|}}$$



確率分布-t分布

- 得られたt統計量がどのくらいの確立で起きうるか
- t分布の確率分布を標本のt統計量と自由度を使って参照



自由度とは？

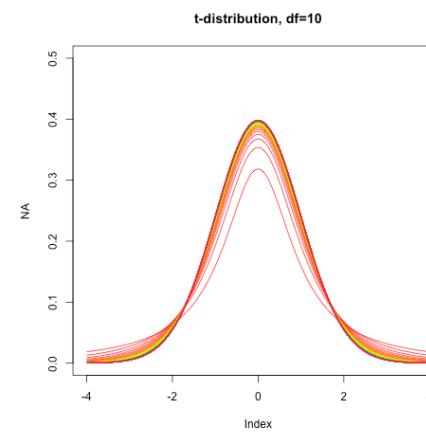
統計量を求めるのに使うことができる「独立」な標本数

我々の測定、検定では：

- 母分散が未知
- よって、確率密度は自由度によって変化

例) 3つの観察で得られた平均値と100観察から得られた平均値はどちらが確からしいか

1から100までの自由度でのt分布

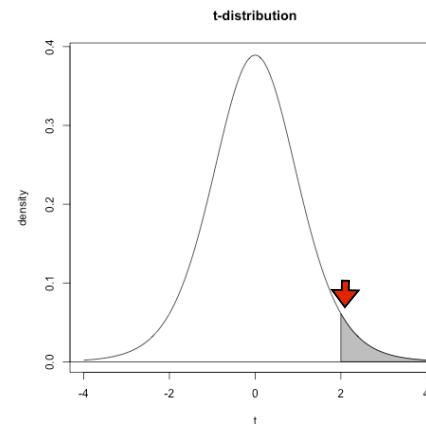
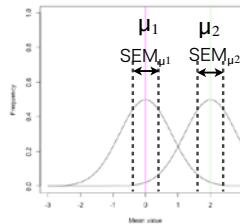


p値とは：

- ・ 標本に基づいた統計量が帰無仮説の下、起きうる確率
- ・ 多くの場合、0.05が危険率

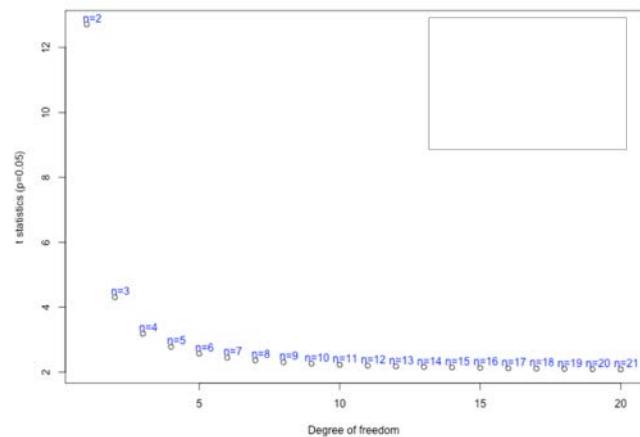
確率分布-t分布

$$t = \frac{|\hat{\mu}_1 - \hat{\mu}_2|}{SEM_{|\hat{\mu}_1 - \hat{\mu}_2|}}$$



t分布表を参照する→Rで求めましょう

補足:
t統計量
自由度
反復
p値
検出力



多重検定の補正

p値とは：

- ・ 標本に基づいた統計量が帰無仮説の下、起きうる確率
- ・ 多くの場合、**0.05**が危険率

p値とは：

- ・ 標本に基づいた統計量が帰無仮説の下、起きうる確率
- ・ 多くの場合、**0.05**が危険率 = **100回に5回起きる**

多重検定の補正

- ・ $p = 0.05$ の検定を100回*繰り返すと、**5回はランダムに間違い**

多重検定の補正

Bonferroniタイプ

False discovery rate (FDR):

- Benjamini-Hochberg
- Storey

*NGS解析では数万回以上繰り返すことになります

Bonferroniタイプの多重検定の補正

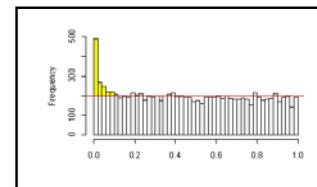
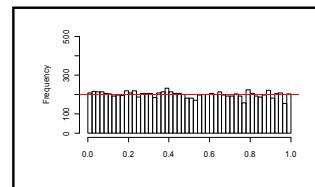
危険率を検定数で調整

$$\text{危険率} = \alpha / k$$

α : 元の危険率、

k: 検定数

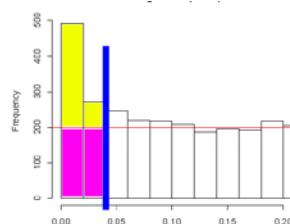
False Discovery Rate (FDR)



False Discovery Rate (FDR)

q値:

補正されたp値。そのq値以下の検定のうち、どのくらいの割合でfalse positiveが含まれているか。

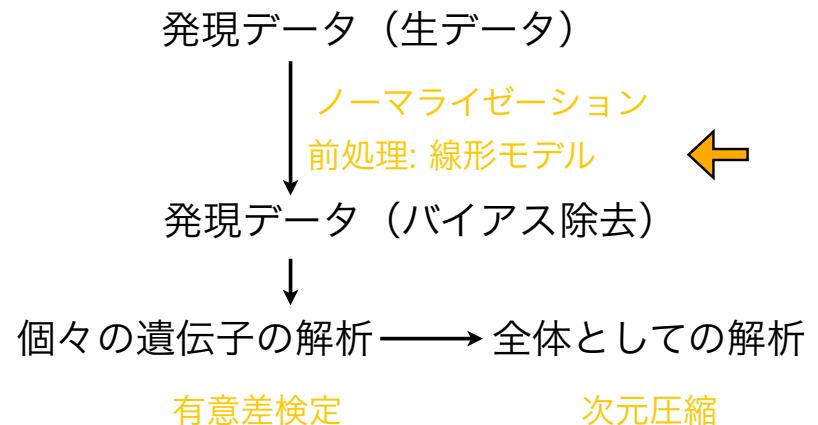


復習／発展学習

- 検定の手順
 - 統計量
 - 自由度
 - p値
- 統計解析の結果は確率 → 多重検定の補正:
ランダムな危険率以下の検定をどう補正するか？
- Storeyの方法によるq値の求め方
- 多重検定の補正における仮定: 時系列データにFDRは使ってよいか？

分散分析・線形モデル:
多変数データを系統立てて解析する
- 実験デザインと統計の連携

解析の流れ

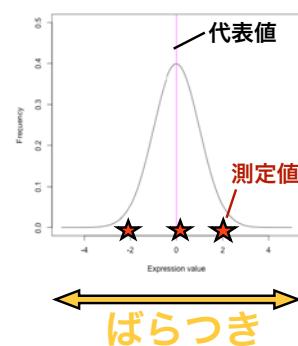


- 線形モデルの概念を掴む
- 実験デザインがどう統計に影響するかを考えるきっかけとする

リマインド:
母集団を推定する統計量

1. (真の値に近い)代表値

2. ばらつきの範囲



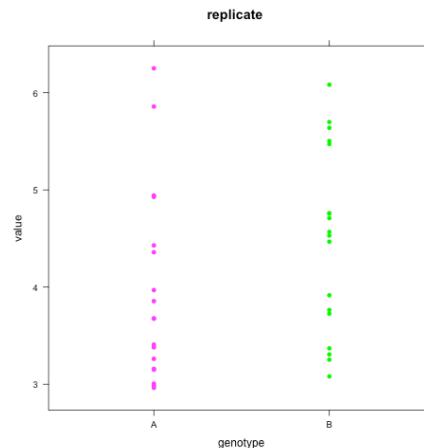
t検定: 平均値の検定

$$x_i = \bar{x} + (x_i - \bar{x})$$

偏差: 平均値からのはらつき

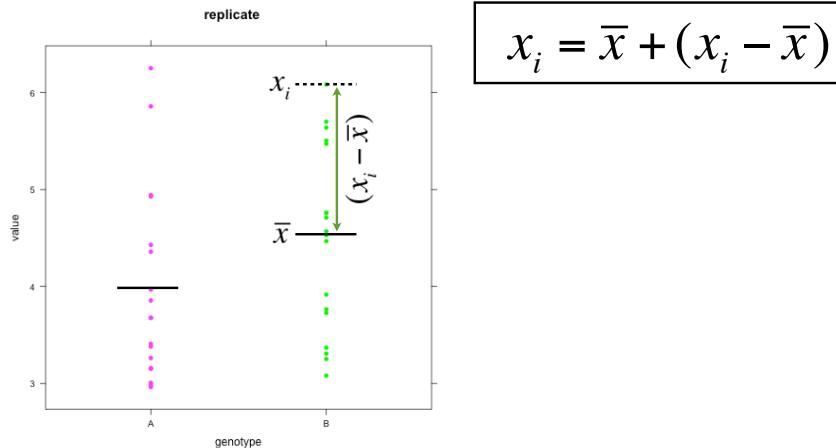
- genotype A, Bについて

6検体ずつ3回反復して計測



- genotype: A, B
- replicate: 1, 2, 3
- value:
計18個/ genotype

線形モデルの枠組みで考えてみる

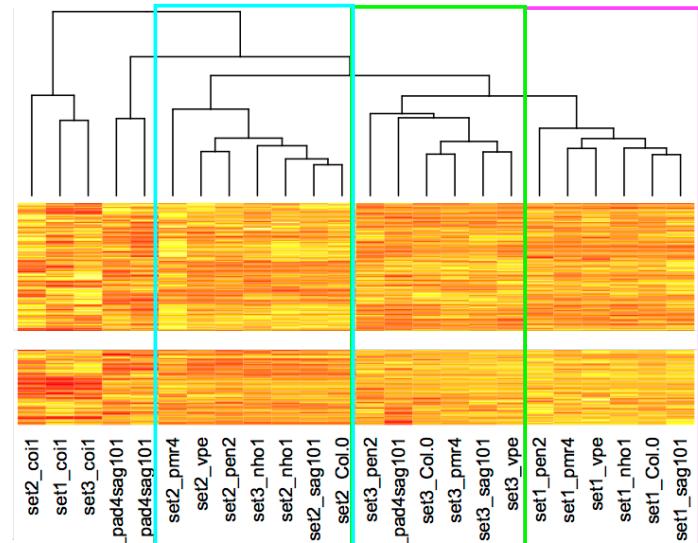


$$x_i = \bar{x} + (x_i - \bar{x})$$

$$x_i = \bar{x} + \varepsilon_i$$

残差 (観察値-推定値):
想定要因では説明できない
データの変動

考慮するのは1要因で良いか？



観察値を複数要因の影響によるものとして分解

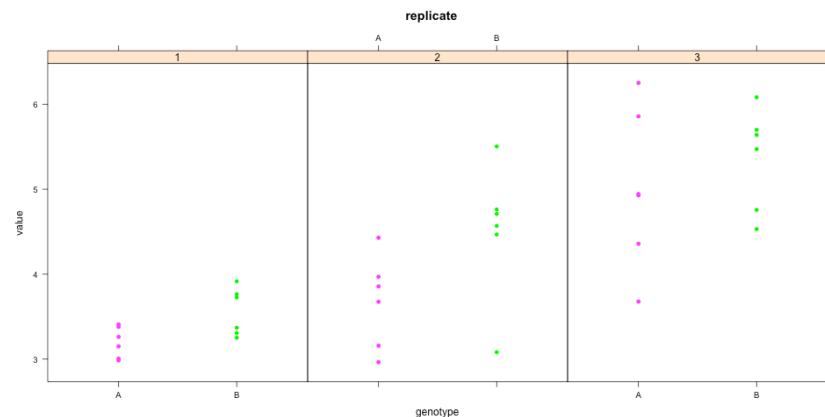
$$x_i = \bar{x} + (x_i - \bar{x})$$

$$x_i = \bar{x} + \varepsilon_i$$

genotypeとreplicateの影響を同時に考えられないか？

$$O_{ij} = x_i + y_j + \varepsilon_{ij}$$

例: 2遺伝子型の測定を3反復したデータ

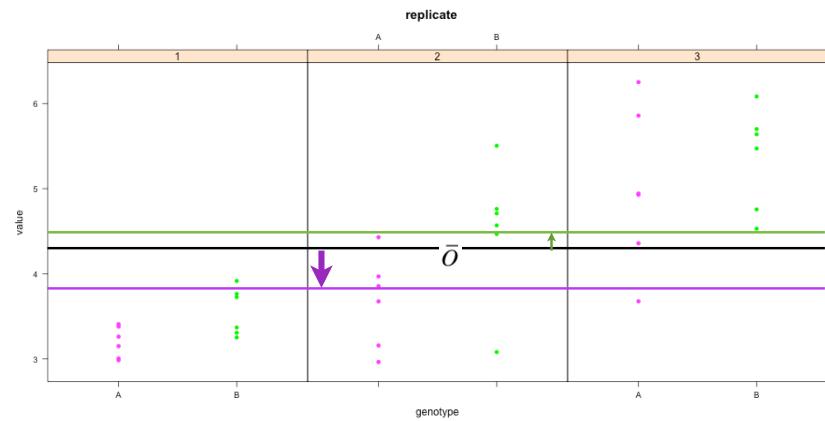


線形モデルの仕組み

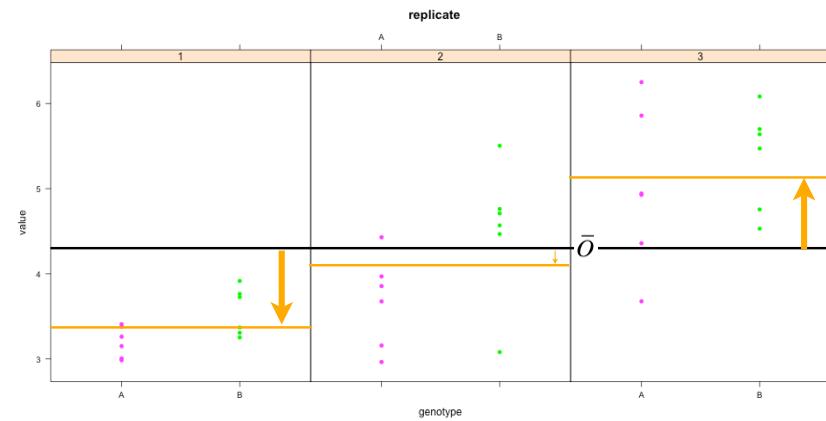
$$O_{ij} = x_i + y_j + \varepsilon_{ij}$$

$$O_{ij} = \bar{O} + (\bar{x}_{i\bullet} - \bar{O}) + (\bar{y}_{\bullet j} - \bar{O}) + \varepsilon_{ij}$$

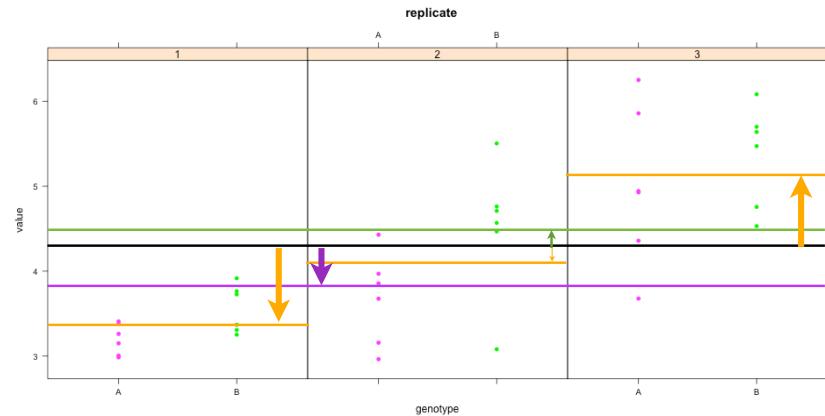
$(\bar{x}_{i\bullet} - \bar{O})$ 遺伝子型による変動



$(\bar{y}_{\bullet j} - \bar{O})$ 反復ごとの変動



各計測値は $O_{ij} = \bar{O} + (\bar{x}_{i\bullet} - \bar{O}) + (\bar{y}_{\bullet j} - \bar{O}) + \varepsilon_{ij}$ と表せる



分散分析・線形モデルの枠組み

$$\begin{aligned}
 O_{ij} &= x_i + y_j + \varepsilon_{ij} \\
 O_{ij} &= \bar{O} + (\bar{x}_{i\bullet} - \bar{O}) + (\bar{y}_{\bullet j} - \bar{O}) + \varepsilon_{ij} \\
 &\quad \downarrow \text{教科書・論文風に書くと} \\
 O_{ij} &= \mu + \alpha_i + \beta_j + \varepsilon_{ij}
 \end{aligned}$$

応答変数 説明変数

線形モデルとは

応答変数 ~ 説明変数1 + 説明変数2 + + 誤差

と観察値を説明する（かもしれない）
変数でそれらの関係性を書き下すこと

- 実際には: Rでlmなどの関数を使う

実験デザインの重要性

- 線形モデルで推定・除去

$$O_{ij} = \alpha_i + \beta_j + \varepsilon_{ij}$$

α_i : 遺伝子型／処理など注目して
いる効果の要因

β_j : 反復（実験日時）／実験者
などバイアス要因

- α_i の推定値、標準誤差のみを使う

実験デザインの重要性

- -omicsデータは”batch effect”という体
系的なバイアスが多くの場合、混入する。
例: 実験時期、餌



Nature Reviews Genetics (2010) 11, 733-

- 線形モデルで推定・除去

再現性のあるデータとは何か？

- 自分自身で再現できる
- いつ行っても再現できる
- どこで行っても再現できる
- 誰が行っても再現できる

実験デザインの重要性

- 線形モデルで推定・除去

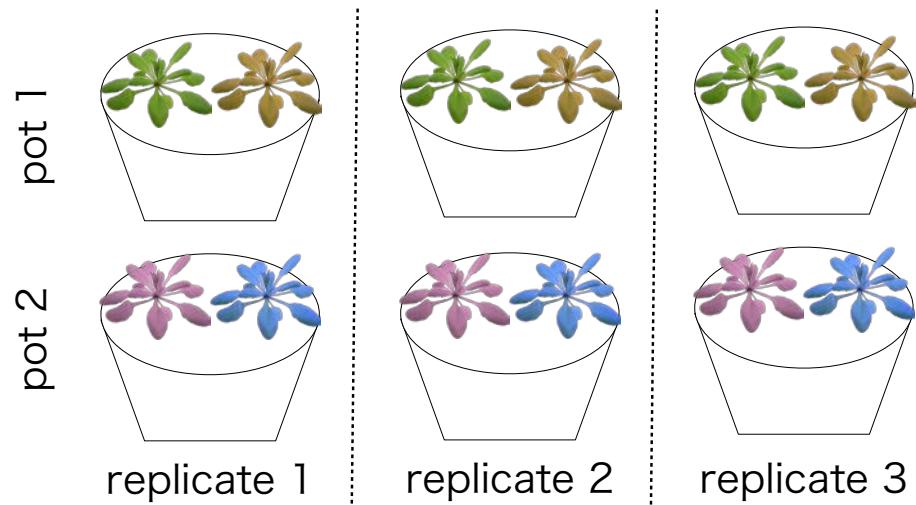
$$O_{ij} = \alpha_i + \beta_j + \varepsilon_{ij}$$

α_i : 遺伝子型／処理など注目している効果の要因

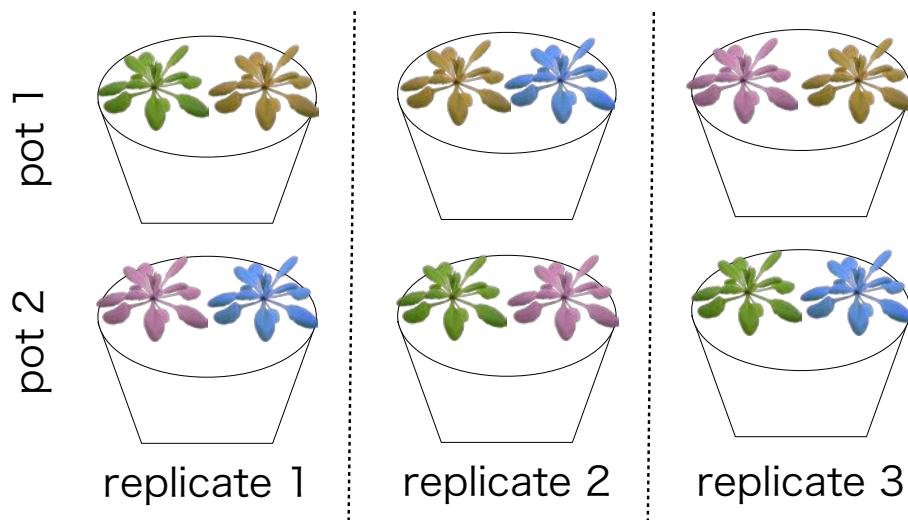
β_j : 反復（実験日時）／実験者などバイアス要因

- α_i の推定値、標準誤差のみを使う

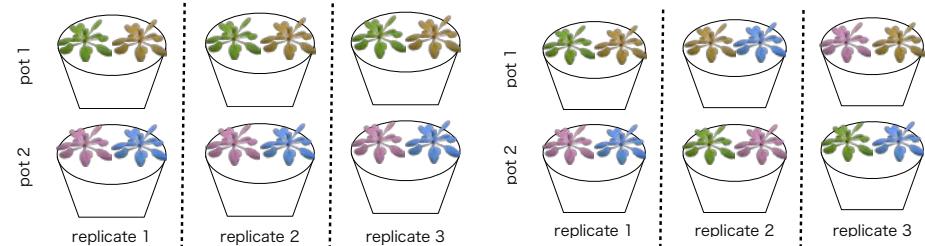
実験デザインの重要性:
genotype+replicate+potモデルを当てはめるには？



実験デザインの重要性:
genotype+replicate+potモデルを当てはめるには？



実験デザインの重要性:
genotype+replicate+potモデルを当てはめるには？

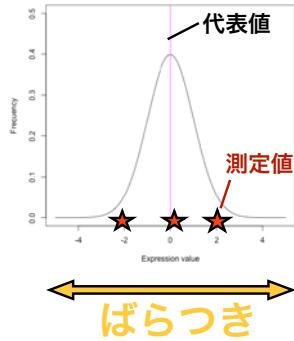


↑
genotypeとpotが独立ではない
(切り分けられない)

リマインド:

母集団を推定する統計量

1. (真の値に近い)代表値



2. ばらつきの範囲

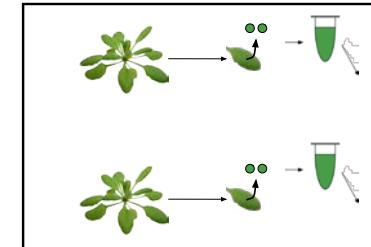
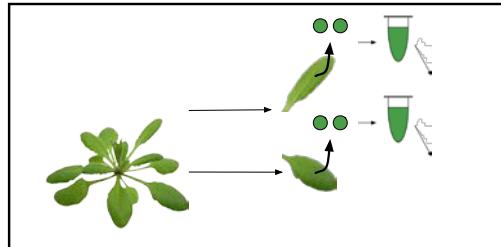
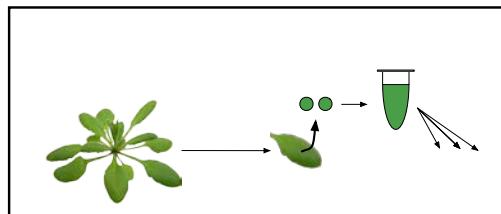
実験デザインの重要性

- 要因効果を推定するための実験デザイン
 - 各実験要因を適切に反復させた実験デザイン
- 実験デザインとモデル
 - 要因: データ取得 「前」 に想定しておくもの
 - データの変動を説明しない要因を解析時に減らすことは可能。実験デザイン時に計画しなかった要因を増せない。

何を知るための実験か?

求めたい代表値は何の代表値か、扱うばらつきは何に由来するか?

ある実験デザインで求めうる代表値・ばらつきは何を表すか?



まとめ

- 計測データセットに影響を与える要因が一つではない場合、分散分析・線形モデルの枠組みが有効
- 理屈は難しいかもしれないが、Rで簡単に実行できるので実験デザインと連動したモデルを立てることが重要

復習／発展学習

- 回帰（最小二乗法）
- 実験計画法
- 交互作用
- Bioconductor: limma、edgeRパッケージ

R入門

基礎生物学研究所
情報管理解析室
内山 郁夫

1

Rとは

- ベル研究所で開発された統計処理言語「S」を基に、フリーソフトとして開発された統計解析環境・プログラミング言語
- コマンドラインインターフェイスが基本。対話的なコマンド実行による解析のほか、スクリプトを書いて一括処理を行うことも可能
- ベクトル・行列演算が簡便かつ効率的に行える
- 独自の関数を作成することによって機能拡張が可能
- 作成した関数等をパッケージ単位でまとめることにより、機能拡張が容易に行える。様々なパッケージを導入することにより、最先端の統計手法を用いることができる。

2

作業ディレクトリの設定

- 作業ディレクトリ：読み込むデータファイルや結果を書き出すファイルを保存するディレクトリ。
- メニューから、その他→作業ディレクトリの変更をえらんでディレクトリを選択する。

作業ディレクトリ data/IU に移動

- 作業ディレクトリを常に固定したい場合は、環境設定から起動タブを開いて初期ディレクトリを設定する。
- 作業ディレクトリをコマンドで設定することも可能
`setwd("ディレクトリ名")` 作業ディレクトリの設定
`getwd()` 作業ディレクトリの確認

3

スカラー演算

```
> 1+3
[1] 4
> 1+3*5
[1] 16
> a <- 1+3*5
> a
[1] 16
> a > 10
[1] TRUE
```

通常の通り、*(かけ算)は+(足し算)より優先する

結果を変数 a に代入する

a とだけタイプすると、変数aの内容が出力される

論理演算は、論理値TRUEまたはFALSEを返す

4

変数と代入

- 計算結果を変数に代入することにより、結果を再利用できる
 - 変数名にはアルファベット、数字、'.' (dot)、'_' (underscore)が利用できる。ただし、先頭はアルファベットまたはドット。
 - 大文字と小文字は区別される。
- 代入を表す演算子には、<- = -> の3通りがある。以下の3つはいずれも a に 4 が代入される。

```
> a <- 1 + 3
> a = 1 + 3
> 1 + 3 -> a
```

5

ヒストリー(履歴)

- コンソール上で、上下の矢印キー(↑↓)によって、コマンドの履歴を前後にたどることができる。
- 左右の矢印キー(←→)によって、カーソルを左右に動かせる。これによってコマンドの編集ができる。
- 以下のコントロールキーを使った操作も可能
 - Control+P 履歴を前に移動(↑と同じ)
 - Control+N 履歴を後ろに移動(↓と同じ)
 - Control+B カーソルを左に移動(←と同じ)
 - Control+F カーソルを右に移動(→と同じ)
 - Control+A カーソルを行の先頭に移動
 - Control+E カーソルを行の最後に移動
- GUIからヒストリーパネルを使って履歴をたどることも可能



ヒストリーパネルの表示・非表示

6

ベクトル

```
> a <- c(1, 3, 7, 4, 6) ベクトルは関数 c を用いて
> a 作成する
[1] 1 3 7 4 6
> b <- 3:7 3から7までの連続した整数
> b
[1] 3 4 5 6 7
> length(a) ベクトルaの長さ(要素数)
[1] 5
> b[3] bの3番目の要素
[1] 5
```

7

ベクトル演算

$a=c(1,3,7,4,6)$; $b=c(3,4,5,6,7)$ と設定されている

```
> 1 + a aの各要素に1を加える
[1] 2 4 8 5 7
> 2 * a aの各要素を2倍する
[1] 2 6 14 8 12
> a + b aとbの要素ごとの和をとる
[1] 4 7 12 10 13
> a * b aとbの要素ごとの積をとる
[1] 3 12 35 24 42
> a > 3 aの要素ごとに3より大きいか比較する
[1] FALSE FALSE TRUE TRUE TRUE
> a < b aとbを要素ごとに大小を比較する
[1] TRUE TRUE FALSE TRUE TRUE
```

8

基本統計量の計算

`a=c(1,3,7,4,6)`と設定されている

> <code>length(a)</code>	aの長さ(要素数)
> <code>sum(a)</code>	aの要素の合計値
> <code>mean(a)</code>	aの要素の平均値
> <code>median(a)</code>	aの要素の中央値
> <code>var(a)</code>	aの要素の不偏分散
> <code>sd(a)</code>	aの要素の標準偏差

問題) `mean(a)`を`sum(a)`と`length(a)`を使って計算してみよう。

9

ベクトル要素の抽出

`a=c(1,3,7,4,6)`と設定されている

> <code>a[2:4]</code>	# 2番目から4番目までの要素
[1] 3 7 4	
> <code>a[c(3,5,2)]</code>	# 3, 5, 2番目の要素
[1] 7 6 3	
> <code>a[c(T,T,F,F,T)]</code>	# Tである要素だけ出力
[1] 1 3 6	
> <code>a[a > 3]</code>	# a>3がTRUEである要素だけ出力
[1] 7 4 6	

10

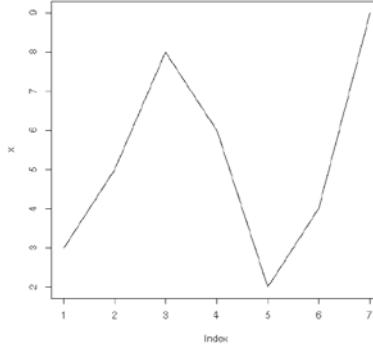
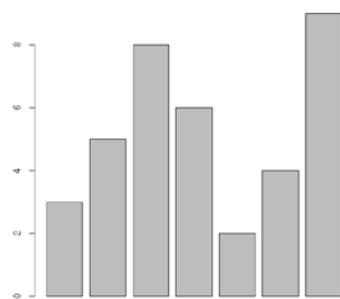
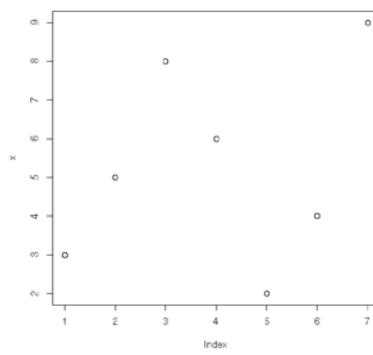
並べかえ(ソート)と順序づけ(ランク)

```
> x <- c(3,5,8,6,2,4,9)
> sort(x)           # 小さい順に並べかえ
[1] 2 3 4 5 6 8 9
> sort(x, decreasing=TRUE)  # 大きい順
[1] 9 8 6 5 4 3 2
> rank(x)           # 小さい順に順序づけ
[1] 2 4 6 5 1 3 7
```

11

プロットの作成(1)

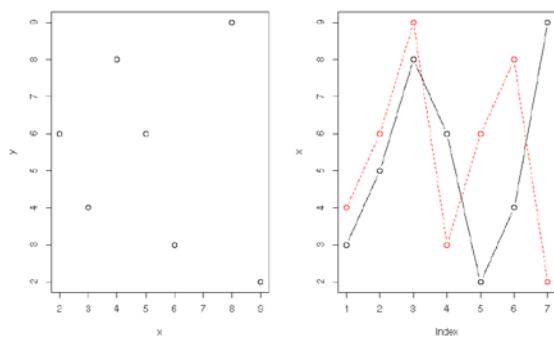
```
> x
[1] 3 5 8 6 2 4 9
> plot(x)
> plot(x,type="l")
> barplot(x)
```



12

プロットの作成(2)

```
# x <- c(3,5,8,6,2,4,9)と設定されている
> y <- c(4,6,9,3,6,8,2)
> par(mfrow=c(1,2)) # 2つのプロットを表示
# x,y をx軸、y軸にとって散布図としてプロット
> plot(x,y)
# x,yを別々にプロット。まずxをプロット。点と線を両方描く。
> plot(x,type="b")
# yを重ねてプロット。linesは枠を描き直さずに線だけを引く。
# lty はline type, col は colorを指定。
> lines(y, type="b", lty=2, col=2)
```



13

Rにおける基本データ型

- Rの「原子データ型」として以下のものがある
 - 数値型 numeric(実数または整数)
 - 論理型 logical(TRUE/FALSE)
 - 文字列型 character("abc", "123" のように、二重引用符で囲まれた文字列として表す)
- Rの原子データはベクトルである。スカラ一値も長さ1のベクトルである。
- 異なる型のデータを集めた構造として「リスト」がある。
- mode(x) によって、変数 x の型を調べられる

14

演算子

- 算術演算子
 - + (加算) - (減算) * (乗算) / (除算),
 - % / % (整数除算) %% (剰余) ^ (累乗)
 - 例) $5 / 2 (=2.5)$ $5 \% / \% 2 (=2)$ $5 \% \% 2 (=1)$
- 論理演算子
 - & (論理積「かつ」) | (論理和「または」) ! a (aの否定)
- 比較演算子
 - >, >=, <, <= (不等号) == (等しい) != (等しくない)

これらの演算子はベクトルの要素ごとにはたらく

(例) $a=c(1,3,7,4,6)$ と設定されている

```
> a >= 2 & a < 5
[1] FALSE TRUE FALSE TRUE FALSE
```

15

ワークスペースとオブジェクト

- コンソール上で `ls()` または `objects()` とすると、変数に保存されたデータ(オブジェクト)のリストを参照できる。

```
> ls()
[1] "a" "b" "x"
```

- ワークスペースブラウザ(メニューバー「ワークスペース」から起動)でより詳細な情報を閲覧可能
- `rm(変数名)` で、オブジェクトを消去できる。

16

関数

関数名(引数1, 引数2, ...)

- 関数は、一般に複数の引数を入力としてとり、何らかの計算を行って一つのオブジェクトを返す(戻り値)。
- 引数には、必須のものと省略可能なものがある。後者は省略すると「デフォルト値」が使用される。
- 引数は、順番によって指定する方法と、「名前=値」の形式で指定する方法がある。通例、必須の引数は前者、選択可能な引数は後者で指定する。

例1) ベクトル x を小さい順(increasing order)でソートする

`sort(x)`

例2) ベクトル x を大きい順 (decreasing order)でソートする

`sort(x, decreasing=TRUE)`

17

マニュアルの表示

- help(関数名)でマニュアルを表示する

> `help(median)`

```
median      package:stats      R Documentation
Median Value

Description:
  Compute the sample median.

Usage:
  median(x, na.rm = FALSE)

Arguments:
  x: an object for which a method has been defined, or a numeric
     vector containing the values whose median is to be computed.
  na.rm: a logical value indicating whether 'NA' values should be
        stripped before the computation proceeds.

Details:
  This is a generic function for which methods can be written.
  However, the default method makes use of 'sort' and 'mean' from
  package 'base' both of which are generic, and so the default
  method will work for most classes (e.g. "Date") for which a
  median is a reasonable concept.

Value:
  The default method returns a length-one object of the same type as
  'x', except when 'x' is integer of even length, when the result
  will be double.

  If there are no values or if 'na.rm = FALSE' and there are 'NA'
  values the result is 'NA' of the same type as 'x' (or more
  generally the result of 'x[FALSE][NA]').

References:
  Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) _The New S
  Language_. Wadsworth & Brooks/Cole.

See Also:
  'quantile' for general quantiles.

Examples:
  median(1:4) # = 2.5 [even number]
  median(c(1:3, 100, 1000)) # = 3 [odd, robust]
```

Description: 関数の簡単な説明

Usage: 関数の呼び出し方

`median(x, na.rm = FALSE)`

x: 必須の引数

na.rm: 省略可能な引数

デフォルト値はFALSE

Arguments: 各引数の詳しい説明

Details: 関数の動作の詳しい説明

Values: 戻り値の説明

Reference: 方法に関する文献

See Also: 関連するコマンド

Examples: 実行例

18

plotのオプション

- main="title" グラフのタイトル
 - xlab(ylab)="label" x軸(y軸)のラベル
 - log="xy" 対数軸の指定
 - xlim(ylim)=c(0,100) x軸(y軸)の値の範囲の設定
 - type="l" "p"(点), "l"(線), "b"(両方), "n"(枠だけ) など
 - lty=1 プロットする線の種類
 - pch=1 プロットする点の種類(文字)
 - col=2 プロットする点および線の色
 - cex=0.8 プロットする文字の大きさ
- ベクトルとして指定することにより、
点ごとに色や種類を変更することも
できる。例) pch=c(1,2,1,3,2)

.	▲	+	×	◇	▽	▣	✳	◆	⊕	
pch	1	2	3	4	5	6	7	8	9	10
col	1	2	3	4	5	6	7	8	9	10
cex	0.25	0.5	0.75	1	1.25	1.5	1.75	2	2.25	2.5

- ◆ plotのオプションは、help(plot)だけでは調べられない。一部のオプションは help(plot.default), plot(par)を参照する必要がある。
- ◆ 既存のグラフ上に線を重ねる場合はlines、点を重ねる場合はpointsを使う。
- ◆ 凡例をつけたいときはlegend関数を使う。

19

行列の作成

一つのベクトルを行列の形に並べる

```
> v <- 1:9                                9つの要素を持つベクトル
> m1 <- matrix(v, 3, 3)                 これを3行3列の行列として定義
> m1
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

複数のベクトルを行または列として束ねる

```
> a <- c(1,3,5)                            3つの要素を持つベクトル3つ
> b <- c(2,4,9)
> c <- c(3,5,7)
> m2 <- cbind(a,b,c)                     これらを列(縦)に並べて行列とする
> m2
   a b c
[1,] 1 2 3
[2,] 3 4 5
[3,] 5 9 7
```

20

行列の要素の取り出し

```

> m1
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> m1[2,3]          2行3列目の要素
[1] 8
> m1[2,]
[1] 2 5 8          2行目の要素すべてをベクトルとして取り出す
> m1[,3]
[1] 7 8 9          3列目の要素すべてをベクトルとして取り出す
> m1[1:2,2:3]
      [,1] [,2]
[1,]    4    7    1,2行目と2,3列目の要素からなる
[2,]    5    8    部分行列を取り出す
> dim(m1)          行列の大きさ
[1] 3 3

```

21

行列の演算

```

> m1+m2
m1 =           a   b   c          要素ごとの足し算
      [1,] 2   6  10
      [2,] 5   9  13
      [3,] 8  15  16
> m1*m2
m2 =           a   b   c          要素ごとのかけ算
      [1,] 1   8  21
      [2,] 6  20  40
      [3,] 15 54  63
> m1 %*% m2
           a   b   c          行列積
      [1,] 48  81  72
      [2,] 57  96  87
      [3,] 66 111 102
> t(m1)
           [,1] [,2] [,3]          転置行列(行と列の入れ換え)
      [1,]    1    2    3
      [2,]    4    5    6
      [3,]    7    8    9

```

22

データフレーム

- Rの統計解析でもっとも基本的となるデータ。
- 行列のような表形式のデータだが、各列が一般に異なる型のデータを持ちうる（行列はすべてが同じ型のデータ）。
- 多くの場合、各行が個体を、各列が個体が持つ属性を表す。
- 通常、タブ区切りテキストやエクセルファイル等から読み込んで処理する。

lung_cancer.txt
肺がん患者の遺伝子発現プロファイルデータ
(GEO: GDS3257)の抜粋

tissue: 腫瘍or正常
smoking: 喫煙歴
stage: 癌のステージ
gender: 性別
gene1 – gene6:
遺伝子発現データ

ID_REF	tissue	smoking	stage	gender	gene1	gene2	gene3
GSM254629	tumor	never	stage I	female	7.4191	5.9318	5.67496
GSM254648	tumor	never	stage I	female	7.5627	6.93398	5.76701
GSM254694	tumor	never	stage I	female	7.54599	7.53287	5.84134
GSM254701	tumor	never	stage I	female	8.31452	7.88291	5.44759
GSM254728	tumor	never	stage I	female	7.19835	6.58398	4.79089
GSM254726	tumor	never	stage I	male	11.9811	8.45595	5.7083
GSM254639	tumor	never	stage II	female	7.41762	7.75681	4.74084
GSM254652	tumor	never	stage II	female	7.62703	7.7446	4.69937
GSM254700	tumor	never	stage II	female	7.40064	10.1866	4.92612
GSM254625	tumor	never	stage II	male	11.9	8.89865	6.69416
GSM254636	tumor	never	stage III	female	7.09852	6.39122	4.54743
GSM254659	tumor	never	stage III	female	7.39159	7.06924	5.12113
GSM254680	tumor	never	stage III	female	7.32462	7.24284	4.90953
GSM254686	tumor	never	stage III	female	7.65883	7.93856	5.09535
GSM254718	tumor	never	stage III	female	7.67937	7.03277	5.64789
GSM254674	tumor	never	stage IV	male	11.0711	6.3368	5.48673
GSM254668	tumor	former	stage I	male	10.9011	6.52462	25.19564

データフレームの読み込み(1)

```
read.table(file, options... )
read.delim(file, options... )
```

タブ区切りテキストファイル lung_cancer.txt からデータの読み込み

```
> cancer <- read.delim("lung_cancer.txt")
```

```
> head(cancer)          データの先頭数行のみの表示
```

	ID_REF	tissue	smoking	stage	gender	gene1	gene2	...
1	GSM254629	tumor	never	stage I	female	7.41910	5.93180	...
2	GSM254648	tumor	never	stage I	female	7.56270	6.93398	...
3	GSM254694	tumor	never	stage I	female	7.54599	7.53287	...
4	GSM254701	tumor	never	stage I	female	8.31452	7.88291	...
5	GSM254728	tumor	never	stage I	female	7.19835	6.58398	...
6	GSM254726	tumor	never	stage I	male	11.98110	8.45595	...

```
> edit(cancer)          エディタ上でデータ表示
```

データフレームの読み込み(2)

	ヘッダなし	ヘッダ行あり	ヘッダ行、ヘッダ列あり																																	
ファイルの形式	<table border="1"> <tr><td>198</td><td>119</td></tr> <tr><td>192</td><td>83</td></tr> <tr><td>191</td><td>110</td></tr> <tr><td>191</td><td>91</td></tr> </table>	198	119	192	83	191	110	191	91	<table border="1"> <thead> <tr><th>Height</th><th>Weight</th></tr> </thead> <tbody> <tr><td>198</td><td>119</td></tr> <tr><td>192</td><td>83</td></tr> <tr><td>191</td><td>110</td></tr> <tr><td>191</td><td>91</td></tr> </tbody> </table>	Height	Weight	198	119	192	83	191	110	191	91	<table border="1"> <thead> <tr><th>ID</th><th>Height</th><th>Weight</th></tr> </thead> <tbody> <tr><td>1</td><td>198</td><td>119</td></tr> <tr><td>2</td><td>192</td><td>83</td></tr> <tr><td>3</td><td>191</td><td>110</td></tr> <tr><td>4</td><td>191</td><td>91</td></tr> </tbody> </table>	ID	Height	Weight	1	198	119	2	192	83	3	191	110	4	191	91
198	119																																			
192	83																																			
191	110																																			
191	91																																			
Height	Weight																																			
198	119																																			
192	83																																			
191	110																																			
191	91																																			
ID	Height	Weight																																		
1	198	119																																		
2	192	83																																		
3	191	110																																		
4	191	91																																		
read.table オプション	header=FALSE	header=TRUE	header=TRUE, row.names=1 (指定なし)																																	

read.table が読み込みのための基本的な関数だが、デフォルト値の異なる読み込み関数がいくつか用意されている

read.table(file)	デフォルトでセパレータは空白文字、header=FALSE
read.delim(file)	デフォルトでセパレータがタブ、header=TRUE
read.csv(file)	デフォルトでセパレータがコンマ、header=TRUE

25

データフレームの情報表示

str(data)	data のデータ構造の表示
summary(data)	data の内容サマリの表示

> **str(cancer)** 各列のデータ型と内容の一部を表示

```
'data.frame': 107 obs. of 11 variables:
 $ ID_REF : Factor w/ 107 levels "GSM254625","GSM254626",...: 5 24 70 77 104 102 15 28
 $ tissue : Factor w/ 2 levels "normal","tumor": 2 2 2 2 2 2 2 2 2 ...
 $ smoking: Factor w/ 3 levels "current ","former ",...: 3 3 3 3 3 3 3 3 3 ...
 $ stage  : Factor w/ 4 levels "stage I","stage II",...: 1 1 1 1 1 1 2 2 2 ...
 $ gender : Factor w/ 2 levels "female","male": 1 1 1 1 1 2 1 1 1 2 ...
 $ gene1  : num  7.42 7.56 7.55 8.31 7.2 ...
 $ gene2  : num  5.93 6.93 7.53 7.88 6.58 ...
 ....
```

> **summary(cancer)** 各列のデータの分布の要約を表示

ID_REF	tissue	smoking	stage	gender	gene1	
GSM254625:	1	normal:49	current :40	stage I :45	female:38	Min. : 6.729
GSM254626:	1	tumor :58	former :36	stage II :35	male :69	1st Qu.: 7.543
GSM254627:	1		never :31	stage III:21		Median :11.224
GSM254628:	1			stage IV : 6		Mean :10.074
GSM254629:	1					3rd Qu.:11.875
GSM254630:	1					Max. :12.659
(Other)	:101					26

因子 factor

- gender (male, female)のように、限られた数のカテゴリーで表現されるもの。
- 因子の取り得る値を水準levelという。
例) genderは2つ、smokingは3つの水準を持つ
- 文字列として表示されるが、内部的には整数値で保持されている。

27

データフレームの要素の取り出し

```
> cancer[5,]          5行目の人のデータの取り出し
   ID_REF tissue smoking    stage gender   gene1   gene2   gene3   gene4
5 GSM254728 tumor never stage I female 7.19835 6.58398 4.79089 7.26575

> cancer[,6]          6列目(gene1)の取り出し
[1] 7.41910 7.56270 7.54599 8.31452 7.19835 11.98110 7.41762 7.62703
[9] 7.40064 11.90000 7.09852 7.39159 7.32462 7.65883 7.67937 11.07110
[17] 10.90110 9.30280 11.42620 12.10840 11.00710 11.43570 9.58576 11.37820
[25] 11.46580 11.98080 7.17584 11.62510 11.22410 7.45557 7.70254 11.65150
[33] 11.01420 12.03040 7.00001 7.53943 10.81220 11.47220 12.05180 11.37000
...
> cancer$gene1        gene1 データの取り出し
(名前によるアクセス。cancer[,6]と同じ)

> cancer[7:10,2:7]     7-10行、2-7列の要素からなる部分データの取り出し
   tissue smoking    stage gender   gene1   gene2
7 tumor never stage II female 7.41762 7.75681
8 tumor never stage II female 7.62703 7.74460
9 tumor never stage II female 7.40064 10.18660
10 tumor never stage II male 11.90000 8.89865
```

28

データフレームの操作

条件式による部分データの抽出

subset(データフレーム, 条件式)

```
# 性別が女性でgene1の発現が8以上
> subset(cancer, gender=="female" & gene1 > 8)
   ID_REF tissue smoking stage gender   gene1   gene2   gene3   gene4
4  GSM254701 tumor  never    stage I female 8.31452 7.88291 5.44759 5.99769
56 GSM254687 tumor current stage III female 10.15150 7.64551 6.61338 7.24318
...
# 喫煙歴がある人
> subset(cancer, smoking %in% c('former','current'))
...
# その数(行数をカウント)
> nrow(subset(cancer, smoking %in% c('former','current')))
[1] 76
```

A %in% B ベクトルAの各要素について、ベクトルBの要素のいずれかと一致すればTRUE, そうでなければFALSEを返す。

```
> c("A","B","C","B","A") %in% c("B","C")
[1] FALSE  TRUE  TRUE  TRUE FALSE
```

29

データフレームのファイルへの書き出し

write.table(data, file="", options...)

```
# サブセットの切り出し
> cancer.subset <- subset(cancer, gene1 > 8)

# タブ区切りテキストとして画面に表示
> write.table(cancer.subset, sep="\t")
# その形式でファイルに保存
> write.table(cancer.subset, sep="\t",
               file="cancer.subset.txt")

# そのまま読み込んで表示してみる
> read.table("cancer.subset.txt")
```

30

Rによるデータ解析

plotによる散布図の作成

各カラム対総当たりの散布図を作成する

```
> plot(cancer)
```

`pairs(cancer)`としても同じ

gene3とgene4の散布図を作成する

```
> plot(cancer$gene3, cancer$gene4)
```

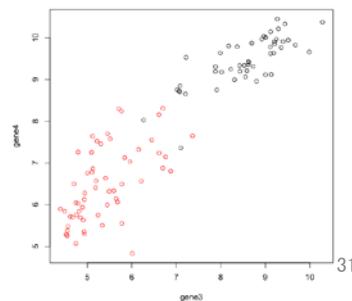
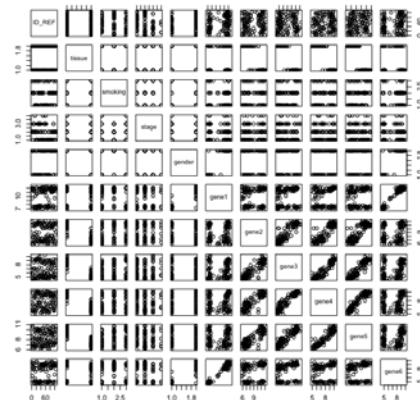
以下もほぼ同じプロットを生成する

```
> plot(gene4 ~ gene3, cancer)
```

tissue (cancer/normal) によって点の色づけする

```
> plot(gene4 ~ gene3, cancer,
       col=tissue)
```

`cancer$tissue` は normal=1, tumor=2 として定義されており、各点が 1=black, 2=red で色づけされる。



Rによるデータ解析

回帰直線の追加

gene3とgene4の散布図(前掲)

```
> plot(gene4 ~ gene3, cancer)
```

線型モデルを用いた直線への当てはめ(回帰直線の決定)

```
> lm(gene4 ~ gene3, cancer)
```

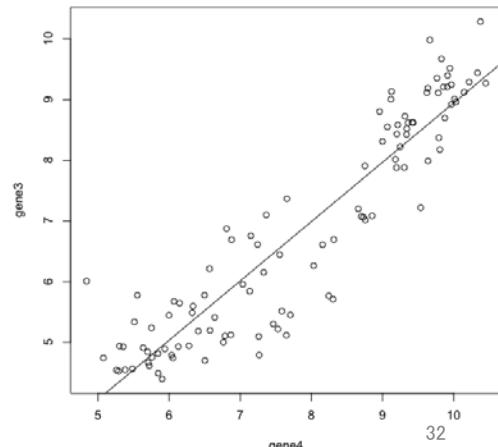
... (結果は画面に出力される) ...

結果を変数 `result.lm` に代入

```
> result.lm <- lm(gene4 ~ gene3,
                    cancer)
```

この結果を使って回帰直線をプロットに追加

```
> abline(result.lm)
```



モデル式と線型モデル

- `lm` の第1引数で指定される式は、データを当てはめるモデル式を簡潔に表現したものとなっている
 例) X_1, X_2 が説明変数、 Y が目的変数として、

$$Y = a * X_1 + b * X_2 + c + \varepsilon \quad (\text{残差})$$
 という式に当てはめる(係数 a, b, c の最適な値を決める)ことを

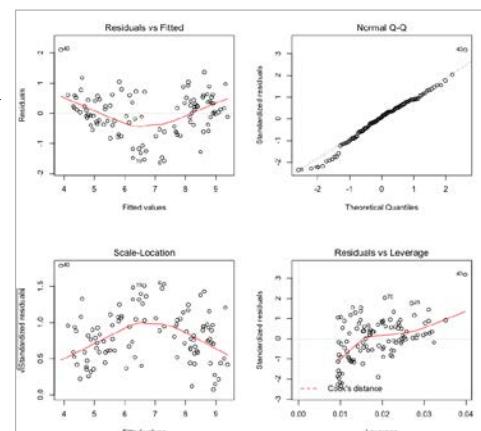
$$Y \sim X_1 + X_2$$
 と記述する。
- モデル式が係数の一次式で表されるモデルを線型モデルといい、最小二乗法で当てはめが行われる。`lm` 関数では、係数の決定とともに、係数が有意に0でないといえるかの検定も行われる。
- 線型モデルへの当てはめは、回帰分析だけでなく、カテゴリ変数を説明変数とした分散分析においても用いられる。

33

Rによるデータ解析 回帰分析結果の詳細表示

```
> par(mfrow=c(2,2)) # 4つのプロットを 2x2 の領域に同時に表示
> plot(result.lm) # 解析結果からプロットの作成
```

プロットから誤差(残差)の分布が適切か(正規分布に従っているか)を確認



```
> summary(result.lm) # 解析結果の詳細表示
Call:
```

```
lm(formula = gene4 ~ gene3, data = cancer)
```

```
Residuals:
    Min      1Q  Median      3Q      Max 
-2.27983 -0.41889 -0.01643  0.45094  1.44834
```

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.86604	0.24801	7.524	1.9e-11 ***
gene3	0.87403	0.03514	24.870	< 2e-16 ***

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.6395 on 105 degrees of freedom
Multiple R-squared:  0.8549, Adjusted R-squared:  0.8535
F-statistic: 618.5 on 1 and 105 DF,  p-value: < 2.2e-16
```

係数が0でない値を持つといえるかを統計的に検定

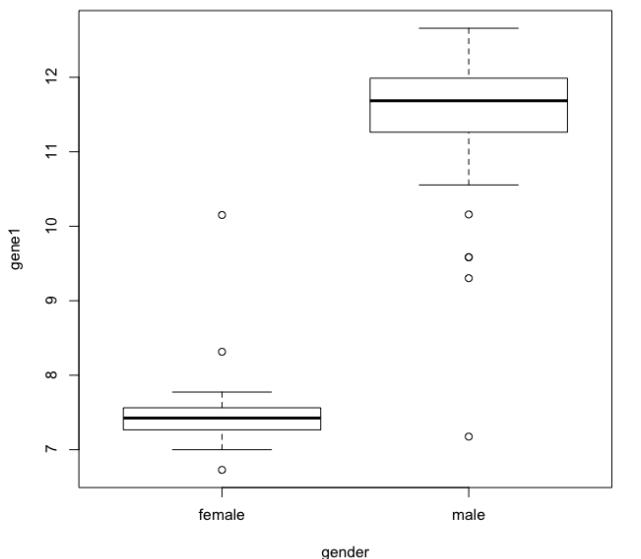
解析が終わったらプロットウィンドウを閉じるか `par(mfrow=c(1,1))` して、プロットの設定を元に戻しておく

Rによるデータ解析

boxplotの作成

性別ごとのgene1の発現量をboxplotで比較する

```
> plot(gene1 ~ gender, cancer)
```



boxplot(gene1 ~ gender, cancer)
としてもほぼ同じ

35

Rによるデータ解析

t検定

2組の標本の平均値に差があるかどうかを検定する

男女でgene1の発現量に違いがあるかどうかをt検定で検定する

男性のgene1発現量

```
> gene1.m <- subset(cancer, gender=="male")$gene1
```

女性のgene1発現量

```
> gene1.f <- subset(cancer, gender=="female")$gene1
```

t検定の実行 (等分散性を仮定しないWelchの検定)

```
> t.test(gene1.m, gene1.f)
```

別法(データフレームから直接データを取り出して検定を実行する)

```
> t.test(gene1 ~ gender, cancer)
```

36

Rによるデータ解析

分散分析

3組以上の標本間で平均値に差があるかどうかを検定する

癌のステージによってgene1の発現に違いがあるかどうかを分散分析で検定する

```
> aov(gene1 ~ stage, cancer)
Call:
  aov(formula = gene1 ~ stage, data = cancer)

Terms:
  stage Residuals
Sum of Squares   47.0675 399.6562
Deg. of Freedom      3        103

Residual standard error: 1.969812
Estimated effects may be unbalanced
```

結果を変数 result.aov に格納してsummaryで出力

```
> result.aov <- aov(gene1 ~ stage, cancer)
> summary(result.aov)
  Df Sum Sq Mean Sq F value    Pr(>F)
stage     3   47.1   15.69   4.043 0.00921 ** ← p-value
Residuals 103 399.7     3.88
---
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

この結果については練習問題3も参照のこと³⁷

解析結果の保存

```
# 指定したオブジェクトを "cancer.Robj" という
# ファイルに保存
> save(cancer, result.lm, result.aov,
       file="cancer.Robj")
```

```
# 保存したオブジェクトをファイルからロードする
> load("cancer.Robj")
```

```
# 現在のワークスペース中のオブジェクトをすべて
# デフォルトの保存ファイル ("RData") に保存。
> save.image()
```

- 明示的にsave.image()コマンドを実行しなくても、終了時にワークスペースを保存するか聞かれるので、そこで保存を選択することにより保存できる。
- ここで保存したデータは、現在の作業ディレクトリが起動時のディレクトリと同じであれば、次回起動時に自動的にロードされるが、ディレクトリを変更した場合はload(".RData")で明示的にロードする必要がある。

リスト

- ベクトルは同一の型のデータを要素とするが、リストは任意の一般に異なる型のデータを要素として含むことができる。

```
> list(1, "a")
```

- リストは、ベクトルや行列のほか、リストをも要素として含むことができるため、複雑なデータを表現できる。

```
> list(c(1,2), c("a","b"), list(1, "a"))
```

- リストの各要素は[[]]で参照できるほか、名前をつけて参照することもできる。

```
> x <- list(first=1, second="b")
```

> x[[1]] # 1番目の要素

[1] 1

> x\$second # second という名前の要素

[1] "b"

- データフレームのほか、各種の統計処理の結果などもリストとして表現されている。

例) データフレームであるcancerをリストとして見る

```
> as.list(cancer)
```

39

オブジェクトの属性とクラス

- 統計解析の結果などは、一般に決まった型と名前の要素を持ち、一定のクラス名をアサインされたリスト(オブジェクト)として返される。

例) aov 関数の結果を格納した変数 result.aov

```
> class(result.aov) クラス名の表示
```

[1] "aov" "lm"

```
> names(result.aov) 要素名の一覧の表示
```

[1] "coefficients" "residuals" "effects"

... (以下略) ...

```
> str(result.aov) データ構造の階層的な表示
```

List of 13

\$ coefficients : Named num [1:4] 10.334 0.195 -1.584 -0.224

... - attr(*, "names")= chr [1:4] "(Intercept)" "stagestage II" ..

... (以下略) ...

- このオブジェクトの詳細は、help(lm)のValueセクションで確認できる

40

総称関数 generic function

- plot, print, summaryなど多くの関数は、引数となるオブジェクトのクラスによって、異なる関数が呼び出されるようになっている。これらを総称関数という
 - 例) plot は、引数のクラスによって plot.data.frame, plot.formula, plot.lm などが呼び出される。デフォルトでは、plot.default が呼び出される。
- これにより、Rでは典型的には以下のようない定型的な処理の流れによって解析が進められる。

```
> result <- some.function(data, opt=value, ...)
> summary(result)    解析結果の要約の表示
> plot(result)       解析結果に基づくプロットの作成
```

参考) コンソールで単に x と入力したときは、print(x) が実行されている。
printは総称関数であり、xのクラスによって表示される内容は違っている。

41

関数の作成(1)

- function 関数によって、新たな関数を作成できる

```
function (引数リスト) { 関数本体 }
```

例) 引数の2乗を計算して返す関数を square として定義

```
> square <- function(x) {
  sq <- x^2
  return(sq)
}

> square(1:5)      引数1:5を与えて関数の呼び出し
[1] 1 4 9 16 25
```

引数: 呼び出し時に指定した値がxに代入され、関数本体が実行される

42

関数の作成(2)

- 関数は、通常はファイル上に作成し、読み込んで使う。ファイル上に作成したRコマンドを読み込むにはsource("ファイル名")とする。

ファイル plotAll.R (plotAll: 指定した2つの部分データフレーム間での総当たりプロットを作成する関数)

```
plotAll <- function(x, y=x) {
  par(mfrow=c(ncol(x), ncol(y)))
  for (i in 1:ncol(x)) {
    x.name=names(x)[i]
    for (j in 1:ncol(y)) {
      y.name=names(y)[j]
      if (x.name == y.name) {
        plot(x[,i], main=x.name)
      } else {
        plot(x[,i], y[,j],
              xlab=x.name, ylab=y.name)
      }
    }
  }
}
```

繰り返し

for (変数 in ベクトル) { 式 }

「変数」に「ベクトル」の要素を順次代入して、「式」を繰り返し実行する。

条件分岐

if (条件式) { 式1 } else { 式2 }

「条件式」がTRUEなら、「式1」を実行し、FALSEなら「式2」を実行する。

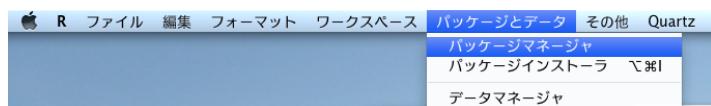
プログラムの読み込みと実行

```
> source("plotAll.R")
> plotAll(cancer[,2:5], cancer[,6:11])
```

43

パッケージの利用

- 様々なパッケージをインストール、ロードすることによりRの機能を拡張できる。
- バイオインフォマティクス関連では、Bioconductorプロジェクトによって、膨大な種類のパッケージが提供されている。



パッケージインストーラ
パッケージのインストール/更新



パッケージマネージャ
パッケージのロード



コマンドラインからの実行

パッケージのインストール
`install.packages(パッケージ名)`

パッケージの更新
`update.packages(パッケージ名)`

パッケージのロード
`library(パッケージ名)`

44

練習問題

cancer データを使って、以下の問題を考えよう。

1. 男性で喫煙歴がある人のデータを抜き出せ。何人いるか。それらの人のgene1の発現データを取り出し、その平均値を計算せよ。
2. gene1 と gene2 の散布図を作成し、genderによって点を色分けせよ。また、回帰直線を引け。この回帰直線の係数は有意であると言えるか。
3. a) gene1 の発現を、gender と stage の 2 因子を使ったモデル式 ($\text{gene1} \sim \text{gender} + \text{stage}$) を用いて分散分析せよ。この場合、まず gender の効果が評価され、次に gender の効果を除いた stage の効果が評価される。このときの stage の効果は有意といえるか。
b) `plot(gender ~ stage, cancer)` によってプロット(mosaicplot)を作成せよ。これにより、データセット中の stage の水準ごとに gender の比率を確認できる。これから stageだけを使ったときの結果と a) の結果の違いが説明できるか。
4. cancer データから各患者の gene1 から gene6 の発現量を抜き出した部分データフレームを作成し、変数 expression に代入せよ。それを用いて `matplot(expression)` を実行せよ。このグラフは各患者を横軸に配置し、各患者における gene1 ~ gene6 の発現量を縦軸に沿ってプロットしたものである。このグラフから何か読み取れるか。(こうした多変量のデータは、多変量解析によって、より明確に特徴を分析できるようになる)

45

RNA-seqの解析パイプライン

RNA-seq Analysis Pipelines

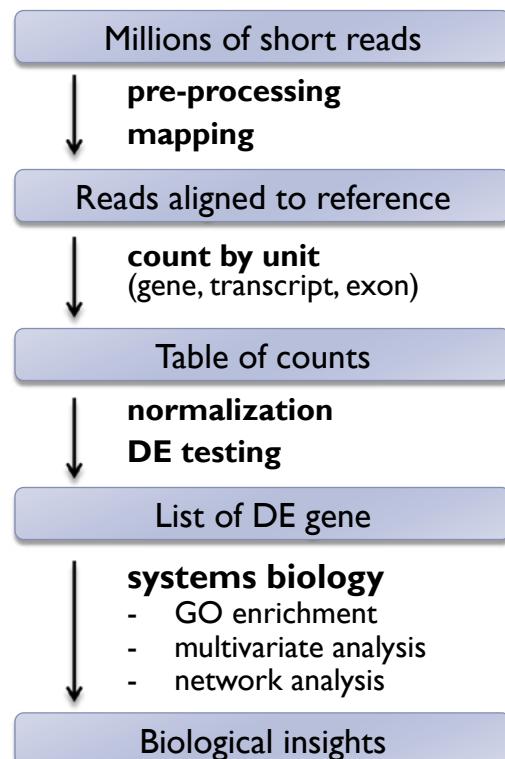
重信秀治 Shuji Shigenobu

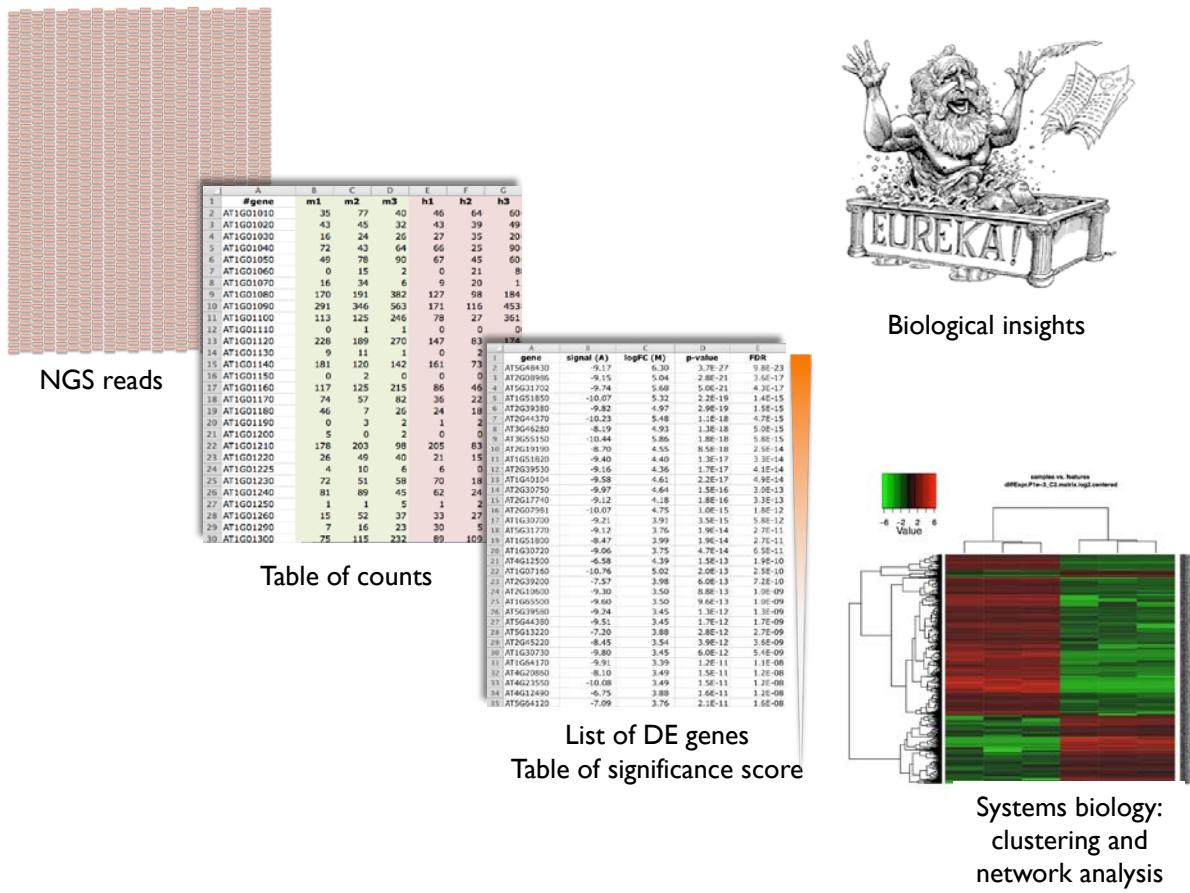
基生研 NIBB

<shige@nibb.ac.jp>

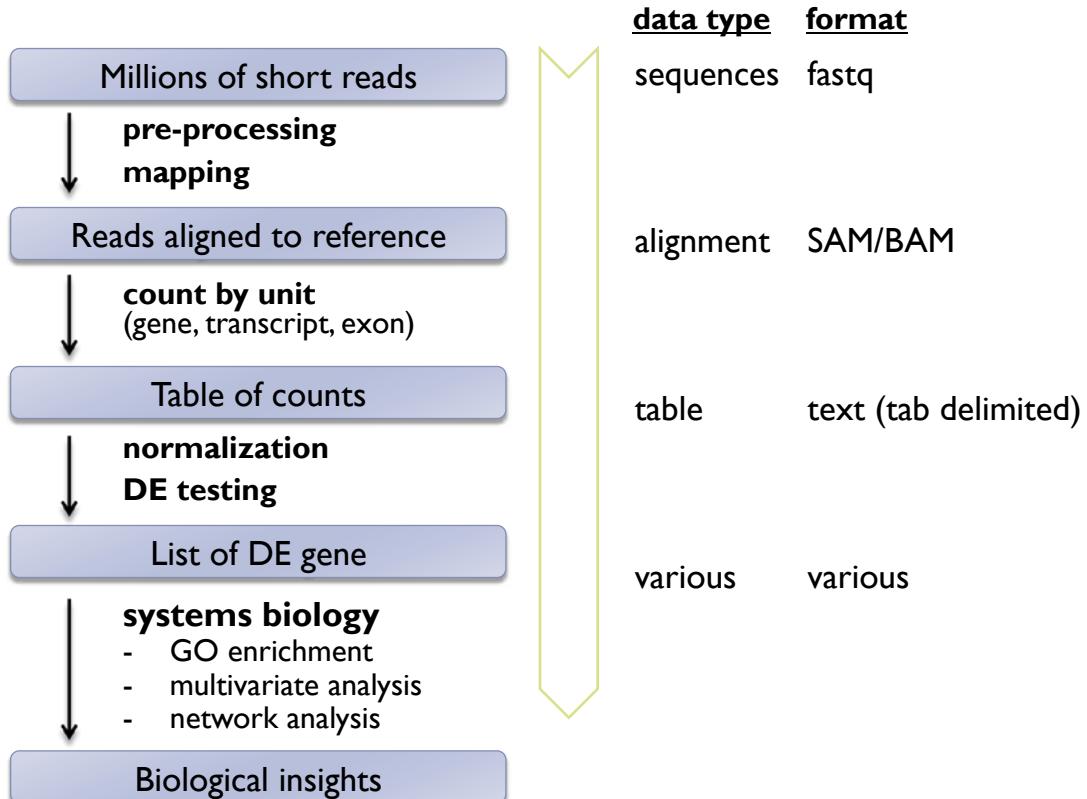
RNA-seq analysis pipeline for DE

Differential Expression analysis





RNA-seq analysis pipeline for DE

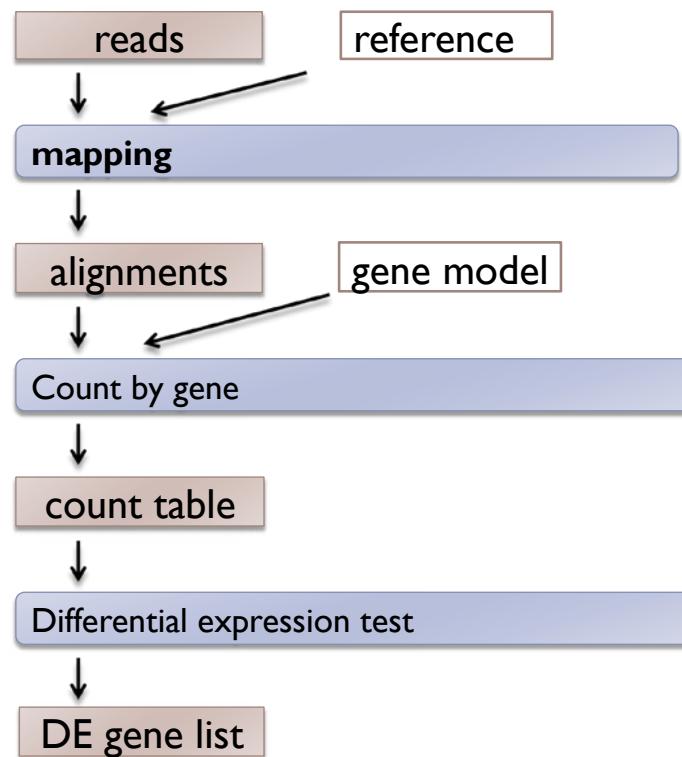


Two Basic Pipelines

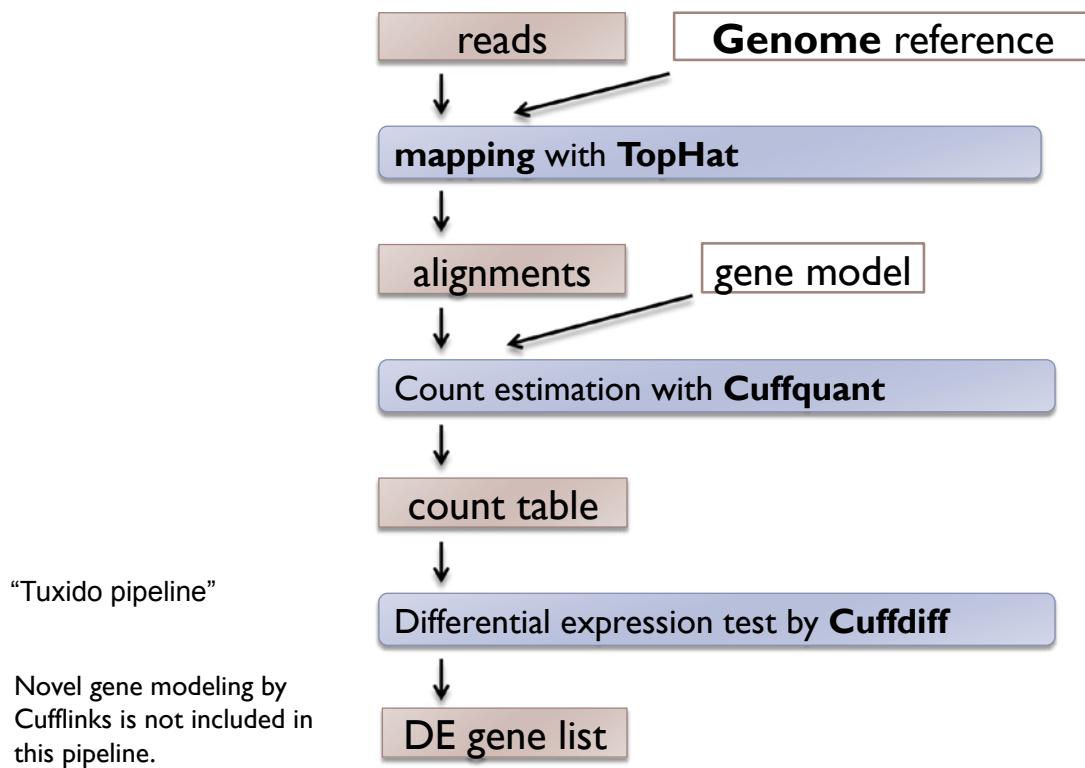
▶ Choice of reference

- ▶ **Genome** – standard for genome-known species
- ▶ **Transcript** – the only way for genome-unknown species
 - can be used for genome-known species

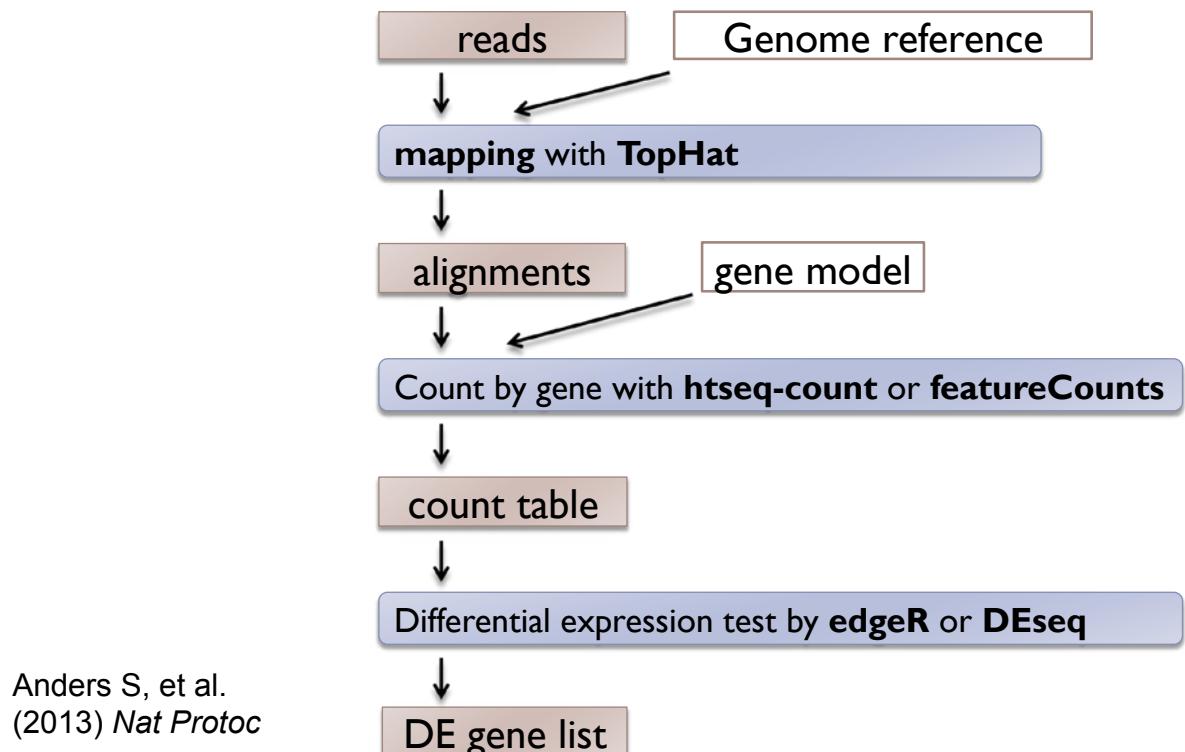
Common workflow



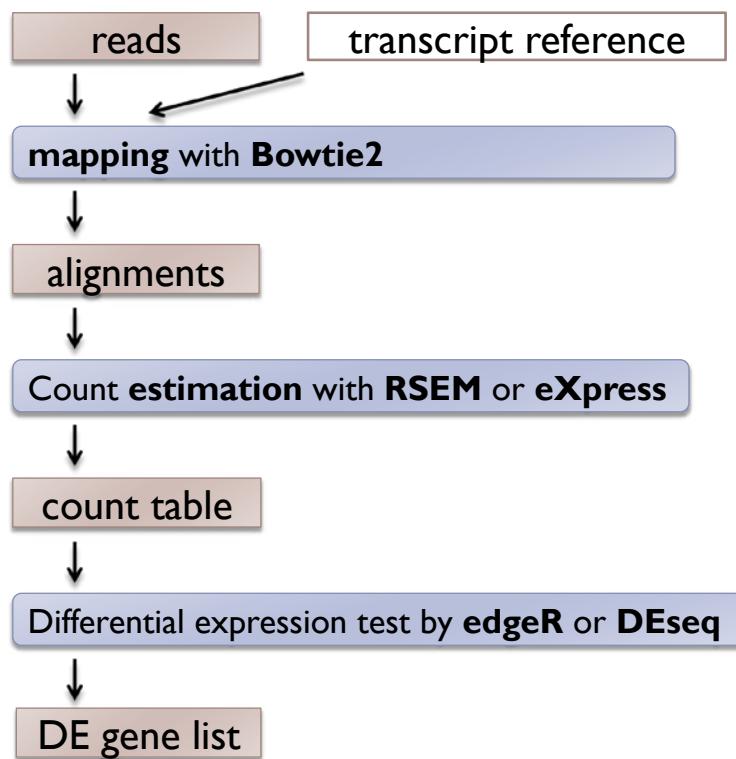
A Pipeline: Genome-based (1)



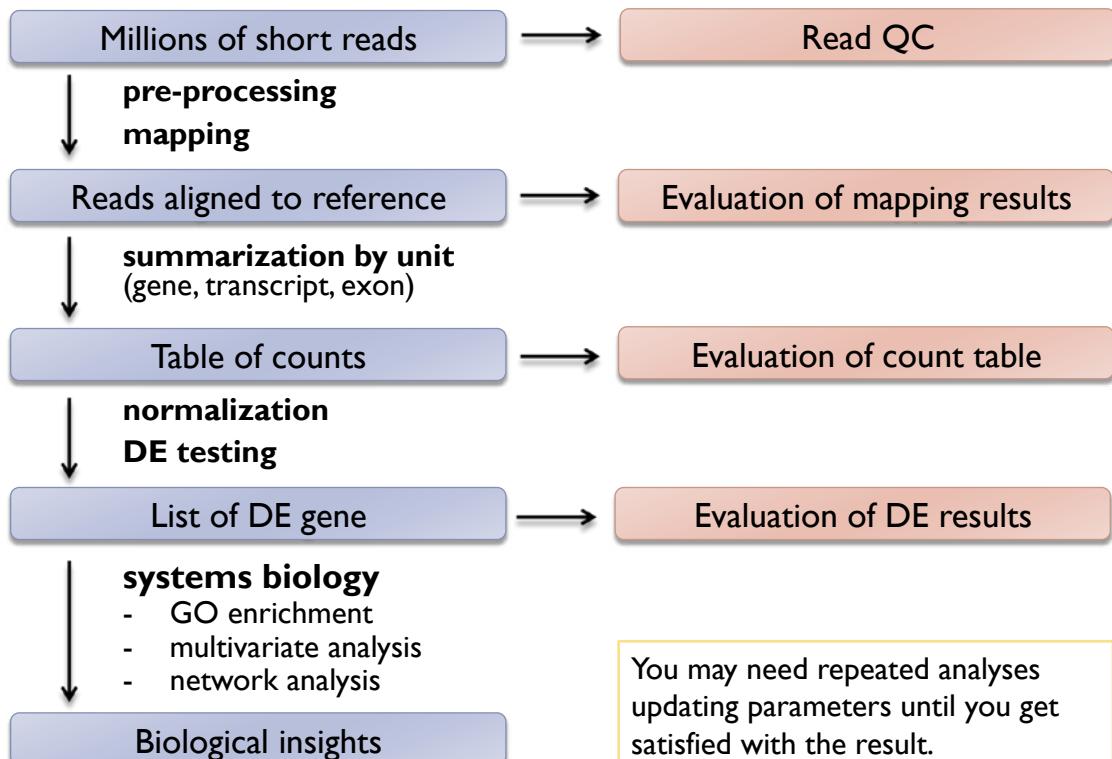
A Pipeline: Genome-based (2)



A Pipeline: Transcript-based



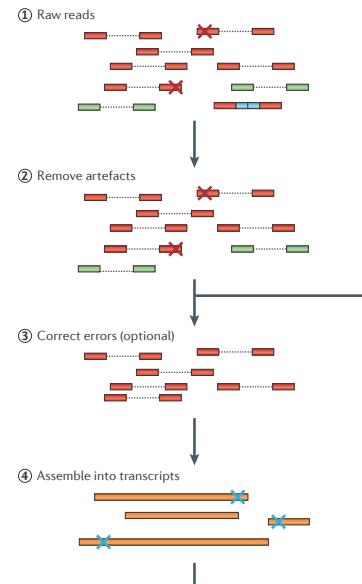
Check Points



Read QC and Pre-processing

- ▶ **Read QC**
 - ▶ Tools: FastQC etc.

- ▶ **Pre-processing**
 - ▶ Filter or trim by base quality
 - ▶ Remove artifacts
 - ▶ adaptors
 - ▶ low complexity reads
 - ▶ PCR duplications (optional)
 - ▶ Remove rRNA and other contaminations (optional)
 - ▶ Sequence error correction (optional)
 - ▶ Tools: cutadapt, trimomatic

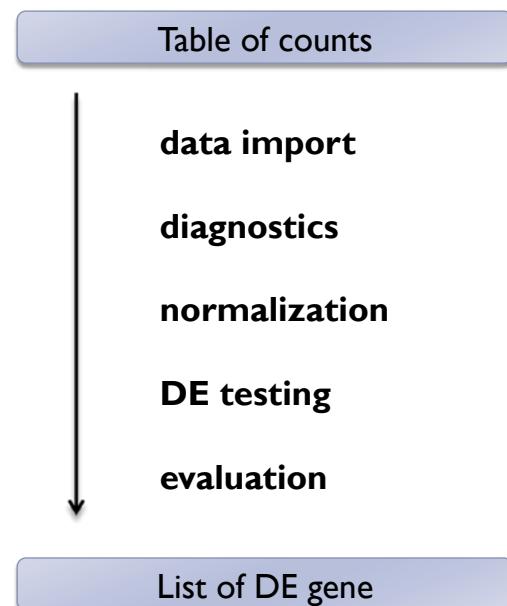
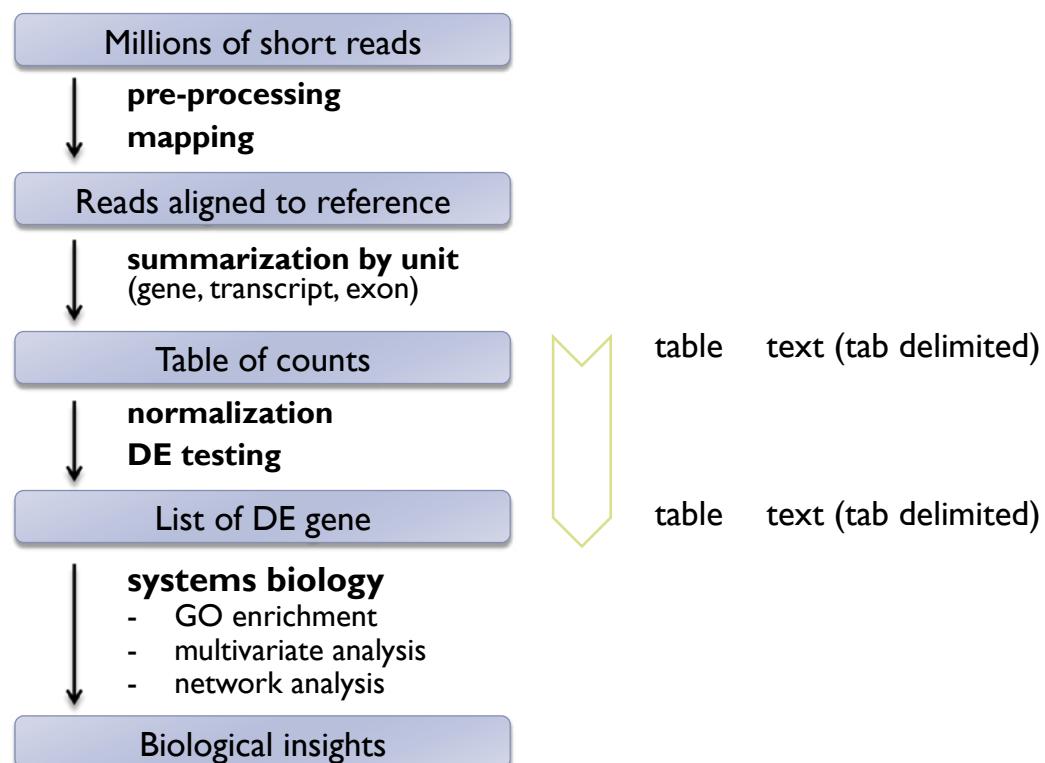


Martin et al (2011) *Nat Rev Genet*

Evaluation of mapping results

- ▶ **Evaluation of SAM/BAM file**
- ▶ **Check statistics**
- ▶ **Visualization**

RNA-seq analysis pipeline for DE



Import count table / diagnostics

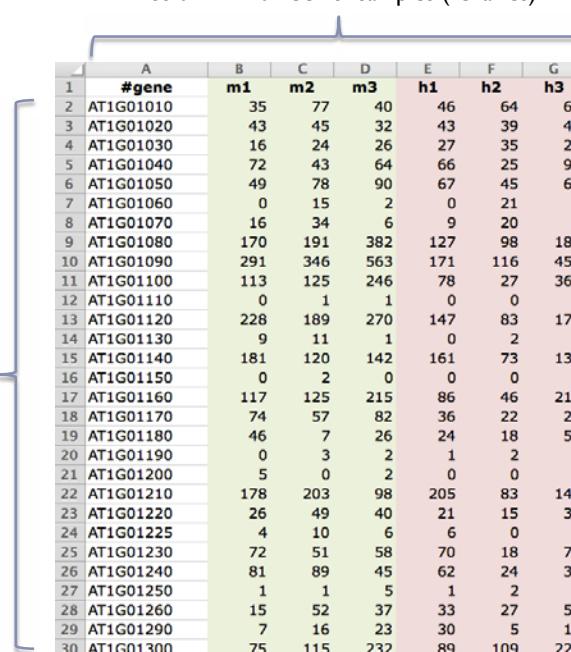
Look into the input data first.

- ▶ Quick view of the table (tools: R, MS Excel etc.)
 - ▶ Check: Format, data structure, data size etc.
- ▶ Scatter plot (tools: R, MS Excel etc.)

Input

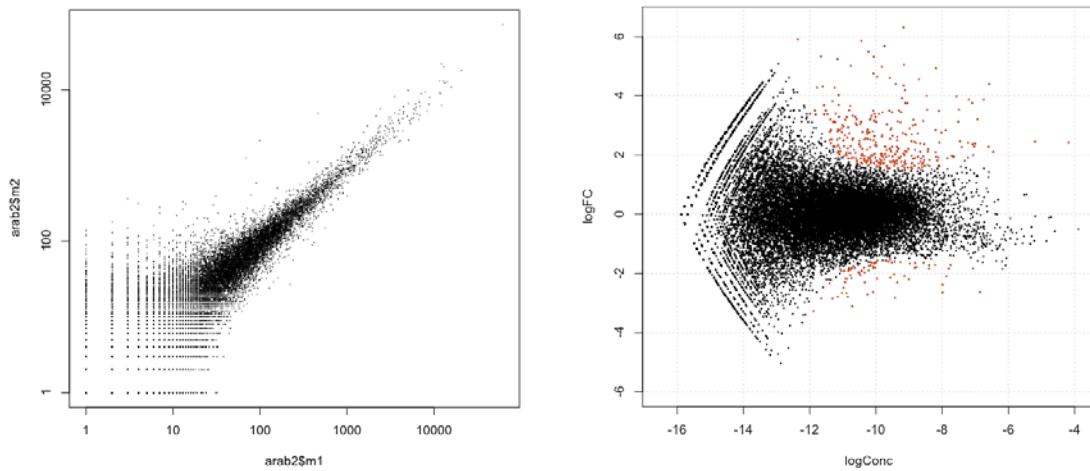
- ▶ Typical primary data = matrix of #genes x #samples

column x number of samples (libraries)

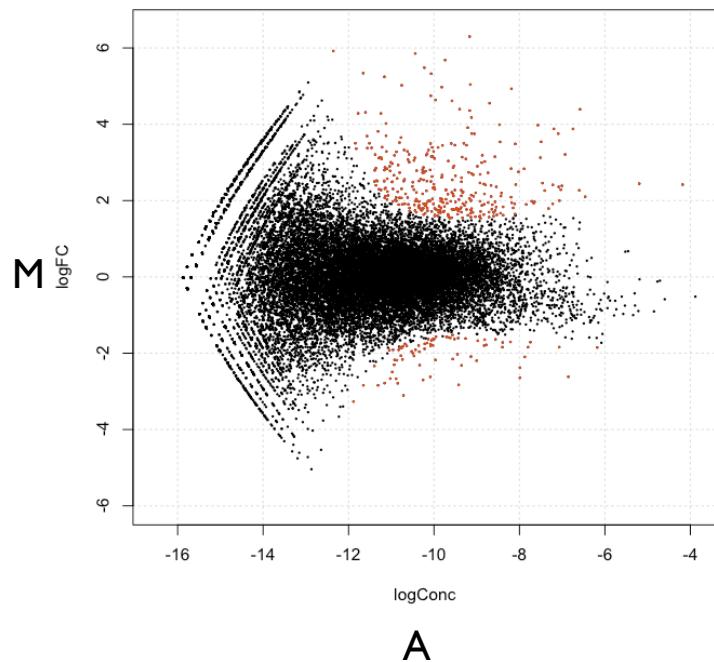


A	B	C	D	E	F	G
#gene	m1	m2	m3	h1	h2	h3
1 AT1G01010	35	77	40	46	64	60
2 AT1G01020	43	45	32	43	39	49
3 AT1G01030	16	24	26	27	35	20
5 AT1G01040	72	43	64	66	25	90
6 AT1G01050	49	78	90	67	45	60
7 AT1G01060	0	15	2	0	21	8
8 AT1G01070	16	34	6	9	20	1
9 AT1G01080	170	191	382	127	98	184
10 AT1G01090	291	346	563	171	116	453
11 AT1G01100	113	125	246	78	27	361
12 AT1G01110	0	1	1	0	0	0
13 AT1G01120	228	189	270	147	83	174
14 AT1G01130	9	11	1	0	2	9
15 AT1G01140	181	120	142	161	73	134
16 AT1G01150	0	2	0	0	0	0
17 AT1G01160	117	125	215	86	46	212
18 AT1G01170	74	57	82	36	22	29
19 AT1G01180	46	7	26	24	18	58
20 AT1G01190	0	3	2	1	2	2
21 AT1G01200	5	0	2	0	0	0
22 AT1G01210	178	203	98	205	83	143
23 AT1G01220	26	49	40	21	15	34
24 AT1G01225	4	10	6	6	0	3
25 AT1G01230	72	51	58	70	18	77
26 AT1G01240	81	89	45	62	24	33
27 AT1G01250	1	1	5	1	2	2
28 AT1G01260	15	52	37	33	27	54
29 AT1G01290	7	16	23	30	5	19
30 AT1G01300	75	115	232	89	109	224

Diagnostics: Scatter plot & MA plot



MA plot



M: log fold-change
A: log intensity average

$$M = \log_2(R/G) = \log_2(R) - \log_2(G)$$

$$A = \frac{1}{2}\log_2(RG) = \frac{1}{2}(\log_2(R) + \log_2(G))$$

R: expression level of sample 1
G: expression level of sample 2

Let's try: data import and quick check

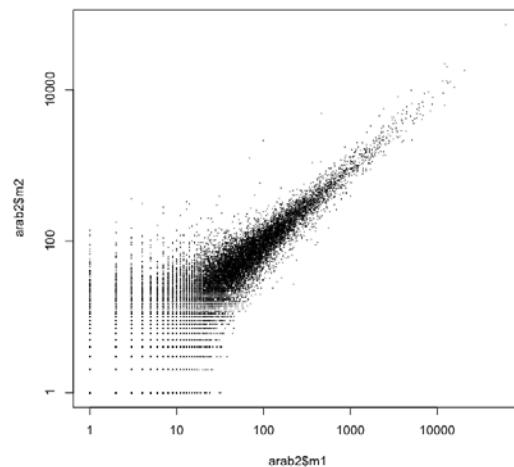
```
> dat <- read.delim("~/data/SS/arab2.txt", row.names=1)
> head(arab2)                                # look at the first several lines
                                                # for checking
  m1  m2  m3  h1  h2  h3
AT1G01010 35  77  40  46  64  60
AT1G01020 43  45  32  43  39  49
AT1G01030 16  24  26  27  35  20
AT1G01040 72  43  64  66  25  90
AT1G01050 49  78  90  67  45  60
AT1G01060  0  15   2   0  21   8

> dim(dat)                                    # get numbers of rows and columns
[1] 26221      6

> colSums(dat)                               # get column sums
  m1      m2      m3      h1      h2      h3
1902032 1934029 3259705 2129854 1295304 3526579
```

Let's try: Scatter plot

```
> plot(dat$m1 + 1, arab2$m2 + 1, log="xy")
```



Scatter plot: R

```
> pairs(dat, log="xy")
```

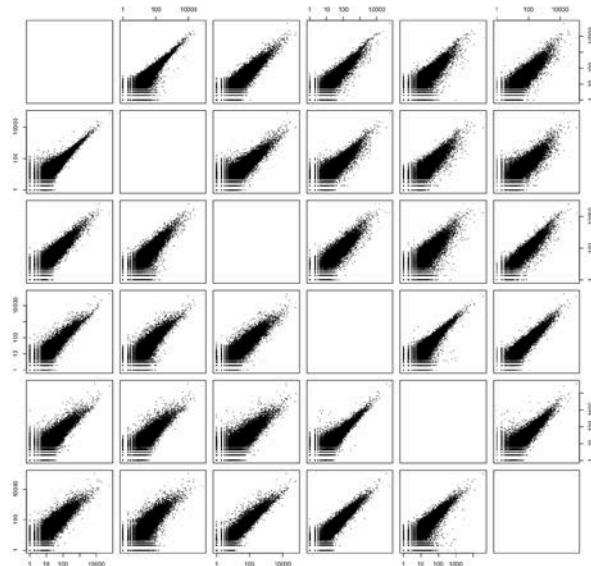


Table of counts

data import

diagnostics

normalization

DE testing

evaluation

List of DE gene

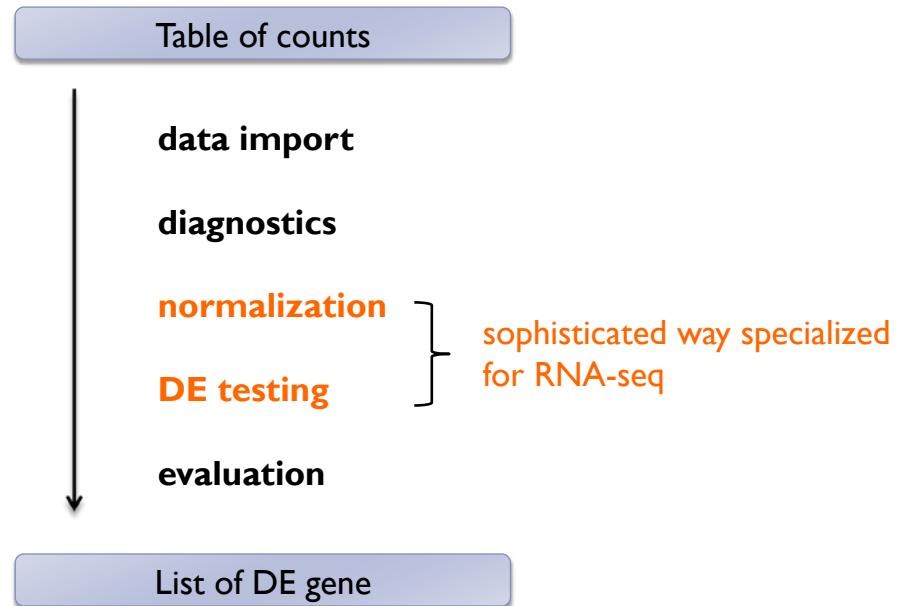
Normalization

What is normalization? Why it is required?

- ▶ Normalization means to adjust transcriptome data for effects which arise from variation in the technology rather than from biological differences between the RNA samples or between genes.
- ▶ Normalization is an essential step in the analysis of DE from RNA-seq data to make them really comparable.

Normalization: two types

- ▶ Between-libraries
 - ▶ Comparing expression (counts) of genes between libraries
- ▶ Within-library
 - ▶ Comparing expression (counts) of genes within a library (should be possible with NGS – in contrast to microarray)



DEG: RNA-seq specific issues

- ▶ RNA-seq count data is Non-Gaussian
- ▶ Normalization: composition effects
- ▶ N (biological replicates) is so small
- ▶ Multiple comparisons (多重検定の問題)

RNA-seq data is Non-Gaussian

- ▶ Microarray data
 - ▶ Log transform intensities
 - ▶ => Analyze as normally distributed random variables allowing parametric analysis
- ▶ RNA-seq data
 - ▶ Not normally distributed random variables
 - ▶ **Poisson distribution** for technical replicates
 - ▶ **Negative binomial distribution** for biological replicates.

RNA-seq issue: Normalization

- ▶ Simple normalization
 - ▶ RPM or RPKM works well, but not best
- ▶ Composition effects
 - ▶ A small number of highly expressed genes can consume a significant amount of the total sequence.
- ▶ Strategies
 - ▶ estimate scaling factors from data and statistical models
 - ▶ quantile normalization
 - ▶ ...

Implementation examples: edgeR and Cuffdiff

edgeR

- ▶ **Model:** An over dispersed Poisson model, **negative binomial (NB) model** is used
- ▶ **Normalization:** **TMM method** (trimmed mean of M values; Robinson et al., 2010), **RLE** (Anders et al., 2010) and **upperquantile** (Bullard et al., 2010)

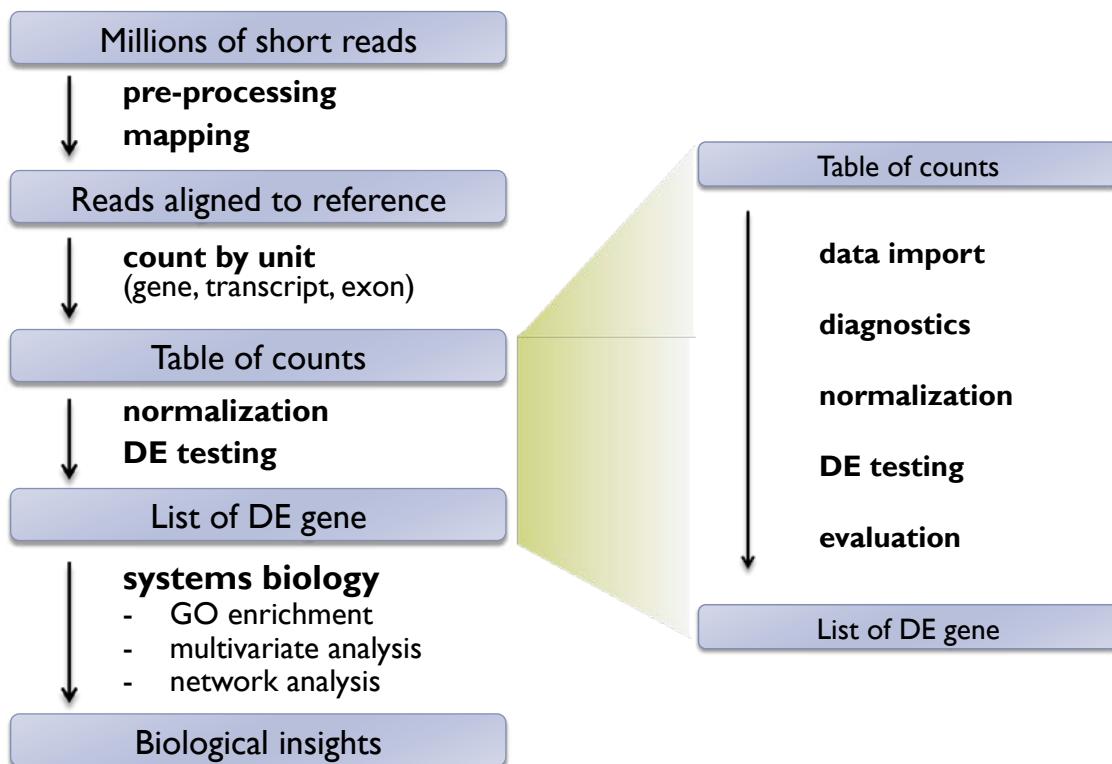
Cuffdiff

- ▶ **Model:** FPKM, Geometric, quartile
- ▶ **Normalization:** Pooled (default), per-condition, blind, Poisson

(example) Cuffdiff

- ▶ **Model**
 - ▶ FPKM, Geometric, quartile
- ▶ **Normalization**
 - ▶ Pooled (default), per-condition, blind, Poisson (not recommended)

RNA-seq analysis pipeline for DE



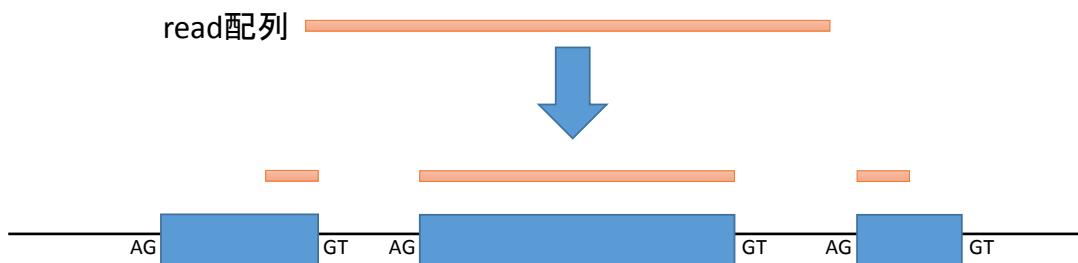
基生研ゲノムインフォマティックストレーニングコース 2015秋
2015.09.09-2015.09.11

RNA-Seqパイプライン ゲノムベースの解析法

基礎生物学研究所・生物機能解析センター
山口勝司

genomeをレファレンスとする場合

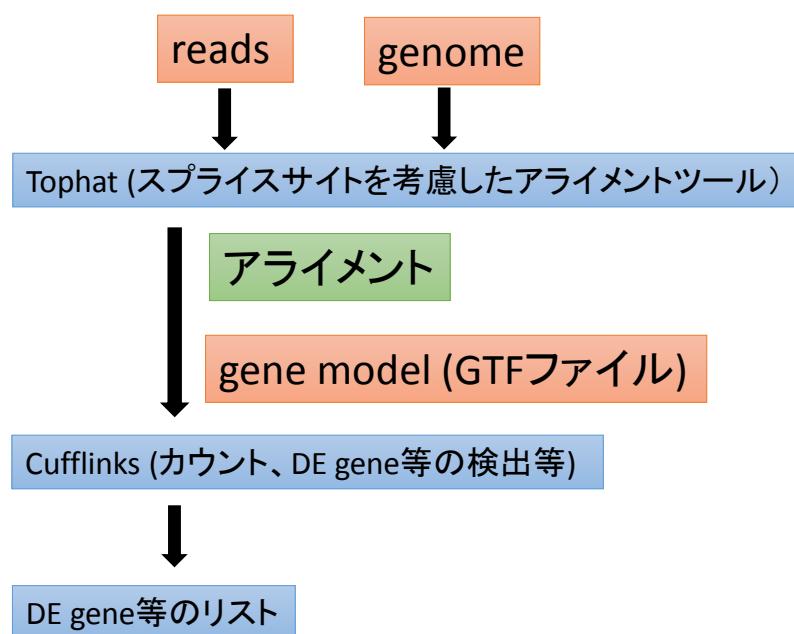
レファレンスがゲノム配列の場合
イントロン配列のスプライシングを考慮した
アライメントを行う必要がある。
TopHatを用いる
他 Blat, SpliceMap, MapSplice, GSMPA, QPALMA



実際こんな感じにアラインされる



本トレーニングコースでの流れ



TopHat2になりalignerとして
Bowtie2に対応
indelを考慮したアライメントが
可能になった 2012.4

TopHat
A spliced read mapper for RNA-Seq

TopHat is a fast splice junction mapper for RNA-Seq reads. It aligns RNA-Seq reads to mammalian-sized genomes using the ultra high-throughput short-read aligner **Bowtie**, and then analyzes the mapping results to identify splice junctions between exons.

TopHat is a collaborative effort among Daehwan Kim and Steven Salzberg in the Center for Computational Biology at Johns Hopkins University, and Cole Trapnell in the Genomics Sciences Department at the University of Washington. TopHat was originally developed by Cole Trapnell at the Center for Bioinformatics and Computational Biology at the University of Maryland, College Park.

» **TopHat 2.1.0 release 6/29/2015**

- TopHat-Fusion algorithm improvements for more sensitive and accurate discovery of fusions, thanks to contributions from Gordon Bean and Ryan Kelley at Illumina.
- This release implements a new algorithm for counting fusion-supporting read pairs that reduces the number of false-positive potential fusions. This algorithm computes the inner distance between read pairs by first converting the pair positions to transcript coordinates using the transcript information in refGene.txt and ensGene.txt. Pairs with small inner distance (suggesting the pair could come from a plausible pair-end insert) are counted as supporting evidence for the fusion. The default threshold for the inner distance is 250 base pairs; this parameter can be set using the `--fusion-pair-dist <int>` flag.
- fixed a few issues with GFF parsing of some annotation files
- fixed a runtime error when using `--no-discordant` option.

Several fixes/improvements thanks to contributors on GitHub:

- new `--max-num-fusions` option allowing the user to specify the maximum number of reported fusions in tophat-fusion-post
- adjusting lower limit for `--fusion-multipairs`
- fixed a few typos, cleaning up python code etc.

» **TopHat source code moved to GitHub 3/31/2015**

TopHat is now available as a public GitHub repository where users are welcome to submit bug reports (issues) and developers are encouraged to submit patches (pull requests).

» **TopHat 2.0.14 release 3/24/2015**

Version 2.0.14 is a maintenance release with the following changes:

- pipeline speed improvements thanks to contributions from Véronique Legrand and Michael Pressigout of Institut Pasteur
- added support for xz compressed read files (thanks to a patch submitted by Ashton Trey Belew)
- applied a couple of Python fixes to prevent potential issues with package handling and some file operations
- fixed a potential linking issue where the wrong libbam.a library could have been linked when building from source

» **TopHat 2.0.13 release 10/2/2014**

Version 2.0.13 is a maintenance release with the following changes:

- removed SAMtools as an external dependency in order to avoid incompatibility issues with recent and future changes of SAMtools and its code library (an older, stable SAMtools version is now packaged with TopHat)
- fixed a few code compatibility issues when compiling on OSX 10.9

» **TopHat 2.0.12 release 6/24/2014**

Version 2.0.12 is a maintenance release with the following simple fix:

Kim et al. *Genome Biology* 2013, 14:R36
<http://genomebiology.com/2013/14/4/R36>



METHOD

Open Access

TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions

Daehwan Kim^{1,2,3*}, Geo Pertea³, Cole Trapnell^{5,6}, Harold Pimentel⁷, Ryan Kelley⁸ and Steven L Salzberg^{3,4}

Abstract

TopHat is a popular spliced aligner for RNA-sequence (RNA-seq) experiments. In this paper, we describe TopHat2, which incorporates many significant enhancements to TopHat. TopHat2 can align reads of various lengths produced by the latest sequencing technologies, while allowing for variable-length indels with respect to the reference genome. In addition to *de novo* spliced alignment, TopHat2 can align reads across fusion breaks, which can occur after genomic translocations. TopHat2 combines the ability to identify novel splice sites with direct mapping to known transcripts, producing sensitive and accurate alignments, even for highly repetitive genomes or in the presence of pseudogenes. TopHat2 is available at <http://ccb.jhu.edu/software/tophat>.

TopHat

A spliced read mapper for RNA-Seq

JOHNS HOPKINS UNIVERSITY
CENTER FOR COMPUTATIONAL BIOLOGY

C C B



TopHat is a fast splice junction mapper for RNA-Seq reads. It aligns RNA-Seq reads to mammalian-sized genomes using the ultra high-throughput short read aligner **Bowtie**, and then analyzes the mapping results to identify splice junctions between exons.

TopHat is a collaborative effort among Daehwan Kim and Steven Salzberg in the [Center for Computational Biology](#) at Johns Hopkins University, and Cole Trapnell in the [Genome Sciences Department](#) at the University of Washington. TopHat was originally developed by Cole Trapnell at the [Center for Bioinformatics and Computational Biology](#) at the University of Maryland, College Park.

» TopHat 2.1.0 release 6/29/2015

- TopHat-Fusion algorithm improvements for more sensitive and accurate discovery of fusions, thanks to contributions from Gordon Bean and Ryan Kelley at Illumina.
 - This release implements a new algorithm for counting fusion-supporting read pairs that reduces the number of false-positive potential fusions. This algorithm computes the inner distance between read pairs by first converting the pair positions to transcript coordinates using the transcript information in refGene.txt and ensGene.txt. Pairs with small inner distance (suggesting the pair could come from a plausible pair-end insert) are counted as supporting evidence for the fusion. The default threshold for the inner distance is 250 base pairs; this parameter can be set using the `--fusion-pair-dist <int>` flag.
 - fixed a few issues with GFF parsing of some annotation files
 - fixed a runtime-error when using `--no-discordant` option.
- Several fixes/improvements thanks to contributors on GitHub:
- new `--max-num-fusions` option allowing the user to specify the maximum number of reported fusions in `tophat-fusion-post`
 - adjusting lower limit for `--fusion-multipairs`
 - fixed a few typos, cleaning up python code etc.

» TopHat source code moved to [GitHub](#) 3/31/2015

TopHat is now available as a public GitHub repository where users are welcome to submit bug reports (issues) and developers are encouraged to submit patches (pull requests).

» TopHat 2.0.14 release 3/24/2015

Version 2.0.14 is a maintenance release with the following changes:

- pipeline speed improvements thanks to contributions from Véronique Legrand and Michaël Pressigout of Institut Pasteur
- added support for xz compressed read files (thanks to a patch submitted by Ashton Trey Belew)
- applied a couple of Python fixes to prevent potential issues with package handling and some file operations
- fixed a potential linking issue where the wrong libbam.a library could have been linked when building from source

» TopHat 2.0.13 release 10/2/2014

Version 2.0.13 is a maintenance release with the following changes:

- removed SAMtools as an external dependency in order to avoid incompatibility issues with recent and future changes of SAMtools and its code library (an older, stable SAMtools version is now packaged with TopHat)
- fixed a few code compatibility issues when compiling on OSX 10.9

» TopHat 2.0.12 release 6/24/2014

Version 2.0.12 is a maintenance release with the following simple fix:

Site Map

[Home](#)
[Getting started](#)
[Manual](#)
[Index and annotation downloads](#)
[FAQ](#)
[Protocol](#)

News and updates

New releases and related tools will be

Getting startedで、
とりあえず使って
みる

Tools Users Google Group. Please use tophat.cufflinks@gmail.com for private communications only. Please do not email technical questions to TopHat contributors directly.

Releases

version 2.1.0	6/29/2015
Source code	
Linux x86_64 binary	
Mac OS X x86_64 binary	

Related Tools

Cufflinks :	Isoform assembly and quantitation for RNA-Seq
Bowtie :	Ultrafast short read alignment
TopHat-Fusion :	An algorithm for Discovery of Novel Fusion Transcripts
CummeRbund :	Visualization of RNA-Seq differential analysis

Getting started

- Install quick-start
- Test the installation
- Preparing your reference
- Preparing your reads
- Running TopHat
- Examining your results

» Install quick-start

Download and extract the latest [Bowtie 2](#) (or [Bowtie](#)) releases.

Note that you can use either Bowtie 2 (the default) or Bowtie (`--bowtie1`) and you will need the following Bowtie 2 (or Bowtie) programs in your PATH:

- `bowtie2` (or `bowtie`)
- `bowtie2-build` (or `bowtie-build`)
- `bowtie2-inspect` (or `bowtie-inspect`)

Installing a pre-compiled binary release

In order to make it easy to install TopHat we provide a few binary packages to save users from the occasionally frustrating process of building TopHat themselves, which requires a certain development environment and the Boost libraries installed. To use the binary packages, simply download the appropriate one for your platform, unpack it, and make sure the `tophat` binaries are in a directory in your PATH environment variable (or create a symbolic link to the included `tophat2` script somewhere in your PATH, see below)

Note: if you want to be able to install and run this new version without overwriting a previous TopHat version already installed on your system, make sure you unpack the new version into a different directory from the old version, then instead of copying the new programs in a directory in your PATH just create a symbolic link from the `tophat2` wrapper script in this new directory to a directory in your shell's PATH. For example, assuming the `~/bin` directory is in your PATH and you unpack `tophat-2.0.0.Linux_x86_64.tar.gz` under your home directory:

```
cd
tar xvzf tophat-2.0.0.Linux_x86_64.tar.gz
cd ~/bin
ln -s ~/tophat-2.0.0.Linux_x86_64/tophat2 .
```

Now you can start the new version of TopHat with the `tophat2` command, while the previous version, if present, can still be launched with the regular "tophat" command (assuming this is how you used it before).

Building TopHat from source

In order to build TopHat2 you must have the following installed on your system:

- the [Boost C++ libraries](#) (we recommend version 1.47 or higher so you can use it for building Cufflinks as well)

インストールの方法・
必要ツールなどの記載・
テストデータ等での極く簡単な
解析手順に関する記載がある

必要ツール

- `bowtie2`
- `samtools`

TopHat2はあらかじめコンパイルした
バイナリーファイルが配布されている
ので、自分でmakeする必要はない。
自分でソースからmakeする場合は
◦ SAMtools lib
◦ Boost C++ library
が必要

testデータが用意されている

```
tar zxvf test_data.tar.gz
cd test_data
tophat -r 20 test_ref reads_1.fq reads_2.fq
```

TopHat
A spliced read mapper for RNA-Seq

TopHat is a fast splice junction mapper for RNA-Seq reads. It aligns RNA-Seq reads to mammalian-sized genomes using the ultra high-throughput short read aligner **Bowtie**, and then analyzes the mapping results to identify splice junctions between exons.

TopHat is a collaborative effort among Daehwan Kim and Steven Salzberg in the **Center for Computational Biology** at Johns Hopkins University, and Cole Trapnell in the **Genome Sciences Department** at the University of Washington. TopHat was originally developed by Cole Trapnell at the **Center for Bioinformatics and Computational Biology** at the University of Maryland, College Park.

TopHat 2.1.0 release 6/29/2015

- TopHat-Fusion algorithm improvements for more sensitive and accurate discovery of fusions, thanks to contributions from Gordon Bean and Ryan Kelley at Illumina.
- This release implements a new algorithm for counting fusion-supporting read pairs that reduces the number of false-positive potential fusions. This algorithm computes the inner distance between read pairs by first converting the pair positions to transcript coordinates using the transcript information in refGene.txt and ensGene.txt. Pairs with small inner distance (suggesting the pair could come from a plausible pair-end insert) are counted as supporting evidence for the fusion. The default threshold for the inner distance is 250 base pairs; this parameter can be set using the `--fusion-pair-dist <int>` flag.
- fixed a few issues with GFF parsing of some annotation files
- fixed a runtime-error when using `--no-discordant` option.

Several fixes/improvements thanks to contributors on GitHub:

- new `--max-num-fusions` option allowing the user to specify the maximum number of reported fusions in tophat-fusion-post
- adjusting lower limit for `--fusion-multipairs`
- fixed a few typos, cleaning up python code etc.

TopHat source code moved to GitHub 3/31/2015

TopHat is now available as a public GitHub repository where users are welcome to submit bug reports (issues) and developers are encouraged to submit patches (pull requests).

TopHat 2.0.14 release 3/24/2015

Version 2.0.14 is a maintenance release with the following changes:

- pipeline speed improvements thanks to contributions from Véronique Legrand and Michaël Pressigout of Institut Pasteur
- added support for xx compressed read files (thanks to a patch submitted by Ashton Trey Belew)
- applied a couple of Python fixes to prevent potential issues with package handling and some file operations
- fixed a potential linking issue where the wrong libbam.a library could have been linked when building from source

TopHat 2.0.13 release 10/2/2014

Version 2.0.13 is a maintenance release with the following change:

- removed SAMtools as an external dependency in order to avoid SAMtools and its code library (an older, stable SAMtools version)
- fixed a few code compatibility issues when compiling on OSX 10.9

TopHat 2.0.12 release 6/24/2014

Version 2.0.12 is a maintenance release with the following simple fix:

of
パラメータの意味など
詳しく知るためには、
必ずManualを見る

Site Map

- Home
- Getting started
- Manual**
- Index and annotation downloads
- FAQ
- Protocol

News and updates

New releases and related tools will be announced through the Bowtie mailing list.

Getting Help

Questions and comments about TopHat can be posted on the **Tuxedo Tools Users Google Group**. Please use tophat.cufflinks@gmail.com for private communications only. Please do not email technical questions to TopHat contributors directly.

Releases

version 2.1.0	6/29/2015
Source code	
Linux x86_64 binary	
Mac OS X x86_64 binary	

Related Tools

- Cufflinks: Isoform assembly and quantitation for RNA-Seq
- Bowtie: Ultrafast short read alignment
- TopHat-Fusion: An algorithm for Discovery of Novel Fusion Transcripts
- CummeRbund: Visualization of RNA-Seq differential analysis

Manual

- [What is TopHat?](#)
- [Prerequisites](#)
- [Using TopHat](#)

What is TopHat?

TopHat is a program that aligns RNA-Seq reads to a genome in order to identify exon-exon splice junctions. It is built on the ultrafast short read mapping program **Bowtie**. TopHat runs on **Linux** and **OS X**.

What types of reads can I use TopHat with?

TopHat was designed to work with reads produced by the Illumina Genome Analyzer, although users have been successful in using TopHat with reads from other technologies. In TopHat 1.1.0, we began supporting Applied Biosystems' Colorspace format. The software is optimized for reads 75bp or longer.

How does TopHat find junctions?

TopHat can find splice junctions without a reference annotation. By first mapping RNA-Seq reads to the genome, TopHat identifies potential exons, since many RNA-Seq reads will contiguously align to the genome. Using this initial mapping information, TopHat builds a database of possible splice junctions and then maps the reads against these junctions to confirm them.

Short read sequencing machines can currently produce reads 100bp or longer but many exons are shorter than this so they would be missed in the initial mapping. TopHat solves this problem mainly by splitting all input reads into smaller segments which are then mapped independently. The segment alignments are put back together in a final step of the program to produce the end-to-end read alignments.

TopHat generates its database of possible splice junctions from two sources of evidence. The first and strongest source of evidence for a splice junction is when two segments from the same read (for reads of at least 45bp) are mapped at a certain distance on the same genomic sequence or when an internal segment fails to map - again suggesting that such reads are spanning multiple exons. With this approach, "GT-AG", "GC-AG" and "AT-AC" introns will be found ab initio. The second source is pairings of "coverage islands", which are distinct regions of piled up reads in the initial mapping. Neighboring islands are often spliced together in the transcriptome, so TopHat looks for ways to join these with an intron. We only suggest users use this second option (`--coverage-search`) for short reads (<45bp) and with a small number of reads (<= 10 million). This latter option will only report alignments across "GT-AG" introns

・75base以上のreadに最適化
・リファレンスannotationなしでも
スプライスジャンクションを見つける

Illumina has provided the RNA-Seq user community with a set of genome sequence indexes (including Bowtie indexes) as well as GTF transcript annotation files. These files can be used with TopHat and Cufflinks to quickly perform expression analysis and gene discovery. The annotation files are augmented with the `tss_id` and `p_id` GTF attributes that Cufflinks needs to perform differential splicing, CDS output, and promoter user analysis. We recommend that you download your Bowtie indexes and annotation files from this page. More information about Illumina's iGenomes project can be found [here](#).

Organism	Data source	Version	Size	Last Modified
Homo sapiens	Ensembl	GRCh37	17297 MB	May 14 17:23
		build36.3	15814 MB	May 14 19:36
		build37.1	15850 MB	May 14 19:04
		build37.2	21450 MB	May 14 17:54
		hg18	17349 MB	May 14 15:31
	NCBI	hg19	21058 MB	May 14 15:36
		hg38	21058 MB	May 14 15:36
	UCSC	hg39	21058 MB	May 14 15:36
		mm10	14193 MB	Jun 14 11:29
		mm12	14537 MB	May 14 21:12
		mm13	15725 MB	May 14 22:52
		mm14	15260 MB	May 15 17:53
Mus musculus	Ensembl	NCBIM37	14428 MB	May 14 22:13
		build37.1	15260 MB	May 15 17:53
		build37.2	13315 MB	May 11 14:18
		mm9	14537 MB	May 14 21:12
	NCBI	mm10	14193 MB	Jun 14 11:29
		rn4	13710 MB	May 15 22:32
		rn5	14234 MB	May 15 23:58
		rn6	13725 MB	May 15 22:33
Rattus norvegicus	Ensembl	RGSC3.4	13725 MB	May 15 22:33
		RGSC_v3.4	14234 MB	May 15 23:58
		rn4	13710 MB	May 15 22:32
	UCSC	Btau_4.0	13315 MB	May 11 14:18
		UMD3.1	14042 MB	May 11 12:41
Bos taurus	Ensembl	Btau_4.2	13357 MB	May 11 14:11
		Btau_4.6.1	13448 MB	May 11 16:09
	NCBI	UMD_3.1	13990 MB	May 11 16:08
		UMD_3.2	13990 MB	May 11 16:08

Site Map
Home
Getting started
Manual
Index and annotation downloads
FAQ
Protocol
News and updates
New releases and related tools will be announced through the Bowtie mailing list .
Getting Help
Questions and comments about TopHat can be posted on the Tuxedo Tools Users Google Group . Please use tophat.cufflinks@gmail.com for private communications only. Please do not email technical questions to TopHat contributors directly.
Releases
version 2.0.12 6/24/2014
Source code
Linux x86_64 binary
Mac OS X x86_64 binary
Related Tools
Cufflinks : Isoform assembly and quantitation for RNA-Seq
Bowtie : Ultrafast short read alignment
TopHat-Fusion : An algorithm for

メジャーな生物種では
indexファイルやannotation
ファイル等が配布されて
いるので有効活用できる

Frequently Asked Questions

- » How to control the alignment of reads in terms of number of mismatches, gap length etc. ?
- » How can I maximize the accuracy of spliced mapping in TopHat?
- » I don't know the mate inner distance (-r/-mate-inner-dist option) for my paired reads, what value should I use?
- » I am not sure which library type to use (fr-firststrand or fr-secondstrand), what should I do?
- » What should I do if I see a message like "Too many open files"?

» How to control the alignment of reads in terms of number of mismatches, gap length etc. ?

You can use three options: `--read-mismatches`, `--read-gap-length` and `--read-edit-dist`. For instance, if you want read alignments with at most 2 base mismatches and no gaps then you can specify:

```
--read-mismatches 2 --read-gap-length 0 --read-edit-dist 2
```

Or if you want read alignments with total length of indels (alignment gaps) of at most 3bp and at most 2 base mismatches you can use these options:

```
--read-mismatches 2 --read-gap-length 3 --read-edit-dist 3
```

FAQも参考に

» How can I maximize the accuracy of spliced mapping in TopHat?

Based on real RNA-seq samples we found out that in the genome mapping step of TopHat a high portion of reads spanning several exons can incorrectly be aligned to processed pseudogenes that are rarely (if any) transcribed or expressed, instead of the genes where they originate from. You can use either of the options below to improve the accuracy of spliced mapping in TopHat:

- If a good gene annotation is available (as the case with the human genome), use it with the `-G` option
 - For poorly annotated genomes you might want to consider using the "`--read-realign-edit-dist 0`" option
- With the realignment option users can choose to remap some (or all) of the mapped reads with mapping edit distance equal to or above user-specified "remapping" edit distance (see `--read-realign-edit-dist` option). Setting "`--read-realign-edit-dist 0`" will map every read against transcriptome, genome, and splice variants (or splice junctions) that are detected by TopHat, no matter whether it is mapped or not in any mapping step. With this remapping strategy, this "pseudogene" problem can be effectively handled. If you use a genome that has processed pseudogenes and you cannot provide good gene annotation to TopHat, you may want to consider using this option for accurate mapping results.

Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks

Cole Trapnell, Adam Roberts, Loyal Goff, Geo Pertea, Daehwan Kim, David R Kelley, Harold Pimentel, Steven L Salzberg, John L Rinn & Lior Pachter

[Affiliations](#) | [Contributions](#) | [Corresponding author](#)

Nature Protocols 7, 562–578 (2012) | doi:10.1038/nprot.2012.016

Published online 01 March 2012

 [Citation](#)  [Reprints](#)  [Rights & permissions](#)  [Article metrics](#)

Abstract

[Abstract](#) • [Accession codes](#) • [References](#) • [Author information](#)

Recent advances in high-throughput cDNA sequencing (RNA-seq) can reveal new genes and splice variants and quantify expression genome-wide in a single assay. The volume and complexity of data from RNA-seq experiments necessitate scalable, fast and mathematically principled analysis software. TopHat and Cufflinks are free, open-source software tools for gene discovery and comprehensive expression analysis of high-throughput mRNA sequencing (RNA-seq) data. Together, they allow biologists to identify new genes and new splice variants of known ones, as well as compare gene and transcript expression under two or more conditions. This protocol describes in detail how to use TopHat and Cufflinks to perform such analyses. It also covers several accessory tools and utilities that aid in managing data, including CummeRbund, a tool for visualizing RNA-seq analysis results. Although the procedure assumes basic informatics skills, these tools assume little to no background with RNA-seq analysis and are meant for novices and experts alike. The protocol begins with raw sequencing reads and produces a transcriptome assembly, lists of differentially expressed and regulated genes and transcripts, and publication-quality visualizations of analysis results. The protocol's execution time depends on the volume of transcriptome sequencing data and available computing resources but takes less than 1 d of computer time for typical experiments and ~1 h of hands-on time.

protocol論文も出ている

ただし今となっては少し古い

Freeではない

tophat基本コマンド

TopHat is a fast splice junction mapper for RNA-Seq reads. It aligns RNA-Seq reads to mammalian-sized genomes using the ultra high-throughput short read aligner [Bowtie](#), and then analyzes the mapping results to identify splice junctions between exons.

```
> tophat -G gene.gtf -o out_dir genome read_1.fastq read_2.fastq
```

-G/--GTF <GTF/GFF3 file>

まずgtfに基づき、トランスクリプトにmapさせ、ゲノム位置として戻す。
mapしないリードはゲノムから探す

tophatの出力

prep_reads.info
 align_summary.txt
 deletions.bed
 insertions.bed
 junctions.bed
 accepted_hits.bam
 unmapped.bam

sam/bam フォーマットのファイル
 accepted_hits.bamファイルがこの後必要

実習1

tophatを用いて2D_1のfastqファイルをgenome_chr4にmapさせよ、
 GTFファイルとしてgenes_chr4.gtfを用いる

例)

```
> tophat -p 4 -G genes_chr4.gtf -o 2D_1 genome_chr4 2D_1_R1.fastq 2D_1_R2.fastq
```

出力を確認しよう。

例えば、align_summary.txtを見ればどの程度mapしたか分かる。
 これでRNA-Seqのリード配列がゲノム配列にアラインできた。

cufflinksを用いてアラインされたreadを数える

定義した方法でのカウントが可能

gene単位

トランスクリプト単位

エキソン単位

- cufflinks
- BEDTools
- HTseq

が利用できる

今回はCufflinksを利用

そもそもTopHat → Cufflinksの解析系は同じ開発元、非常に良く使われている。

ローカスアノテーション情報を記載したgtfファイルを用意しておけば、
 それに基づいて、genes単位、isoforms単位での解析を進めてくれる。

簡易的に、特定ローカスの解析などを進めたい場合や、
 gtfファイルがない場合などは、BEDToolsも有用
 gtfファイル自分で作製するのは結構大変だが、bedファイルは比較的容易

<http://cole-trapnell-lab.github.io/cufflinks/>

Cufflinks

Transcriptome assembly and differential expression analysis for RNA-Seq.

Cufflinks assembles transcripts, estimates their abundances, and tests for differential expression and regulation in RNA-Seq samples. It accepts aligned RNA-Seq reads and assembles the alignments into a parsimonious set of transcripts. Cufflinks then estimates the relative abundances of these transcripts based on how many reads support each one, taking into account biases in library preparation protocols.

Cufflinks was originally developed as part of a collaborative effort between the [Laboratory for Mathematical and Computational Biology](#), led by Lior Pachter at UC Berkeley, Steven Salzberg's [computational genomics group](#) at the Institute of Genetic Medicine at Johns Hopkins University, and [Barbara Wold's lab](#) at Caltech. The project is now maintained by [Cole Trapnell's lab](#) at the University of Washington.

Cufflinks is provided under the OSI-approved [Boost License](#)

News

To get the latest updates on the Cufflinks project and the rest of the "Tuxedo tools", please subscribe to our [mailing list](#)

Cufflinks has moved to GitHub	DECEMBER 10, 2014
Cufflinks 2.2.1 released	MAY 05, 2014
Cufflinks 2.2.0 released	MARCH 25, 2014
Cufflinks 2.1.1 released	APRIL 11, 2013

Protocol

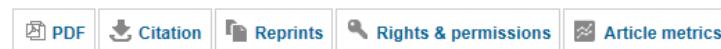
Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation

[Cole Trapnell, Brian A Williams, Geo Pertea, Ali Mortazavi, Gordon Kwan, Marijke J van Baren, Steven L Salzberg, Barbara J Wold & Lior Pachter](#)

[Affiliations](#) | [Contributions](#) | [Corresponding author](#)

Nature Biotechnology **28**, 511–515 (2010) | doi:10.1038/nbt.1621

Received 02 February 2010 | Accepted 22 March 2010 | Published online 02 May 2010



High-throughput mRNA sequencing (RNA-Seq) promises simultaneous transcript discovery and abundance estimation^{1, 2, 3}. However, this would require algorithms that are not restricted by prior gene annotations and that account for alternative transcription and splicing. Here we introduce such algorithms in an open-source software program called Cufflinks. To test Cufflinks, we sequenced and analyzed >430 million paired 75-bp RNA-Seq reads from a mouse myoblast cell line over a differentiation time series. We detected 13,692 known transcripts and 3,724 previously unannotated ones, 62% of which are supported by independent expression data or by homologous genes in other species. Over the time series, 330 genes showed complete switches in the dominant transcription start site (TSS) or splice isoform, and we observed more subtle shifts in 1,304 other genes. These results suggest that Cufflinks can illuminate the substantial regulatory flexibility and complexity in even this well-studied model of muscle development and that it can improve transcriptome-based genome annotation.

[INSTALL](#)[MANUAL](#)[GETTING STARTED](#)[TOOLS](#)[HELP](#)[HOW IT WORKS](#)[PROTOCOL](#)[BENCHMARKS](#)[CODE](#)[FEED](#)

Cufflinks

Transcriptome assembly and differential expression analysis for RNA-Seq.

Cufflinks is available for Linux and Mac OS X. You can find the full list of releases below.

The Cufflinks source code for each point release is available below as well. If you want to grab the current code, check out the [Cufflinks GitHub repository](#).

Star 23 Fork 12

Cufflinks Releases

Version	Date	Linux	Mac OS X	Source
2.2.1	May 05, 2014	Linux	Mac OS X	Source
2.2.0	March 25, 2014	Linux	Mac OS X	Source
2.1.1	April 11, 2013	Linux	Mac OS X	Source
2.1.0	April 10, 2013	Linux	Mac OS X	Source

[INSTALL](#)[MANUAL](#)[GETTING STARTED](#)[TOOLS](#)[HELP](#)[HOW IT WORKS](#)[PROTOCOL](#)[BENCHMARKS](#)[CODE](#)[FEED](#)

Cufflinks

Transcriptome assembly and differential expression analysis for RNA-Seq.

- Install quick-start
 - Installing a pre-compiled binary release
- Building Cufflinks from source
 - Installing Boost
 - Installing the SAM tools
 - Installing the Eigen libraries
 - Building Cufflinks
 - Testing the installation
- Common uses of the Cufflinks package
- Using pre-built annotation packages

自分でソースからmakeする場合は
 • Samtools
 • Boost C++ library
 が必要

cufflinks ./test_data.sam

これでツールが動くことを確認

Install quick-start

Installing a pre-compiled binary release

In order to make it easy to install Cufflinks, we provide a few binary packages to save users from occasionally frustrating process of building Cufflinks, which requires that you install the Boost libraries. To use the binary packages, simply download the appropriate one for your machine, untar it, and make sure the cufflinks,cuffdiff and cuffcompare binaries are in a directory in your PATH environment variable.

Cufflinks

Transcriptome assembly and differential expression analysis for RNA-Seq.

Bowtie: ultrafast short read alignment

Bowtie is an ultrafast and memory-efficient tool for aligning sequencing reads to long reference sequences. It is particularly good at aligning reads of about 50 up to 100s or 1,000s of characters, and particularly good at aligning to relatively long (e.g. mammalian) genomes. Bowtie 2 indexes the genome with an FM Index to keep its memory footprint small: for the human genome, its memory footprint is typically around 3.2 GB. Bowtie 2 supports gapped, local, and paired-end alignment modes.

Bowtie is provided under the OSI-approved Artistic License 2.0.

TopHat: alignment of short RNA-Seq reads

TopHat is a fast splice-junction mapper for RNA-Seq reads. It aligns RNA-Seq reads to mammalian-sized genomes using the ultra high-throughput short read aligner Bowtie, and then analyzes the mapping results to identify splice junctions between exons.

TopHat is provided under the OSI-approved Artistic License 2.0.

CummeRbund: visualization of RNA-Seq differential analysis

CummeRbund is an R package that is designed to aid and simplify the task of analyzing Cufflinks RNA-Seq output.

CummeRbund is provided under the OSI-approved Artistic License 2.0.

Monocle: Differential expression for single-cell RNA-Seq and qPCR.

Monocle is a toolkit for analyzing single-cell gene expression experiments. Monocle was designed for RNA-Seq, but can also work with single cell qPCR. It performs differential expression analysis, and can find genes that differ between cell types or between cell states. When used to study an ongoing biological process such as cell differentiation, Monocle learns that process and places cells in order according to their progress through it. Monocle finds genes that are dynamically regulated during that process.

Monocle is provided under the OSI-approved Artistic License (version 2.0)

Cufflinksの関連ツール
Bowtie, TopHatは説明済み

Cufflinks

Transcriptome assembly and differential expression analysis for RNA-Seq.

The Cufflinks RNA-Seq workflow

The Cufflinks suite of tools can be used to perform a number of different types of analyses for RNA-Seq experiments. The Cufflinks suite includes a number of different programs that work together to perform these analyses. The complete workflow, performing all the types of analyses Cufflinks can execute, is summarized in the graph below. The left side illustrates the "classic" RNA-Seq workflow, which includes read mapping with **TopHat**, assembly with Cufflinks, and visualization and exploration of results with **CummeRbund**. A newer, more advanced workflow was introduced with Cufflinks version 2.2.0, and is shown on the right. Both are still supported. You can read about the classic workflow in detail in our [protocol paper](#).



Cufflinks

Cufflinks is both the name of a suite of tools and a program within that suite. Cufflinks the program assembles transcriptomes from RNA-Seq data and quantifies their expression.

Cuffcompare

After assembling a transcriptome from one or more samples, you'll probably want to compare your assembly to known transcripts. Even if there is no "reference" transcriptome for the organism you're studying, you may want to compare the transcriptomes assembled from different RNA-Seq libraries. Cuffcompare helps you perform these comparisons and assess the quality of your assembly.

Cuffmerge

When you have multiple RNA-Seq libraries and you've assembled transcriptomes from each of them, we recommend that you merge these assemblies into a master transcriptome. This step is required for a differential expression analysis of the new transcripts you've assembled. Cuffmerge performs this merge step.

Cuffquant

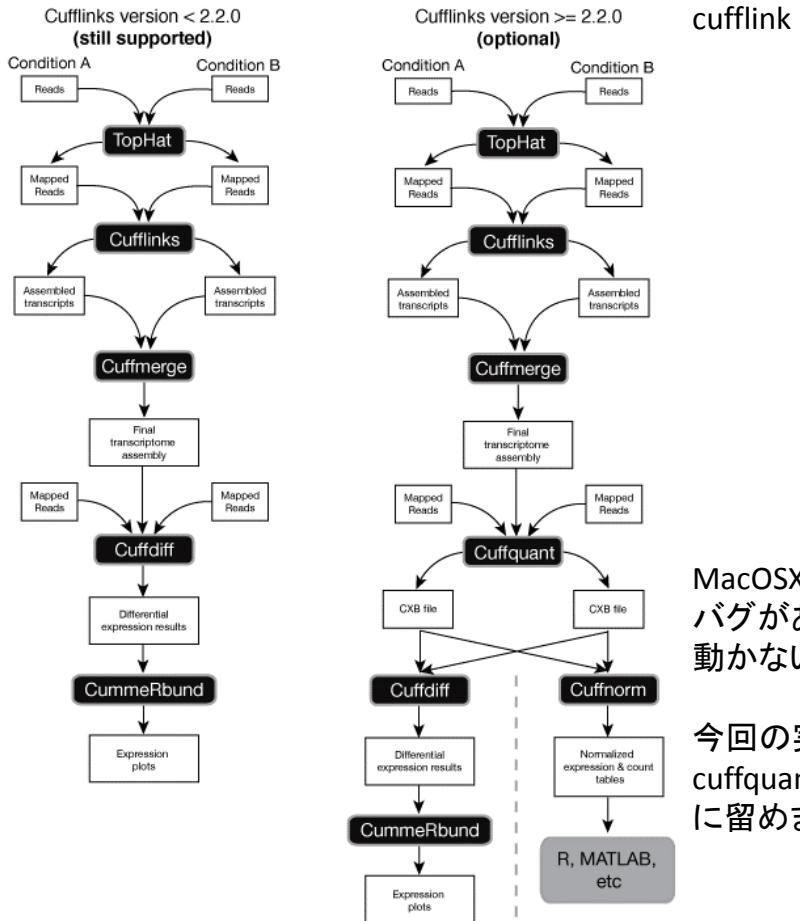
Quantifying gene and transcript expression in RNA-Seq samples can be computationally expensive. Cuffquant allows you to compute the gene and transcript expression profiles and save these profiles to files that you can analyze later with Cuffdiff or Cuffnorm. This can help you distribute your computational load over a cluster and is recommended for analyses involving more than a handful of libraries.

Cuffdiff

Comparing expression levels of genes and transcripts in RNA-Seq experiments is a hard problem. Cuffdiff is a highly accurate tool for performing these comparisons, and can tell you not only which genes are up- or down-regulated between two or more conditions, but also which genes are differentially spliced or are undergoing other types of isoform-level regulation.

Cuffnorm

Sometimes, all you want to do is normalize the expression levels from a set of RNA-Seq libraries so that they're all on the same scale, facilitating downstream analyses such as clustering. Expression levels reported by Cufflinks in FPKM units are usually comparable between samples, but in certain situations, applying an extra level of normalization can remove sources of bias in the data. Cuffnorm normalizes a set of samples to be on as similar scales as possible, which can improve the results you obtain with other downstream tools.



cufflink
cufflinks
cuffmerge
cuffcompare
cuffquant
cuffnorm
cuffdiff
の6つのプログラムから構成

cuffquant, cuffnormは
ver2.2.0(20140325)
から実装

MacOSX版のバイナリーはver2.2.0以降は
バグがありsegmentation errorでまともに
動かないようです。

今回の実習ではver2.1.1を使用し、
cuffquant, cuffnormは簡単な説明のみ
に留めます。

INSTALL MANUAL GETTING STARTED TOOLS HELP HOW IT WORKS PROTOCOL BENCHMARKS CODE FEED

Cufflinks

Transcriptome assembly and differential expression analysis for RNA-Seq.

Cufflinks is an ongoing research project as well as a suite of tools. Here are the papers that describe the science behind the programs. If you use Cufflinks, [please cite these papers](#) in your work!

Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation

Cole Trapnell, Brian Williams, Geo Pertea, Ali Mortazavi, Gordon Kwan, Jeltje van Baren, Steven Salzberg, Barbara Wold, Lior Pachter.

Nature Biotechnology, 2010

High-throughput mRNA sequencing (RNA-Seq) promises simultaneous transcript discovery and abundance estimation. However, this would require algorithms that are not restricted by prior gene annotations and that account for alternative transcription and splicing. Here we introduce such algorithms in an open-source software program called Cufflinks. To test Cufflinks, we sequenced and analyzed ~430 million paired 75-bp RNA-Seq reads from a mouse myoblast cell line over a differentiation time series. We detected 13,692 known transcripts and 3,724 previously unannotated ones, 62% of which are supported by independent expression data or by homologous genes in other species. Over the time series, 330 genes showed complete switches in the dominant transcription start site (TSS) or splice isoform, and we observed more subtle shifts in 1,304 other genes. These results suggest that Cufflinks can illuminate the substantial regulatory flexibility and complexity in even this well-studied model of muscle development and that it can improve transcriptome-based genome annotation.

doi:10.1038/nbt.1621

Note: This is the original Cufflinks paper. Please cite this paper if you use Cufflinks in your work.

Improving RNA-Seq expression estimates by correcting for fragment bias

Adam Roberts, Cole Trapnell, Julie Donaghey, John L. Rinn, Lior Pachter.

Genome Biology, 2011

どうやって動いているか

まず動いて使えそうな感じになつたら詳細を把握していく

cufflinks基本コマンド

Cufflinksコマンド

```
cufflinks -o out_directory -G hoge.gtf tophat_directory/accepted_hits.bam
```

cufflinksを実行してパラメータを確認しよう。

考慮すべきパラメーター例

- o 出力の指定、TopHatの出力と同じ場所にしておくのが分かりやすいだろう
- p CPUスレッド数の指定(デフォルトは1)、結構時間がかかるので使える数を指定すると良いだろう
- G GTFファイルに記載されたアノテーションのみについて計算
- g GTFファイルに記載されたアノテーションをガイドにしてアセンブルする
- M 無視したいトランスクript(rRNAなど)を指定

cufflinks出力

出力

skipped.gtf
transcripts.gtf
genes.fpkm_tracking
isoforms.fpkm_trancing

実習2

先のtophatの結果を用いてcufflinksにかけてみよう

例)

```
> cufflinks -p 4 -o 2D_1 -G genes_chr4.gtf accepted_hits.bam
```

出力を確認しよう。

geneごと、isoformごとにFPKM値が計算されているのが分かる。

-gを用いてcufflinksにかけると新規の発現領域が存在するのが分かる

cuffcompareコマンド

Cufflinks includes a program that you can use to help analyze the transfrags you assemble. The program cuffcompare helps you:

Compare your assembled transcripts to a reference annotation

Track Cufflinks transcripts across multiple experiments (e.g. across a time course)

From the command line, run cuffcompare as follows:

```
cuffcompare [options]* <cuff1.gtf> [cuff2.gtf] ... [cuffN.gtf]
```

今回はすでにあるgtfファイルの情報を用いるので、意識的に使う必要はない。

cuffmergeコマンドと出力

個々のサンプルのアセンブルモデルを統合する。

```
Usage:
  cuffmerge [Options] <assembly_GTF_list.txt>

Options:
  -h/--help                         Prints the help message and exits
  -o                                <output_dir>          Directory where merged assembly will be written [ default: ./merged_asm ]
  -g/--ref-gtf                        An optional "reference" annotation GTF.
  -s/--ref-sequence                  <seq_dir>/<seq_fasta> Genomic DNA sequences for the reference.
  --min-isoform-fraction <0-1.0>    Discard isoforms with abundance below this [ default: 0.05 ]
  -p/--num-threads                   <int>                  Use this many threads to merge assemblies. [ default: 1 ]
  --keep-tmp
```

統合ファイルリストを事前に作製する必要がある(例 assemblies.txt)

```
cuffmerge -s $REFSEQ -g $GTF assemblies.txt
```

例 assemblies.txt

```
~/arabi_2D_2/transcripts.gtf
~/arabi_2D_3/transcripts.gtf
~/arabi_2D2L_2/transcripts.gtf
~/arabi_2D2L_3/transcripts.gtf
```

出力

merged.gtf

Cufflinks includes a script called cuffmerge that you can use to merge together several Cufflinks assemblies. It handles also running Cuffcompare for you, and automatically filters a number of transfrags that are probably artifacts. If you have a reference GTF file available, you can provide it to the script in order to gracefully merge novel isoforms and known isoforms and maximize overall assembly quality. The main purpose of this script is to make it easier to make an assembly GTF file suitable for use with Cuffdiff.

cuffdiffコマンド

DE gene等を統計計算で取り出す
コマンド入力して使用法を確認してみよう

```
Usage: cuffdiff [options] <transcripts.gtf> <sample1_hits.sam> <sample2_hits.sam> [... sampleN_hits.sam]
Supply replicate SAMs as comma separated lists for each condition:
sample1_rep1.sam,sample1_rep2.sam,...sample1_repM.sam
General Options:
-o/--output-dir           write all output files to this directory          [ default: ./ ]
-L/--labels                comma-separated list of condition labels
--FDR                      False discovery rate used in testing          [ default: 0.05 ]
```

```
cuffdiff -o out_file merged.gtf bam1,bam2,bam3 bam4,bam5,bam6
```

Version 2.2.0以降は先のcuffquantで得られたcxbファイルをbamファイルの代わりに用いる。
cuffdiffにかかる時間やメモリー使用量が軽減される。

cuffdiffの出力

bias_params.info	gene_exp.diff
run.info	cds_exp.diff
read_groups.info	cds.diff
var_model.info	isoform_exp.diff
cds.read_group_tracking	promoters.diff
cds.fpkm_tracking	splicing.diff
cds.count_tracking	tss_group_exp.diff
genes.read_group_tracking	
genes.fpkm_tracking	
genes.count_tracking	
isoforms.read_group_tracking	
isoforms.count_tracking	
isoforms.fpkm_tracking	
tss_groups.read_group_tracking	
tss_groups.fpkm_tracking	
tss_groups.count_tracking	

diffの付いたファイルがそれぞれの
違いの情報を記載したファイル

.diffファイルの内容

Column number	Column name	Example	Description
1	Tested id	XLOC_000001	A unique identifier describing the transcript, gene, primary transcript, or CDS being tested
2	gene	Lypla1	The gene_name(s) or gene_id(s) being tested
3	locus	chr1:4797771-4835363	Genomic coordinates for easy browsing to the genes or transcripts being tested.
4	sample 1	Liver	Label (or number if no labels provided) of the first sample being tested
5	sample 2	Brain	Label (or number if no labels provided) of the second sample being tested
6	Test status	NOTEST	Can be one of OK (test successful), NOTEST (not enough alignments for testing), LOWDATA (too complex or shallowly sequenced), HIDATA (too many fragments in locus), or FAIL, when an ill-conditioned covariance matrix or other numerical exception prevents testing.
7	FPKM _x	8.01089	FPKM of the gene in sample x
8	FPKM _y	8.551545	FPKM of the gene in sample y
9	log ₂ (FPKM _y /FPKM _x)	0.06531	The (base 2) log of the fold change y/x
10	test stat	0.860902	The value of the test statistic used to compute significance of the observed change in FPKM
11	p value	0.389292	The uncorrected p-value of the test statistic
12	q value	0.985216	The FDR-adjusted p-value of the test statistic
13	significant	no	Can be either "yes" or "no", depending on whether p is greater than the FDR after Benjamini-Hochberg correction for multiple-testing

cuffquantコマンドと出力(ver2.2.0以降)

bamの内容からgene/transcriptレベルで定量化し、バイナリー出力する

cuffquant -o out_directory hoge.gtf accepted_hits.bam

cuffquantを実行してパラメータを確認しよう。

考慮すべきパラメーター例

- o 出力ディレクトリーの指定
- p CPUスレッド数の指定(デフォルトは1)、結構時間がかかるので使える数を指定
- M 無視したいトランスクリプト(rRNAなど)を指定
- 他にもestimationに関わる -b -u パラメータがある。

出力

abundances.cxb

```
> cuffquant -p 4 -o 2D_1 genes_chr4.gtf accepted_hits.bam
```

新たにcxbファイルが作製されていることが分かる。

出力ファイルはこの1つだけ

cuffdiffの前にcuffquantを行い、cxbファイルを作製することで cuffdiffを速くできる。

cuffnormコマンドと出力(ver2.2.0以降)

Cuffnormコマンド

Cuffnorm, which simply computes
a normalized table of expression values for genes and transcripts.

```
> cuffnorm -o out_file genes_chr4.gtf bam1,bam2,bam3 bam4,bam5,bam6
```

```
cuffnorm [options]* <transcripts.gtf>
<sample1_replicate1.sam[,...,sample1_replicateM.sam]>
<sample2_replicate1.sam[,...,sample2_replicateM.sam]>...
[sampleN.sam_replicate1.sam[,...,sample2_replicateM.sam]]
```

sam/bamかcxbファイルどちらも入力可能。ただし混在は不可

cuffnormの出力(ver2.2.0以降)

```
cds.attr_table
cds.count_table
cds.fpkm_table
cuffnorm.tree
genes.attr_table
genes.count_table
genes.fpkm_table
isoforms.attr_table
isoforms.count_table
isoforms.fpkm_table
run.info
samples.table
tss_groups.attr_table
tss_groups.count_table
tss_groups.fpkm_table
```

たくさんのサンプルで発現プロットやクラスター図を書きたい場合便利。

tophat -> cufflinksの解析系を使用する際の注意

It does not perform differential expression analysis. To assess the significance of changes in expression for genes and transcripts between conditions, use Cuffdiff. Cuffnorm's output files are useful when you have many samples and you simply want to cluster them or plot expression levels of genes important in your study.

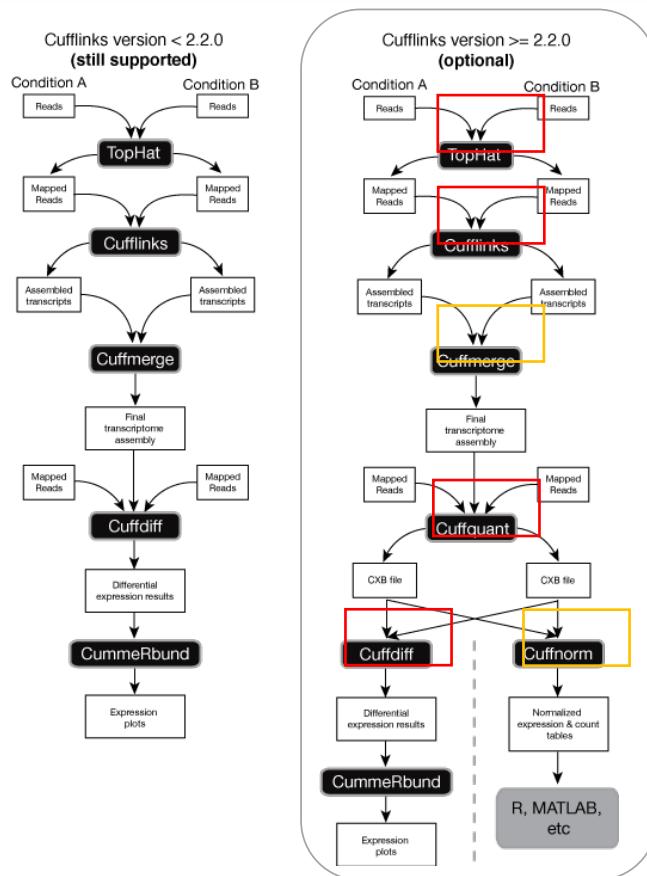
Cuffnorm will report both FPKM values and **normalized**, estimates for the number of fragments that originate from each gene, transcript, TSS group, and CDS group. Note that because these counts are already normalized to account for differences in library size, they should not be used with downstream differential expression tools that require **raw** counts as input.

tophat -> cufflinksは一連の解析系

cufflinksの出力はすでにノーマライズされたものなので、rawデータを要求するedgeRなどの別のツールのinputには利用できない。

The screenshot shows the Nature Protocols website with the title "Count-based differential expression analysis of RNA sequencing data using R and Bioconductor". It includes the authors (Simon Anders, Davis J McCarthy, Yunshun Chen, Michał Okoniewski, Gordon K Smyth, Wolfgang Huber & Mark D Robinson), affiliations, and a brief abstract. The abstract discusses the use of RNA-seq for profiling transcriptomes across different conditions, mentioning the need for quality control checks and statistical modeling. It highlights the workflow, which is largely based on the free open-source R language and Bioconductor software, and notes that hands-on time for typical small experiments (e.g., 4–10 samples) can be <1 h, with computation time <1 d using a standard desktop PC.

versionによる違いまとめ



tophat, cufflinksの実習

1. TopHatを用いて、paired-endのtest data

2D_2_R1.fastq, 2D_2_R2.fastq

をリファレンスgenome_chr4にマップさせよ

オプション -Gの有無に

による違いを確認しよう。

2.Cufflinksを用いて、

2D_2のカウントをしよう。

-Gと-gの違いを確認しよう。

結果をIGVで可視化してみよう

TAIR10の配列を呼び出し、TopHatで得られたBAMファイルを読み込む



Excelを使って結果を確認してみよう

gene_exp.diffファイルを読み込んでみる
tab区切りテキストファイルなのでそのまま読み込める
Excelのsort機能を使ってq値でsortしてみる

q値でsort


test_id	gene_id	gene	locus	sample_1	sample_2	status	value_1	value_2	log2(fold_chan	test_stat	p_value	q_value	significant
XLOC_000047	XLOC_000047	KEA1	1:284609-291094	q1	q2	OK	12.8356	47.6879	1.89347	4.44122	5.00E-05	0.000325	yes
XLOC_000091	XLOC_000091	BXL2	1:564204-567769	q1	q2	OK	112.839	21.5634	-2.38762	-6.02938	5.00E-05	0.000325	yes
XLOC_000148	XLOC_000148	PSB27	1:898875-899655	q1	q2	OK	194.744	691.64	1.82844	7.10401	5.00E-05	0.000325	yes
XLOC_000310	XLOC_000310	PSBP-1	1:2047824-2049418	q1	q2	OK	588.195	3147.84	2.42	7.92975	5.00E-05	0.000325	yes
XLOC_000404	XLOC_000404	NPQ1	1:2706923-2709531	q1	q2	OK	21.2494	78.5734	1.88662	3.26377	5.00E-05	0.000325	yes
XLOC_000419	XLOC_000419	CSD1	1:2827060-2838469	q1	q2	OK	503.523	181.545	-1.47173	-5.38312	5.00E-05	0.000325	yes
XLOC_000450	XLOC_000450	CSP41B	1:3015327-3018234	q1	q2	OK	113.687	650.406	2.51627	8.83387	5.00E-05	0.000325	yes
XLOC_000487	XLOC_000487	LRR_XI-23	1:3252239-3255693	q1	q2	OK	26.4081	49.6396	0.910512	2.30664	5.00E-05	0.000325	yes
XLOC_000598	XLOC_000598	ATGLX1	1:3995168-3997907	q1	q2	OK	60.1583	162.387	1.4326	3.26419	5.00E-05	0.000325	yes
XLOC_000600	XLOC_000600	AT1G11860	1:4001112-4003442	q1	q2	OK	319.6	756.582	1.24323	4.18318	5.00E-05	0.000325	yes
XLOC_000614	XLOC_000614	AT1G12080	1:4084161-4085045	q1	q2	OK	1884.29	67.9613	-4.79316	-9.20293	5.00E-05	0.000325	yes
XLOC_000616	XLOC_000616	CHL1-1	1:4105232-4109545	q1	q2	OK	107.267	57.7917	-0.892267	-2.70294	5.00E-05	0.000325	yes
XLOC_000624	XLOC_000624	AT1G12230	1:4147961-4151056	q1	q2	OK	102.049	50.9296	-1.00268	-2.40566	5.00E-05	0.000325	yes
XLOC_000680	XLOC_000680	CYP71B7	1:4467219-4469033	q1	q2	OK	17.1443	84.588	2.30272	4.53043	5.00E-05	0.000325	yes
XLOC_000724	XLOC_000724	AT1G13930	1:4761011-4762666	q1	q2	OK	94.6747	2483.48	4.71324	10.4968	5.00E-05	0.000325	yes
XLOC_000749	XLOC_000749	AT1G14345	1:4899144-4899979	q1	q2	OK	38.3992	157.145	2.03295	4.49341	5.00E-05	0.000325	yes
XLOC_000765	XLOC_000765	AT1G14670	1:5037611-5040528	q1	q2	OK	84.8105	44.439	-0.932415	-2.66978	5.00E-05	0.000325	yes
XLOC_000835	XLOC_000835	NDF1	1:5489297-5493772	q1	q2	OK	20.0548	104.567	2.3824	4.27443	5.00E-05	0.000325	yes
XLOC_000884	XLOC_000884	HCF173	1:5723087-5727312	q1	q2	OK	7.34039	112.227	3.93442	5.2414	5.00E-05	0.000325	yes
XLOC_000916	XLOC_000916	FUG1	1:5885082-5890470	q1	q2	OK	48.9638	105.457	1.10687	3.5512	5.00E-05	0.000325	yes
XLOC_001003	XLOC_001003	NDF6	1:6460597-6462224	q1	q2	OK	45.3045	185.555	2.03412	2.97075	5.00E-05	0.000325	yes
XLOC_001030	XLOC_001030	LHCA6	1:6612748-6613972	q1	q2	OK	52.6816	153.395	1.54188	4.09397	5.00E-05	0.000325	yes
XLOC_001063	XLOC_001063	PUP14	1:6832346-6833837	q1	q2	OK	37.731	91.5568	1.27892	3.13218	5.00E-05	0.000325	yes
XLOC_001076	XLOC_001076	ATLFNR2	1:6942716-6945018	q1	q2	OK	87.7487	1025.37	3.54662	10.0816	5.00E-05	0.000325	yes
XLOC_001099	XLOC_001099	AT1G20390	1:7065493-7071561	q1	q2	OK	45.6232	15.9769	-1.51378	-4.22277	5.00E-05	0.000325	yes
XLOC_001170	XLOC_001170	AT1G21680	1:7613004-7615339	q1	q2	OK	27.146	80.96	1.57647	3.93831	5.00E-05	0.000325	yes

GTFファイルに記載された遺伝子ごとの発現カウントに対して倍率、p値、q値が計算される。

Rを使ってMA plotを書いて見よう

gene_exp.diffファイルを読み込んでみる
tab区切りテキストファイルなのでread.delim関数で読み込む
M, Aをそれぞれ計算する
plot関数を使って描画
colorのパラメータをsignifitureの値で色分けさせてみる。

例)

```
dat <- read.delim("gene_exp.diff")
A<-1/2*(log2(dat$value_1+1)+log2(dat$value_2+1))
M<-log2(dat$value_1+1)-log2(dat$value_2+1)
plot(A,M,col=dat$significant, pch=16, cex=0.4, ylim=c(-8,8))
```

簡易スクリプトを使って、結果を成形してみよう

Awkは便利な簡易スクリプト
1行記述でもできる

例)

q_valueが0.05以下のもののみリストアップするには?
q_valueの記載は13列目だから…

awk '\$13<=0.05 {print \$0}' gene_exp.diff
と記述すればOK
\$で列番号を指定できる
\$0は行全体を意味する

その他

grep, head, sort, cut, uniq等のUnixコマンドも活用しよう

実践演習課題

データセット

2D_1, 2D_2, 2D_3と2D2L_1, 2D2L_2, 2D2L_3をTopHat→Cufflinksの系を用いて、
2D(2days dark条件で育てた芽生え)
2D2L(その後2days light条件で育てた芽生え)
でのDE gene等を確認せよ。

GTFファイルとしてgenes_chr4.gtf
fastaファイルとしてgenome_chr4.fa
を利用する。
(アラビドプシスTAIR10の配列だが計算時間を考慮して、
それぞれChr4のみになっている)

RNA-Seqパイプライン -ゲノムベースの解析法-の最終3スライドを参考に、
マッピングデータのIGVでの可視化、
エクセルでの確認、
Rを用いたM-A plotの描画、
簡易スクリプトを用いたデータ抽出をせよ。

基礎生物学研究所 ゲノムインフォマティクス・トレーニングコース 2015秋

RNA-seq解析パイプライン： Transcript-based pipeline

Shuji Shigenobu
重信 秀治

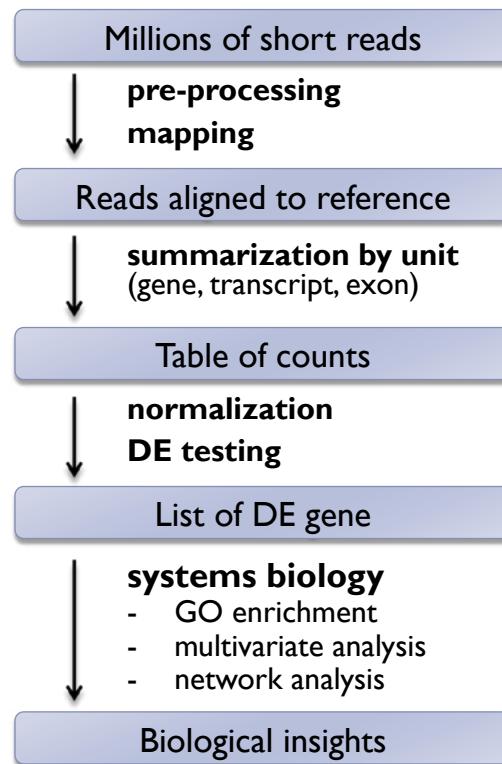
基礎生物学研究所
生物機能解析センター



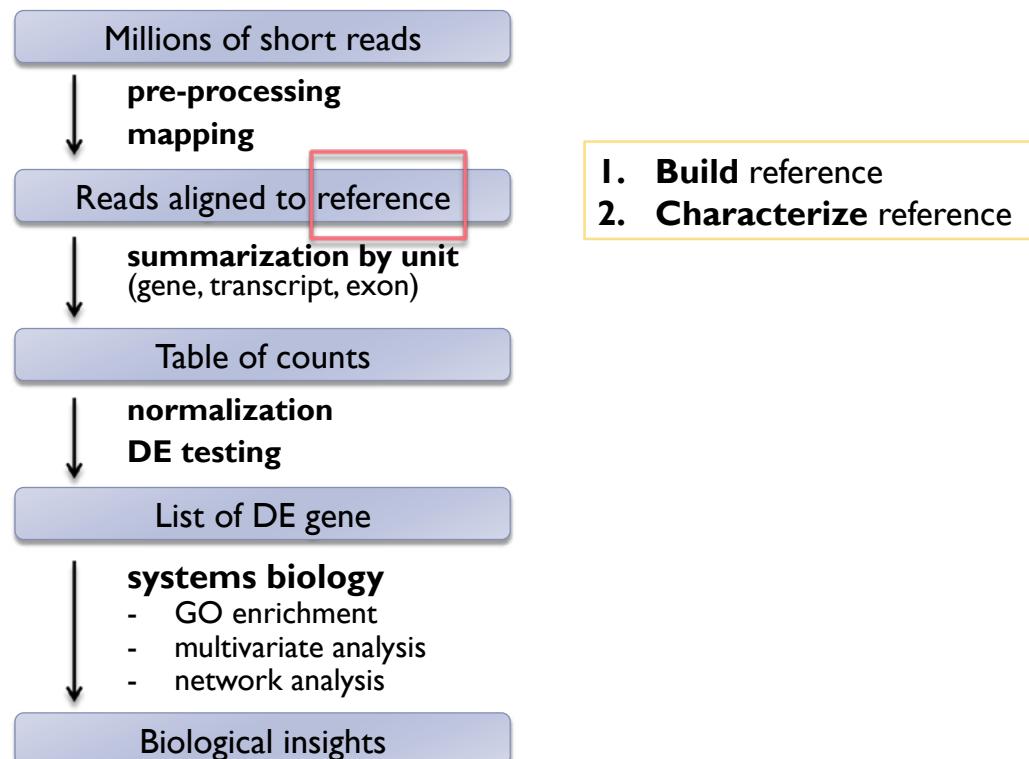
Two Basic Pipelines

- ▶ Choice of reference
- ▶ **Genome** – standard for genome-known species
- ▶ **Transcript** – the only way for genome-unknown species
 - can be used for genome-known species

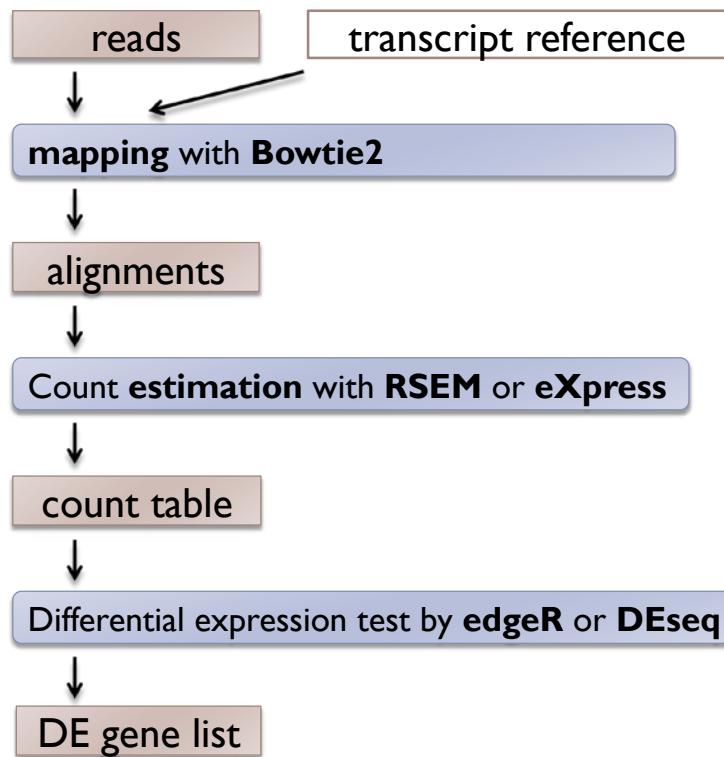
RNA-seq workflow with reference genome



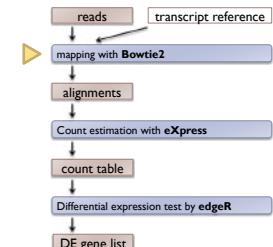
RNA-seq workflow **without** reference genome



A Pipeline: Transcript-based



Mapping – alignment software



- ▶ For mapping reads onto transcript reference
short read mapper (unspliced read aligner) is used
- ▶ **Bowtie2** – basic mapping to reference sequence

<http://bowtie-bio.sourceforge.net/bowtie2/index.shtml>

Bowtie 2 is an ultrafast and memory-efficient tool for aligning sequencing reads to long reference sequences. It is particularly good at aligning reads of about 50 up to 100s or 1,000s of characters, and particularly good at aligning to relatively long (e.g. mammalian) genomes. Bowtie 2 indexes the genome with an FM Index to keep its memory footprint small: for the human genome, its memory footprint is typically around 3.2 GB. Bowtie 2 supports gapped, local, and paired-end alignment modes.

Hiring Postdocs

- The Langmead and Salzberg labs currently have open positions for postdoctoral researchers. See [the posting](#) and please apply if you're interested in working with either or both of us.

Version 2.1.0 - February 21, 2013

- Improved multithreading support so that Bowtie 2 now uses native Windows threads when compiled on Windows and uses a faster mutex. Threading performance should improve on all platforms.
- Improved support for building 64-bit binaries for Windows x64 platforms.
- Bowtie 2 uses a lightweight mutex by default.
- Test option `--nospin` is no longer available. However bowtie2 can always be recompiled with `EXTRA_FLAGS="-DNQ_SPINLOCK"` in order to drop the default spinlock usage.

Version 2.0.6 - January 27, 2013

- Fixed issue whereby spurious output would be written in `--no-unal` mode.
- Fixed issue whereby multiple input files combined with `--reorder` would cause truncated output and a memory spike.
- Fixed spinlock datatype for Win64 API (LLP64 data model) which made it crash when compiled under Windows 7 x64.
- Fixed bowtie2 wrapper to handle filename/paths operations in a more platform independent manner.
- Added pthread as a default library option under cygwin, and ptheadGC for MinGW.
- Fixed some minor issues that made MinGW compilation fail.

Version 2.0.5 - January 4, 2013

- Fixed an issue that would cause excessive memory allocation when aligning to very repetitive genomes.
- Fixed an issue that would cause a pseudo-randomness-related assert to be thrown in debug mode under rare circumstances.

Version 2.0.4 - December 17, 2012

- Updated manual's discussion of the `-1` and `-X` options to mention that setting them farther apart makes Bowtie 2 slower.
- Renamed `COPYING` to `LICENSE` and created a `README` to be GitHub-friendly.

Version 2.0.4 - December 17, 2012

- Fixed issue whereby `--un`, `--al`, `--un-conc`, and `--al-conc` options would incorrectly suppress SAM output.

bowtie2

Bowtie is an ultrafast, memory-efficient short read aligner.

<http://bowtie-bio.sourceforge.net/bowtie2/index.shtml>

(example)

```
$ bowtie2 -x transcript.fa -U reads.fq -a -S out.sam
```

▶ Output

▶ Alignment in SAM format : **out.sam**

(ex1) Let's Try Bowtie2

Align 75-bp Illumina reads with a transcript reference using Bowtie2.

Prepare reads and reference genome

Sequences for this exercise are stored in `~/data/ss/`.

IlluminaReads1.fq – Illumina reads in fastq format
minimouse_mRNA.fa – a set of transcript sequences

Build index of reference sequence

```
$bowtie2-build minimouse_mRNA.fa myref
```

Align reads with reference

```
$bowtie2 -x myref -U IlluminaReads1.fq -a -S out.sam
```

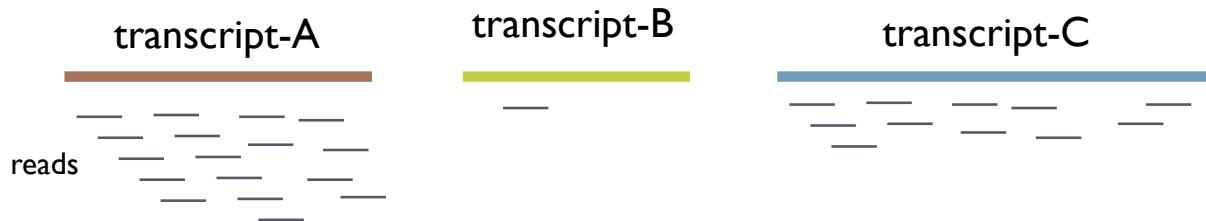
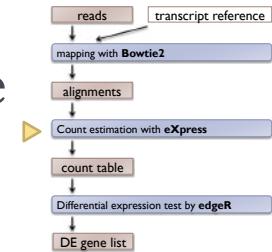
Evaluation of mapping results

- ▶ Evaluation of SAM/BAM file
 - ▶ Check statistics
 - ▶ Visualization

(example)

```
$ samtools view bowtieout.bam
```

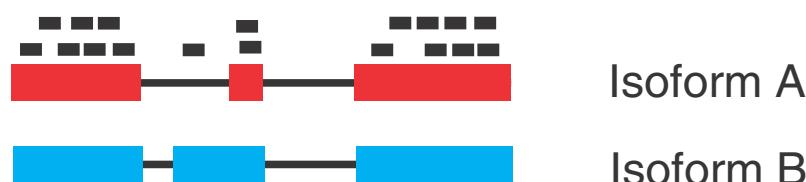
Count Reads by Transcript/gene



- ▶ The simplest way: just count reads by contig.
But...
- ▶ Mapping ambiguity should be taken into consideration.

Estimate Abundance

- ▶ **Multimapping issues**
 - ▶ Isoforms
 - ▶ Very similar paralogs
 - ▶ Repetitive sequences
 - ▶ => cannot align reads uniquely
- ▶ Mapping ambiguity should be taken into consideration.



- ▶ Critical for RNA-seq de novo analysis
- ▶ Software: RSEM and eXpress (EM algorithm)

eXpress

eXpress is a streaming tool for quantifying the abundances of a set of target sequences from sampled subsequences.

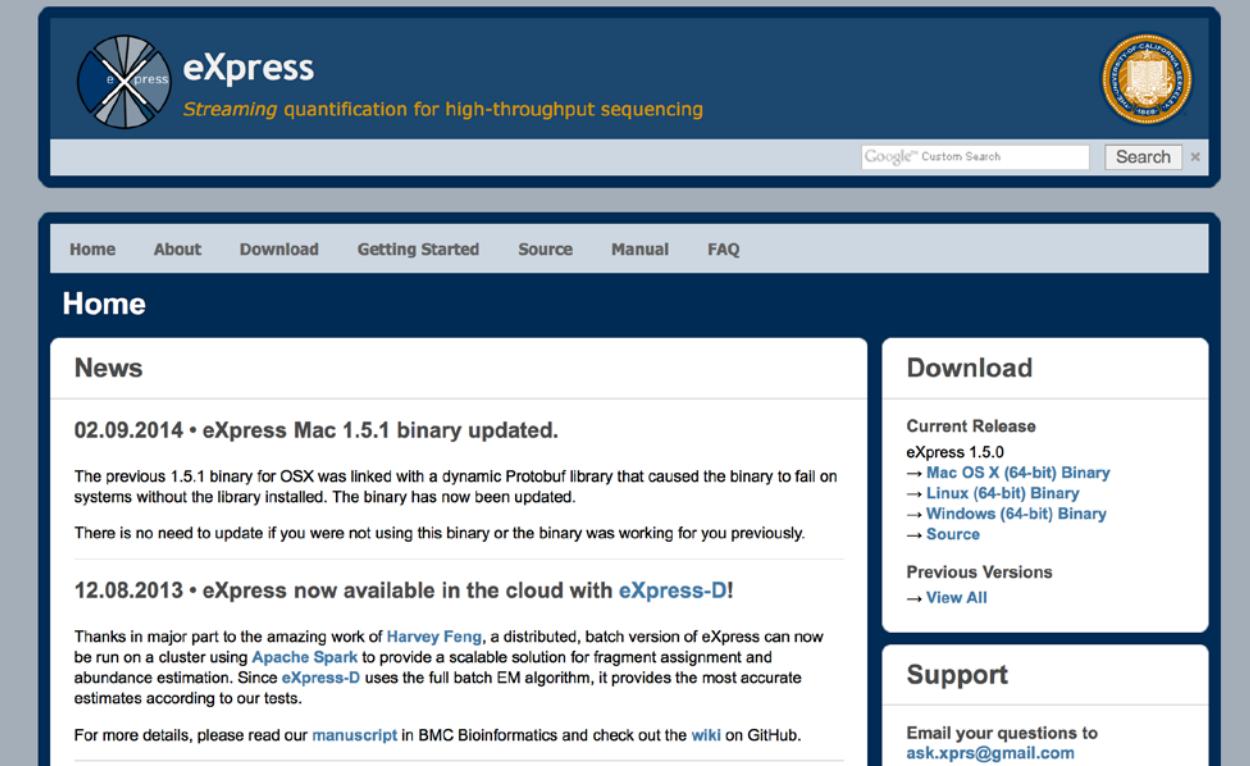
<http://bio.math.berkeley.edu/eXpress/>

(example)

```
$ express transcripts.fasta hits.bam
```

▶ Output

- ▶ Count estimation table: **results.xprs**



The screenshot shows the official website for eXpress. At the top, there's a dark blue header bar with the eXpress logo (a stylized 'X' with 'e' and 'X' inside), the text 'eXpress' in white, and 'Streaming quantification for high-throughput sequencing' in yellow. To the right is the University of California Berkeley seal. Below the header is a navigation menu with links: Home, About, Download, Getting Started, Source, Manual, and FAQ. The 'Home' link is highlighted in white. The main content area has a dark blue background. On the left, under 'News', there are two entries: '02.09.2014 • eXpress Mac 1.5.1 binary updated.' and '12.08.2013 • eXpress now available in the cloud with eXpress-D!'. The 'Download' section on the right lists the 'Current Release' as eXpress 1.5.0 with links for Mac OS X, Linux, Windows, and Source. It also has a 'Previous Versions' section with a 'View All' link. The 'Support' section at the bottom right includes an email address: ask.xprs@gmail.com.

<http://bio.math.berkeley.edu/eXpress/index.html>

(ex1) Let's Try eXpress

Align 75-bp Illumina reads with a transcript reference using Bowtie2.

Prepare alignments and reference genome

Sequences for this exercise are stored in `~/data/ss/`.

```
IlluminaReads1.fq - Illumina reads in fastq format
out.sam - this file should be generated in the previous bowtie practice
```

Run eXpress

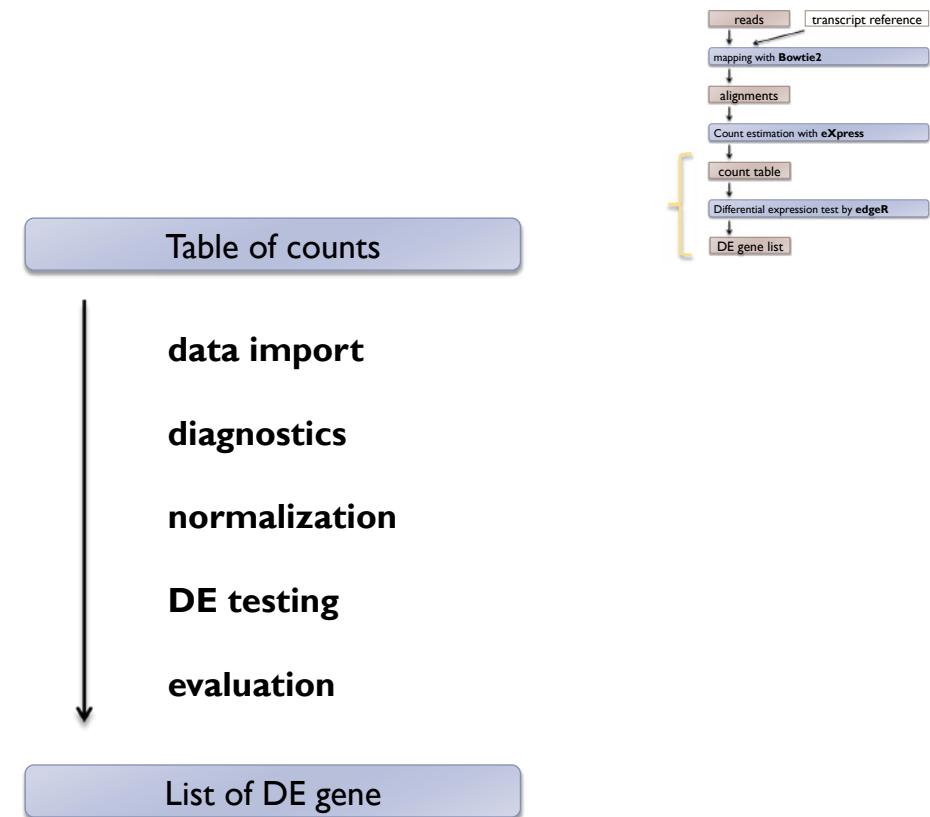
```
$ express minimouse_mRNA.fa out.sam
```

```
Output : results.xprs, params.xprs
```

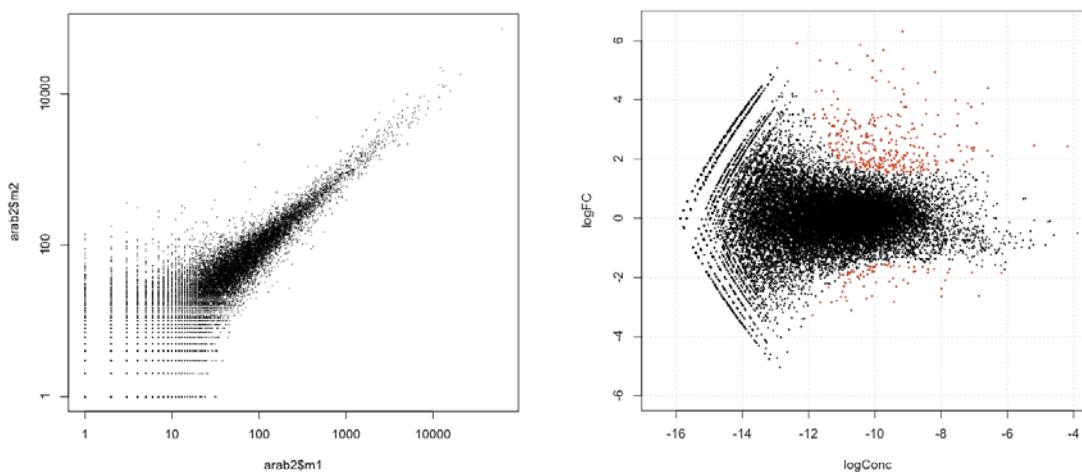
eXpress: output

`results.xprs`

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	bundle_id	target_id	length	eff_length	tot_counts	uniq_counts	est_counts	eff_counts	ambig_distr_alpha	ambig_distr_beta	fpkm	fpkm_conf_low	fpkm_conf_high	solvable
2	1.m.245853	621	398.1	807	15	86.2	134.4	9.83E+01	9.96E+02	2.34E+01	1.88E+01	2.80E+01 T		
3	1.m.245856	660	442.0	991	199	919.8	1373.4	5.53E+01	5.46E+00	2.25E+02	2.12E+02	2.38E+02 T		
4	2.m.42076	1959	1591.7	156	156	156.0	192.0	0.00E+00	0.00E+00	1.06E+01	1.06E+01	1.06E+01 T		
5	3.m.60782	291	83.0	12	12	12.0	42.1	0.00E+00	0.00E+00	1.57E+01	1.57E+01	1.57E+01 T		
6	4.m.158451	282	64.5	0	0	0.0	0.0	0	0	0.00E+00	0.00E+00	0.00E+00	0.00E+00 T	
7	5.m.337354	219	39.4	0	0	0.0	0.0	0	0	0.00E+00	0.00E+00	0.00E+00	0.00E+00 T	
8	6.m.338934	261	82.3	0	0	0.0	0.0	0	0	0.00E+00	0.00E+00	0.00E+00	0.00E+00 T	
9	7.m.5973	822	719.9	4	4	4.0	4.6	0.00E+00	0.00E+00	6.01E-01	6.01E-01	6.01E-01 T		
10	8.m.337793	219	38.7	0	0	0.0	0.0	0	0	0.00E+00	0.00E+00	0.00E+00	0.00E+00 T	
11	9.m.340910	210	40.5	0	0	0.0	0.0	0	0	0.00E+00	0.00E+00	0.00E+00	0.00E+00 T	
12	10.m.289784	3177	2521.4	350	350	441.0	0.00E+00	0.00E+00	1.50E+01	1.50E+01	1.50E+01	1.50E+01 T		
13	11.m.248666	240	61.8	1	1	1.0	3.9	0.00E+00	0.00E+00	1.75E+00	1.75E+00	1.75E+00 T		
14	12.m.90727	240	55.7	13	13	13.0	56.1	0.00E+00	0.00E+00	2.53E+01	2.53E+01	2.53E+01 T		
15	13.m.338727	216	48.1	0	0	0.0	0.0	0	0	0.00E+00	0.00E+00	0.00E+00	0.00E+00 T	
16	14.m.123519	225	43.2	0	0	0.0	0.0	0	0	0.00E+00	0.00E+00	0.00E+00	0.00E+00 T	
17	15.m.328661	251	50.8	1	1	1.0	4.9	0.00E+00	0.00E+00	2.13E+00	2.13E+00	2.13E+00 T		
18	16.m.26062	642	356.1	1	1	1.0	1.8	0.00E+00	0.00E+00	3.04E-01	3.04E-01	3.04E-01 T		
19	17.m.1295	240	53.6	0	0	0.0	0.0	0	0	0.00E+00	0.00E+00	0.00E+00	0.00E+00 T	
20	18.m.307626	201	220.2	4	3	3.0	2.7	8.33E+00	4.07E+04	1.47E+00	1.46E+00	1.49E+00 T		
21	18.m.307625	204	35.7	301	300	301.0	1718.3	1.02E+01	2.10E-03	9.12E+02	9.05E+02	9.18E+02 T		
22	19.m.49789	237	51.9	3	3	3.0	13.7	0.00E+00	0.00E+00	6.26E+00	6.26E+00	6.26E+00 T		
23	20.m.33508	162	151.3	1	1	1.0	1.1	0.00E+00	0.00E+00	7.15E-01	7.15E-01	7.15E-01 T		
24	21.m.109341	183	286.3	2	2	2.0	1.3	0.00E+00	0.00E+00	7.56E-01	7.56E-01	7.56E-01 T		
25	22.m.331919	564	277.3	0	0	0.0	0.0	0	0	0.00E+00	0.00E+00	0.00E+00	0.00E+00 T	
26	23.m.23766	303	98.5	3	3	3.0	9.2	0.00E+00	0.00E+00	3.30E+00	3.30E+00	3.30E+00 T		
27	24.m.246777	1149	1152.1	631	29	202.5	202.0	1.58E+02	3.90E+02	1.90E+01	1.65E+01	2.15E+01 T		
28	24.m.246852	1323	1315.4	761	156	588.8	592.2	1.22E+02	4.85E+01	4.84E+01	4.50E+01	5.19E+01 T		
29	24.m.246633	207	31.8	10	4	5.7	37.1	1.29E+04	3.27E+04	1.94E+01	1.05E+01	2.82E+01 T		
30	24.m.246662	192	200.4	6	3	3.0	2.9	1.20E+01	3.22E+03	1.63E+00	1.51E+00	1.74E+00 T		
31	25.m.99743	1641	1387.9	470	470	470.0	555.7	0.00E+00	0.00E+00	3.66E+01	3.66E+01	3.66E+01 T		
32	26.m.335620	234	58.9	0	0	0.0	0.0	0	0	0.00E+00	0.00E+00	0.00E+00	0.00E+00 T	
33	27.m.16882	528	297.5	14	14	14.0	24.9	0.00E+00	0.00E+00	5.09E+00	5.09E+00	5.09E+00	5.09E+00 T	
34	28.m.77438	255	81.4	9	9	9.0	28.2	0.00E+00	0.00E+00	1.20E+01	1.20E+01	1.20E+01	1.20E+01 T	
35	29.m.131505	450	263.2	18	11	15.8	27.1	8.87E+00	3.95E+00	6.51E+00	4.68E+00	8.35E+00 T		
36	29.m.131517	170	195.9	6	0	1.8	1.5	8.17E+00	1.96E+01	9.74E+01	0.00E+00	2.46E+00 T		
37	29.m.131504	705	528.2	15	14	14.4	19.2	6.51E+01	1.01E+02	2.05E+00	2.69E+00	3.21E+00 T		



Diagnostics: Scatter plot & MA plot



edgeR

- ▶ A Bioconductor package for differential expression analysis of digital gene expression data
- ▶ **Model:** An over dispersed Poisson model, negative binomial (NB) model, is used
- ▶ **Normalization:** TMM method (trimmed mean of M values) to deal with composition effects
- ▶ **DE test:** exact test and generalized linear models (GLM)

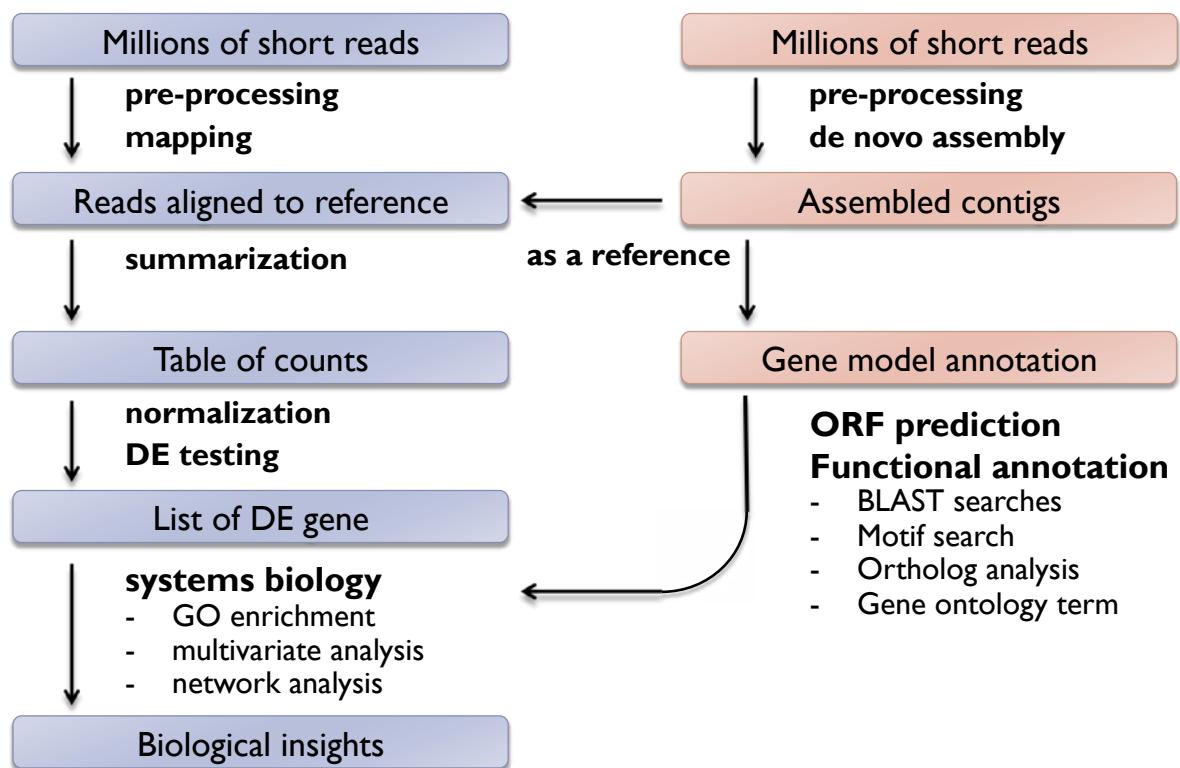
edgeR

- ▶ input: count data (not RPKM)
- ▶ output: gene table with DE significance statistics (FDR)

(example)

```
$ R
> library(edgeR)                      #load edgeR library
> dat <- read.delim("count_data.txt", ...) #import count table to R
> group <- c(rep("M", 3), rep("H", 3))   #assign groups
> D <- DGEList(dat, group=group)         #import data to edgeR
> D <- calcNormFactors(D)               #normalization (TMM)
> D <- estimateCommonDisp(D)           #estimate common dispersion
> D <- estimateTagwiseDisp(D, ...)     #estimate tagwise dispersion
> de <- exactTest(D, pair=c("M", "H")) #DE test
> topTags(de)
Comparison of groups: H-M
      logConc    logFC      P.Value        FDR
AT5G48430 -15.36821 6.255498 9.919041e-12 2.600872e-07
AT5G31702 -15.88641 5.662522 3.637593e-10 4.083773e-06
AT3G55150 -17.01537 5.870635 4.672331e-10 4.083773e-06
...
```

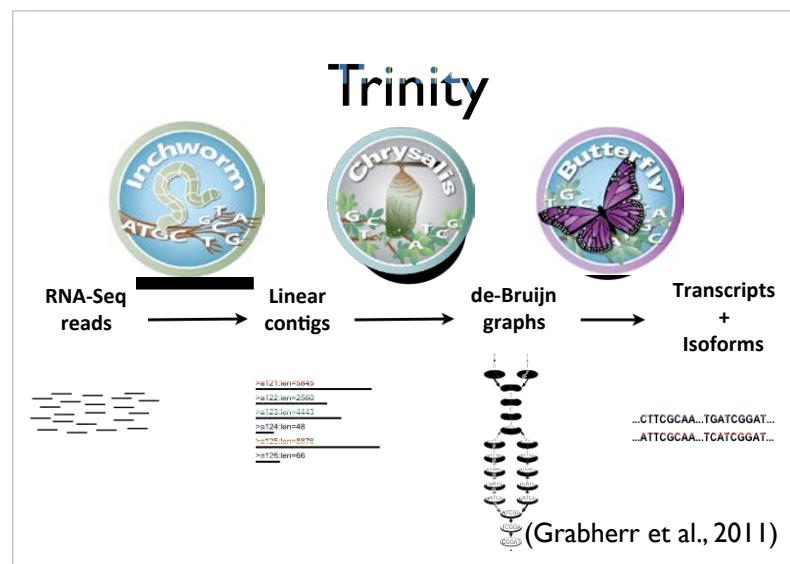
RNA-seq analysis pipeline (*de novo* strategy)



de novo assemblers of RNA-seq

De novo assemblers use reads to assemble transcripts directly, which does not depend on a reference genome.

- ▶ Trinity
- ▶ Oases
- ▶ TransAbyss
- ▶ EBARDenovo
- ▶ ...



<http://trinityrnaseq.sourceforge.net/>

Trinity example

- ▶ Input: Illumina short reads in FASTQ | FASTA format
- ▶ Output: assembled contigs in FASTA format

```
# prepare input reads
$ cat *.R1.fastq > left_all.fq
$ cat *.R2.fastq > right_all.fq

# Run Trinity
$ Trinity --seqType fq --left left_all.fq --right right_all.fq --
CPU 8 --max_memory 20G
```

(Trinity is supported on only Linux)

optional

ORF prediction

- ▶ Special consideration in ORF prediction after de novo RNA-seq assembly
 - ▶ Sometimes partial: Start Met or terminal codon may be missing.
 - ▶ Ideally one ORF is present per contig, but erroneously joined contigs may include multiple ORFs.
 - ▶ Possible frame shifts.
 - ▶ Frame shifts do not occur so often in Illumina, while it happens very frequently in 454 and IonProton.

optional

Functional Annotation of Predicted ORFs

- ▶ **BLAST**
 - ▶ NCBI NR (or UniProt)
 - ▶ species of interest (model organisms, close relatives etc)
 - ▶ specific DB (SwissProt, rRNA DB, CEGMA etc)
 - ▶ self (assembly v.s. assembly)
- ▶ **Motif search**
 - ▶ Pfam, SignalP etc.
- ▶ **Ortholog analysis**
 - ▶ vs model organism
 - ▶ ortholog database (OrthoDB, eggNOG, OrthoMCL etc)
 - ▶ close relatives
- ▶ **Gene Ontology term assignment**

optional

Quick annotation by BLASTX

- ▶ **Query:** assembled contigs
(nucleotide sequences in multi-fasta format)
- ▶ **DB:** Protein sequences of a model organism

Format DB

```
$ makeblastdb -in protein.fa -dbtype prot
```

Search

```
$ blastx -query trinity_contigs -db protein.fa \
-num_threads 8 -evalue 1.0e-8 -outfmt 0 > blastxout.txt
```

optional

Let's try BLASTX

- ▶ Query: minimouse_mRNA.fa
- ▶ DB: human.protein.faa (human RefSeq protein)

I. Format DB

```
$ makeblastdb -in human.protein.faa -dbtype prot -parse_seqids
```

2. Search

```
$ blastx -query minimouse_mRNA.fa -db human.protein.faa \
-num_threads 8 -evalue 1.0e-8 -outfmt 0 > blastxout.txt
```

```
$ blastx -query minimouse_mRNA.fa -db human.protein.faa \
-num_threads 8 -evalue 1.0e-8 -outfmt 7 > blastxout.tab
```

多変量解析

(特徴空間分割・次元圧縮)

佐藤昌直

モチベーション:

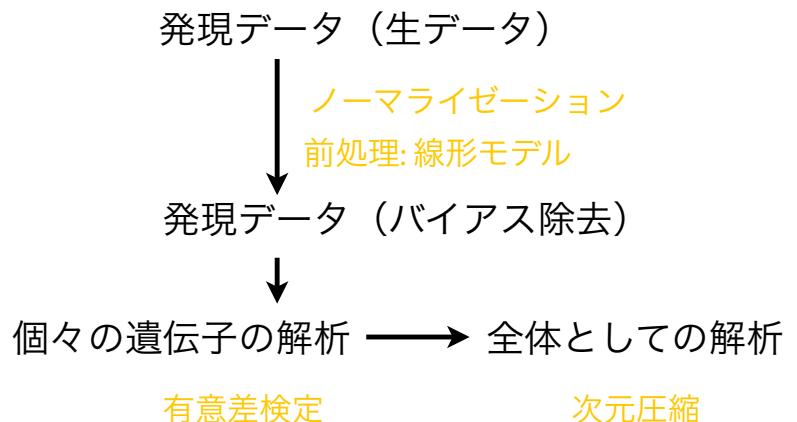
多次元(例: 多パラメーター)をより少ない指標を使って理解する



N個のサンプルをM個($M < N$)のグループに分類する

→ 人間が新たな解釈を与える

解析の流れ



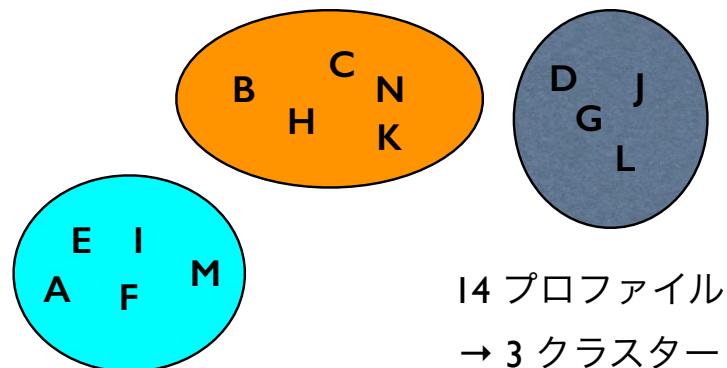
下記のデータセットに含まれる数値を俯瞰してみましょう。データの特徴を読み取れるでしょうか？

```

inputMatrix<- read.delim("~/data/MS/Sato_A_thaliana-P_syri
gae_arvRpt2_6h_expRatio_small.txt", header=TRUE, row.names=1
)
head(inputMatrix) #読み込みデータの一部を表示
image(t(inputMatrix)) #カラーコードによって可視化
heatmap(as.matrix(inputMatrix)) #階層クラスタリングで解析し、簡易
表示
  
```

この高次元（多パラメーター）
の問題をどう扱うか？

I. クラスタリングによる分類



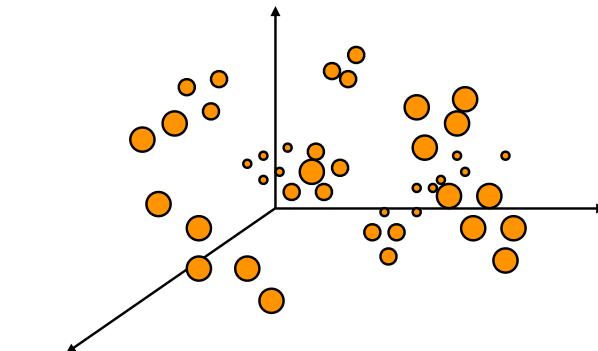
多変量解析のポイント

教師有りか無しか
(supervised or unsupervised) ?

どのような距離行列を使うか？

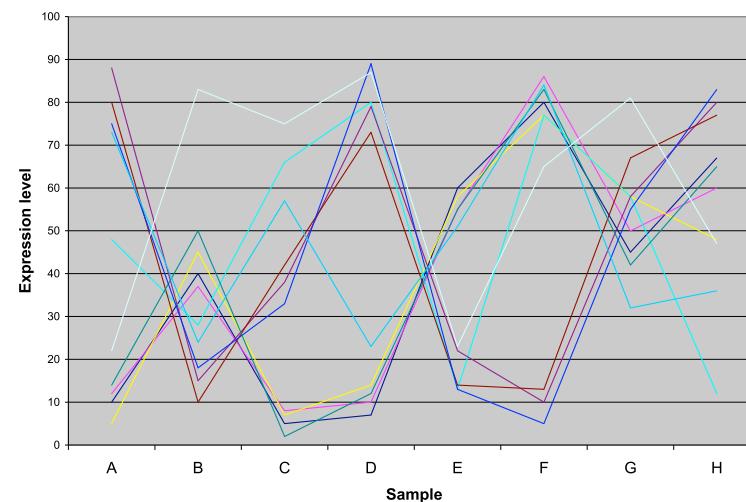
この高次元（多パラメーター）
の問題をどう扱うか？

2. パラメーター数を減らす

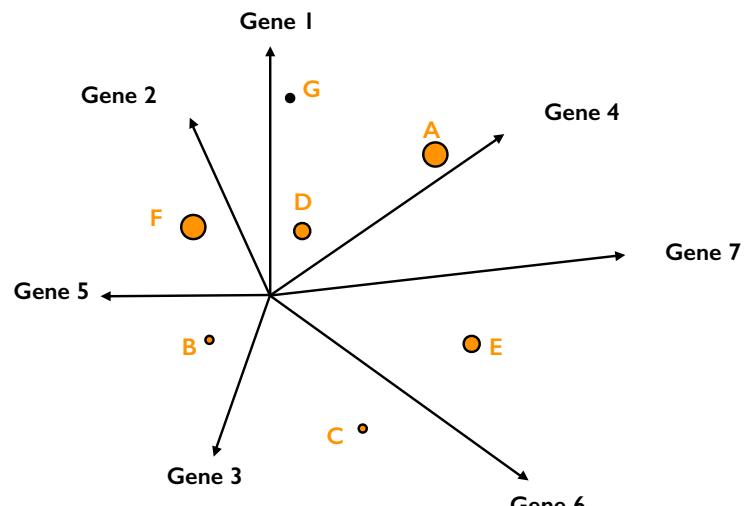


多パラメーター → 3パラメーター (次元圧縮)

トランスクリプトームデータの
ある一部について可視化してみる



7次元の遺伝子発現データセット



コンピューターにどうデータを渡せば
この問題をどう扱えるか？

人間

遺伝子発現プロファイル間の
パターンの比較

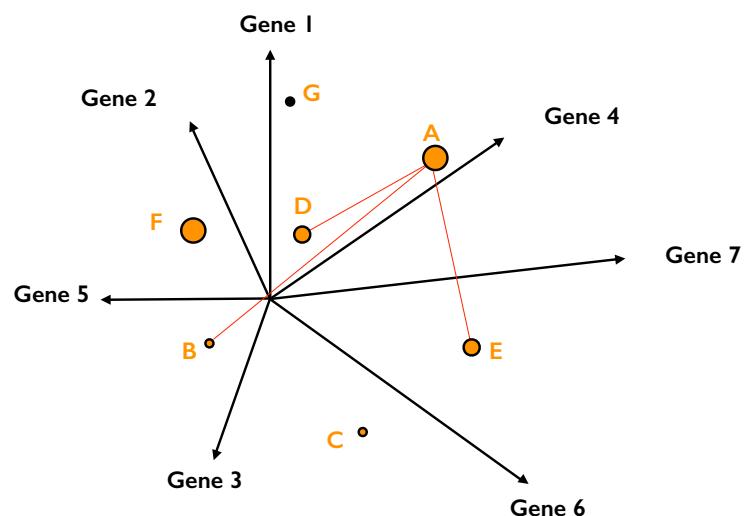
問題の定義

(生物学の問題を数学の問題に置き換える)

コンピューター

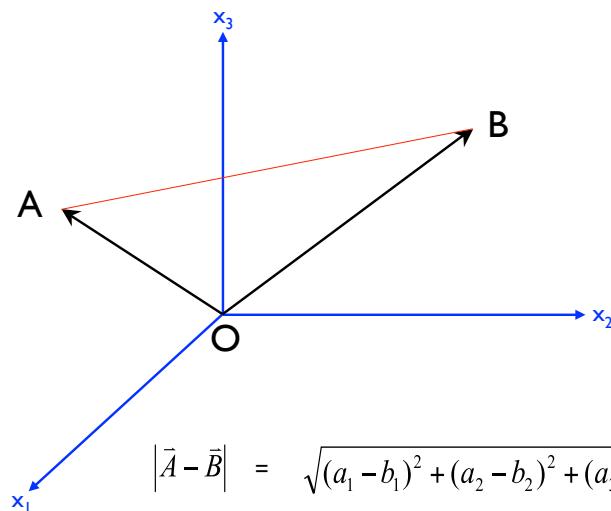
ある次元の空間における
データポイントの分布の比較

7遺伝子の発現プロファイル間の類似性は
7次元空間での距離によって決まる

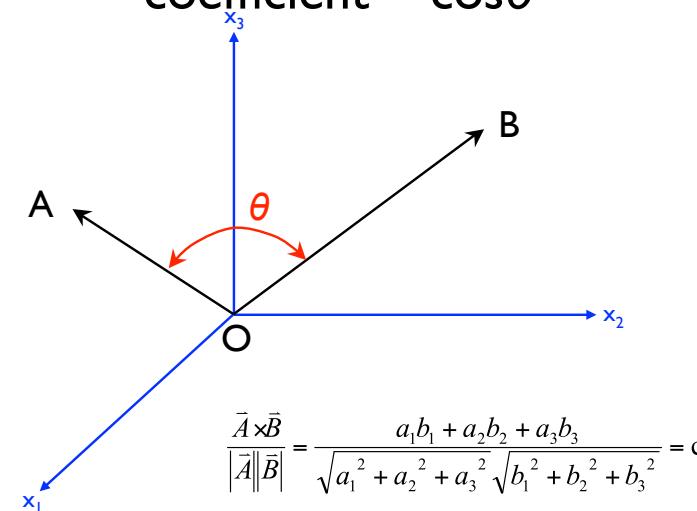


距離の基準にするか？
距離尺度

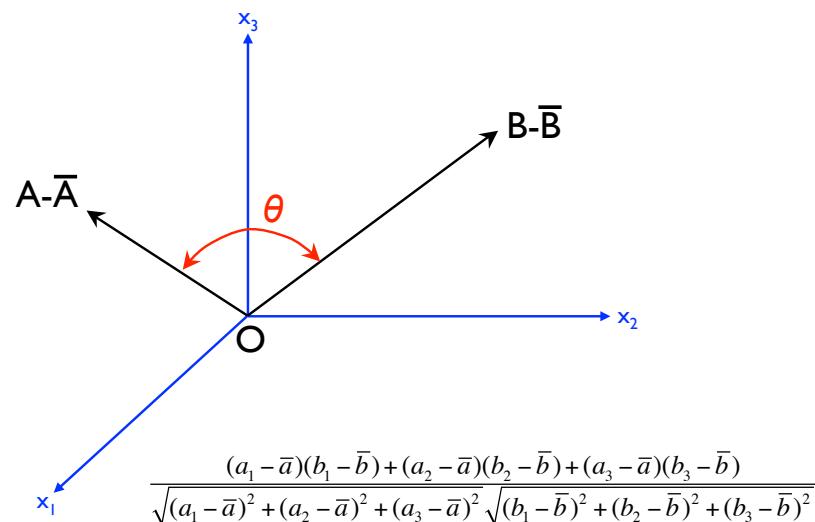
ユークリッド距離



Uncentered Pearson correlation coefficient = $\cos\theta$

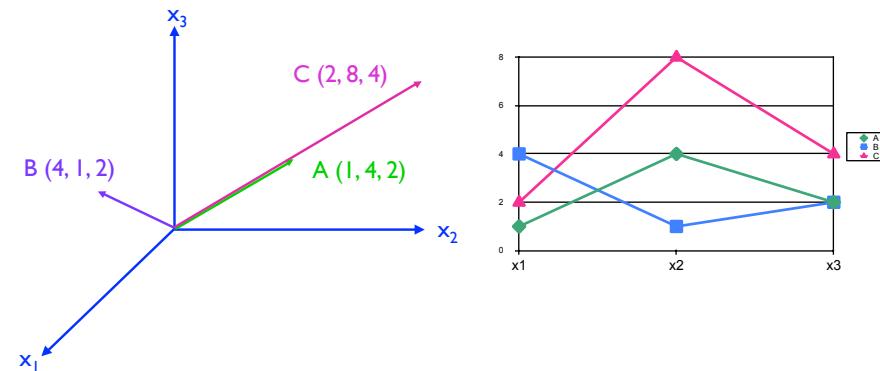


相関係数 Pearson correlation coefficient



遺伝子発現プロファイルの形と大きさ

- 形: ベクトルの方向
- 大きさ: ベクトルのサイズ



どの距離係数を使うか？

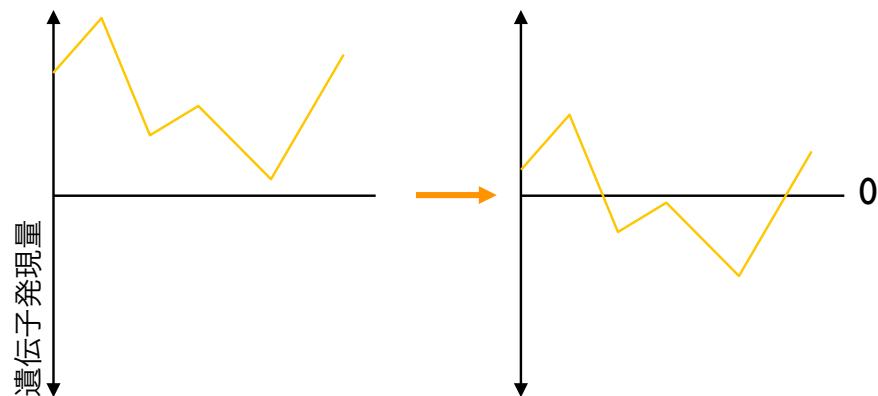
- どんなプロファイルを同じプロファイルと定義するか？
- 距離係数計算の背後にあるものを意識して選択する。

距離係数計算の過程には

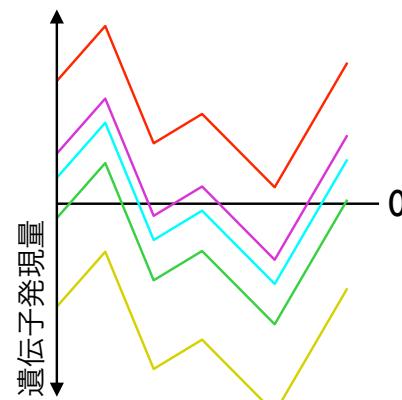
- **Centering:** 平均値をゼロにする
- **Normalization***: ベクトルの大きさを1にする

* トランскryptオームデータのnormalizationとは異なることに注意

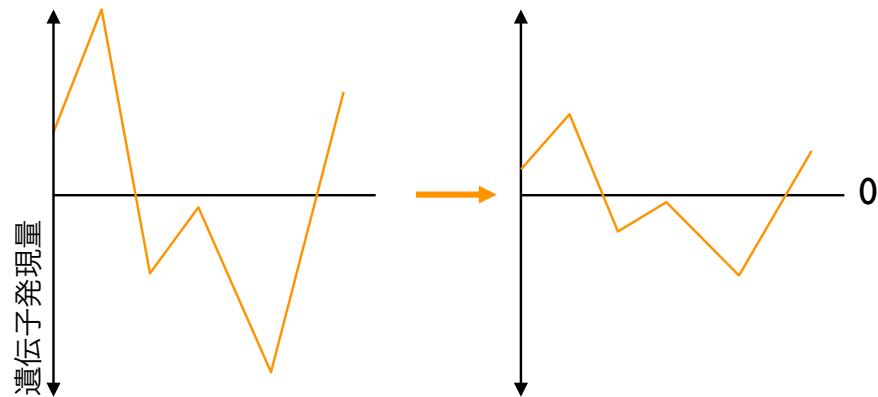
Centering



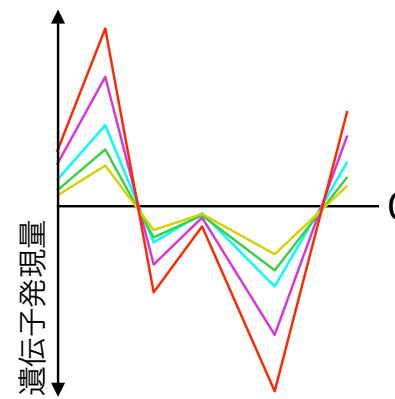
これらはcentering後は
全く同じプロファイルになる



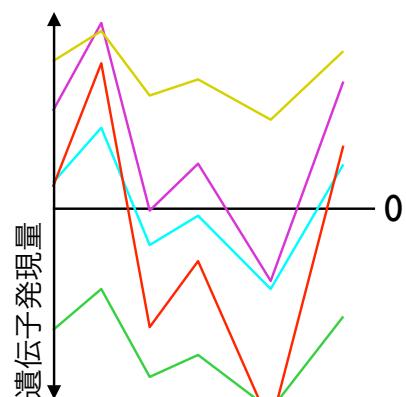
Normalization



これらはnormalization後は
全く同じプロファイルになる



これらはcentering, normalization後は
全く同じプロファイルになる



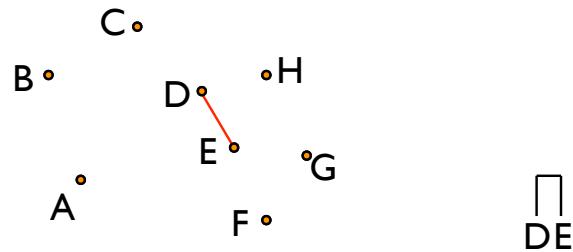
多変量解析における注意点

方法依存的に抽出される特徴なので、
どんな特徴を認識したいのか注意が必要

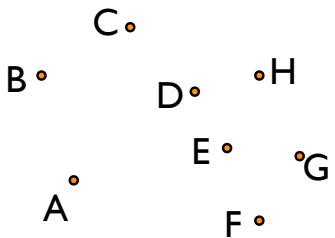
多変量解析の実際

階層クラスタリング

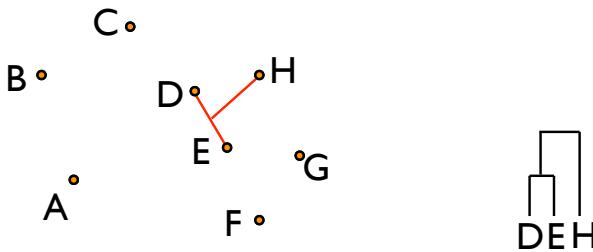
Agglomerative hierarchical clustering



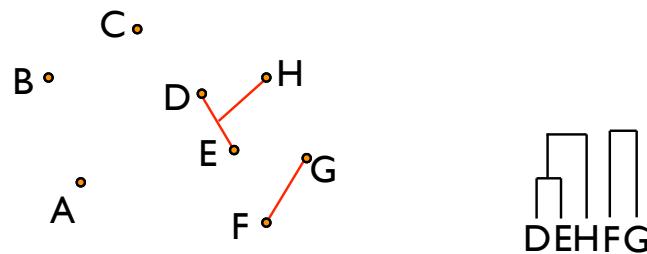
Agglomerative hierarchical clustering



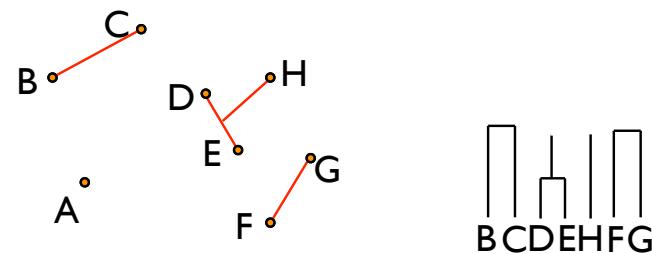
Agglomerative hierarchical clustering



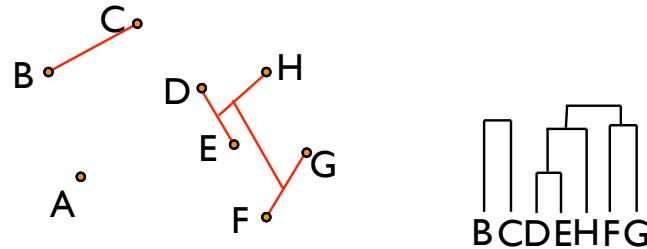
Agglomerative hierarchical clustering



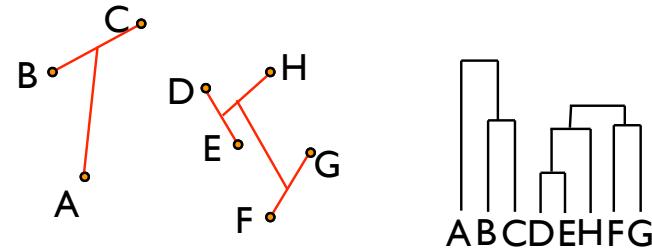
Agglomerative hierarchical clustering



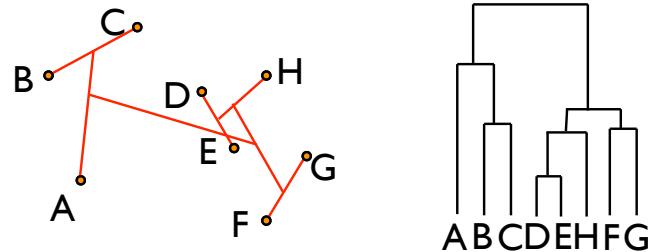
Agglomerative hierarchical clustering



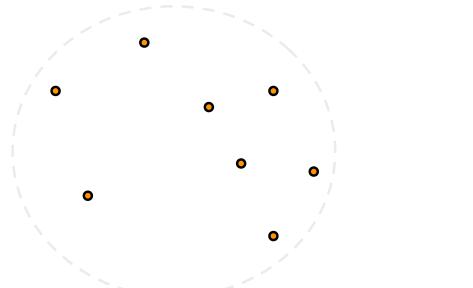
Agglomerative hierarchical clustering



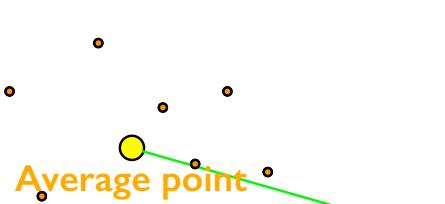
Agglomerative hierarchical clustering



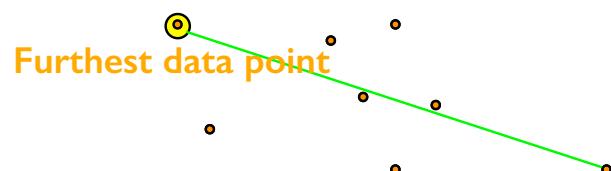
クラスター定義手法



Average linkage



Complete linkage



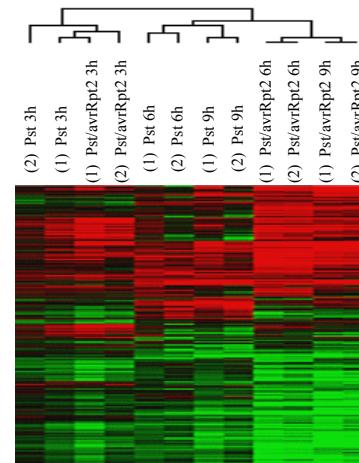
Single linkage



階層クラスタリングの欠点

- Bottom-up: 非常に「手順」依存性
- 一つの距離のみを指標としたクラスタリング

階層クラスタリングの利点



- クラスター化してより少数のカテゴリーを示す
- 人間が認識可能なパターンを示す

「手順依存的」な方法の欠点を補うには？

- 偶然、観察されているクラスターを推定する
 - 同じ手順を繰り返す
 - クロスバリデーション

クロスバリデーション

- あるクラスターは必然か偶然か？
- leave-one out validation: サンプルを一つ抜いてクラスタリングしてみる
- 少数の特定遺伝子がクラスタリングに影響していないか？
- Bootstrap: 遺伝子サブセットでクラスリングを繰り返してみる

主成分分析

多変量解析(I)のまとめ

**教師有りか無しか
(supervised or unsupervised) ?**

- 事前情報、前提はあるか？
- ある場合はk-means法などの利用を検討

どのような距離行列を使うか？

- プロファイルの大きさ
- プロファイルの角度 など

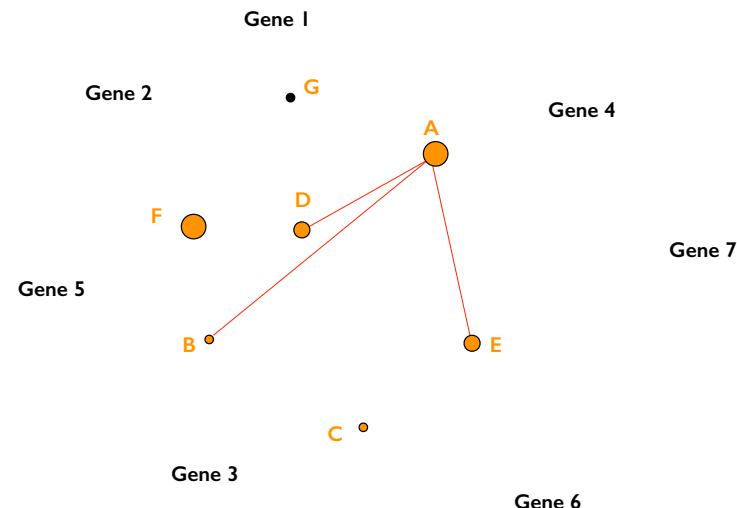
主成分分析とは？

モチベーション:

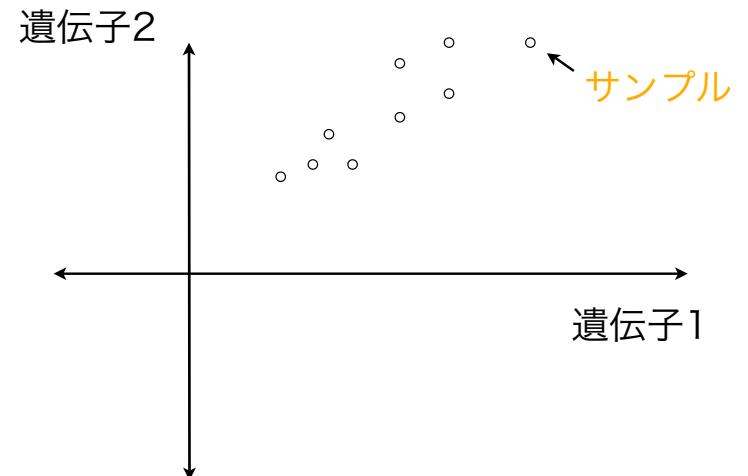
多数の遺伝子で構成される多次元データ
(サンプル) の中で相関のある遺伝子群を使って
新たな軸を作り、データを見直す

→ **人間が新たな解釈を与える**

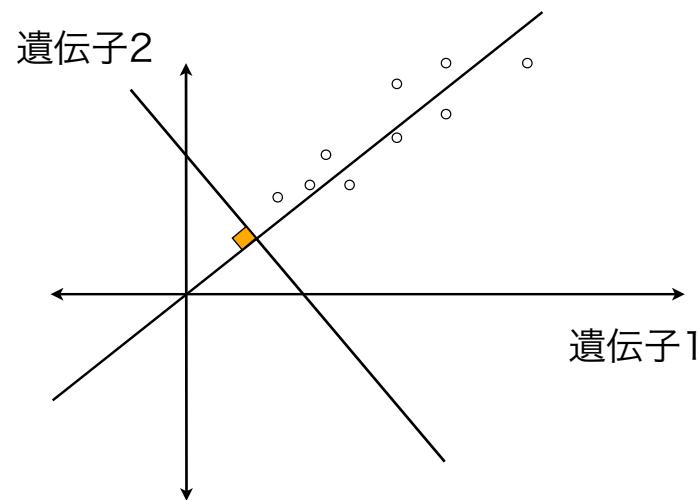
階層クラスタリング、k-means法:
プロファイル間の類似性は
空間での1つの距離によって決まる



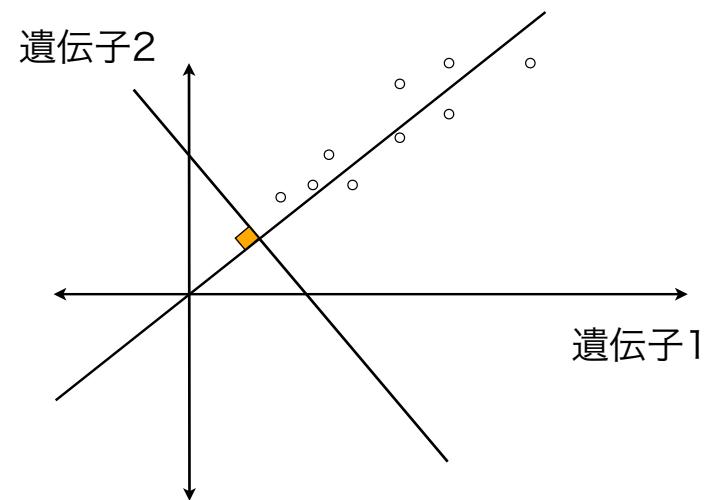
PCAは何をするのか？



PCAは何をするのか？



PCAは何をするのか？



PCAの概略(2次元)

1. 各サンプル ($1..n$) の観察値(x_n, y_n)を

$$\begin{aligned} u_n &= a_1 x_n + b_1 y_n \\ v_n &= a_2 x_n + b_2 y_n \end{aligned}$$

とおく

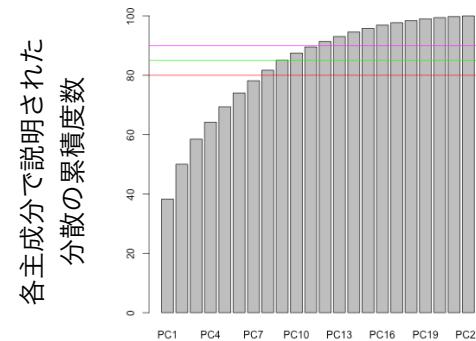
2. $a^2 + b^2 = 1$, u と v の相関係数0という制約の下でこれを解いて a_n, b_n を求める。

PCAで得られる重要な統計量

- 寄与率
- 因子負荷量
- 主成分得点

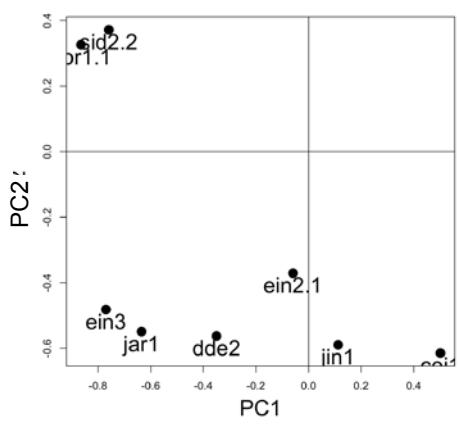
寄与率

- 各主成分が説明する分散の割合



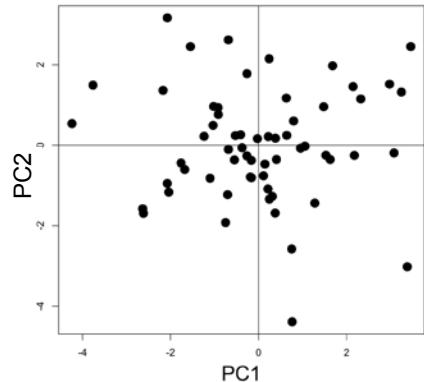
負荷量 loadings

- 得られた主成分と元データのパラメーターの相関
- 各パラメーターがもとのデータの情報をどれだけ有するか



主成分得点 scores

- 各パラメーターの値を各主成分について標準化したもの



標準化: 平均0, SD=1

主成分分析(まとめ)

- 主成分分析はデータの分散を説明する新たな軸を計算する方法
 - 寄与率
 - 因子負荷量
 - 主成分得点

ポイント

- デフォルトのprincompでは返り値 loadingsは因子負荷量ではない。
- 相関を使うか、分散共分散行列を使うか

多次元尺度構成法

Multi-dimensional scaling(MDS),
Principle coordinate analysis

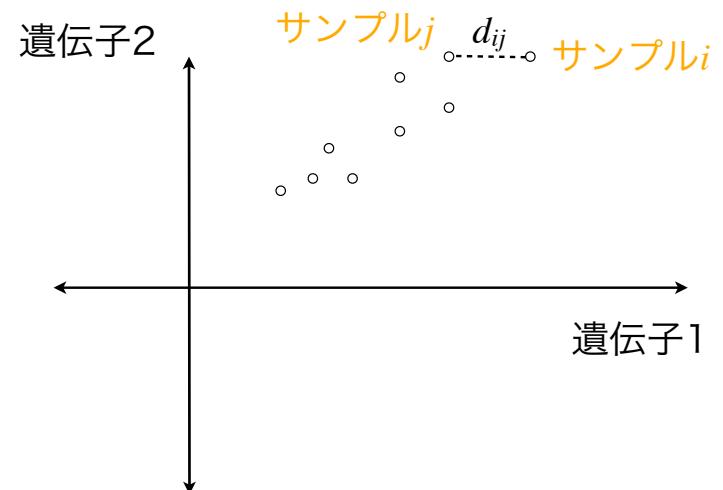
多次元尺度構成法とは？

モチベーション(PCAと同様) :

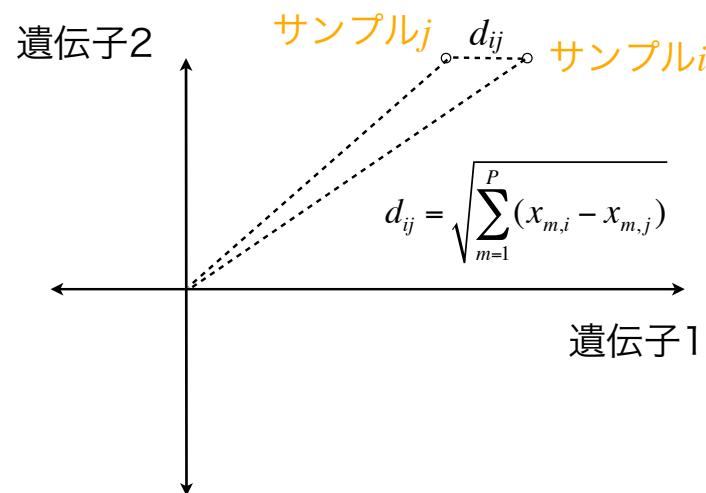
多数の遺伝子で構成される多次元データ（サンプル）の中で各サンプル間の違いを表現する座標を作る

→ 人間が新たな解釈を与える

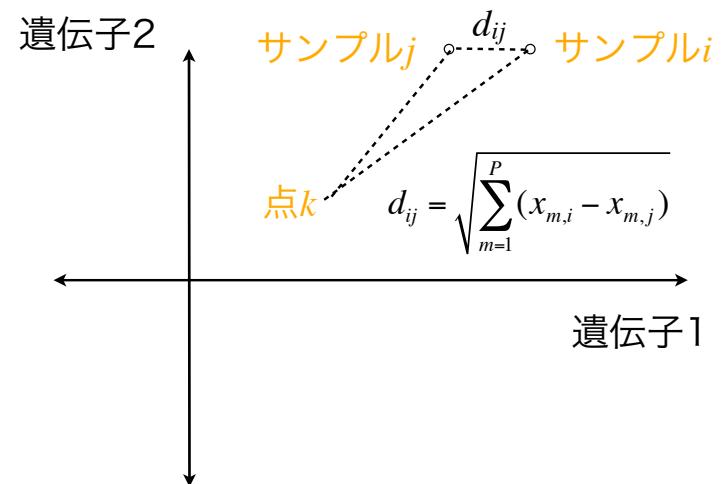
MDSは何をするのか？



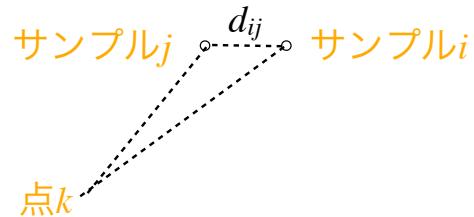
サンプル間の距離をまず計算する



この定理はサンプル*i,j*に対し、どこを原点（点*k*）としても成り立つ

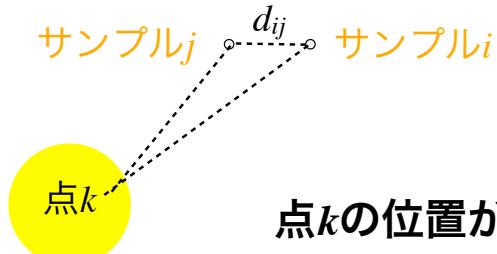


この定理はサンプル*i,j*に対し、どこを原点（点*k*）としても成り立つ



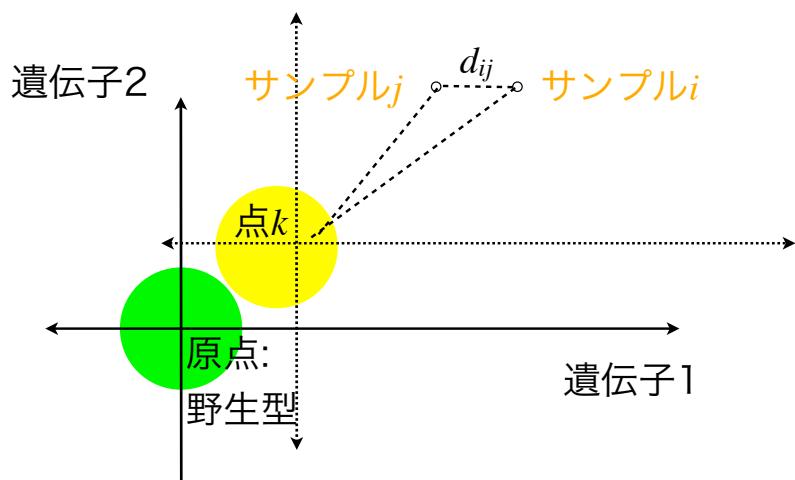
$$d_{ij}^2 = d_{ik}^2 + d_{jk}^2 - 2d_{ik}d_{jk} \cos\theta$$

この定理はサンプル*i,j*に対し、どこを原点（点*k*）としても成り立つ



点*k*の位置が意味を持つことはないのか？

例:入力データが野生型・変異体プロファイルの比であったら？



多変量解析(2)のまとめ

PCA/MDS

- データをそれがもつ次元に分解して評価・可視化する
- 重心の置き方に違い: 入力データをどのように前処理するか

研究の目的、実験デザイン、多変量解析

目的

- ・何を知りたいか
- 明確に
- ・実施の制約
 - ・予算
 - ・時間、労力

実験デザイン

- ・線形モデル
- ・比較、因子
- ・検出力

多変量解析

- ・入力データ前処理
- ・距離尺度
- ・アルゴリズム

多変量解析をもう一步進めて:

人間の解釈をアシストするデータ取得を心がける

多変量解析の枠組み

モチベーション:

多次元(例: 多パラメーター)をより少ない指標を使って理解する



N個のサンプルをM個($M < N$)のグループに分類する

→ 人間が新たな解釈を与える

コントロール、
指標サンプルは
含められるか?

今回のトレーニングコースで 扱わなかった重要項目

- ・確率分布
- ・回帰、相関
- ・線形モデルにおける交互作用
- ・非線形クラスタリング・次元圧縮
 - ・self-organization mapなど