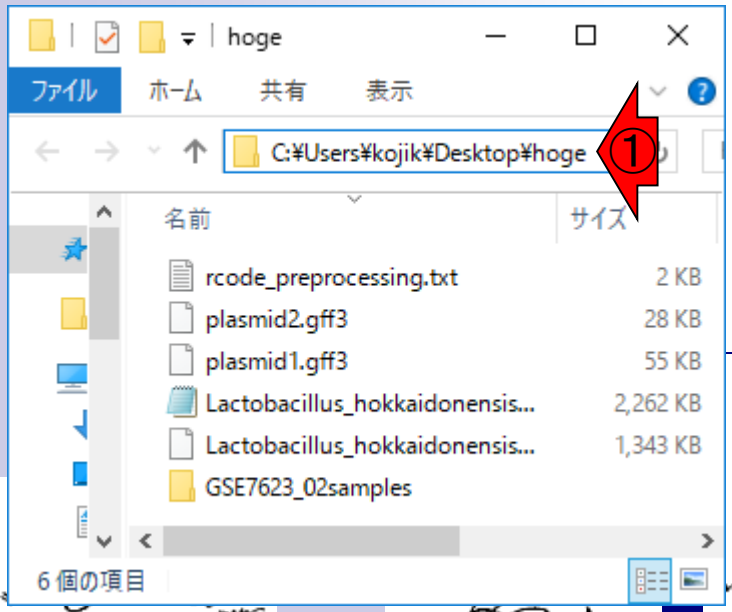


①持ち込みPCのヒトは後ろにある
USBメモリ中のhogeフォルダをデスクトップにコピーしておいてください

機能ゲノム学第1回

大学院農学生命科学研究科
アグリバイオインフォマティクス教育研究プログラム
門田幸二（かどた　こうじ）
kadota@iu.a.u-tokyo.ac.jp
http://www.iu.a.u-tokyo.ac.jp/~kadota/

講義予定

細胞中で発現している全転写物（トランスクリプトーム）の解析技術は、マイクロアレイから次世代シーケンサ（RNA-seq）に移行しつつあります。しかしRNA-seq解析の多くは、マイクロアレイの知識を前提としています。本科目では、マイクロアレイデータを主な例として、各種トランスクリプトーム解析手法について解説します。また、Rのスキルアップを目指します

- 第1回（2017年05月08日）
  - 原理、各種データベース、生データ取得
  - 教科書の1.2節、2.2節周辺
- 第2回（2017年05月15日）
  - 数値行列作成、クラスタリング、実験デザイン
  - 教科書の3.2節周辺
- 第3回（2017年05月22日）
  - 発現変動解析（多重比較問題とFDR）、各種プロット（M-A plot）
  - 教科書の3.2節と4.2節周辺
- 第4回（2017年05月29日）
  - 発現変動解析（デザイン行列や3群間比較）
  - 機能解析（Gene Ontology解析やパスウェイ解析）

Contents

- トランスクリプトーム解析技術の原理や特徴
  - マイクロアレイとRNA-seq、遺伝子≠転写物
  - 様々な解析目的、トランスクリプトーム（転写物）配列取得
  - アノテーションファイルの読み込み（Rで転写物配列取得のイントロ）
  - Rで転写物配列取得（アノテーションファイルとゲノム情報ファイルから）
  - マイクロアレイの特徴
  - 発現データベース（DB）
- 発現DBからのプローブレベルデータ取得
  - Affymetrix GeneChip
  - R経由（教科書の§2.2.1）
- Affymetrix GeneChipデータ前処理法を実行

イントロダクション

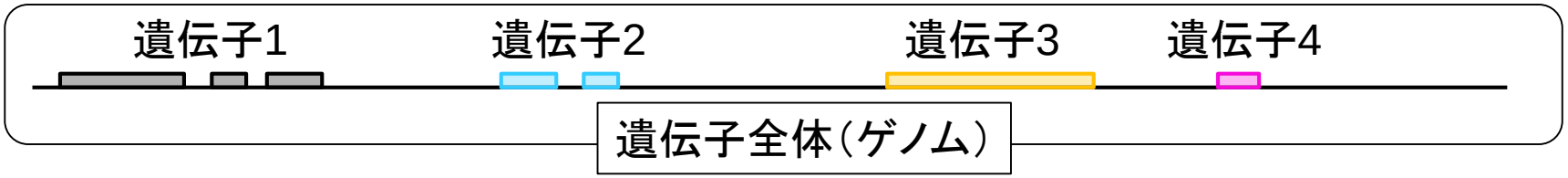
調べたいサンプルでゲノム中のどの領域が、どういう時期に、どの程度転写されている(発現している)かを調べるのがトランスクリプトーム解析。**遺伝子発現解析**や**発現解析**は、トランスクリプトーム解析の一部

- トランスクリプトームとは
  - ある特定の状態の組織や細胞中に存在する全RNA(転写物、transcripts)の総体
- 様々なトランスクリプトーム解析技術
  - **マイクロアレイ**(配列既知の生物種)
    - Affymetrix GeneChip、Illumina BeadArrayなど
  - 配列決定に基づく方法(配列未知でもよい)
    - EST、SAGE、CAGE、**RNA-seq**など

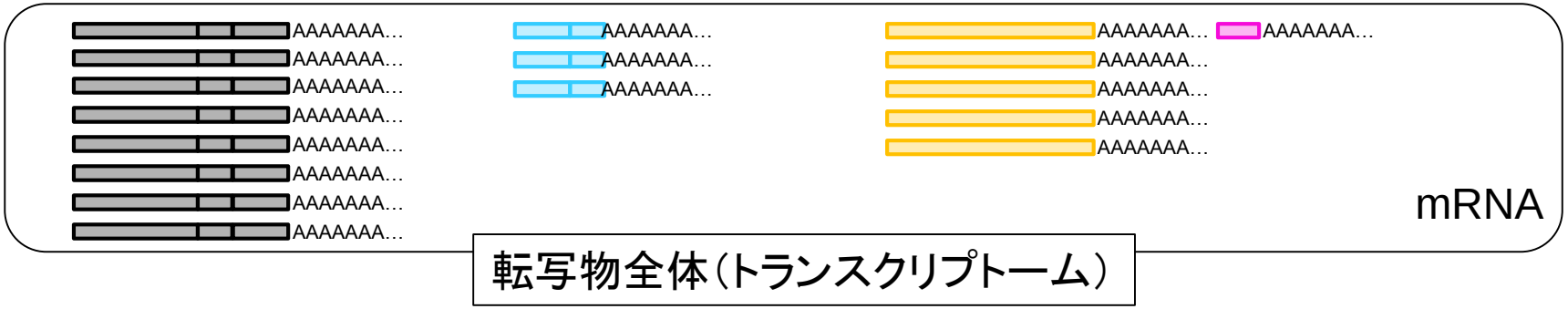
働いているRNAの種類や量を調べるのが目的

トランスクリプトーム解析

- ある状態のあるサンプル（例：目）のあるゲノムの領域



・どの染色体上のどの領域にどの遺伝子があるかは調べる個体（例：ヒト）が同じなら不変（目だろうが心臓だろうが…）

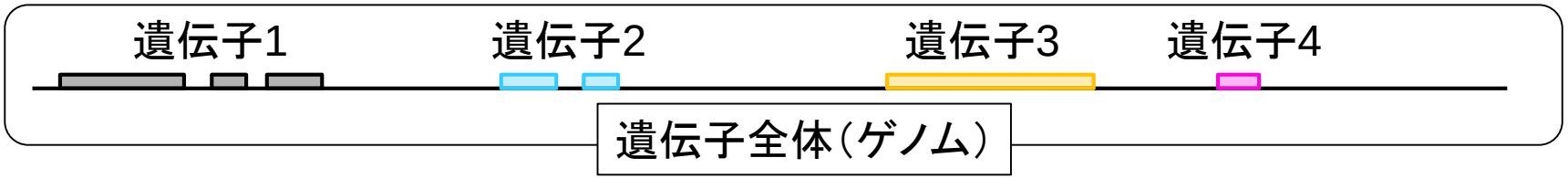


・遺伝子1は沢山転写されている（発現している）
・遺伝子4はごくわずかしか転写されてない
・…

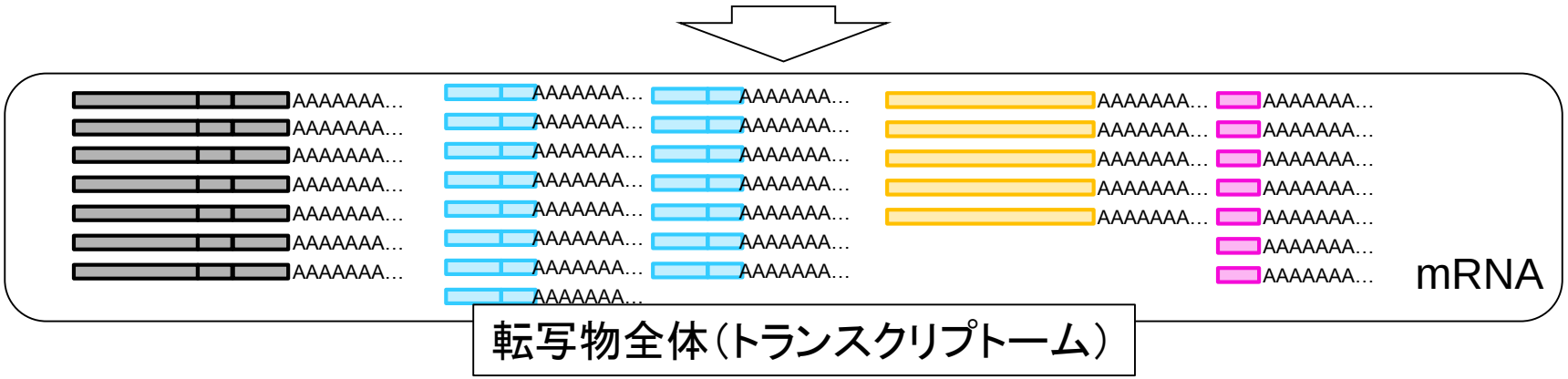
トランスクリプトーム解析

働いているRNAの種類や量を調べるのが目的

- ある状態のあるサンプル（例：目）のあるゲノムの領域



・どの染色体上のどの領域にどの遺伝子があるかは調べる個体（例：ヒト）が同じなら不変（目だろうが心臓だろうが…）

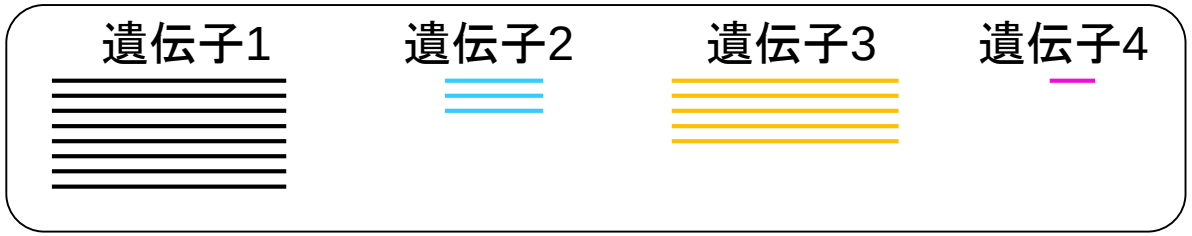


・光刺激に応答して発現亢進するのは遺伝子2と4

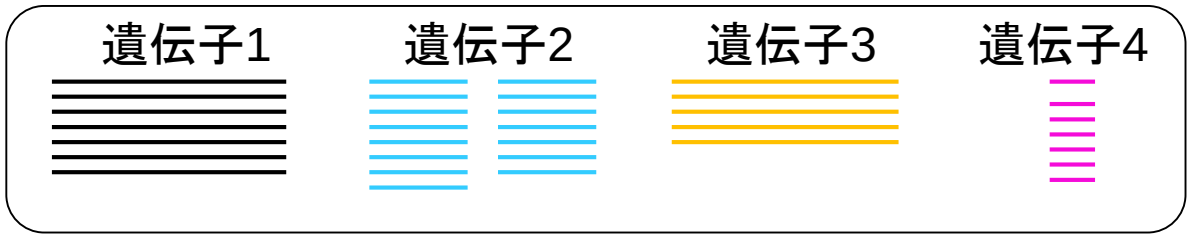
トランスクリプトーム解析

状態の異なる複数サンプルのデータを取得して解析するのが一般的。サンプル間比較。

- 光刺激前（T1）の目のトランスクリプトーム



- 光刺激後（T2）の目のトランスクリプトーム

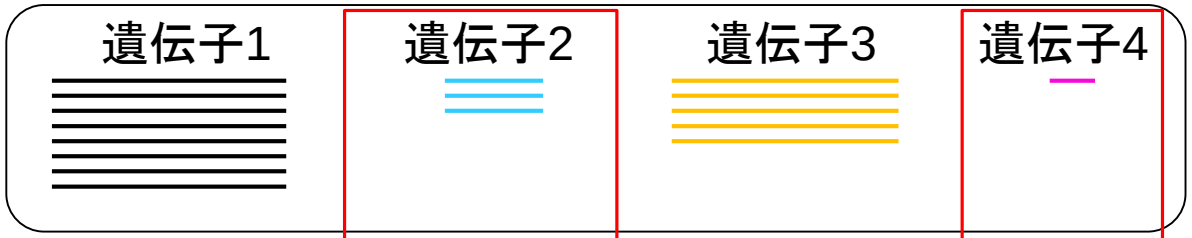


	T1	T2
遺伝子1	8	7
遺伝子2	3	15
遺伝子3	5	5
遺伝子4	1	7
…	…	…

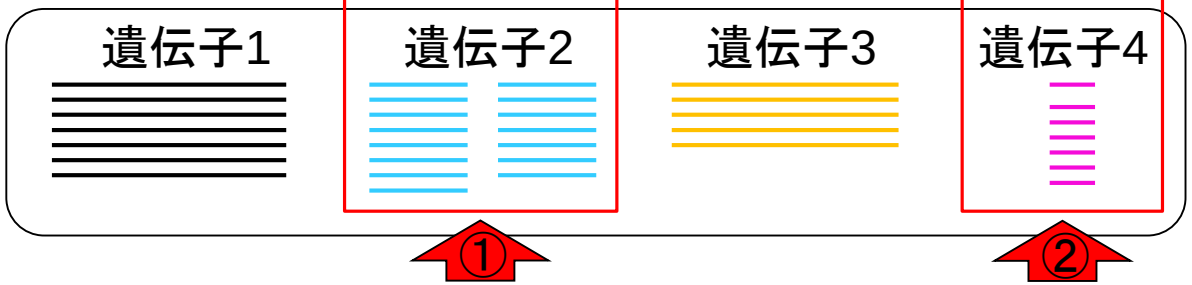
トランスクリプトーム解析

具体的な目的は、①や②の発現変動遺伝子同定など。

- 光刺激前（T1）の目のトランスクリプトーム



- 光刺激後（T2）の目のトランスクリプトーム



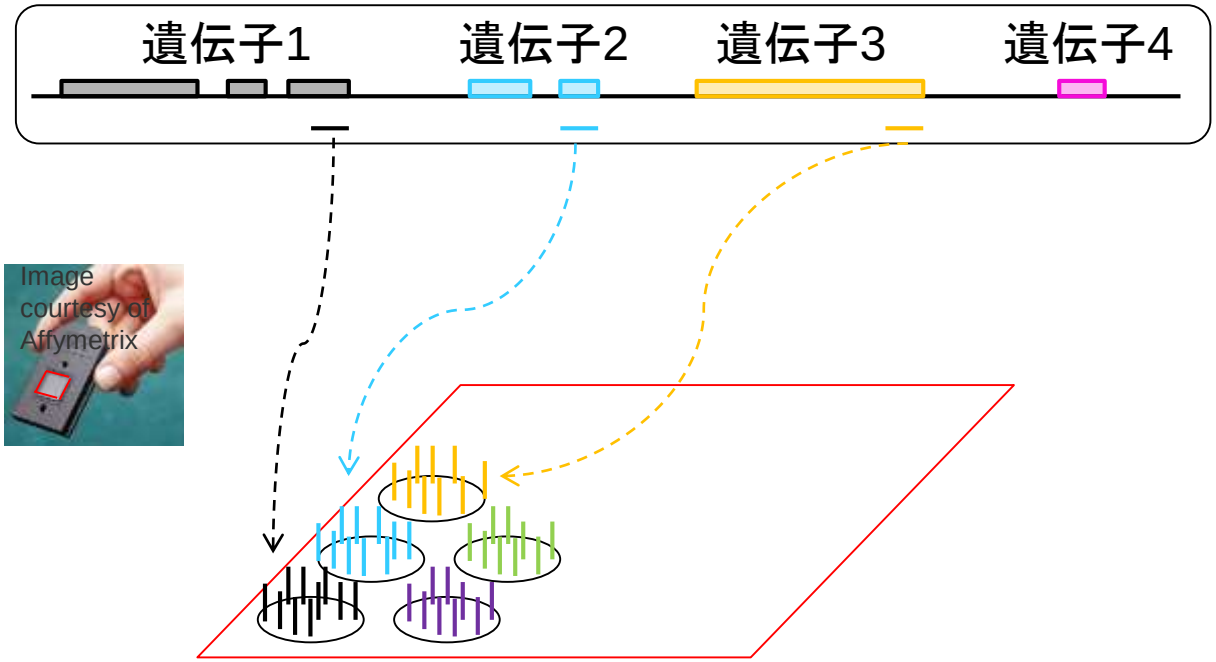
これがいわゆる「遺伝子発現行列」

	T1	T2
遺伝子1	8	7
遺伝子2	3	15
遺伝子3	5	5
遺伝子4	1	7
…	…	…

マイクロアレイ

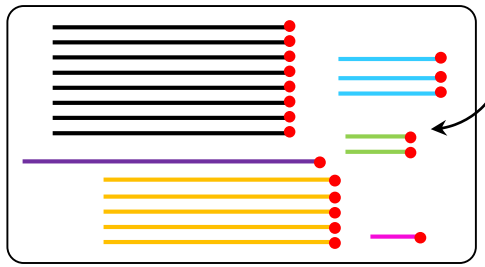
搭載遺伝子数や種類はメーカー次第。
遺伝子4など、搭載されていない遺伝子や未知遺伝子の発現情報は測定不可…

- よく研究されている生き物は多数の遺伝子（の配列情報）がわかっている

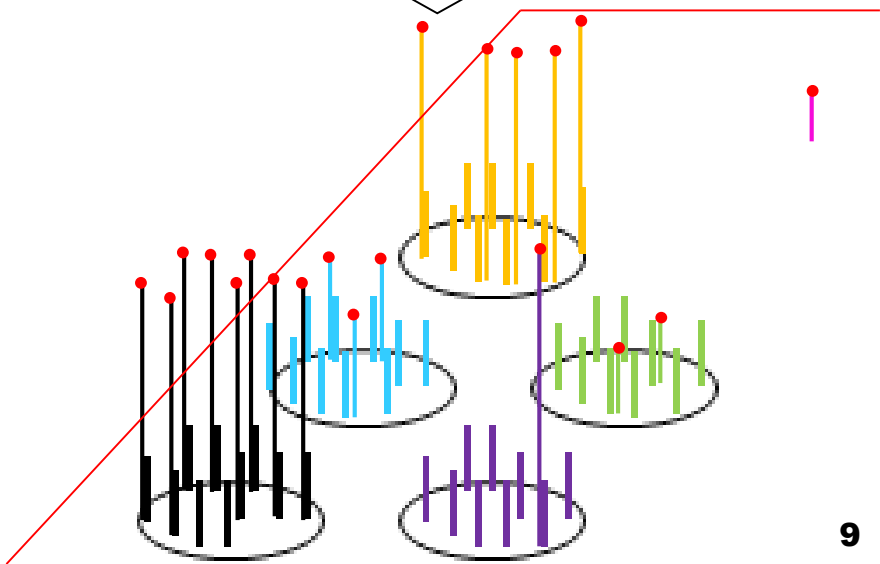


既知遺伝子（の配列の相補鎖）のプローブを搭載した”チップ”。12mm×12mm程度

光刺激前（T1）の目のトランスクリプトーム



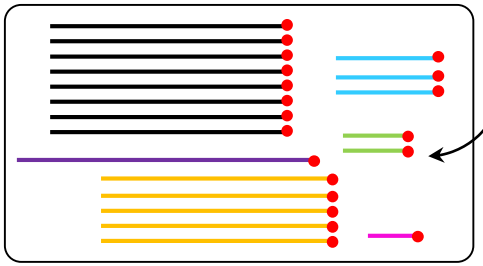
ハイブリダイゼーション



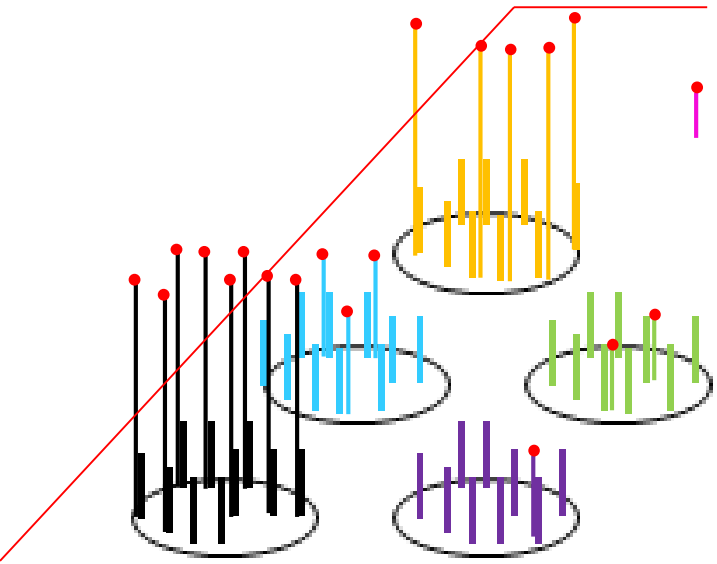
マイクロアレイ

光刺激前(T1)と光刺激後(T2)の状態の数値データを比較して、サンプル(状態)間で発現に差がある遺伝子(発現変動遺伝子;DEG)を同定

光刺激前(T1)の目のトランスクリプトーム



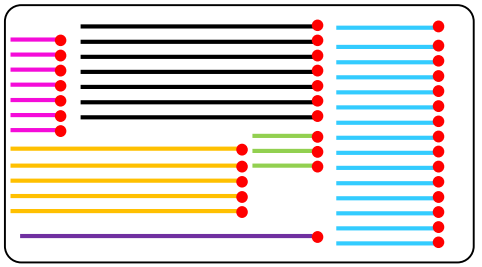
ハイブリダイゼーション



専用の検出器で各遺伝子に対応する領域の蛍光シグナル強度を測定

	T1
遺伝子1	8
遺伝子2	3
遺伝子3	5
遺伝子4	?
遺伝子5	…
…	…

光刺激後(T2)の目のトランスクリプトーム



ハイブリダイゼーションとシグナル検出

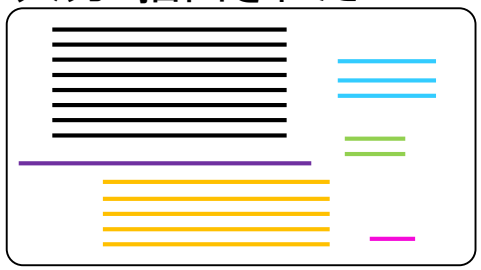
T2
7
15
5
?
…
…

データ解析

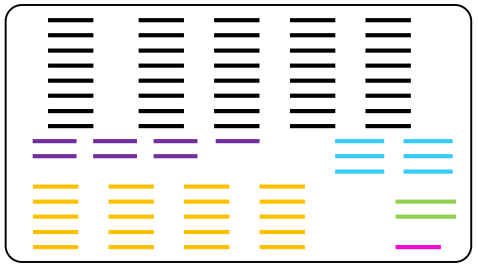
入力：サンプルのRNA
出力：大量塩基配列データ

RNA-seq

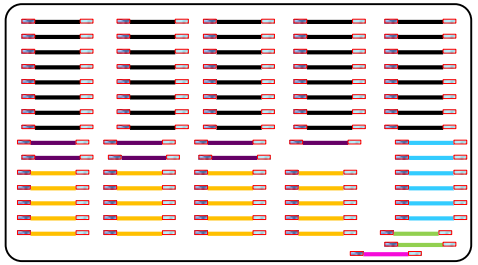
入力：抽出されたRNA



断片化

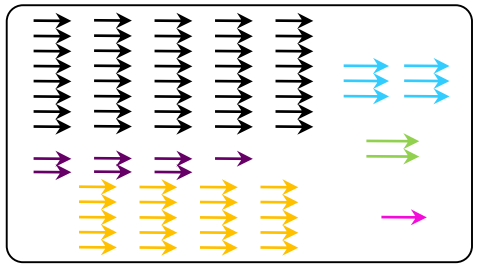


アダプター付加



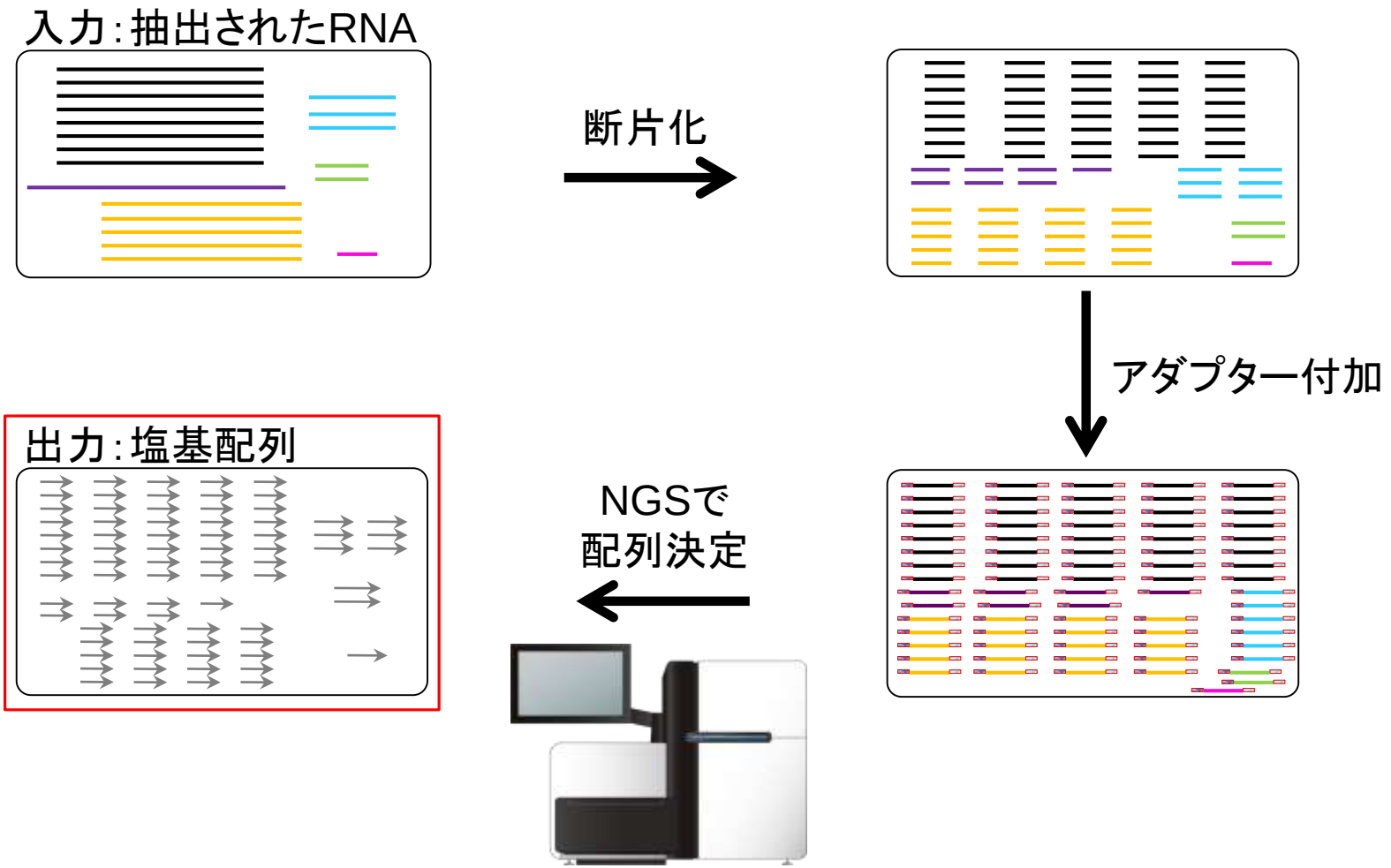
NGSで
配列決定

出力：塩基配列



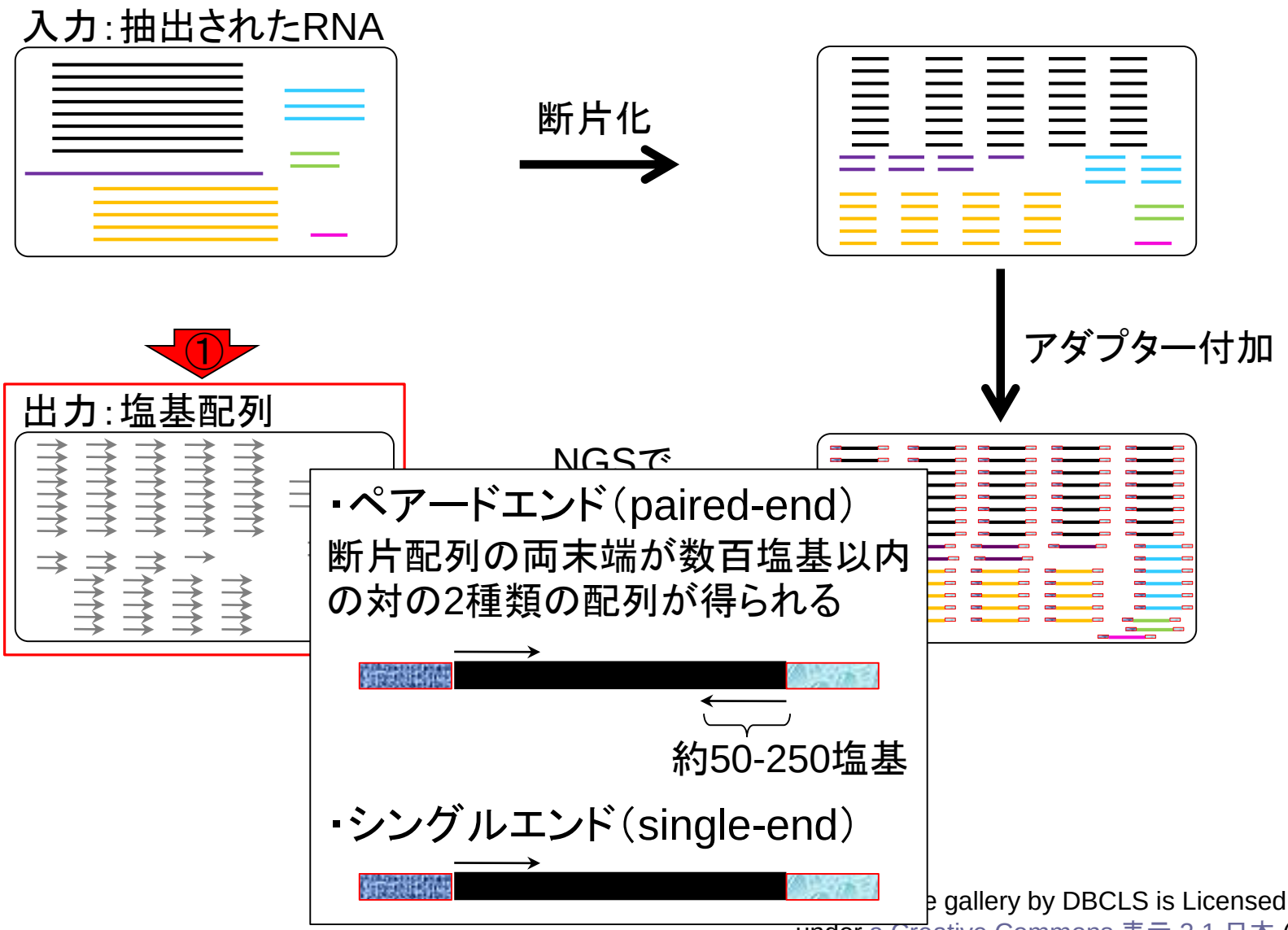
RNA-seq

NGSの出力は、リードと呼ばれる数百塩基程度の配列が延々と続く巨大なファイル。各矢印が1つのリードに相当。この段階では、まだどのリードがどの転写物由来かは不明（なので灰色一色）



RNA-seq

Illuminaの場合は、両側から読むpaired-endと片側のみ読むsingle-endの2つのやり方が存在する。①の出力イメージはsingle-endの場合

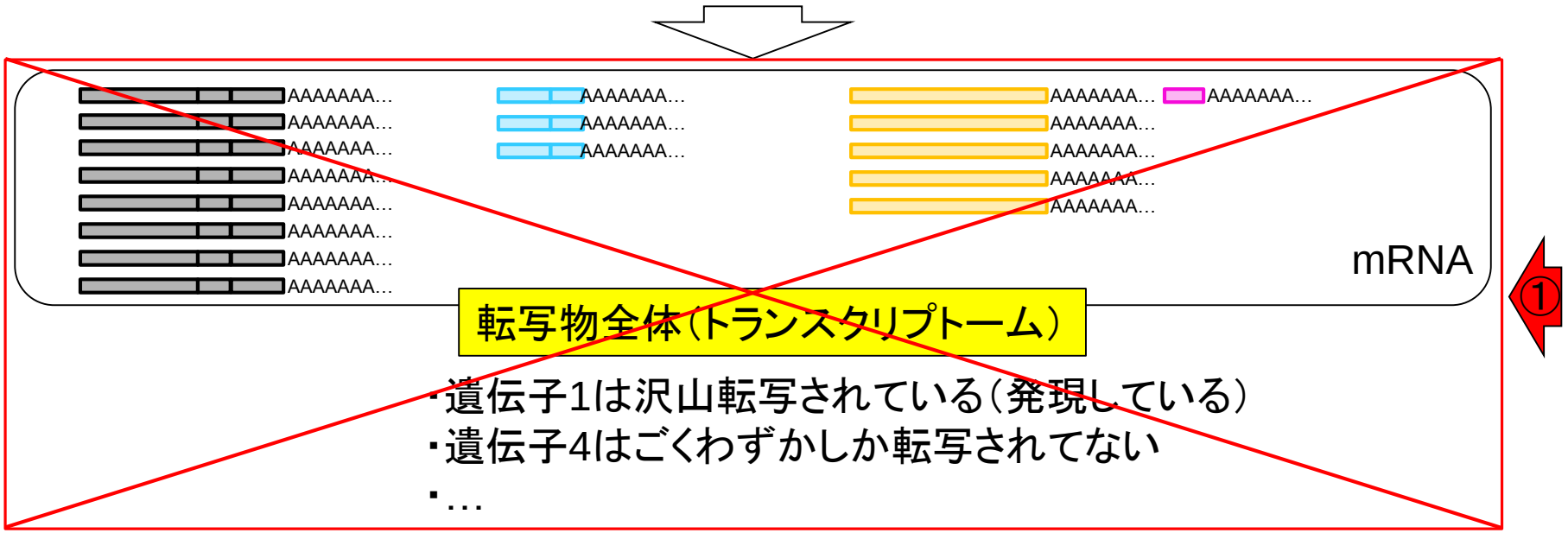


遺伝子 ≠ 転写物

①赤枠部分の表現は、本当は不正確。昔は実験機器の解像度が事実上**遺伝子レベル**だった。**遺伝子発現解析**という表現はその名残り

- ある状態のあるサンプル（例：目）のあるゲノムの領域

・どの染色体上のどの領域にどの遺伝子があるかは調べる個体（例：ヒト）が同じなら不変（目だろうが心臓だろうが…）



・遺伝子1は沢山転写されている（発現している）
・遺伝子4はごくわずかしか転写されてない
・…

遺伝子 ≠ 転写物

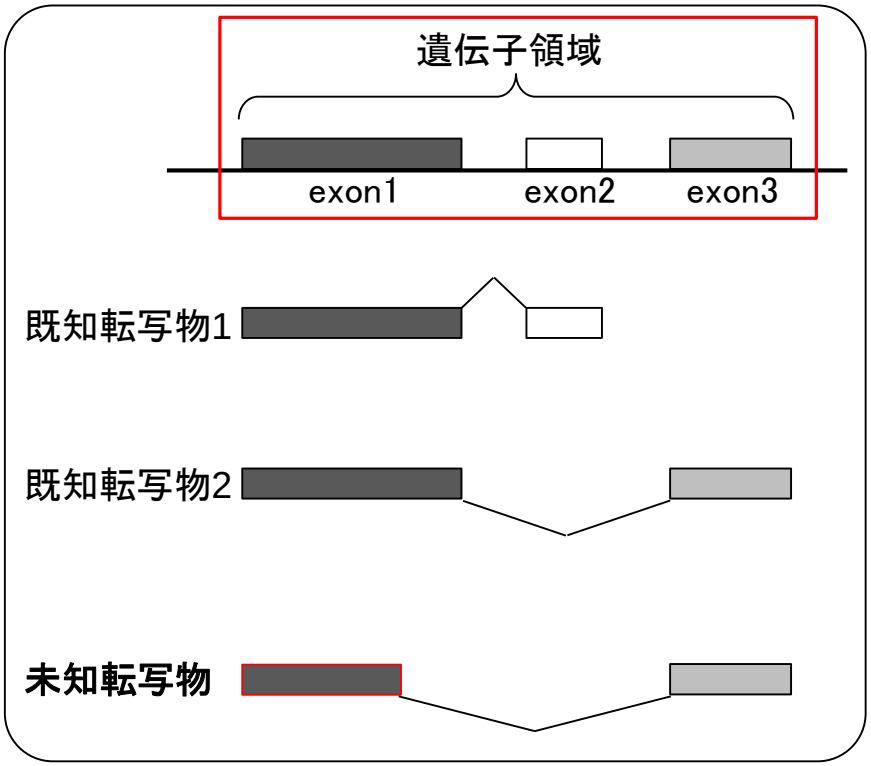
ある遺伝子領域から転写(transcription)されている転写物(transcript)は、1種類とは限らない

- ある状態のあるサンプル(例:目)のあるゲノムの領域

遺伝子 ≠ 転写物

ある遺伝子領域から転写(transcription)されている転写物(transcript)は、1種類とは限らない。
例えば、①遺伝子1の領域では、3種類の真の転写物が存在し、そのうち2種類は既知とする

- ある状態のあるサンプル(例:目)の

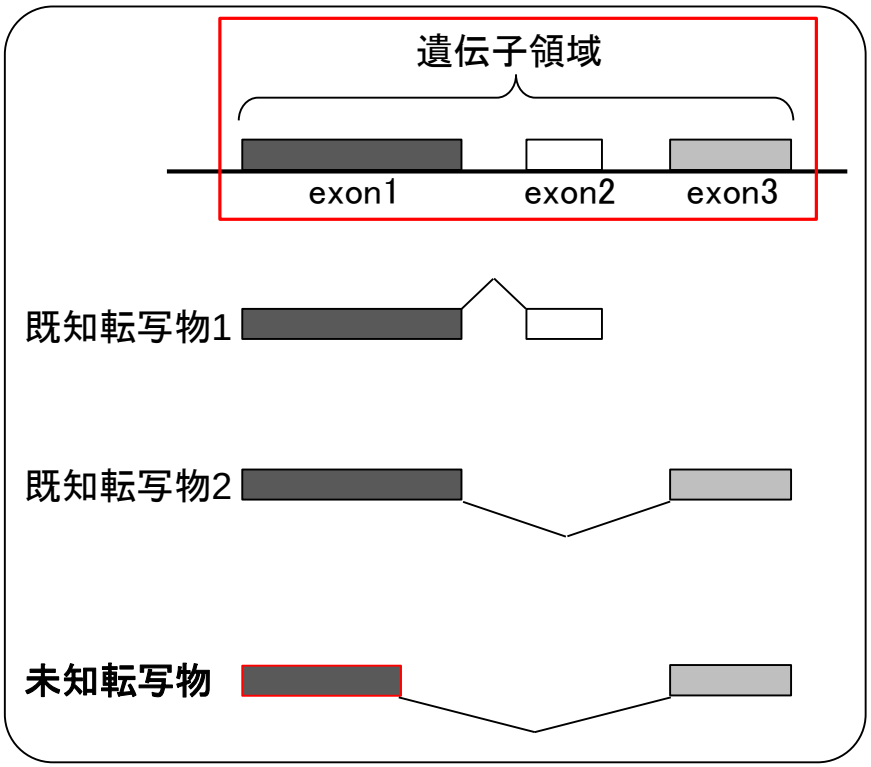


真の転写物情報

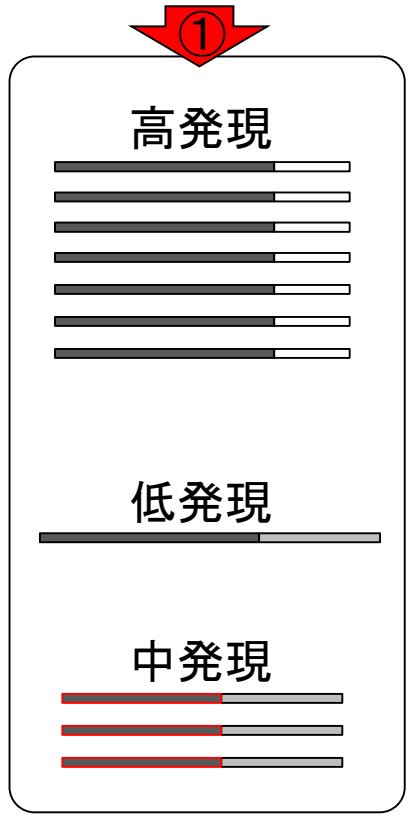
遺伝子 ≠ 転写物

実際の細胞内(例:目のサンプル)での発現情報(働いている度合い)が①のような感じだったとする

- ある状態のあるサンプル(例:目)のあるゲノムの領域



真の転写物情報

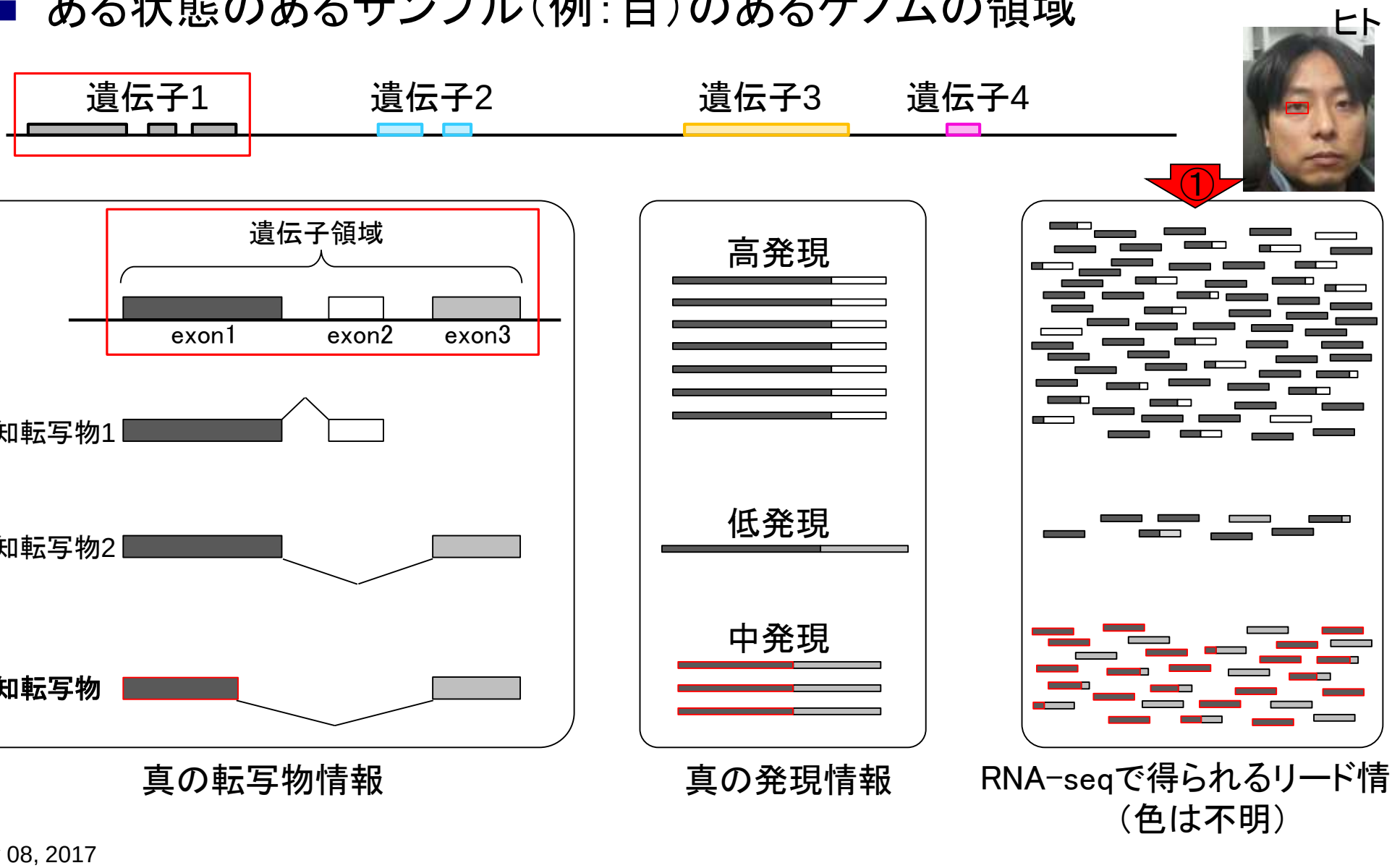


真の発現情報

遺伝子 ≠ 転写物

①NGS機器を用いて転写されているmRNA配列決定（RNA-seq）をした結果のイメージ

- ある状態のあるサンプル（例：目）のあるゲノムの領域



Contents

- トランスクリプトーム解析技術の原理や特徴
  - マイクロアレイとRNA-seq、遺伝子≠転写物
  - 様々な解析目的、トランスクリプトーム（転写物）配列取得
  - アノテーションファイルの読み込み（Rで転写物配列取得のイントロ）
  - Rで転写物配列取得（アノテーションファイルとゲノム情報ファイルから）
  - マイクロアレイの特徴
  - 発現データベース（DB）
- 発現DBからのプローブレベルデータ取得
  - Affymetrix GeneChip
  - R経由（教科書の § 2.2.1）
- Affymetrix GeneChipデータ前処理法を実行

非モデル生物(non-model organisms)を
取扱う場合には、選択肢はRNA-seqのみ

様々な解析目的

- トランスクリプトーム(転写物)配列取得
  - RNA-seqを利用
  - ゲノム配列既知の場合:遺伝子構造推定、新規isoform同定など
  - ゲノム配列未知の場合:トランスクリプトーム用アセンブラを実行
- 遺伝子または転写物ごとの発現量の正確な推定
  - 主にRNA-seq。ヒトやマウスなどのモデル生物はマイクロアレイも利用可能
- 比較するサンプル間で発現変動している遺伝子または転写物の同定
  - マイクロアレイ
    - 用いるアレイの種類(3'発現解析用アレイ、エクソンアレイ、トランスクリプトームアレイなど)によって発現変動解析の解像度(遺伝子、exon、転写物レベルなど)が異なる。
    - アレイが提供されていない生物種の解析は不可能
  - RNA-seq
    - 基本的に生物種非依存。任意のリファレンス配列(ゲノムまたはトランスクリプトーム)にリードをマップし、カウントデータ取得、統計解析。ゲノム配列がなくてもトランスクリプトーム配列をアセンブリで取得すればリファレンスとして利用可能。

参考

様々な解析目的

- トランスクリプトーム配列取得
  - ゲノム配列**既知**の場合

新規転写物同定などに相当。①がメインプログラム。多くのメインプログラム内部で、②や③のサブプログラムが動作する。例えばBowtie-Tophat-Cufflinksパイプラインは、①Cufflinks内部で②ジャンクションリードもマップ可能なTophat(やTophat2)が動作しており、そのさらに内部で基本マッピングプログラムである③Bowtie(やBowtie2)が動作している。最近ではTophat2からHISAT2に代わっている

- 解析 | 一般 | 上流配列解析 | Relative Appearance Ratio(Yamamoto_20
- 解析 | 基礎 | k-mer | ゲノムサイズ推定(基礎) | grqc (last modified 2016
- 解析 | 基礎 | 平均-分散プロット | について (last modified 2015/11/11)
- 解析 | 基礎 | 平均-分散プロット | Technical replicates (last modified 2014/02/18)
- 解析 | 基礎 | 平均-分散プロット | Biological re[illegible]cates (last modified 2014/02/21)
- 解析 | 新規転写物同定(ゲノム配列を利用) [illegible]modified 2015/08/25)
- 解析 | 発現量推定(トランスクリプトーム配列を利用) (last modified 2015/11/10)
- 解析 | クラスタリング | について (last modified 2014/02/05)
- 解析 | クラスタリング | サンプル間 | hclust (last modified 2015/02/26)
- 解析 | クラスタリング | サンプル間 | TCC(Sun_2013) (last modified 2015/11/15)
- 解析 | クラスタリング | 遺伝子間(基礎) | MBCluster.Seq(Si_2014) (last modified 2015/03/17)
- 解析 | クラスタリング | 遺伝子間(応用) | MBCluster.Seq(Si_2014)+TCC正規化(Sun_2013) (last modif
- 解析 | シミュレーションカウントデータ | について (last modified 2015/11/10)
- 解析 | シミュレーションカウントデータ | Technical rep.(ポアソン分布) (last modified 2015/01/23)
- 解析 | シミュレーションカウントデータ | Biological rep. | 基礎 (last modified 2015/01/23)

- アセンブル | について (last modified 2014/06/20)
- アセンブル | ゲノム用 (last modified 2015/08/20)
- アセンブル | トランスクリプトーム(転写物)用 (last modified 2015/08/18)
- マッピング | について (la[illegible]dified 2015/11/11)
- マッピング | basic aligner (last modified 2014/08/08)
- マッピング | splice-aware aligner (last modified 2015/11/11)
- マッピング | Bisulfite sequencing[illegible]st modified 2014/07/09)
- マッピング | (ESTレベルの長さ)contig (last modified 2014/06/24)
- マッピング | 基礎 (last modified 2013/06/19)

トランスクリプトーム配列の *de novo*アセンブリに相当。多くのプログラムは発現量(FPKM値)も出力してくれます

様々な解析目的

- トランスクリプトーム配列取得
  - ゲノム配列**未知**の場合

- アセンブル | について (last modified 2014/06/20)
- アセンブル | ゲノム用 (last modified 2015/08/2
- アセンブル | トランスクリプトーム(転写物)用 ① modified 2015/08/18)
- マッピング | について (last modified 2015/11/1
- マッピング | basic aligner (last modified 2014/08/08)
- マッピング | splice-aware aligner (last modified 2015/11/11)
- マッピング | Bisulfite sequencing用 (last modified 2014/07/09)
- マッピング | (ESTレベルの長さの)contig (last modified 2014/06/24)
- マッピング | 基礎 (last modified 2013/06/19)

アセンブル | トランスクリプトーム(転写物)用

Rパッケージはおそらくありません。

プログラム:

- Multiple-k: Surget-Groba and Montoya-Burgos, Genome Res., 2010
- Trans-ABySS: Robertson et al., Nat Methods, 2010
- Rnnotator: Martin et al., BMC Genomics, 2010
- Trinity: Grabherr et al., Nat Biotechnol, 2011
- Oases: Schulz et al., Bioinformatics, 2012
- EBARDenovo: Chu et al., Bioinformatics, 2013
- BRANCH: Bao et al., Bioinformatics, 2013
- IDBA-tran: Peng et al., Bioinformatics, 2013
- SOAPdenovo-Trans: Xie et al., Bioinformatics, 2014
- VTBuilder: Archer et al., BMC Bioinformatics, 2014
- Rockhopper 2(バクテリア用): Tjaden B, Genome Biol., 2015
- DETONATE(RSEM-EVAL): Li et al., Genome Biol., 2014
- Bridger: Chang et al., Genome Biol., 2015
- IFRAT: Mbandi et al., BMC Bioinformatics, 2015

Review、ガイドライン、パイプライン系:

- Review: Martin and Wang, Nat Rev Genet., 2011

参考

様々な解析目的

- 発現量の正確な推定

転写物の発現量を正確に推定したい場合は、専用のプログラムを使うべし。②RSEMが有名。③Sailfishも高速なアルゴリズムとして有名。④**TIGER2**は日本語で質問できる上、最近の手法比較論文(Kanitz et al. *Genome Biol.*, 2015)でも高評価でおススメ

- 解析 | 基礎 | 平均-分散プロット | Biological replicates (last modified 2014
- 解析 | 新規転写物同定(ゲノム配列を利用) (last modified 2015/08/25)
- 解析 | 発現量推定(トランスクリプトーム配列を利用) (last modified 2015/11/10)
- 解析 | クラスタリング | について (last modified 2014/02/25)
- 解析 | クラスタリング | サンプル
- 解析 | クラスタリング | サンプル
- 解析 | クラスタリング | 遺伝子間
- 解析 | クラスタリング | 遺伝子間
- 解析 | シミュレーションカウント
- 解析 | シミュレーションカウント
- 解析 | シミュレーションカウント

① ②

解析 | 発現量推定(トランスクリプトーム配列を利用)

新規転写物(新規isoform)の発見などが目的でなく、既知転写物の発現量を知りたいだけの場合には、やたらと時間がかかるゲノム配列へのマッピングを避けるのが一般的です。有名なCufflinksも一応GTF形式のアノテーションファイルを与えることでゲノム全体にマップするのを避けるモードがあるらしいので、一応リストアップしています。転写物へのマッピングの場合には、splice-aware alignerを用いたジャンクションリードのマッピングを行う必要がないので、高速にマッピング可能なbasic alignerで十分です。但し、複数個所にマップされるリードは考慮する必要があり、確率モデルのパラメータを最尤法に基づいて推定するexpectation-maximization (EM)アルゴリズムがよく用いられます。マッピングを行わずに、k-merを用いてalignment-freeで行う発現量推定を行うSailfishやRNA-Skimは従来法に比べて劇的に高速化がなされているようです。間違いがいくつか含まれているとは思います。2015年11月に調べた結果をリストアップします:

プログラム:

- Cufflinks: Trapnell et al., Nat Biotechnol., 2010
- NEUMA: Lee et al., Nucleic Acids Res., 2011
- IsoEM: Nicolae et al., Algorithms Mol. Biol., 2011
- RSEM: Li and Dewey, BMC Bioinformatics, 2011
- eXpress: Roberts and Pachter, Nat Methods, 2013
- ReXpress: Roberts et al., Bioinformatics, 2013
- TIGAR: Nariai et al., Bioinformatics, 2013
- eXpress-D: Roberts et al., BMC Bioinformatics, 2013
- PennSeq: Hu et al., Nucleic Acids Res., 2014
- Sailfish: Patro et al., Nat Biotechnol., 2014
- RNA-Skim: Zhang and Wang, Bioinformatics, 2014
- TIGER2: Nariai et al., BMC Genomics, 2014
- EMSAR: Lee et al., BMC Bioinformatics, 2015
- NLDMseq: Liu et al., BMC Bioinformatics, 2015

② ③ ④

参考

様々な解析目的

- 発現変動解析(2群間比較)

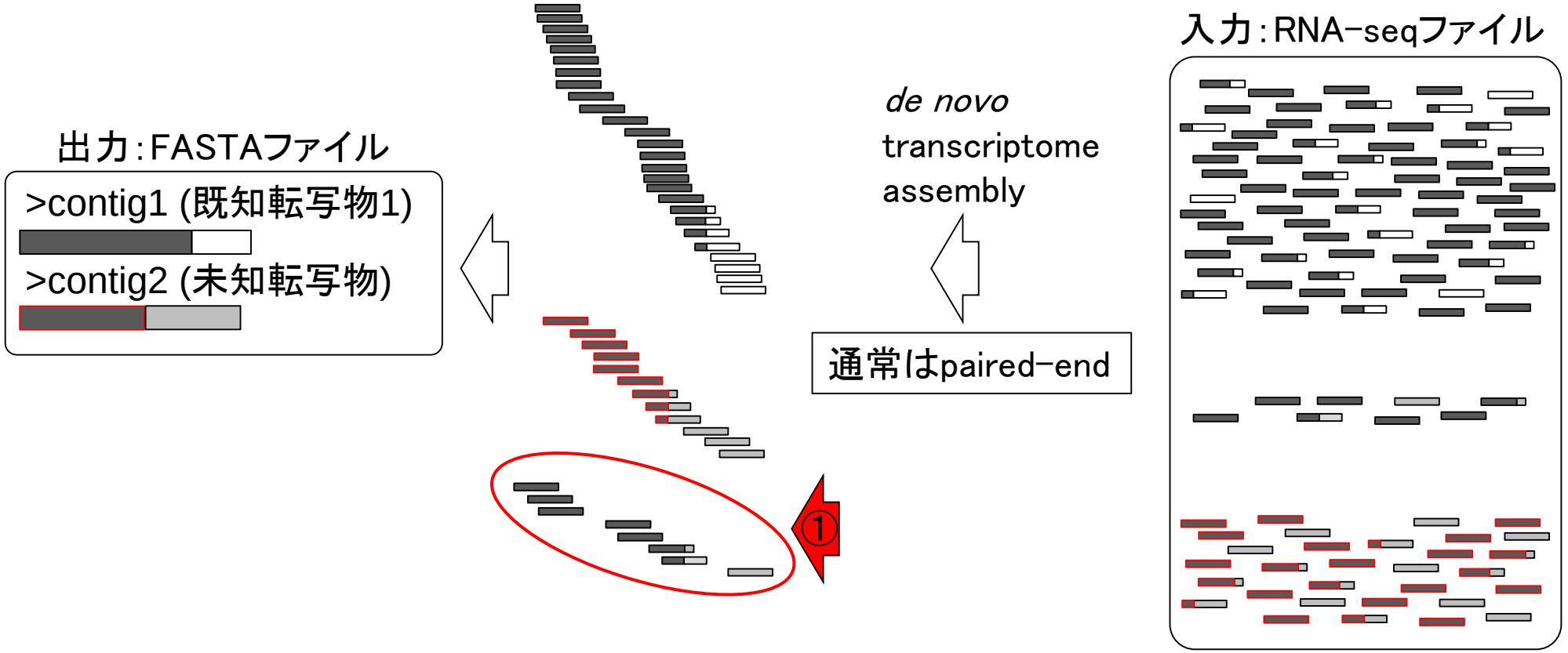
- 解析 | フィルタリング | について (last modified 2015/11/10)
- 解析 | 発現変動 | について (last modified 2014/07/10)
- 解析 | 発現変動 | 2群間 | 対応なし | について (last modified 2015/11/13)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | DESeq2(Love_2014) (last modified 2015/11/15)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | TCC(Sun_2013) (last modified 2015/07/07)推奨 ①
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | Blekhmanデータ | TCC(Sun_2013) (last modified
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | SAMseq(Li_2013) (last modified 2014/02/07)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | edgeR(Robinson_2010) (last modified 2014/07/24)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | WAD(Kadota_2008) (last modified 2015/03/30)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製なし | TCC(Sun_2013) (last modified 2014/03/05)推奨 ②
- 解析 | 発現変動 | 2群間 | 対応なし | 複製なし | DESeq(Anders_2010) (last modified 2014/03/20)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製なし | edgeR(Robinson_2010) (last modified 2014/03/20)

2群間比較で①反復あり(複製あり)データの場合はedgeR、②反復なしの場合はDESeq2を内部的に用いて頑健な結果を返すTCCがおススメ。反復の有無に応じて、内部的に用いるパッケージを自動で切り替える

様々な解析目的

- トランスクリプトーム配列取得
  - ゲノム配列**未知**の場合

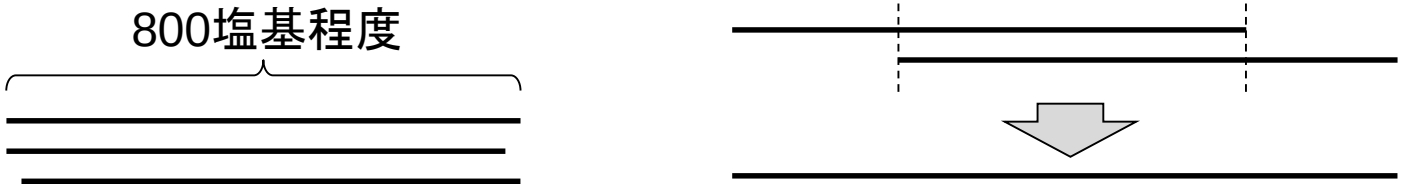
①ターゲットサンプル中でそれほど発現していない転写物は、*de novo*（1から、最初から、の意味）アセンブリが原理的に困難。これはIllumina short-readデータをイメージしたもの



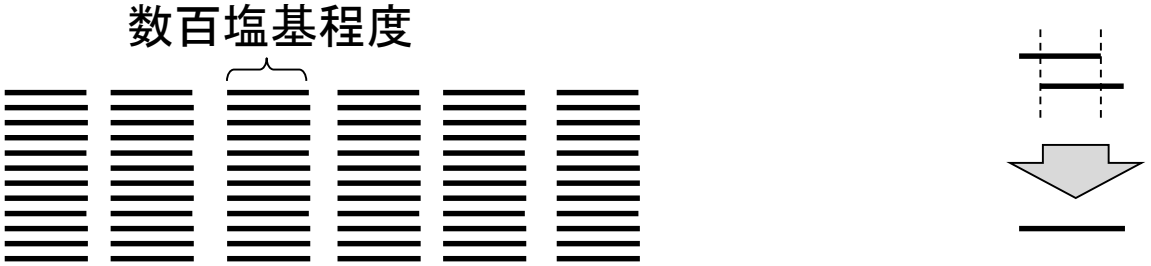
ロングリードも…

①第3世代の1分子シークエンサの代表格である PacBio RS II/Sequel Systemは、ゲノム配列決定で注目されているが、転写物配列を得る方向性も存在する

- 旧世代シーケンサー（ABI3730など）：～1,000塩基



- NGS (short-read; Illumina)：～数百塩基



- NGS (long-read; PacBio)：～数万塩基

ロングリードも…

①おそらくこれがPacBioシステムを用いて転写物配列を取得するという代表的な論文。これを引用している文献を見るなどすれば、最近の傾向が把握できる。例えば②など

Nat Biotechnol. 2013 Nov;31(11):1009-14. doi: 10.1038/nbt.2705. Epub 2013 Oct 13.

A single-molecule long-read survey of the human transcriptome.

Sharon D[1], Tilgner H, Grubert F, Snyder M.

Author information

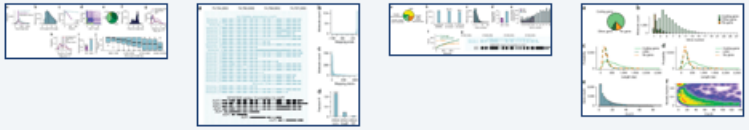
Abstract

Global RNA studies have become central to understanding biological processes, but methods such as microarrays and short-read sequencing are unable to describe an entire RNA molecule from 5' to 3' end. Here we use single-molecule long-read sequencing technology from Pacific Biosciences to sequence the polyadenylated RNA complement of a pooled set of 20 human organs and tissues without the need for fragmentation or amplification. We show that full-length RNA molecules of up to 1.5 kb can readily be monitored with little sequence loss at the 5' ends. For longer RNA molecules more 5' nucleotides are missing, but complete intron structures are often preserved. In total, we identify ~14,000 spliced GENCODE genes. High-confidence mappings are consistent with GENCODE annotations, but >10% of the alignments represent intron structures that were not previously annotated. As a group, transcripts mapping to unannotated regions have features of long, noncoding RNAs. Our results show the feasibility of deep sequencing full-length RNA from complex eukaryotic transcriptomes on a single-molecule level.

PMID: 24108091 PMCID: PMC4075632 DOI: 10.1038/nbt.2705

[Indexed for MEDLINE] Free PMC Article

Images from this publication. See all images (4) Free text



Publication type, MeSH terms, Substances, Grant support

LinkOut - more resources

PMC FREE Full text

Save items

Add to Favorites

Similar articles

Accurate identification and analysis of human mRNA i: [G3 (Bethesda). 2013]

Combining RT-PCR-seq and RNA-seq to catalog all [Genome Res. 2012]

Knowledge-based reconstruction of mRNA transcripts w [PLoS One. 2012]

Review Whole transcriptome analysis with sequenci [Cell Mol Life Sci. 2015]

Review Single-molecule direct RNA seq [Wiley Interdiscip Rev RNA. 2011]

See reviews...

See all...

Cited by 60 PubMed Central articles

Review Update on hypoxia-inducible factors and hy [Hypoxia (Auckl). 2017]

Genome-wide analysis of complex wheat gliadins, the dor [Sci Rep. 2017]

Hybrid sequencing and map finding (HySeMaFi): optional s [Sci Rep. 2017]

See all...

①ERP003225 (Sharon et al., Nat Biotechnol., 31: 1009–1014, 2013)

Contents

- トランスクリプトーム解析技術の原理や特徴
  - マイクロアレイとRNA-seq、遺伝子≠転写物
  - 様々な解析目的、トランスクリプトーム（転写物）配列取得
  - アノテーションファイルの読み込み（Rで転写物配列取得のイントロ）
  - Rで転写物配列取得（アノテーションファイルとゲノム情報ファイルから）
  - マイクロアレイの特徴
  - 発現データベース（DB）
- 発現DBからのプローブレベルデータ取得
  - Affymetrix GeneChip
  - R経由（教科書の § 2.2.1）
- Affymetrix GeneChipデータ前処理法を実行

アノテーション

アノテーションファイルの形式は、GFF/GTFが有名

- イントロ | NGS | 配列取得 | FASTQ or SRA | SRAdb(Zhu 2013) (last modified 2015/02/24)
- イントロ | NGS | 配列取得 | シミュレーションデータ | について (last modified 2015/01/18)
- イントロ | NGS | 配列取得 | シミュレーションデータ | ランダムな塩基配列の生成から (last modified 2015/01/18)
- イントロ | NGS | アノテーション情報取得 | について (last modified 2014/03/26)
- イントロ | NGS | アノテーション情報取得 | GFF/GTF形式ファイル (last modified 2017/04/10) NEW
- イントロ | NGS | アノテーション情報取得 | refFlat形式ファイル (last modified 2013/09/25)
- イントロ | NGS | アノテーション情報取得 | biomaRt(Durinck 2009) (last modified 2013/09/26)
- イントロ | NGS | アノテーション情報取得 |
- イントロ | NGS | アノテーション情報取得 |
- イントロ | NGS | アノテーション情報取得 |
- イントロ | NGS | アノテーション情報取得 |
- イントロ | NGS | 読み込み | BSgenome |
- イントロ | NGS | 読み込み | FASTA形式
- イントロ | NGS | 読み込み | FASTA形式
- イントロ | NGS | 読み込み | FASTQ形式
- イントロ | NGS | 読み込み | FASTQ形式
- イントロ | NGS | 読み込み | FASTQ形式
- イントロ | NGS | 読み込み | SAM/BAM形

①

イントロ | NGS | アノテーション情報取得 | GFF/GTF形式ファイル NEW

多くの生物種についてEnsembl (Yates et al., Nucleic Acids Res., 2016) のFTPサイトから GTF形式(GFF ver. 2)の遺伝子アノテーションファイルを得ることができます。2017年4月10日現在、上記のFTPサイトへのアクセス時にユーザ名とパスワードを聞かれますが、「匿名でログオンする」にチェックを入れればOKです。GTFはGeneral Transfer Formatまたは Gene Transfer Formatの略で、GTFの派生版としてGTF2というフォーマットもあるようです。また、General Feature Format ver. 3 (GFF3)という形式も存在するなど、GFF/GTF形式として総称されている中で様々なバリエーションがあります。いずれもrefFlat形式同様、どの領域にどの遺伝子があるのかという座標(Coordinates)情報を含みます。ゲノム配列のバージョンと同じであることを確認した上で用いましょう。

- Ensembl (Yates et al., Nucleic Acids Res., 2016)
  圧縮(gzip)ファイル形式です。基本はFTPサイトです。代表的なものを以下にリストアップしています。
  - ヒト; Human(H.sapiens)
  - ラット; Rat(R.norvegicus)
  - ネコ; Cat(F.catus)
  - ウサギ; Rabbit(O.cuniculus)
  - ニワトリ; Chicken(G.gallus)
  - イヌ; Dog(C.familiaris)
  - ウマ; Horse(E.caballus)
  - ゼブラフィッシュ; Zebrafish (D.rerio)
  - ...
- イネ: RAP-DB (Sakai et al., Plant Cell Physiol., 2013)
  - 「ダウンロード」-「Gene set」-「Gene structure and function information in GFF format」-「Download」。IRGSP-1.0_representative_2014-03-05.tar.gz (12.4MB程度)の圧縮ファイルが得られます。
- シロイヌナズナ: The Arabidopsis Information Resource (TAIR) (Lamesch et al., Nucleic Acids Res., 2012)
  - 「ダウンロード」-「Genes」-「TAIR10_genome_release」-「TAIR10_gff3」の TAIR10_GFF3_genes.gff (42MB程度)

他にrefFlat形式など様々なファイル形式が存在します

GFF/GTF形式ファイルの例

GFF3形式（シロイヌナズナ; TAIR10_GFF3_genes.gff）

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

GTF形式（ゼブラフィッシュ; Danio_rerio.Zv9.75.gtf）

	A	B	C	D	E	F	G	H	I
1	#!genome-build Zv9								
2	#!genome-version Zv9								
3	#!genome-date 2010-04								
4	#!genome-build-accession NCBI:GCA_000002035.2								
5	#!genebuild-last-updated 2014-02								
6	7	protein_coding	gene	100958	101715	.	+	.	gene_id "ENSDARG00000076051"; gene_name "CABZ01062994.1"; gene
7	7	protein_coding	transcript	100958	101715	.	+	.	gene_id "ENSDARG00000076051"; transcript_id "ENSDART00000113409
8	7	protein_coding	exon	100958	100975	.	+	.	gene_id "ENSDARG00000076051"; transcript_id "ENSDART00000113409
9	7	protein_coding	CDS	100958	100975	.	+	0	gene_id "ENSDARG00000076051"; transcript_id "ENSDART00000113409
10	7	protein_coding	exon	101077	101715	.	+	.	gene_id "ENSDARG00000076051"; transcript_id "ENSDART00000113409
11	7	protein_coding	CDS	101077	101715	.	+	0	gene_id "ENSDARG00000076051"; transcript_id "ENSDART00000113409
12	7	protein_coding	gene	116160	117573	.	+	.	gene_id "ENSDARG00000088691"; gene_name "BX511027.1"; gene_sour
13	7	protein_coding	transcript	116160	117573	.	+	.	gene_id "ENSDARG00000088691"; transcript_id "ENSDART00000129330

GFFの読み込み

読み込み段階でコケる、読み込みはうまくいったが、その後の解析段階でコケるなど、Linux上での解析同様、一筋縄ではいきません。過去の受講生など多方面からの情報提供のおかげでだいぶ分かってきました。尚、教科書執筆当時のTranscriptDbという記述はTxDbに変更されています（p91あたり）

① ←

イントロ | NGS | アノテーション情報取得 | TxDb | GFF/GTF形式ファイルから NEW

QuasRパッケージを用いてゲノムへのマッピング結果からカウント情報を得たいときに、"TxDb"という形式のオブジェクトを利用する必要があります。ここでは、GenomicFeatures パッケージを用いて手元にあるGFF/GTF形式ファイルを入力としてTxDbオブジェクトを得るやり方を示します。基本的にはGenomicFeaturesパッケージ中のmakeTxDbFromGFF関数を用いてGFF/GTF形式ファイルを読み込むことで TxDbオブジェクトをエラーなく読み込むこと自体は簡単にできます。しかし、得られたTxDbオブジェクトとゲノムマッピング結果ファイルを用いてカウント情報を得る場合に、ゲノム配列提供元とアノテーション情報提供元が異なっているとエラーとなります。具体的には、GFF/GTFファイル中にゲノム配列中にない染色体名があるとエラーが出る場合があります。

1. TAIR(Lamesch et al., Nucleic Acids Res., 2012) から提供されているArabidopsisのGFF3形式ファイル(TAIR10_GFF3_genes.gff)の場合:

基本形です。エラーは出ませんが、2015年3月4日現在、ChrCが環状ではないと認識されてしまっています。

```
in_f <- "TAIR10_GFF3_genes.gff"          #入力ファイル名を指定してin_fに格納(GFF/GTFファイル)

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

#本番(TxDbオブジェクトの作成)
txdb <- makeTxDbFromGFF(in_f)            #txdbオブジェクトの作成
txdb                                     #確認してるだけです
```

GFFの読み込み

①例題7。②ここで用いているGFF形式の入力ファイルは、③から取得しました。③をクリックしたつもりで次のスライドを眺める

イントロ | NGS | アノテーション情報取得 | TxDb | GFF/GTF形式ファイルから NEW

QuasRパッケージを用いてゲノムへのマッピング結果からカウント情報を得たいときに、"TxDb"という形式のオブジェクトを利用する必要があります。ここでは、GenomicFeatures パッケージを用いて手元にあるGFF/GTF形式ファイルを入力としてTxDbオブジェクトを得るやり方を示します。基本的にはGenomicFeaturesパッケージ中のmakeTxDbFromGFF関数を用いてGFF/GTF形式ファイルを読み込むことで TxDbオブジェクトをエラーなく読み込むこと自体は簡単にできます。しかし、得られたTxDbオブジェクトとゲノムマッピング結果ファイルを用いてカウント情報を得る場合に、ゲノム配列提供元とアノテーション情報提供元が異なっているとエラーとなります。具体的には、GFF/GTFファイル中にゲノム配列中にない染色体名があるとエラーが出る場合があります。

1. TAIR(Lamesch et al., Nucleic Acids Res., 2012) から提供されているArabidopsisのGFF3形式ファイル(TAIR10_GFF3_genes.gff)の場合:

基本形です。エラーは出ませんが、2015年3月4日現在、ChrCが環状ではないと認識されてしまっています。

```
in_f <- "TAIR10_GFF3_genes.gff"          #入力ファイル名を指定してin_fに格納(GFF/GTFファイル)
```

①

7. GFF3形式ファイル(Lactobacillus_hokkaidonensis_jcm_18461.GCA_000829395.1.30.chromosome.Chromosome.gff3)の場合:

Ensembl (Flicek et al., 2014)から提供されている Lactobacillus hokkaidonensis JCM 18461 (Tanizawa et al., 2015) のデータです。

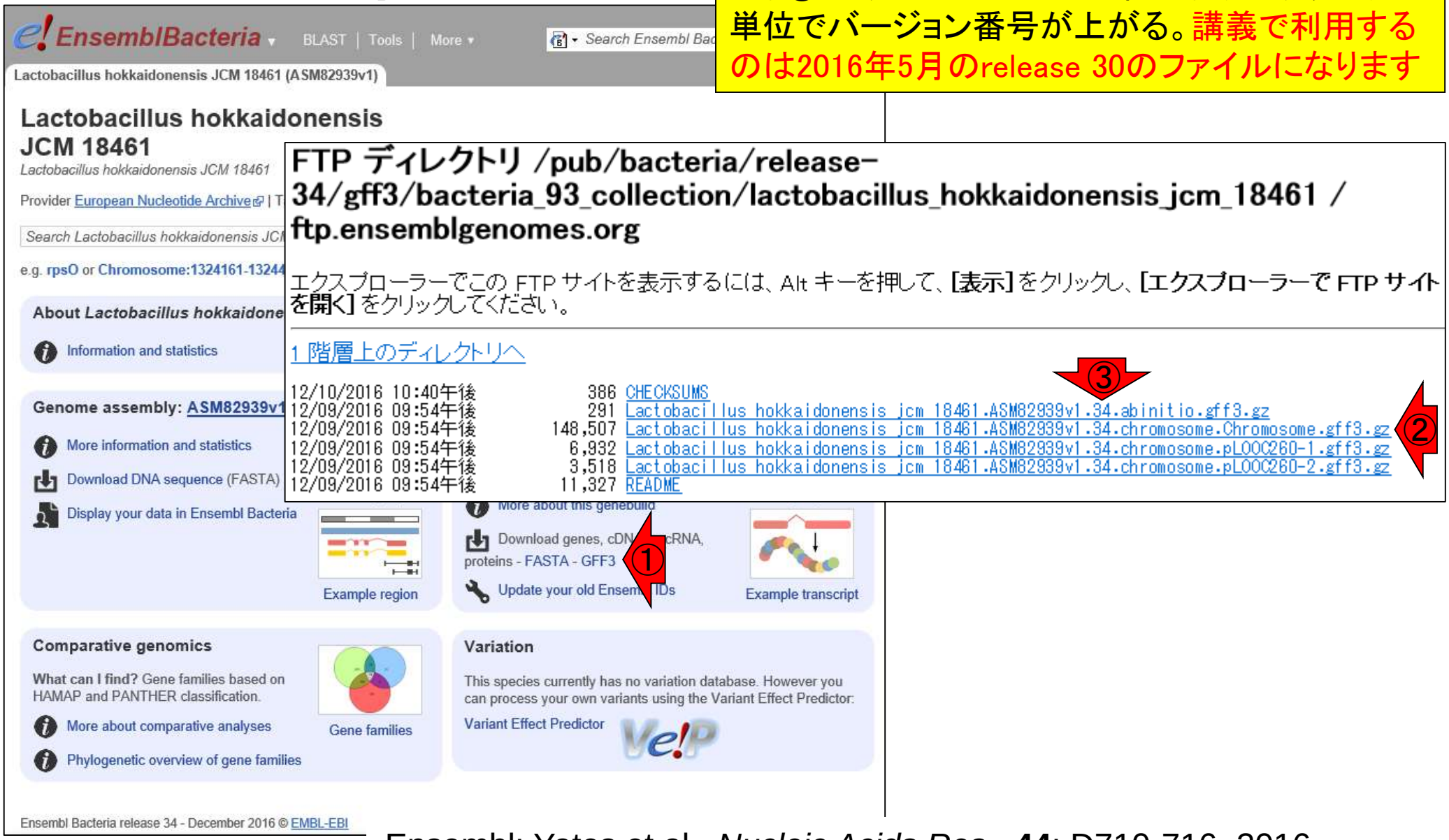
③

```
in_f <- "Lactobacillus_hokkaidonensis_jcm_18461.GCA_000829395.1.30.chromosome.Chromosome.gff3"#入

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

#本番(TxDbオブジェクトの作成)
txdb <- makeTxDbFromGFF(in_f, format="auto")#txdbオブジェクトの作成
txdb                                     #確認してるだけです
```

②

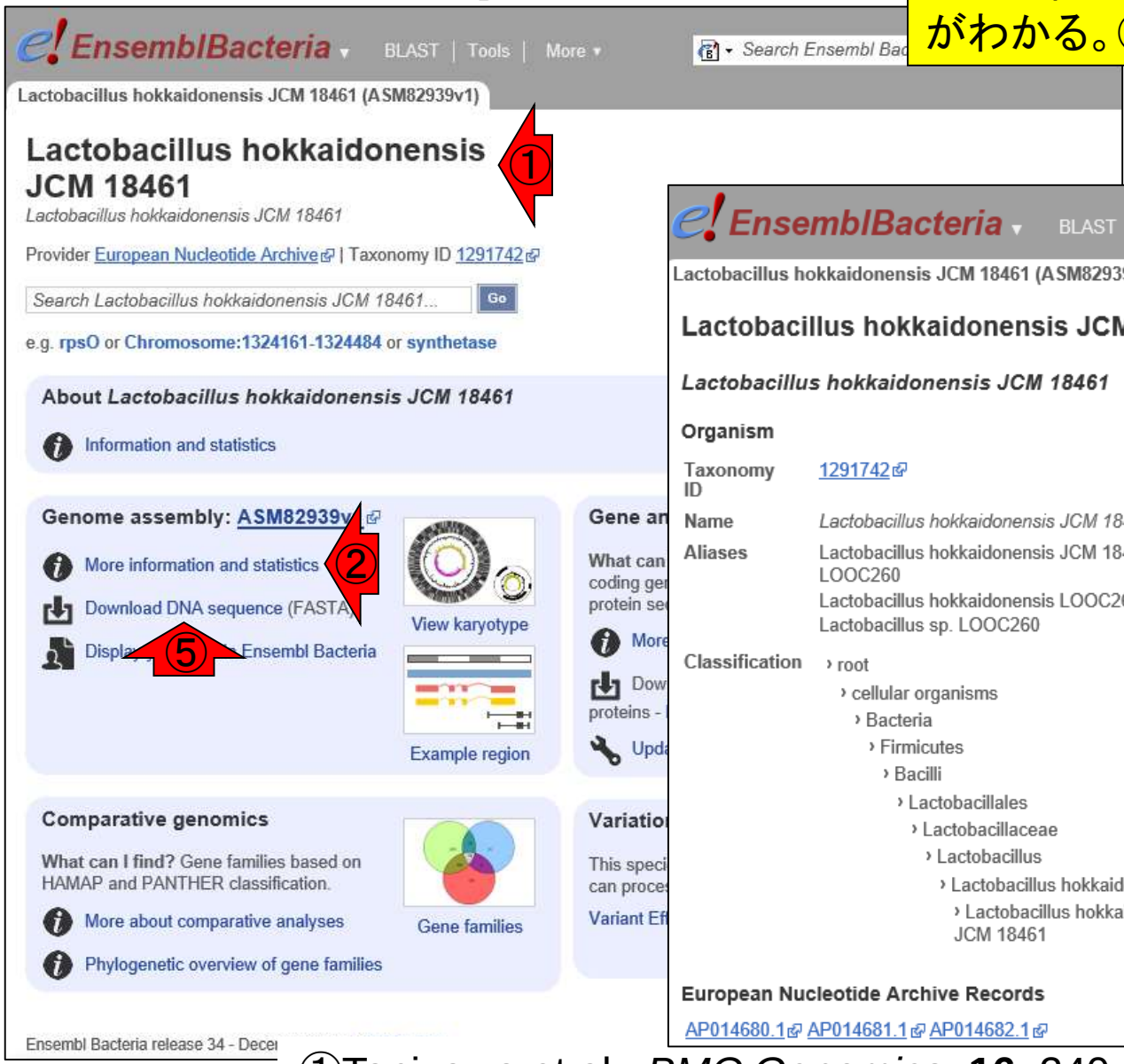
Ensembl解説

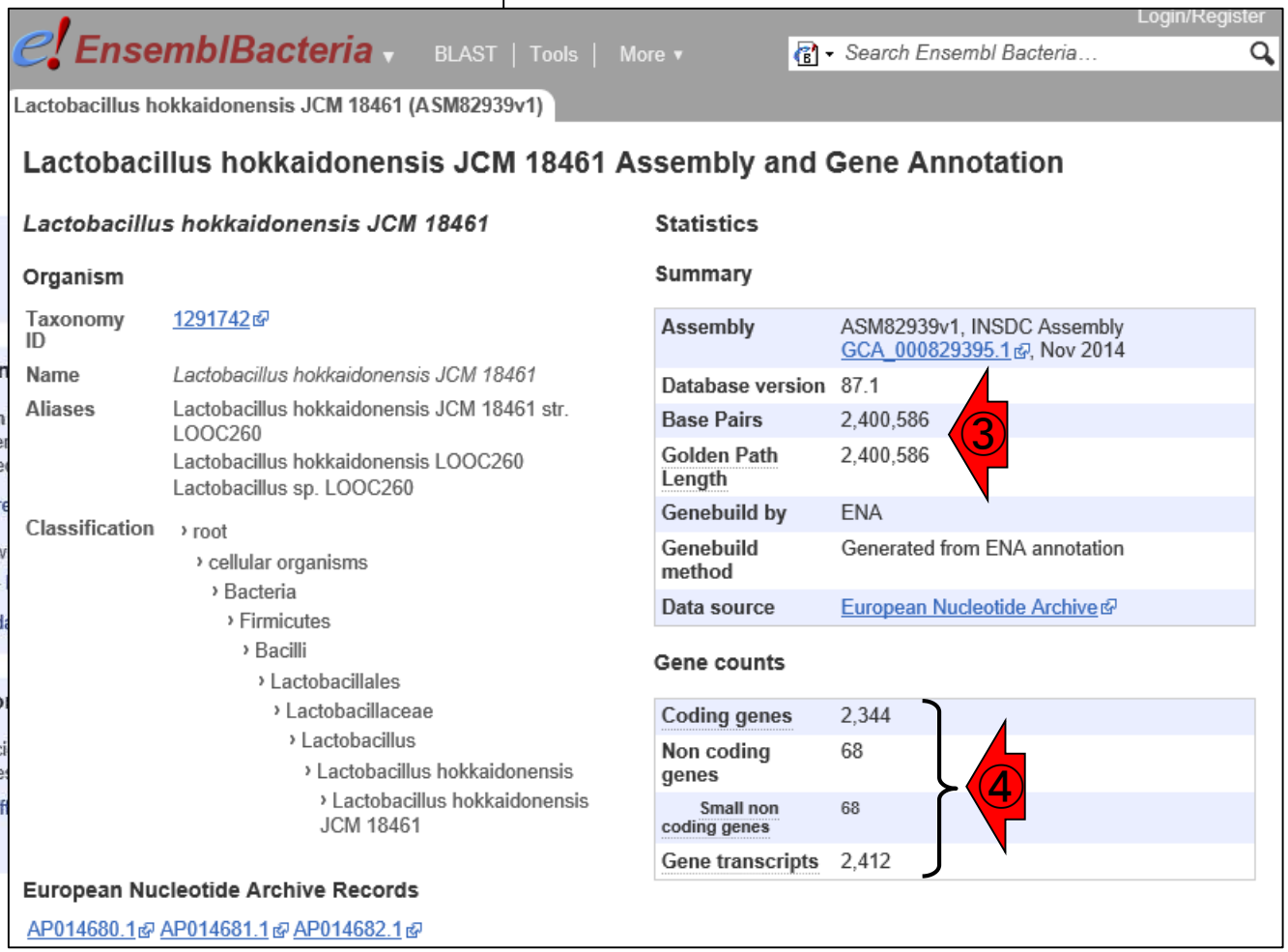
①GFFファイルはここから取得。②のgzip圧縮ファイルをダウンロードして解凍したものが入力ファイル。③のあたりがバージョン番号。概ね、数か月単位でバージョン番号が上がる。講義で利用するのは2016年5月のrelease 30のファイルになります



Ensembl解説

①このゲノムの全貌は②である程度把握可能。原著論文の情報なども合わせることで、③ゲノムサイズが約2.4MB、④2,344 coding genesなどの情報がわかる。⑤でゲノム配列をダウンロードできる





①Tanizawa et al., *BMC Genomics*, **16**: 240, 2015

Ensembl解説

いろんなものがあって私はよくわかりませんが、GFFファイルと一緒に取り扱いたいときには、GFFファイルと似た名前の①を採用します。正確には、このゲノムは1つの染色体と②2つのプラスミド(pLOOC260-1とpLOOC260-2)からなっています。①はそのうちの染色体配列のみになります。③ファイルサイズ的に、これが3つの配列がまとめられたものなのでしょう

FTP ディレクトリ /pub/bacteria/releas
34/fasta/bacteria_93_collection/lactol
ftp.ensemblgenomes.org

エクスプローラーでこの FTP サイトを表示するには、Alt キー を押して、[表示] をクリックし、[エクスプローラー で FTP サイトを開く] をクリックしてください。

1 階層上のディレクトリへ

12/10/2016 06:59午後	1,132	CHECKSUMS
12/09/2016 06:09午後	706,228	Lactobacillus hokkaidonensis jcm 18461.ASM82939v1.dna.chromosome.Chromosome.fa.gz
12/09/2016 06:09午後	26,023	Lactobacillus hokkaidonensis jcm 18461.ASM82939v1.dna.chromosome.pLOOC260-1.fa.gz
12/09/2016 06:09午後	13,356	Lactobacillus hokkaidonensis jcm 18461.ASM82939v1.dna.chromosome.pLOOC260-2.fa.gz
12/09/2016 06:09午後	745,607	Lactobacillus hokkaidonensis jcm 18461.ASM82939v1.dna.toplevel.fa.gz
12/09/2016 06:09午後	706,237	Lactobacillus hokkaidonensis jcm 18461.ASM82939v1.dna rm.chromosome mosome.fa.gz
12/09/2016 06:09午後	26,032	Lactobacillus hokkaidonensis jcm 18461.ASM82939v1.dna rm.chromosome OOC260-1.fa.gz
12/09/2016 06:09午後	13,364	Lactobacillus hokkaidonensis jcm 18461.ASM82939v1.dna rm.chromosome.pLOOC260-2.fa.gz
12/09/2016 06:09午後	745,633	Lactobacillus hokkaidonensis jcm 18461.ASM82939v1.dna rm.toplevel.fa.gz
12/09/2016 06:09午後	706,237	Lactobacillus hokkaidonensis jcm 18461.ASM82939v1.dna sm.chromosome.Chromosome.fa.gz
12/09/2016 06:09午後	26,031	Lactobacillus hokkaidonensis jcm 18461.ASM82939v1.dna sm.chromosome.pLOOC260-1.fa.gz
12/09/2016 06:09午後	13,364	Lactobacillus hokkaidonensis jcm 18461.ASM82939v1.dna sm.chromosome.pLOOC260-2.fa.gz
12/09/2016 06:09午後	745,632	Lactobacillus hokkaidonensis jcm 18461.ASM82939v1.dna sm.toplevel.fa.gz
12/09/2016 06:09午後	4,923	README

GFFの読み込み

①例題7が読み込みの基本形。②GenomicFeaturesというパッケージが提供する③makeTxDbFromGFF関数を用いてGFFファイルを読み込んで、TxDbという独特の形式で取り扱えるようにする

①

7. GFF3形式ファイル(Lactobacillus_hokkaidonensis_jcm_18461.GCA_000829

Ensembl (Flicek et al., 2014)から提供されている Lactobacillus hokkaidonensis JCM 18461 (Tanizawa et al., 2015)のデータです。

```
in_f <- "Lactobacillus_hokkaidonensis_jcm_18461.GCA_000829395.1.30.chromosome.Chromosome.gff3"#入

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

#本番(TxDbオブジェクトの作成)
txdb <- makeTxDbFromGFF(in_f, format="auto")#txdbオブジェクトの作成
txdb                                        #確認してるだけです
```

②

③

GFFの読み込み

①makeTxDbFromGFF関数での読み込み時に②警告メッセージが出ている。これは、内部的に使っている「③RSQLiteパッケージ中の④dbGetPreparedQuery関数は非推奨なので、⑤DBIパッケージ中の…を使うように変えてね」という趣旨の警告メッセージです。エラーじゃないのでとりあえず無視。こういうのはよくあります

7. GFF3形式ファイル(Lactobacillus_hokkaidonensis_jcm_18461.GCA_00

Ensembl (Flicek et al., 2014)から提供されている Lactobacillus hokkaidone

```
in_f <- "Lactobacillus_hokkaidonensis_jcm_18461.GCA_000

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

#本番①:TxDbオブジェクトの作成)
txdb <- makeTxDbFromGFF(in_f, format="auto")#txdbオブジェクトの作成
txdb                                        #確認してるだけです
```

R Console

```
    'citation("Biobase")', and for packages 'citation("pkgname")'.

>
> #本番①:TxDbオブジェクトの作成)
> txdb <- makeTxDbFromGFF(in_f, format="auto")#txdbオブジェクトの作成
Import genomic features from the file as a GRanges object ... OK
Prepare the 'metadata' data frame ... OK
Make the TxDb object ... OK
 警告メッセージ:
1: RSQLite::dbGetPreparedQuery() is deprecated, please switch to DBI:$
2: Named parameters not used in query: internal_chrom_id, chrom, ...$
3: Named parameters not used in query: internal_id, name, type, chrom$
4: Named parameters not used in query: internal_id, name, chrom, stra$
5: Named parameters not used in query: internal_id, name, chrom, stra$
6: Named parameters not used in query: internal_tx_id, exon_rank, int$
7: Named parameters not used in query: gene_id, internal_tx_id
> txdb                                        #確認してるだけです
TxDb object:
```

GFFの読み込み

①読み込み後のtxdbオブジェクトの中身を表示

7. GFF3形式ファイル(Lactobacillus_hokkaidonensis_jcm_18461.GCA_000829395.1.30.chromosome.Chromosome.gff3)の場合:

Ensembl (Flicek et al., 2014)から提供されている Lactobacillus hokkaidonensis JCM 18461 (Tanizawa et al., 2015) のデータです。

```
in_f <- "Lactobacillus_hokkaidonensis_jcm_18461.GCA_000829395.1.30.chromosome.Chromosome.gff3"#入

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

#本番(TxDbオブジェクトの作成)
txdb <- makeTxDbFromGFF(i
txdb
```

①

R Console

```
> txdb                                      #確認してるだけです
TxDb object:
# Db type: TxDb
# Supporting package: GenomicFeatures
# Data source: Lactobacillus_hokkaidonensis_jcm_18461.GCA_000829395.1$
# Organism: NA
# Taxonomy ID: NA
# miRBase build ID: NA
# Genome: NA
# transcript_nrow: 2262
# exon_nrow: 2262
# cds_nrow: 2194
# Db created by: GenomicFeatures package from Bioconductor
# Creation time: 2017-04-18 14:34:46 +0900 (Tue, 18 Apr 2017)
# GenomicFeatures version at creation time: 1.26.3
# RSQLite version at creation time: 1.1-2
# DBSCHEMAVERSION: 1.1
>
```

矛盾?!

このゲノムは、1つの染色体と2つのプラスミド(pLOOC260-1とpLOOC260-2)からなっています。
①の結果は染色体のみの数値です。②のEnsemblウェブサイト上で見られる数値と一致していません

7. GFF3形式ファイル(Lactobacillus_hokkaidonensis_jcm_18461.GCA_0008293

Ensembl (Flicek et al., 2014)から提供されている Lactobacillus hokkaidonensis JCM 18461 (Tanizawa et al

```
in_f <- "Lactobacillus_hokkaidonensis_jcm_18461.GCA_000829395.1.30.chromosome.

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

#本番(TxDbオブジェクトの作成
txdb <- makeTxDbFromGFF(i
txdb
```

R Console

```
> txdb
TxDb object:
# Db type: TxDb
# Supporting package: GenomicFeatures
# Data source: Lactobacillus_hokkaidone
# Organism: NA
# Taxonomy ID: NA
# miRBase build ID: NA
# Genome: NA
# transcript_nrow: 2262
# exon_nrow: 2262
# cds_nrow: 2194
# Db created by: GenomicFeatures packag
# Creation time: 2017-04-18 14:34:46 +0
# GenomicFeatures version at creation t
# RSQLite version at creation time: 1.1-2
# DBSCHEMAVERSION: 1.1
>
```

①

Assembly	ASM82939v1, INSDC Assembly GCA_000829395.1, Nov 2014
Database version	87.1
Base Pairs	2,400,586
Golden Path Length	2,400,586
Genebuild by	ENA
Genebuild method	Generated from ENA annotation
Data source	European Nucleotide Archive

Gene counts

Coding genes	2,344
Non coding genes	68
Small non coding genes	68
Gene transcripts	2,412

②

課題1

このゲノムは、1つの染色体と2つのプラスミド(pLOOC260-1とpLOOC260-2)からなっています。①の結果は染色体のみの数値です。②のEnsemblウェブサイト上で見られる数値と一致していません。プラスミドのgffファイル(plasmid1.gff3とplasmid2.gff3)をそれぞれ読み込んで①transcript_nrow (Gene transcripts)と②cds_nrow (Coding genes)の情報を得て、③Ensemblウェブサイト上の数値と絡めて簡単に考察せよ

7. GFF3形式ファイル(Lactobacillus_hokkaidonensis_jcm_18461.GCA_0008293

Ensembl (Flicek et al., 2014)から提供されている Lactobacillus hokkaidonensis JC

```
in_f <- "Lactobacillus_hokkaidonensis_jcm_18461.GCA_0008293

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

#本番(TxDbオブジェクトの作成
txdb <- makeTxDbFromGFF(i
txdb
```

R Console

```
> txdb
TxDb object:
# Db type: TxDb
# Supporting package: GenomicFeatures
# Data source: Lactobacillus_hokkaidone
# Organism: NA
# Taxonomy ID: NA
# miRBase build ID: NA
# Genome: NA
① transcript_nrow: 2262
# exon_nrow: 2262
② cds_nrow: 2194
# Db created by: GenomicFeatures packag
# Creation time: 2017-04-18 14:34:46 +0
# GenomicFeatures version at creation t
# RSQLite version at creation time: 1.1-2
# DBSCHEMAVERSION: 1.1
>
```

Length	
Genebuild by	ENA
Genebuild method	Generated from ENA annotation
Data source	European Nucleotide Archive

Gene counts

Coding genes	2,344 ③
Non coding genes	68
Small non coding genes	68
Gene transcripts	2,412 ③

課題1

①プラスミドのgffファイル（plasmid1.gff3とplasmid2.gff3）はこちら

Contents

- トランスクリプトーム解析技術の原理や特徴
  - マイクロアレイとRNA-seq、遺伝子≠転写物
  - 様々な解析目的、トランスクリプトーム（転写物）配列取得
  - アノテーションファイルの読み込み（Rで転写物配列取得のイントロ）
  - Rで転写物配列取得（アノテーションファイルとゲノム情報ファイルから）
  - マイクロアレイの特徴
  - 発現データベース（DB）
- 発現DBからのプローブレベルデータ取得
  - Affymetrix GeneChip
  - R経由（教科書の§2.2.1）
- Affymetrix GeneChipデータ前処理法を実行

転写物配列取得

multi-FASTAファイル（ゲノム配列情報）とGFFファイル（アノテーション情報）を同時に読み込むことで、①トランスクリプトーム（転写物）配列情報を一気に取得することも可能。②例題5。③はhogeフォルダ中にあります

- イントロ | 一般 | 配列取得 | プロモーター配列 | BSgenomeとTxDbから(last
- イントロ | 一般 | 配列取得 | プロモーター配列 | GenomicFeatures(Lawrenc
- イントロ | 一般 | 配列取得 | トランスクリプトーム配列 | 公共DBから(last modified 2015/05/
- イントロ | 一般 | 配列取得 | トランスクリプトーム配列 | GenomicFeatures(Lawrence_2013) (last modified 2016/02/09)
- イントロ | 一般 | 配列取得 | トランスクリプトーム配列 | biomaRt(Durinck_2009)(last modifie 2015/02/20)
- イントロ | 一般 | 読み込み | xlsx形式 | openxlsx(last modified 2015/11/15)
- イントロ | NG
- イントロ | NG
- イントロ | NG
- イントロ | NG
- イントロ | NG
- イントロ | NG
- イントロ | NG
- イントロ | NG

① ②

イントロ | 一般 | 配列取得 | トランスクリプトーム配列 | GenomicFeatures(Lawrence_2013)

NEW

GenomicFeaturesパッケージを主に用いてトランスクリプトーム配列を得るやり方を示します。「?extractTranscriptSeqs」を行うことで、様々な例題を見ることができます。transcriptsBy関数部分は、exonsBy, cdsBy, intronsByTranscript, fiveUTRsByTranscript, threeUTRsByTranscriptなど様々な他の関数で置き換えることができます。

「ファイル

1. ヒト (BSg

対応するア
果は82,960

```
out_f <-
param_bsg
param_txd

#必要なパ
library(G

#前処理(ゲ
library(p
tmp <- ls
genome <-
```

5. GFF3形式のアノテーションファイルとFASTA形式のゲノム配列ファイルを読み込む場合:

GFF3形式ファイル(Lactobacillus_hokkaidonensis_jcm_18461.GCA_000829395.1.30.chromosome.Chromosome.gff3)とFASTA形式ファイル(Lactobacillus_hokkaidonensis_jcm_18461.GCA_000829395.1.30.dna.chromosome.Chromosome.fa)を読み込むやり方です。Ensembl (Flicek et al., 2014)から提供されている Lactobacillus hokkaidonensis JCM 18461 (Tanizawa et al., 2015) のデータです。

```
in_f1 <- "Lactobacillus_hokkaidonensis_jcm_18461.GCA_000829395.1.30.dna.chromosome.Chromosome.fa"
in_f2 <- "Lactobacillus_hokkaidonensis_jcm_18461.GCA_000829395.1.30.chromosome.Chromosome.gff3"#入
out_f <- "hoge5.fasta"                    #出力ファイル名を指定してout_fに格納

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

#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="auto")#txdbオブジェクトの作
txdb                                      #確認してるだけです

#前処理(欲しい領域の座標情報取得)
hoge <- transcripts(txdb)                 #指定した範囲の座標情報を取得
hoge                                      #確認してるだけです

#本番(配列取得)
```

③

講義日程（平成29年度）

1. 平成29年05月08日
   - 講義資料PDF
   - .gff3ファイル (約1.3MB)
   - .faファイル (約2.2MB)
   - (Rで)塩基配列解析
   - (Rで)マイクロアレイデータ解析
   - plasmid1.gff3(課題用)
   - plasmid2.gff3(課題用)
2. 平成29年05月15日

③

転写物配列取得

①は、GFFファイル情報を保持したtxdbオブジェクトから、transcriptsという関数を用いて抽出したい転写物の座標情報を取得した結果をhogeに保存している

5. GFF3形式のアノテーションファイルとFASTA形式のゲノム配列ファイルを読み込む場合:

GFF3形式ファイル(Lactobacillus_hokkaidonensis_jcm_18461.GCA_000829395.1.30.chromosome.Chromosome.gff3)とFASTA形式ファイル(Lactobacillus_hokkaidonensis_jcm_18461.GCA_000829395.1.30.dna.chromosome.Chromosome.fa)を読み込むやり方です。
Ensembl (Flicek et al., 2014)から提供されている Lactobacillus hokkaidonensis JCM 18461 (Tanizawa et al., 2015) のデータです。

```
in_f1 <- "Lactobacillus_hokkaidonensis_jcm_18461.GCA_000829395.1.30.dna.chromosome.Chromosome.fa"#ノ
in_f2 <- "Lactobacillus_hokkaidonensis_jcm_18461.GCA_000829395.1.30.chromosome.Chromosome.gff3"#入力
out_f <- "hoge5.fasta"                    #出力ファイル名を指定してout_fに格納

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

#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_
txdb

#前処理(欲しい領域の座標情報取
hoge <- transcripts(txdb)
hoge

#本番(配列取得)
fasta <- getSeq(FaFile(in_f
fasta

#後処理(description部分を変更
```

R Console

```
> #前処理(欲しい領域の座標情報取得)
> hoge <- transcripts(txdb)                    #指定した範囲の座標情報を取得
> hoge                                         #確認してるだけです
GRanges object with 2262 ranges and 2 metadata columns:
          seqnames              ranges strand   |     tx_id     tx_name
             <Rle>           <IRanges>  <Rle>   | <integer> <character>
      [1] Chromosome       [ 360, 1676]      +   |         1       dnaA-1
      [2] Chromosome       [1852, 2991]      +   |         2       dnaN-1
      [3] Chromosome       [3233, 3457]      +   |         3         <NA>
      [4] Chromosome       [3467, 4588]      +   |         4       recF-1
      [5] Chromosome       [4588, 6531]      +   |         5       gyrB-1
      ...         ...                ...    ... ...       ...          ...
   [2258] Chromosome [2273924, 2275312]      -   |      2258       trmE-1
   [2259] Chromosome [2275488, 2276288]      -   |      2259         <NA>
   [2260] Chromosome [2276455, 2277288]      -   |      2260         <NA>
   [2261] Chromosome [2277304, 2277648]      -   |      2261         <NA>
   [2262] Chromosome [2277719, 2277853]      -   |      2262       rpmH-1
   -------
   seqinfo: 1 sequence from an unspecified genome; no seqlengths
>
```

転写物配列取得

①GFFファイルの見方がよくわかっていなくても、うまく読み込めているらしいことはわかる

```
##gff-version  3
##sequence-region   Chromosome 360 2277853
#!genome-build European Nucleotide Archive ASM82939
#!genome-version GCA_000829395.1
#!genome-date 2014-11
#!genome-build-accession GCA_000829395.1
#!genebuild-last-updated 2014-11
Chromosome  ena  gene        360  1676  .  +  .  ID=ge
Chromosome  ena  transcript  360  1676  .  +  .  ID=tr
Chromosome  ena  exon        360  1676  .  +  .  Pare
Chromosome  ena  CDS         360  1676  .  +  0  ID=C
###
Chromosome  ena  gene        1852 2991  .  +  .  ID=ge
Chromosome  ena  transcript  1852 2991  .  +  .  ID=tr
Chromosome  ena  exon        1852 2991  .  +  .  Pare
Chromosome  ena  CDS         1852 2991  .  +  0  ID=C
###
Chromosome  ena  gene        3233 3457  .  +  .  ID=ge
Chromosome  ena  transcript  3233 3457  .  +  .  ID=tr
Chromosome  ena  exon        3233 3457  .  +  .  Pare
Chromosome  ena  CDS         3233 3457  .  +  0  ID=C
###
Chromosome  ena  gene        3467 4588  .  +  .  ID=ge
```

①

```
標情報取得)
ts(txdb)                    #指定した範囲の座標情報を取得
                            #確認してるだけです
2262 ranges and 2 metadata columns:
          ranges strand   |      tx_id     tx_name
       <IRanges>  <Rle>   | <integer> <character>
    [ 360, 1676]      +   |         1       dnaA-1
    [1852, 2991]      +   |         2       dnaN-1
    [3233, 3457]      +   |         3         <NA>
    [3467, 4588]      +   |         4       recF-1
    [4588, 6531]      +   |         5       gyrB-1
             ...    ... ...       ...          ...
[2273924, 2275312]    -   |      2258       trmE-1
[2275488, 2276288]    -   |      2259         <NA>
[2276455, 2277288]    -   |      2260         <NA>
[2277304, 2277648]    -   |      2261         <NA>
[2277719, 2277853]    -   |      2262       rpmH-1
  -------
  seqinfo: 1 sequence from an unspecified genome; no seqlengths
> |
```

②

転写物配列取得

①in_f1で指定したゲノム配列情報はここで登場。①ゲノム配列から、②のhogeで指定した座標の塩基配列を③(Biostringsパッケージが提供する)getSeq関数を用いて取得。④(Rsamtoolsパッケージが提供する)FaFile関数は、getSeq関数利用時に必要

5. GFF3形式のアノテーションファイルとFASTA形式のゲノム配列ファイルを読

GFF3形式ファイル(Lactobacillus_hokkaidonensis_jcm_18461.GCA_000829395
ファイル(Lactobacillus_hokkaidonensis_jcm_18461.GCA_000829395.1.30.dna.c
Ensembl (Flicek et al., 2014)から提供されている Lactobacillus hokkaidonensis JCM 18461 (Tanizawa et al., 2015) のデータです。

```
in_f1 <- "Lactobacillus_hokkaidonensis_jcm_18461.GCA_000829395.1.30.dna.chromosome.Chromosome.fa"#ノ
in_f2 <- "Lactobacillus_hokkaidonensis_jcm_18461.GCA_000829395.1.30.chromosome.Chromosome.gff3"#入力
out_f <- "hoge5.fasta"                    #出力ファイル名を指定してout_fに格納

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

#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="auto")#txdbオブジェクトの作成
txdb                                      #確認してるだけです

#前処理(欲しい領域の座標情報取得)
hoge <- transcripts(txdb)                 #指定した範囲の座標情
hoge                                      #確認してるだけです

#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)      #配列情報を取得した結果
fasta                                     #確認してるだけです

#後処理(description部分を変更)
```

④ ③ ① ②

R Console

```
> FaFile(in_f1)
class: FaFile
path: .../Lactobacillus_hokkaidonensis_jcm_$
index: .../Lactobacillus_hokkaidonensis_jcm$
isOpen: FALSE
yieldSize: NA
>
```

転写物配列取得

①getSeq実行後のfastaオブジェクトが、欲しいトランスクリプトーム配列情報ではあるが…

5. GFF3形式のアノテーションファイルとFASTA形式のゲノム配列ファイルを読み込む場合:

GFF3形式ファイル(Lactobacillus_hokkaidonensis_jcm_18461.GCA_000829395.1.30.chromosome.Chromosome.gff3)とFASTA形式ファイル(Lactobacillus_hokkaidonensis_jcm_18461.GCA_000829395.1.30.dna.chromosome.Chromosome.fa)を読み込むやり方です。
Ensembl (Flicek et al., 2014)から提供されている Lactobacillus hokkaidonensis JCM 18461 (Tanizawa et al., 2015) のデータです。

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

#入力ファイルの読み込み
txdb <- makeTxDbFromGFF(in_f2, format="auto")#txdbオブジェクトの作成
txdb                                         #確認してるだけです

#前処理(欲しい領域の座標情報取得)
hoge <- transcripts(txdb)
hoge

#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)
fasta

#後処理(description部分を変更)
names(fasta) <- paste(seqnames(hoge),
                      end(ranges(hoge
fasta

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

R Console

```
> fasta <- getSeq(FaFile(in_f1), hoge)    #配列情報を取得した$
> fasta                                   #確認してるだけです
  A DNAStringSet instance of length 2262
       width seq                                    names          $
   [1]  1317 GTGACTGATTTAGAA...AGCTAAAGCCATAG Chromosome
   [2]  1140 ATGAAATTTACAATT...TTAGAACTTACTAA Chromosome
   [3]   225 GTGCAAGAAGCAAAA...TTCAAAATGAGTAG Chromosome
   [4]  1122 ATGATTTTAAAAGAA...AGGAGGAACCATAG Chromosome
   [5]  1944 GTGAGCGATAAAAAA...ACTTAGATCTATAG Chromosome
   ...   ... ...
[2258]  1389 GTGGCACAGACAGAG...GTTTAGGTAAATAG Chromosome
[2259]   801 ATGGCAATTTTTACT...CTAGTGAGATGTAA Chromosome
[2260]   834 GTGAAAAAGCACTTA...GTAGGCGCAAGTGA Chromosome
[2261]   345 ATGAGAAAGTCATAT...TAGATGAGCATTAA Chromosome
[2262]   135 ATGAAGCGCACATTT...TATTATCTGCATAG Chromosome
>
```

転写物配列取得

①のfastaオブジェクトをそのままFASTA形式で保存すると、②で見えているがままのdescription情報が書きだされる。つまり、すべて”Chromosome”になってしまう

```
#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)    #配列情報を取得した結果をfastaに格納
fasta                                   #確認してるだけです

#後処理(description部分を変更)
names(fasta) <- paste(seqnames(hoge), start(ranges(hoge)),#"染色体名_start_end"に変更
                      end(ranges(hoge)), sep="_")#"染色体名_start_end"に変更
fasta                                   #確認してるだけです

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

①

R Console

```
> fasta <- getSeq(FaFile(in_f1), hoge)    #配列情報を取得した$
> fasta                                   #確認してるだけです
  A DNAStringSet instance of length 2262
        width seq                                         names               $
    [1]  1317 GTGACTGATTTAGAA...AGCTAAAGCCATAG            Chromosome
    [2]  1140 ATGAAATTTACAATT...TTAGAACTTACTAA            Chromosome
    [3]   225 GTGCAAGAAGCAAAA...TTCAAAATGAGTAG            Chromosome
    [4]  1122 ATGATTTTAAAAGAA...AGGAGGAACCATAG            Chromosome
    [5]  1944 GTGAGCGATAAAAAA...ACTTAGATCTATAG            Chromosome
    ...   ... ...
 [2258]  1389 GTGGCACAGACAGAG...GTTTAGGTAAATAG            Chromosome
 [2259]   801 ATGGCAATTTTTACT...CTAGTGAGATGTAA            Chromosome
 [2260]   834 GTGAAAAAGCACTTA...GTAGGCGCAAGTGA            Chromosome
 [2261]   345 ATGAGAAAGTCATAT...TAGATGAGCATTAA            Chromosome
 [2262]   135 ATGAAGCGCACATTT...TATTATCTGCATAG            Chromosome
> |
```

②

転写物配列取得

```
#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)    #配列情報を取得し
fasta                                   #確認してるだけで

#後処理(description部分を変更)
names(fasta) <- paste(seqnames(hoge), start(ranges(hoge)),#"染色体名
                      end(ranges(hoge)), sep="_")#"染色体名_start_en
fasta                                   #確認してるだけです

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

①

②

赤枠部分で行っているのは、description部分の記述内容を“Chromosome_start_end”として、どこの座標由来の塩基配列かがわかるようにしている。①pasteは、文字列を②sepオプションで指定した文字を間に挟んで連結する関数。③の例をみれば挙動がわかると期待

③

R Console

```
> paste("uge", "age", sep="_")
[1] "uge_age"
> seqnames(hoge)
factor-Rle of length 2262 with 1 run
  Lengths:       2262
  Values : Chromosome
Levels(1): Chromosome
> ranges(hoge)
IRanges of length 2262
          start     end width
[1]         360    1676  1317
[2]        1852    2991  1140
[3]        3233    3457   225
[4]        3467    4588  1122
[5]        4588    6531  1944
...         ...     ...   ...
[2258] 2273924 2275312  1389
[2259] 2275488 2276288   801
[2260] 2276455 2277288   834
[2261] 2277304 2277648   345
[2262] 2277719 2277853   135
>
```

転写物配列取得

①description部分が変わっていることがわかる。これを眺めるだけで、出力ファイルをみなくてもうまくいっていると判断できる（と油断していると時々落とし穴があるので注意）

```
#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)    #配列情報を取得した結果をfastaに格納
fasta                                   #確認してるだけです

#後処理(description部分を変更)
names(fasta) <- paste(seqnames(hoge), start(ranges(hoge)),#"染色体名_start_end"に変更
                      end(ranges(hoge)), sep="_")#"染色体名_start_end"に変更
fasta                                   #確認してるだけです

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

R Console

```
> #後処理(description部分を変更)
> names(fasta) <- paste(seqnames(hoge), start(ranges(hoge))$
+                       end(ranges(hoge)), sep="_")#"染色$
> fasta                                   #確認してるだけで$
  A DNAStringSet instance of length 2262
       width seq                          names
   [1]  1317 GTGACTGATTT...AAAGCCATAG Chromosome_360_1676
   [2]  1140 ATGAAATTTAC...AACTTACTAA Chromosome_1852_2991
   [3]   225 GTGCAAGAAGC...AAATGAGTAG Chromosome_3233_3457
   [4]  1122 ATGATTTTAAA...GGAACCATAG Chromosome_3467_4588
   [5]  1944 GTGAGCGATAA...AGATCTATAG Chromosome_4588_6531
    ...   ... ...
[2258]  1389 GTGGCACAGAC...AGGTAAATAG Chromosome_227392...
[2259]   801 ATGGCAATTTT...TGAGATGTAA Chromosome_227548...
[2260]   834 GTGAAAAAGCA...GCGCAAGTGA Chromosome_227645...
[2261]   345 ATGAGAAAGTC...TGAGCATTAA Chromosome_227730...
[2262]   135 ATGAAGCGCAC...ATCTGCATAG Chromosome_227771...
>
```

①

課題2

①このfastaオブジェクトを入力として、転写物数、塩基長の最大(max)・最小(min)・平均(mean)を示せ

```
#本番(配列取得)
fasta <- getSeq(FaFile(in_f1), hoge)    #配列情報を取得した結果をfastaに格納
fasta                                   #確認してるだけです

#後処理(description部分を変更)
names(fasta) <- paste(seqnames(hoge), start(ranges(hoge)),#"染色体名_start_end"に変更
                      end(ranges(hoge)), sep="_")#"染色体名_start_end"に変更
fasta                                   #確認してるだけです

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

R Console

```
> #後処理(description部分を変更)
> names(fasta) <- paste(seqnames(hoge), start(ranges(hoge))$
+                       end(ranges(hoge)), sep="_")#"染色$
> fasta                                    #確認してるだけで$
  A DNAStringSet instance of length 2262
       width seq                          names
    [1]  1317 GTGACTGATTT...AAAGCCATAG Chromosome_360_1676
    [2]  1140 ATGAAATTTAC...AACTTACTAA Chromosome_1852_2991
    [3]   225 GTGCAAGAAGC...AAATGAGTAG Chromosome_3233_3457
    [4]  1122 ATGATTTTAAA...GGAACCATAG Chromosome_3467_4588
    [5]  1944 GTGAGCGATAA...AGATCTATAG Chromosome_4588_6531
    ...   ... ...
 [2258]  1389 GTGGCACAGAC...AGGTAAATAG Chromosome_227392...
 [2259]   801 ATGGCAATTTT...TGAGATGTAA Chromosome_227548...
 [2260]   834 GTGAAAAAGCA...GCGCAAGTGA Chromosome_227645...
 [2261]   345 ATGAGAAAGTC...TGAGCATTAA Chromosome_227730...
 [2262]   135 ATGAAGCGCAC...ATCTGCATAG Chromosome_227771...
> 
```

Contents

- トランスクリプトーム解析技術の原理や特徴
  - マイクロアレイとRNA-seq、遺伝子≠転写物
  - 様々な解析目的、トランスクリプトーム（転写物）配列取得
  - アノテーションファイルの読み込み（Rで転写物配列取得のイントロ）
  - Rで転写物配列取得（アノテーションファイルとゲノム情報ファイルから）
  - マイクロアレイの特徴
  - 発現データベース（DB）
- 発現DBからのプローブレベルデータ取得
  - Affymetrix GeneChip
  - R経由（教科書の§2.2.1）
- Affymetrix GeneChipデータ前処理法を実行

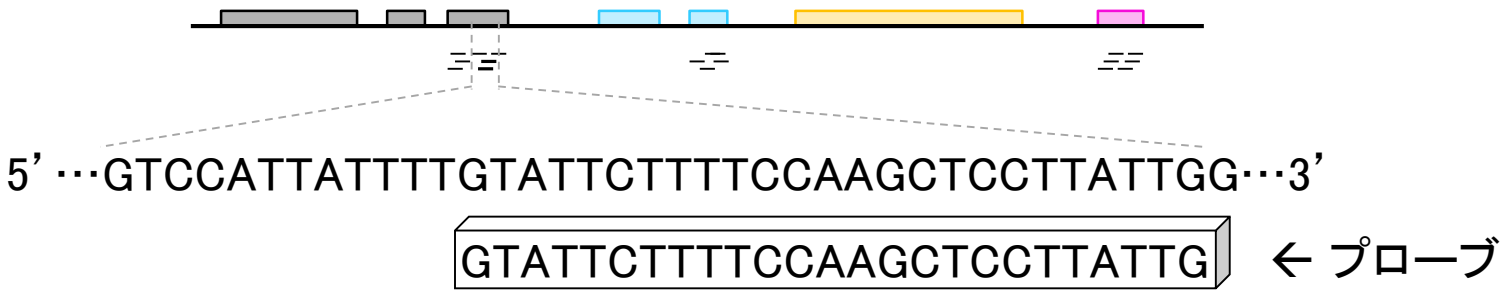
①(Rで)マイクロアレイデータ解析のほうです

発現DB

ステレオタイプなイメージ

マイクロアレイ(microarray)が提供されている生物種で機能解析を目的とする場合はお手軽かも。①2016年の講義資料まではこの偏りを実際に調べましたが、今年はもうやりません

- マイクロアレイの長所
  - 取り扱いやすいデータ量(~100Mb程度)
  - 長年の実績:解析手法がほぼ確立。(WindowsのRのみで解析可能)
  - 検査用チップが利用可能(MammaPrintなど)
- マイクロアレイの短所
  - 解析可能範囲が搭載転写物に限定
  - プローブが3'末端に偏っている(**3'発現解析用アレイ**) ①
  - ダイナミックレンジが狭い



①公共発現DBを眺めてみる

発現DB

(Rで)マイクロアレイデータ解析

(last modified 2015/04/24, since 2005)

What's

- 門田
  知見
  マイク
  す。(
- お知
  料な

- はじめ
- 過去
- イン
- イン
- イン
- イン
- イン
- イン
- イン
- イン

- (削除予定)Rのインストールと起動(last modified 2014/05/14)
- (削除予定)Rの昔のバージョンのインストール(last modified 2012/04/07)
- 使用例(初心者向け)(last modified 2011/09/15)
- サンプルデータ (last modified 2014/06/02)
- 書籍 | について (last modified 2014/05/12)
- 書籍 | トランスクリプトーム解析 | 1.1 はじめに (last modified 2014/05/09)
- 書籍 | トランスクリプトーム解析 | 1.2.1 原理(Affymetrix 3'発現アレイ) (last modified 2014/05/12)
- 書籍 | トランスクリプトーム解析 | 1.2.2 最近の知見 (last modified 2014/05/09)
- 書籍 | トランスクリプトーム解析 | 2.2.1 生データ(プローブレベルデータ)取得 (last modified 2014/04/18)
- 書籍 | トランスクリプトーム解析 | 2.2.2 データの正規化(基礎) (last modified 2014/04/17)
- 書籍 | トランスクリプトーム解析 | 2.2.3 データの正規化(計算例) (last modified 2014/04/18)
- 書籍 | トランスクリプトーム解析 | 2.2.4 データの正規化(その他) (last modified 2014/04/18)
- 書籍 | トランスクリプトーム解析 | 2.2.5 アノテーション情報 (last modified 2014/04/18)
- 書籍 | トランスクリプトーム解析 | 3.2.1 クラスタリング(データ変換や距離の定義など) (last modified 2014/05/16)
- 書籍 | トランスクリプトーム解析 | 3.2.2 実験デザイン、データ分布、統計解析との関係 (last modified 2014/04/18)
- 書籍 | トランスクリプトーム解析 | 3.2.3 多重比較問題 (last modified 2014/04/19)
- 書籍 | トランスクリプトーム解析 | 3.2.4 各種プロット(M-A plotや平均-分散プロットなど) (last modified 2014/04/19)
- 書籍 | トランスクリプトーム解析 | 4.2.1 2群間比較 (last modified 2014/04/19)
- 書籍 | トランスクリプトーム解析 | 4.2.2 他の実験デザイン(paired, multi-factor, 3群間) (last modified 2014/04/19)
- 書籍 | トランスクリプトーム解析 | 4.2.3 多群間比較(特異的発現パターン) (last modified 2014/04/20)
- イントロ | 発現データ取得 | 公共DBから ① modified 2014/05/11)
- イントロ | 発現データ取得 | inSilicoDb(Tan mau 2011) (last modified 2013/08/20)
- イントロ | 発現データ取得 | ArrayExpress(Kauffmann 2009) (last modified 2014/05/15)推奨
- イントロ | 発現データ取得 | GEOquery(Davis 2007) (last modified 2013/08/20)
- イントロ | アノテーション情報取得 | 公共DB(GEO)から (last modified 2013/08/18)

発現DB

①NCBI（アメリカ）が提供する②GEOで、どれだけのデータが登録されているかを眺めるのは③ここ

イントロ | 発現データ取得 | 公共DBから

遺伝子発現（主にマイクロアレイ）データベースをリストアップします。

一次データベース

- GEO: Barrett et al., Nucleic Acids
  - GSE7623（ラット24サンプル
  - GSE30533（ラット10サンプル
  - GSE2361（ヒト36サンプル,
  - GSE10246（マウス182サン
  - GSE1133（ヒトとマウス438
  - GSE5364（ヒト341サンプル,
  - GSE15998（マウス106サン
- ArrayExpress: Rustici et al., Nucle
  - GSE7623（ラット24サンプル
  - GSE30533（ラット10サンプル
  - GSE2361（ヒト36サンプル,
  - GSE10246（マウス182サン
  - GSE1133（リンク先なし）: Su
  - GSE5364（ヒト341サンプル,
  - GSE15998（マウス106サン

二次データベース

- inSilico Db: Coletta et al., Genome
- BioGPS: Wu et al., Nucleic Acids
- Expression Atlas: Petryszak et al.,
- CellFinder: Stachelscheid et al., N

①

②

NCBI Resources How To Sign in to NCBI

GEO Home | Documentation | Query & Browse | Email GEO

Gene Expression Omnibus

GEO is a public functional genomics data repository supporting MIAME-compliant data submissions. Array- and sequence-based data are accepted. Tools are provided to help users query and download experiments and curated gene expression profiles.

Gene Expression Omnibus

Keyword or GEO Accession Search

Getting Started

- Overview
- FAQ
- About GEO DataSets
- About GEO Profiles
- About GEO2R Analysis
- How to Construct a Query
- How to Download Data

Tools

- Search for Studies at GEO DataSets
- Search for Gene Expression at GEO Profiles
- Search GEO Documentation
- Analyze a Study with GEO2R
- GEO BLAST
- Programmatic Access
- FTP Site

Browse Content

- Repository Browser
- DataSets: 4348
- Series: 83648
- Platforms: 17128
- Samples: 2050483

③

Clough and Barrett., *Methods Mol Biol.*, **1418**: 93-110, 2016

発現DB

①EMBL-EBI（ヨーロッパ）が提供する②ArrayExpressで、どれだけのデータが登録されているかを眺めるのは③ここ

イントロ | 発現データ取得 | 公共DBから

遺伝子発現（主にマイクロアレイ）データベースをリストアップします。

一次データベース

- GEO: Barrett et al., Nucleic Acids Res., 2013
  - GSE7623（ラット24サンプル, 62MB）: Nakai et al., BBB, 2008
  - GSE30533（ラット10サンプル, 25MB）: Kamei et al., PLoS One, 2013
  - GSE2361（ヒト36サンプル, 130MB）: Ge et al., Genomics, 2005
  - GSE10246（マウ
  - GSE1133（ヒトと
  - GSE5364（ヒト34
  - GSE15998（マウ
- ArrayExpress: Rustici e
  - GSE7623（ラット2
  - GSE30533（ラット
  - GSE2361（ヒト36
  - GSE10246（マウ
  - GSE1133（リンク
  - GSE5364（ヒト34
  - GSE15998（マウ

二次データベース

- inSilico Db: Coletta et a
- BioGPS: Wu et al., Nuc
- Expression Atlas: Petry
- CellFinder: Stachelsche

②

EMBL-EBI ① Services Research Training About us EMBL-EBI Hinxton

ArrayExpress

Search

Examples: E-MEXP-31, cancer, p53, Geuvadis

advanced search

Home Browse Submit Help About ArrayExpress Contact Us Login

ArrayExpress – functional genomics data

ArrayExpress Archive of Functional Genomics Data stores data from high-throughput functional genomics experiments, and provides these data for reuse to the research community.

Data Content

Updated ③erday at 03:00

- 69875 experiments
- 2209892 assays
- 45.34 TB of archived data

Kolesnikov et al., *Nucleic Acids Res.*, **43**: D1113-1116, 2015

発現DB

2つのDB間で用語の統一はなされていないことがわかる

NCBI GEO

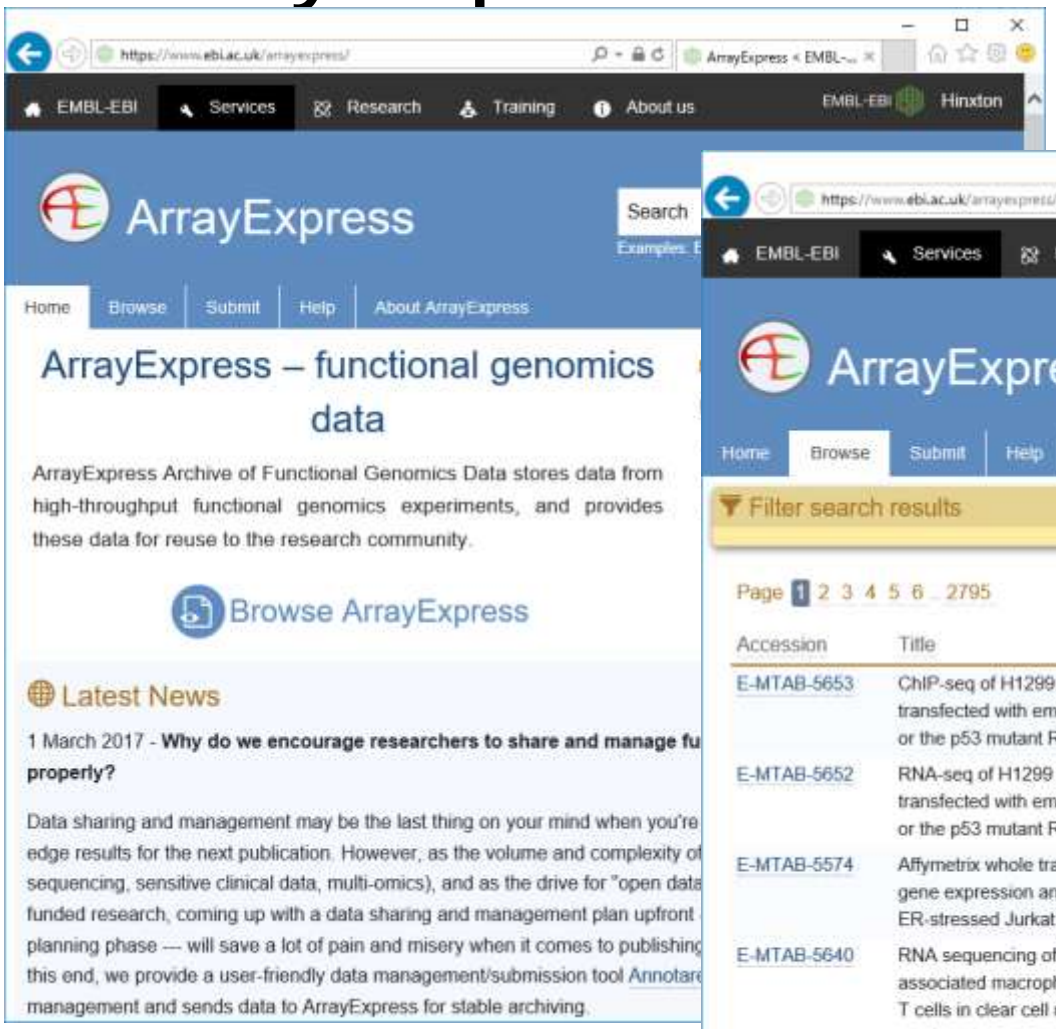
DataSets:	4348
Series:	83648
Platforms:	17128
Samples:	2050483

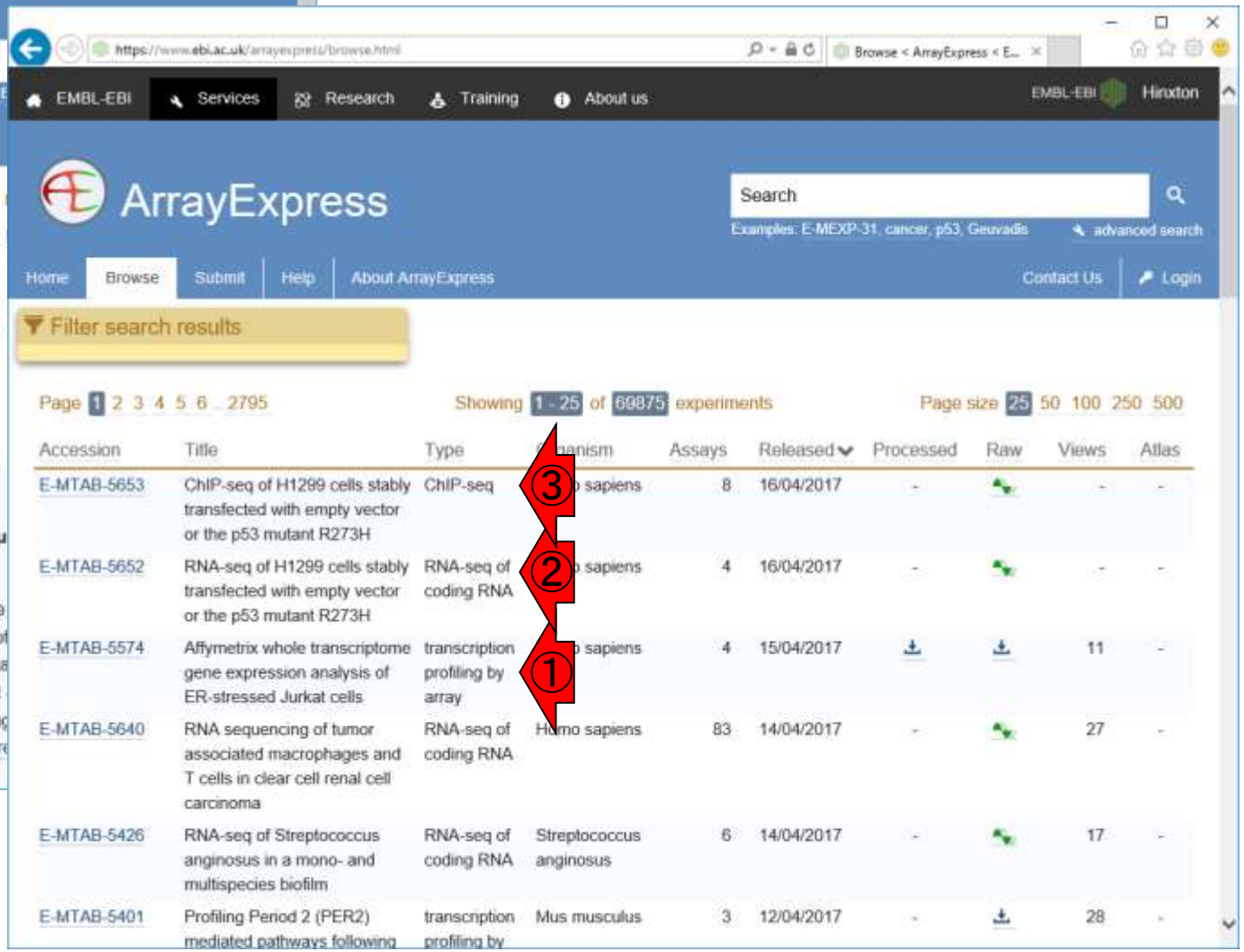
EMBL-EBI ArrayExpress

- 69875 experiments
- 2209892 assays
- 45.34 TB of archived data

ArrayExpressには…

①マイクロアレイデータだけでなく②RNA-seqデータ、そして③ChIP-seqデータも格納されています





発現DB

①GEO中で使われている②用語（DataSets, Series, Platforms, Samples）の説明は③ここ

イントロ | 発現データ取得 | 公共DBから

遺伝子発現（主にマイクロアレイ）データベースをリストアップします。

一次データベース

- GEO: Barrett et al., Nucleic Acids
  - GSE7623(ラット24サンプル
  - GSE30533(ラット10サンプル
  - GSE2361(ヒト36サンプル,
  - GSE10246(マウス182サン
  - GSE1133(ヒトとマウス438
  - GSE5364(ヒト341サンプル,
  - GSE15998(マウス106サン
- ArrayExpress: Rustici et al., Nucle
  - GSE7623(ラット24サンプル
  - GSE30533(ラット10サンプル
  - GSE2361(ヒト36サンプル,
  - GSE10246(マウス182サン
  - GSE1133(リンク先なし): Su
  - GSE5364(ヒト341サンプル,
  - GSE15998(マウス106サン

二次データベース

- inSilico Db: Coletta et al., Genome
- BioGPS: Wu et al., Nucleic Acids
- Expression Atlas: Petryszak et al.,
- CellFinder: Stachelscheid et al., N

① ②

NCBI Resources How To Sign in to NCBI

GEO Home | Documentation | Query & Browse | Email GEO

Gene Expression Omnibus

GEO is a public functional genomics data repository supporting MIAME-compliant data submissions. Array- and sequence-based data are accepted. Tools are provided to help users query and download experiments and curated gene expression profiles.

Gene Expression Omnibus

Keyword or GEO Accession Search

Getting Started

Overview ③
FAQ
About GEO DataSets
About GEO Profiles
About GEO2R Analysis
How to Construct a Query
How to Download Data

Tools

Search for Studies at GEO DataSets
Search for Gene Expression at GEO Profiles
Search GEO Documentation
Analyze a Study with GEO2R
GEO BLAST
Programmatic Access
FTP Site

Browse Content

Repository Browser

DataSets:	4348
Series:	83648
Platforms:	17128
Samples:	2050483

発現DB

①Platformsは、**大まかにはアレイやNGS機器のこと。**2017年4月19日現在、17,128種類登録されている

NCBI

GEO Gene Expression Omnibus

GEO Publications | FAQ

NCBI » GEO » Info » GEO Overview

GEO Overview

Platform

Platform records are supplied by submitters

A Platform record is composed of a summary description of the array or sequencer and, for array-based Platforms, a data table defining the array template.Each Platform record is assigned a unique and stable GEO accession number (GPLxxx). A Platform may reference many Samples that have been submitted by multiple submitters.

Example Platform record »

Sample

Sample records are supplied by submitters

A Sample record describes the conditions under which an individual Sample was handled, the manipulations it underwent, and the abundance measurement of each element derived from it. Each Sample record is assigned a unique and stable GEO accession number (GSMxxx). A Sample entity must reference only one Platform and may be included in multiple Series.

Example Sample record »

Series records are supplied by submitters

DataSets:	4348
Series:	83648
Platforms: ①	17128
Samples:	2050483

A	T… a…
B	T… ta… te…
C	**Text description of the biological sample and protocols to which it was subjected**
D	**Text tab-delimited table of processed hybridization result** (may optionally include raw data columns)
E	**Original raw data file, or processed sequence data file**
F	**Text description of the**

Platformsの例(2017年4月)

同じメーカー、同じ生物種でも様々なバージョンのアレイが存在する

- Affymetrix GeneChip
  - Affymetrix Human Genome U133 Plus 2.0 Array: **GPL570**
    - 2003年11月リリース、54,675 probesets、129,096枚の利用実績
  - Affymetrix Human Genome U133A Array: **GPL96**
    - 2002年3月リリース、22,283 probesets、40,059枚
  - Affymetrix Mouse Genome 430 2.0 Array: **GPL1261**
    - 2004年5月リリース、45,101 probesets、50,782枚
  - Affymetrix Rat Genome 230 2.0 Array: **GPL1355**
    - 2004年6月リリース、31,099 probesets、19,455枚
- Illumina BeadChip
  - Illumina HumanHT-12 V4.0 expression beadchip: **GPL10558**
    - 2010年6月リリース、47,323 probes、62,544枚
  - Illumina HumanHT-12 V3.0 expression beadchip: **GPL6947**
    - 2008年6月リリース、49,576 probes、23,066枚
- Agilent Microarray
  - Agilent-014850 Whole Human Genome Microarray 4x44K G4112F: **GPL6480**
    - 2008年2月リリース、41,108 probes、18,886枚

Platformsの例(2016年5月)

同じメーカー、同じ生物種でも様々なバージョンのアレイが存在する

- Affymetrix GeneChip
  - Affymetrix Human Genome U133 Plus 2.0 Array: **GPL570**
    - 2003年11月リリース、54,675 probesets、120,878枚の利用実績
  - Affymetrix Human Genome U133A Array: **GPL96**
    - 2002年3月リリース、22,283 probesets、37,702枚
  - Affymetrix Mouse Genome 430 2.0 Array: **GPL1261**
    - 2004年5月リリース、45,101 probesets、48,043枚
  - Affymetrix Rat Genome 230 2.0 Array: **GPL1355**
    - 2004年6月リリース、31,099 probesets、18,882枚
- Illumina BeadChip
  - Illumina HumanHT-12 V4.0 expression beadchip: **GPL10558**
    - 2010年6月リリース、47,323 probes、49,402枚
  - Illumina HumanHT-12 V3.0 expression beadchip: **GPL6947**
    - 2008年6月リリース、49,576 probes、22,287枚
- Agilent Microarray
  - Agilent-014850 Whole Human Genome Microarray 4x44K G4112F: **GPL6480**
    - 2008年2月リリース、41,108 probes、16,647枚

Platformsの例(2015年5月)

同じメーカー、同じ生物種でも様々なバージョンのアレイが存在する

- Affymetrix GeneChip
  - Affymetrix Human Genome U133 Plus 2.0 Array: **GPL570**
    - 2003年11月リリース、54,675 probesets、105,000枚以上の利用実績
  - Affymetrix Human Genome U133A Array: **GPL96**
    - 2002年3月リリース、22,283 probesets、37,000枚以上
  - Affymetrix Mouse Genome 430 2.0 Array: **GPL1261**
    - 2004年5月リリース、45,101 probesets、43,000枚以上
  - Affymetrix Rat Genome 230 2.0 Array: **GPL1355**
    - 2004年6月リリース、31,099 probesets、17,000枚以上
- Illumina BeadChip
  - Illumina HumanHT-12 V4.0 expression beadchip: **GPL10558**
    - 2010年6月リリース、47,323 probes、33,000枚以上
  - Illumina HumanHT-12 V3.0 expression beadchip: **GPL6947**
    - 2008年6月リリース、49,576 probes、20,000枚以上
- Agilent Microarray
  - Agilent-014850 Whole Human Genome Microarray 4x44K G4112F: **GPL6480**
    - 2008年2月リリース、41,108 probes、14,000枚以上

発現DB

Seriesは、1つの研究プロジェクトなどで用いた複数サンプルからなるグループをまとめたもの。大まかには論文ごとのIDという理解でよい

Platform

Platform records are supplied by submitters

A Platform record is composed of a summary description of the array or sequencer and, for array-based Platforms, a data table defining the array template.Each Platform record is assigned a unique and stable GEO accession number (GPLxxx). A Platform may reference many Samples that have been submitted by multiple submitters.

Example Platform record »

Sample

Sample records are supplied by submitters

A Sample record describes the conditions under which an individual Sample was handled, the manipulations it underwent, and the abundance measurement of each element derived from it. Each Sample record is assigned a unique and stable GEO accession number (GSMxxx). A Sample entity must reference only one Platform and may be included in multiple Series.

Example Sample record »

Series

Series records are supplied by submitters

A Series record links together a group of related Samples and provides a focal point and description of the whole study. Series records may also contain tables describing extracted data, summary conclusions, or analyses. Each Series record is assigned a unique and stable GEO accession number (GSExxx).

Example Series record »

A	Text description of the array or sequencer
B	Text tab-d… table of th… template
C	Text descr… biological protocols … was subje…
D	Text tab-d… table of pr… hybridizat… (may optiona… data columns)
E	Original raw data file, or processed sequence data file
F	Text description of the overall experiment
G	Tar archive of original raw data files, or processed sequence data files

DataSets:	4348
① Series:	83648
Platforms:	17128
Samples:	2050483

Seriesの例

・NGSデータも登録されている
・1論文で1 GSE IDとは限らない
・1 GSE IDで1 GPL IDとは限らない

- Affymetrix GeneChip
  - Ge et al., *Genomics*, 86: 127–141, 2005
    - **GSE2361**、ヒト36サンプル、GPL96を利用
  - Nakai et al., *Biosci Biotechnol Biochem.*, 72: 139–148, 2008
    - **GSE7623**、ラット24サンプル、GPL1355を利用
  - Kamei et al., *PLoS One*, 8: e65732, 2013
    - **GSE30533**、ラット10サンプル、GPL1355を利用
- Illumina BeadChip
  - Sharma et al., *Cancer Cell*, 23: 35–47, 2013
    - **GSE28680**、ヒト24サンプル、GPL10558を利用
- NGSデータも…
  - Neyret-Kahn et al., *Genome Res.*, 23: 1563–1579, 2013
    - **GSE42213**、ヒト26サンプル、GPL10999とGPL11154を利用
      - **GSE42211**、ヒト20サンプル、GPL10999とGPL11154を利用（ChIP-seq）
      - **GSE42212**、ヒト6サンプル、GPL10999を利用（RNA-seq）
  - Huang et al., *Development*, 139: 2161–2169, 2012
    - **GSE36469**、シロイヌナズナ8サンプル、GPL13222を利用

発現DB

Samplesは、登録されているサンプル数。大まかには、使われたアレイの枚数という理解でよい

Platform

Platform records are supplied by submitters

A Platform record is composed of a summary description of the array or sequencer and, for array-based Platforms, a data table defining the array template.Each Platform record is assigned a unique and stable GEO accession number (GPLxxx). A Platform may reference many Samples that have been submitted by multiple submitters.

Example Platform record »

Sample

Sample records are supplied by submitters

A Sample record describes the conditions under which an individual Sample was handled, the manipulations it underwent, and the abundance measurement of each element derived from it. Each Sample record is assigned a unique and stable GEO accession number (GSMxxx). A Sample entity must reference only one Platform and may be included in multiple Series.

Example Sample record »

Series

Series records are supplied by submitters

A Series record links together a group of related Samples and provides a focal point and description of the whole study. Series records may also contain tables describing extracted data, summary conclusions, or analyses. Each Series record is assigned a unique and stable GEO accession number (GSExxx).

Example Series record »

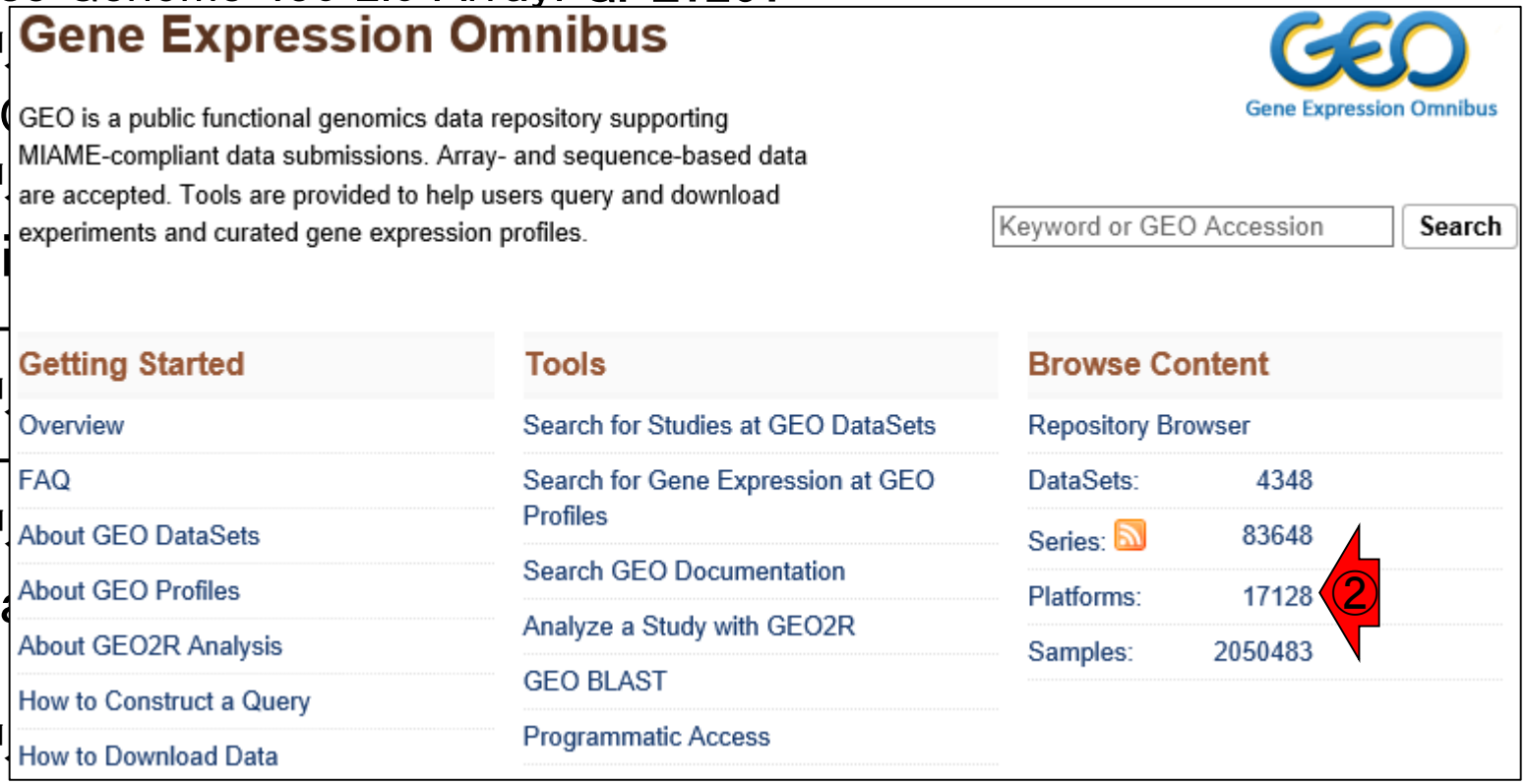
A	Text description of the array or sequencer
B	Text tab-d table of th template
C	Text descr biological protocols was subje
D	Text tab-d table of pr hybridizat (may optiona data columns)
E	Original raw data file, or processed sequence data file
F	Text description of the overall experiment
G	Tar archive of original raw data files, or processed sequence data files

DataSets:	4348
Series:	83648
Platforms:	17128
Samples: ①	2050483

Platformsの例(2017年4月)

①GPL570が129,096枚利用されたという事実は、②から辿って調べました

- Affymetrix GeneChip
  - Affymetrix Human Genome U133 Plus 2.0 Array: **GPL570**
    - 2003年11月リリース、54,675 probesets、129,096枚の利用実績 ①
  - Affymetrix Human Genome U133A Array: **GPL96**
    - 2002年3月リリース、22,283 probesets、40,059枚
  - Affymetrix Mouse Genome 430 2.0 Array: **GPL1261**
    - 2004年5月リ
  - Affymetrix Rat
    - 2004年6月リ
- Illumina BeadChi
  - Illumina Humanh
    - 2010年6月リ
  - Illumina Humanh
    - 2008年6月リ
- Agilent Microarra
  - Agilent-014850
    - 2008年2月リ



Platformsの例

①デフォルトはRelease dateになっている。用いられたサンプル数が多い順にソートして眺める場合は②Samplesのところを2回クリック。（クリックごとに昇順と降順が繰り返される）

Series | Samples | Platforms | DataSets | Summary | Adva

Search | 17,128 platforms | Export | All publications | Page 1 of 857 > >> | Page 1

②

①

Accession	Title	Technology	Organism(s)	Data rows	Samples	Series	Contact	Release date
GPL5132	Galbraith Lab Maize Unigene cDNA microarray slide UG1-2.2	spotted DNA/cDNA	*Zea mays*	16,560	12	1	Tim L. Setter	Apr 19, 2017
GPL21108	Custom NimbleGen 12-plex Bufo marinus, Bufo boreas, Xenopus tropicalis Oligo Expression Array	in situ oligonucleotide	*Anaxyrus boreas* *Rhinella marina* *Xenopus tropicalis*	31,367	71	1	Thomas Poorten	Apr 18, 2017
GPL23307	Agilent-081421 Horse_60K_2016_01_22 021322	in situ oligonucleotide	*Equus caballus*	62,976	96	3	Aline FOURY	Apr 18, 2017
GPL23309	AAFC Clostridium perfringens 3K v1.0	spotted oligonucleotide	*Clostridium perfringens*	3,335	3	1	Dion Lepp	Apr 18, 2017
GPL23312	Agilent-020871 Eucommia ulmoides Eul_4x44K_A	in situ oligonucleotide	*Eucommia ulmoides*	45,220	102	1	Yoshihisa Nakazawa	Apr 18, 2017
GPL23314	Multi-species Autoantigen Microarray 94	spotted peptide or protein	*Homo sapiens* *Mus musculus* *Rattus norvegicus*	94	20	1	yun lian	Apr 18, 2017
GPL23320	Illumina HiSeq 2500 (Streptococcus pyogenes serotype M3)	high-throughput sequencing	*Streptococcus pyogen...*				GEO	Apr 18, 2017
GPL23321	Illumina HiSeq 2500 (Streptococcus pyogenes serotype M1)	high-throughput sequencing	*Streptococcus pyogen...*				GEO	Apr 18, 2017
GPL23306	Exiqon miRCURY LNA microRNA Array, 7th generation - hsa, mmu & rno (miRBase v18.0)	spotted oligonucleotide	*Homo sapiens*	3,542	3	2	SHUAI HUANG	Apr 17, 2017
GPL23308	Illumina HiSeq 2500 (Oryzias latipes)	high-throughput sequencing	*Oryzias latipes*				GEO	Apr 17, 2017
GPL23310	Illumina HiSeq 2000 (coral metagenome)	high-throughput sequencing	*coral metagenome*				GEO	Apr 17, 2017
GPL23311	Illumina MiSeq (1,4-dioxane degrading enrichment culture)	high-throughput sequencing	*1,4-dioxane-degrading...*				GEO	Apr 17, 2017
GPL22601	nCounter Mouse v.1.5 miRNA Expression Assay (Nanostring Technologies)	other	*Mus musculus*	626	11	2	Phillip J Bridges	Apr 14, 2017
GPL22602	nCounter GX Mouse Inflammation Kit (Nanostring Technologies)	other	*Mus musculus*	199	12	2	Phillip J Bridges	Apr 14, 2017
GPL23304	PacBio RS II (Human alphaherpesvirus 1)	high-throughput sequencing	*Human alphaherpesvir...*				GEO	Apr 14, 2017

Platformsの例

①（NGS機器も含まれるため、もはや正確な言い回しではないが…）そのアレイで計測されたサンプル数。トップはGPL570というIDが付与された、2003年11月リリースのAffymetrix Human Genome U133 Plus 2.0 Arrayという正式名称の、**3’発現アレイ**

Series | Samples | Platforms | DataSet

Search | 17,128 platforms | Export | Page 1 of 857 | Page size 20

①

Accession	Title	Technology	Organism(s)	Data rows	Samples	Series	Contact	Release date
GPL570	[HG-U133_Plus_2] Affymetrix Human Genome U133 Plus 2.0 Array	in situ oligonucleotide	*Homo sapiens*	54,675	129096	4594	Affymetrix, Inc.	Nov 07, 2003
GPL13112	Illumina HiSeq 2000 (Mus musculus)	high-throughput sequencing	*Mus musculus*		68755	3807	GEO	Feb 02, 2011
GPL10558	Illumina HumanHT-12 V4.0 expression beadchip	oligonucleotide beads	*Homo sapiens*	48,107	62544	1850	Illumina Inc.	Jun 17, 2010
GPL11154	Illumina HiSeq 2000 (Homo sapiens)	high-throughput sequencing	*Homo sapiens*		62057	4944	GEO	Nov 02, 2010
GPL13534	Illumina HumanMethylation450 BeadChip (HumanMethylation450_15017482)	oligonucleotide beads	*Homo sapiens*	485,577	56226	858	Illumina Inc.	May 13, 2011
GPL1261	[Mouse430_2] Affymetrix Mouse Genome 430 2.0 Array	in situ oligonucleotide	*Mus musculus*	45,101	50782	3869	Affymetrix, Inc.	May 25, 2004
GPL17021	Illumina HiSeq 2500 (Mus musculus)	high-throughput sequencing	*Mus musculus*		46406	1921	GEO	Apr 16, 2013
GPL96	[HG-U133A] Affymetrix Human Genome U133A Array	in situ oligonucleotide	*Homo sapiens*	22,283	40059	1073	Affymetrix, Inc.	Mar 11, 2002
GPL16791	Illumina HiSeq 2500 (Homo sapiens)	high-throughput sequencing	*Homo sapiens*		39606	2151	GEO	Mar 14, 2013
GPL6244	[HuGene-1_0-st] Affymetrix Human Gene 1.0 ST Array [transcript (gene) version]	in situ oligonucleotide	*Homo sapiens*	33,297	30649	1503	Affymetrix, Inc.	Dec 05, 2007
GPL18573	Illumina NextSeq 500 (Homo sapiens)	high-throughput sequencing	*Homo sapiens*		25292	473	GEO	Apr 15, 2014
GPL6246	[MoGene-1_0-st] Affymetrix Mouse Gene 1.0 ST Array [transcript (gene) version]	in situ oligonucleotide	*Mus musculus*	35,557	24034	1950	Affymetrix, Inc.	Dec 05, 2007
GPL6947	Illumina HumanHT-12 V3.0 expression beadchip	oligonucleotide beads	*Homo sapiens*	49,576	23066	493	Illumina Inc.	Jun 10, 2008
GPL1355	[Rat230_2] Affymetrix Rat Genome 230 2.0 Array	in situ oligonucleotide	*Rattus norvegicus*	31,099	19455	631	Affymetrix, Inc.	Jul 20, 2004
GPL6480	Agilent-014850 Whole Human Genome Microarray 4x44K G4112F (Probe Name version)	in situ oligonucleotide	*Homo sapiens*	41,108	18886	767	Agilent Technologies	Feb 11, 2008
GPL8490	Illumina HumanMethylation27 BeadChip (HumanMethylation27_270596_v.1.2)	oligonucleotide beads	*Homo sapiens*	27,578	18708	333	Illumina Inc.	Apr 27, 2009

Platformsの例

①と②を眺めることで、Illumina社のNGS機器であるHiSeq 2000に対して1つのGPL IDが付与されているわけではなく、「NGS機器と適用した生物種」でGPL IDが付与されていることがわかる。3年前は、①Illumina HiSeq 2000 (Mus musculus)に対して、別のID (GPL18672)も割り当てられていた…

Series | Samples | Platforms | DataSets | Summ

Search | 17,128 platforms

Accession	Title	Technology	Organism					
GPL570	[HG-U133_Plus_2] Affymetrix Human Genome U133 Plus 2.0 Array	in situ oligonucleotide	*Homo sapiens*	54,675	129096	4594	Affymetrix, Inc.	Nov 07, 2003
GPL13112	Illumina HiSeq 2000 (Mus musculus)	high-throughput sequencing	*Mus musculus*		68755	3807	GEO	Feb 02, 2011
GPL10558	Illumina HumanHT-12 V4.0 expression beadchip	oligonucleotide beads	*Homo sapiens*	48,107	62544	1850	Illumina Inc.	Jun 17, 2010
GPL11154	Illumina HiSeq 2000 (Homo sapiens)	high-throughput sequencing	*Homo sapiens*		62057	4944	GEO	Nov 02, 2010
GPL13534	Illumina HumanMethylation450 BeadChip (HumanMethylation450_15017482)	oligonucleotide beads	*Homo sapiens*	485,577	56226	858	Illumina Inc.	May 13, 2011
GPL1261	[Mouse430_2] Affymetrix Mouse Genome 430 2.0 Array	in situ oligonucleotide	*Mus musculus*	45,101	50782	3869	Affymetrix, Inc.	May 25, 2004
GPL17021	Illumina HiSeq 2500 (Mus musculus)	high-throughput sequencing	*Mus musculus*		46406	1921	GEO	Apr 16, 2013
GPL96	[HG-U133A] Affymetrix Human Genome U133A Array	in situ oligonucleotide	*Homo sapiens*	22,283	40059	1073	Affymetrix, Inc.	Mar 11, 2002
GPL16791	Illumina HiSeq 2500 (Homo sapiens)	high-throughput sequencing	*Homo sapiens*		39606	2151	GEO	Mar 14, 2013
GPL6244	[HuGene-1_0-st] Affymetrix Human Gene 1.0 ST Array [transcript (gene) version]	in situ oligonucleotide	*Homo sapiens*	33,297	30649	1503	Affymetrix, Inc.	Dec 05, 2007
GPL18573	Illumina NextSeq 500 (Homo sapiens)	high-throughput sequencing	*Homo sapiens*		25292	473	GEO	Apr 15, 2014
GPL6246	[MoGene-1_0-st] Affymetrix Mouse Gene 1.0 ST Array [transcript (gene) version]	in situ oligonucleotide	*Mus musculus*	35,557	24034	1950	Affymetrix, Inc.	Dec 05, 2007
GPL6947	Illumina HumanHT-12 V3.0 expression beadchip	oligonucleotide beads	*Homo sapiens*	49,576	23066	493	Illumina Inc.	Jun 10, 2008
GPL1355	[Rat230_2] Affymetrix Rat Genome 230 2.0 Array	in situ oligonucleotide	*Rattus norvegicus*	31,099	19455	631	Affymetrix, Inc.	Jul 20, 2004
GPL6480	Agilent-014850 Whole Human Genome Microarray 4x44K G4112F (Probe Name version)	in situ oligonucleotide	*Homo sapiens*	41,108	18886	767	Agilent Technologies	Feb 11, 2008
GPL8490	Illumina HumanMethylation27 BeadChip (HumanMethylation27_270596_v.1.2)	oligonucleotide beads	*Homo sapiens*	27,578	18708	333	Illumina Inc.	Apr 27, 2009

Contents

- トランスクリプトーム解析技術の原理や特徴
  - マイクロアレイとRNA-seq、遺伝子≠転写物
  - 様々な解析目的、トランスクリプトーム（転写物）配列取得
  - アノテーションファイルの読み込み（Rで転写物配列取得のイントロ）
  - Rで転写物配列取得（アノテーションファイルとゲノム情報ファイルから）
  - マイクロアレイの特徴
  - 発現データベース（DB）
- 発現DBからのプローブレベルデータ取得
  - Affymetrix GeneChip
  - R経由（教科書の§2.2.1）
- Affymetrix GeneChipデータ前処理法を実行

デバイスも進歩しているが、①3’アレイで蓄積された過去のデータと比較しやすいため、3’アレイが今でもよく利用されます

Affymetrix GeneChip

- **3’発現アレイ** → exon array → transcriptome array
  - Affymetrix Human Transcriptome Array (HTA 2.0)
  - Furney et al., *Cancer Discov*., **3**: 1122-1129, 2013.
  - GPL17585(exon level)
  - GPL17586(gene level)

Affymetrix GeneChip

- **3’発現アレイ** → exon array → transcrip
  - Affymetrix Human Transcriptome Array (H
  - Furney et al., *Cancer Discov*., **3**: 1122-112
  - GPL17585(exon level)
  - GPL17586(gene level)

①赤枠内で示すように、1つの遺伝子（転写物）の発現レベルを調べるのに、通常10個程度のプローブを利用。これを**プローブセット**（probeset）という。プローブごとに測定されたシグナル情報からなる数値ベクトルをスカラー値としてまとめる必要がある。プローブセット(≒遺伝子or転写物)の発現量算出に相当。

生データ取得

GSE7623のプローブレベルデータ取得を行うやり方を示します

- Affymetrix GeneChip
  - Ge et al., *Genomics*, 86: 127-141, 2005
    - **GSE2361**、ヒト36サンプル、GPL96を利用
  - Nakai et al., *BBB.*, 72: 139-148, 2008
    - **GSE7623**、ラット24サンプル、GPL1355を利用
  - Kamei et al., *PLoS One*, 8: e65732, 2013
    - **GSE30533**、ラット10サンプル、GPL1355を利用
- Illumina BeadChip
  - Sharma et al., *Cancer Cell*, 23: 35-47, 2013
    - **GSE28680**、ヒト24サンプル、GPL10558を利用
- NGSデータも…
  - Neyret-Kahn et al., *Genome Res.*, 23: 1563-1579, 2013
    - **GSE42213**、ヒト26サンプル、GPL10999とGPL11154を利用
      - **GSE42211**、ヒト20サンプル、GPL10999とGPL11154を利用（ChIP-seq）
      - **GSE42212**、ヒト6サンプル、GPL10999を利用（RNA-seq）
  - Huang et al., *Development*, 139: 2161-2169, 2012
    - **GSE36469**、シロイヌナズナ8サンプル、GPL13222を利用

• イントロ | 発現データ取得 | ArrayExpress(Kauffmann_2009) 参考

結論としては①ArrayExpressというRパッケージを利用するR経由のほうがお手軽。②例題3

R経由で生データ取得

- 書籍 | トランスクリプトーム解析 | 4.2.3 多群間比較(特異的発現パターン) (last modified 2014/04/20)
- イントロ | 発現データ取得 | 公共DBから (last modified 2014/05/11)
- イントロ | 発現データ取得 | inSilicoDb(Taminau_2011) (last modified 2015/05/11) NEW
- イントロ | 発現データ取得 | ArrayExpress(Kauffmann_2009) (last modified 2014/05/15)推奨
- イントロ | 発現データ取得 | GEOquery(Davis_2007) (last modified 2013/08/20)
- イントロ | アノテーション情報取得 | 公共DB(GEO)から (last modified 2013/08/18)
- イントロ |
- イントロ |
- イントロ |

①

イントロ | 発現データ取得 | ArrayExpress(Kauffmann_2009) NEW

マイクロアレイデータベースArrayExpressに登録されているデータをArrayExpressというRパッケージで取得するやり方を示します。GEO IDでも検索可能であり、CELファイルデータも取得可能、任意のpreprocessing法を適用可能、などの利点からこのパッケージ経由での利用をお勧めします。
「ファイル」-「ディレクトリの変更」でファイルを保存したいディレクトリに移動し以下をコピペ。

1. AffymetrixデータE-MEXP-1422 (Bourgon et al., PNAS, 2010)のCELファイルを取得し、RMA法(Irizarry et al., Biostatistics, 2003)を実行して得られた発現情報を取得したい場合:

以下のArrayExpress関数のオプションをsave=Fからsave=Tに変更すると、CELファイルなどを含む全データのダウンロードも同時に行ってくれます。が、そんなことをいちいちやらなくてもReadAffy関数を用いて読み込んだ状態と同じなので直接RMA(Irizarry et al., Biostatistics, 2003)などの任意の正規化法を適用可能です。

```
out_f <-
param <- E

#必要なパッケ
library(Arr
library(aff
```

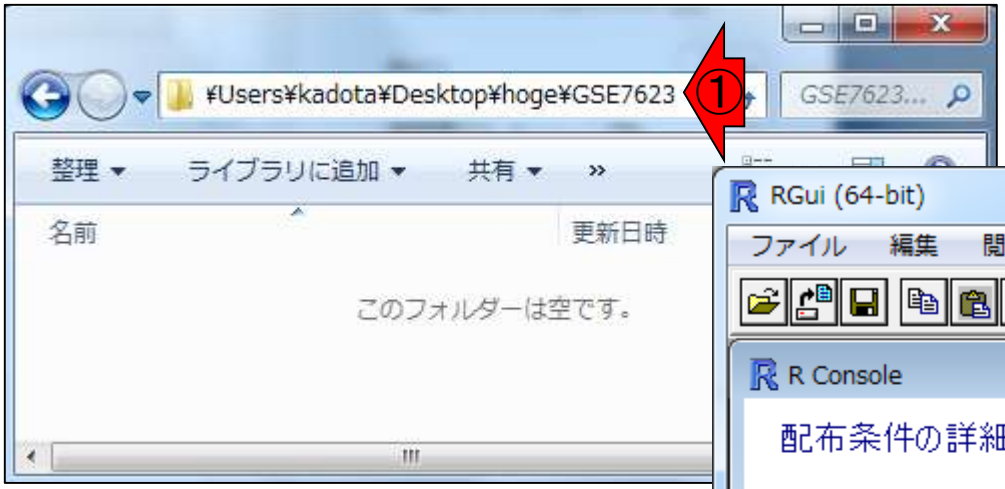
②

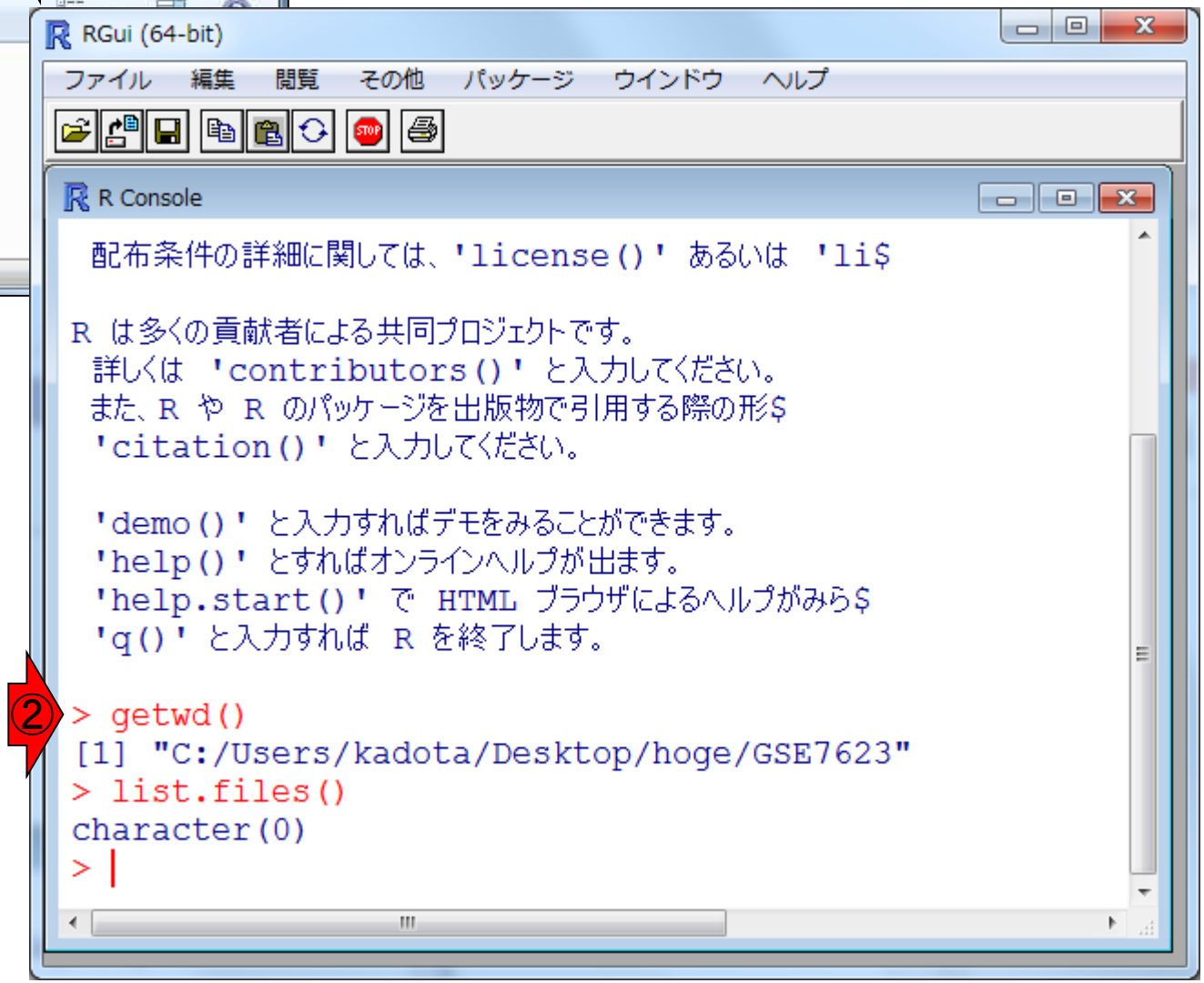
3. AffymetrixデータGSE7623 (Nakai et al., BBB, 2008)のCELファイルを取得したい場合:

```
param <- "GSE7623"                          #入手したいIDを指定

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

#前処理(データ取得)
hoge <- getAE(param, type="raw", extract=F)#paramで指定したIDの生データをダウン
```

R経由で生データ取得

①デスクトップにhogeフォルダ、およびその中にGSE7623フォルダを作成。Rを起動し、②作業ディレクトリをそこに変更





コピペ。実習ではやらないで!

R経由で生データ取得

3. AffymetrixデータGSE7623 (Nakai et al., BBB, 2008)のCELファイルを取得したい場合:

```
param <- "GSE7623"                    #入手したいIDを指定

#必要なパッケージをロー
library(ArrayExpress                  み込み

#前処理(データ取得)
hoge <- getAE(param,                  定したIDの生データをダウン
```

切り取り(T)
コピー(C) ①
貼り付け
すべて選択(A)
印刷(I)...
印刷プレビュー(N)...

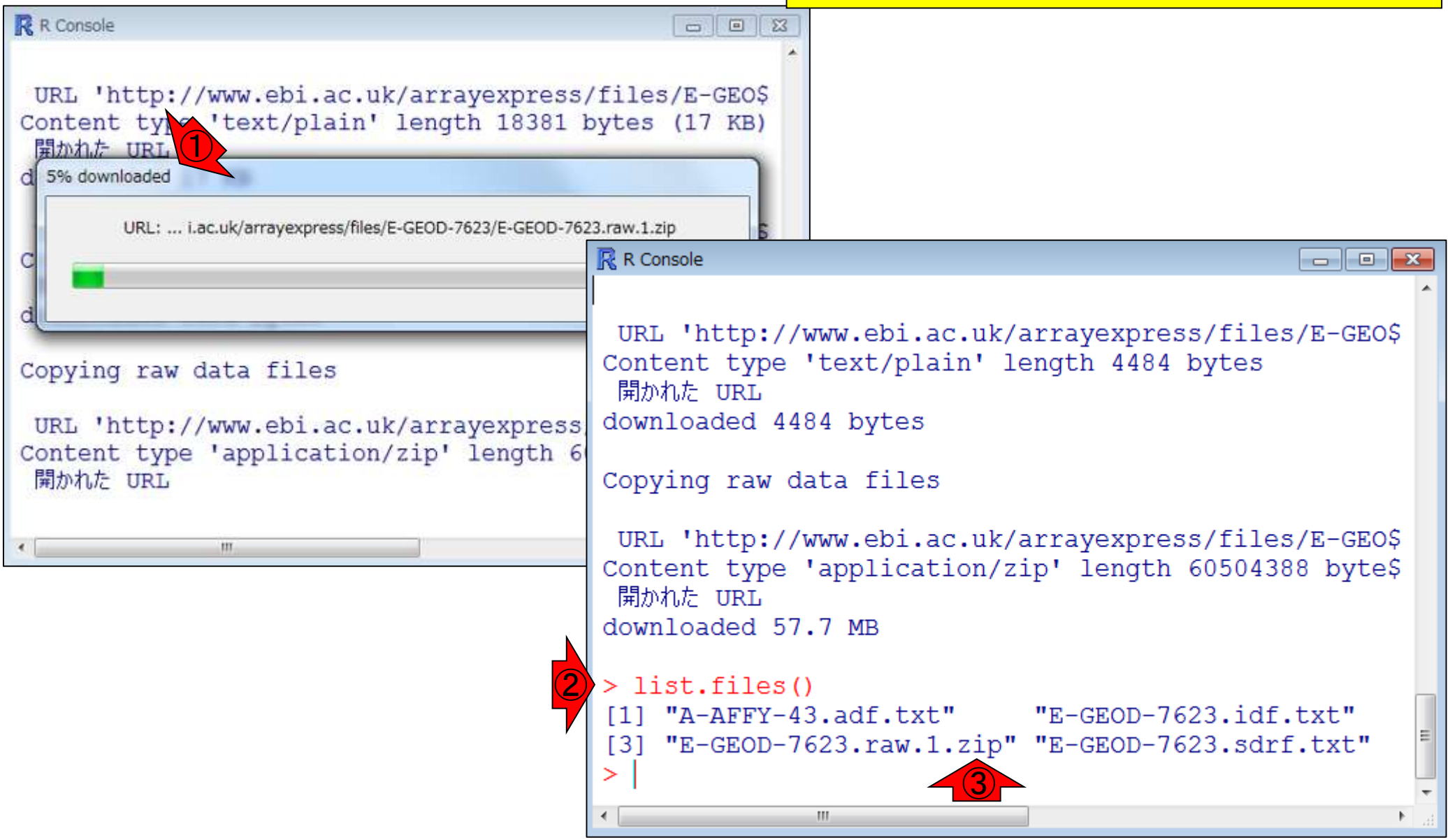
R Console

```
 配布条件の詳細に関しては、'license()' あるいは 'li$

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

 'demo()' と入力すればデモを
 'help()' とすればオンライン
 'help.start()' で HTM
 'q()' と入力すれば R を終

> getwd()
[1] "C:/Users/kadota/
> list.files()
character(0)
>
```

コピー Ctrl+C
ペースト Ctrl+V ②
コマンドのみペースト
コピー&ペースト Ctrl+X
ウインドウの消去 Ctrl+L
全て選択
✓ バッファに出力 Ctrl+W
ウィンドウを常にトップに置く

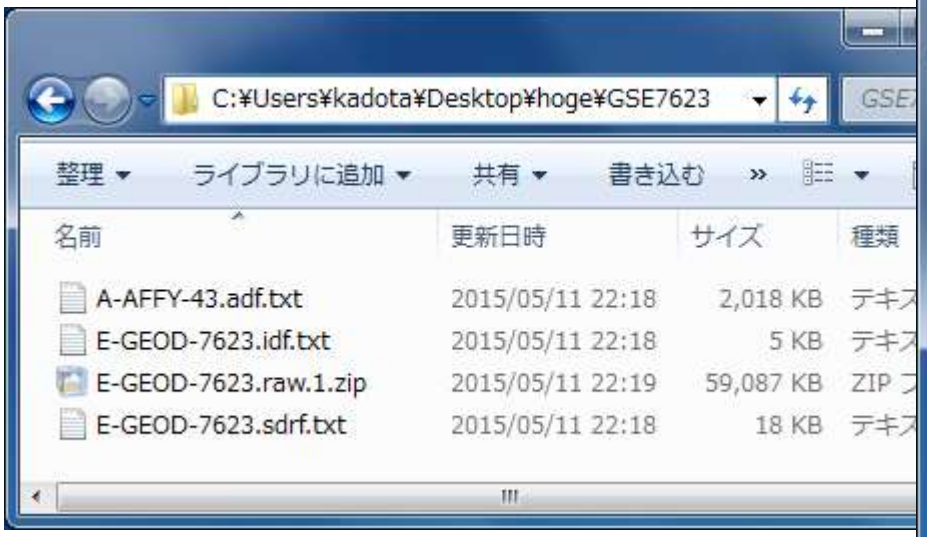
R経由で生データ取得

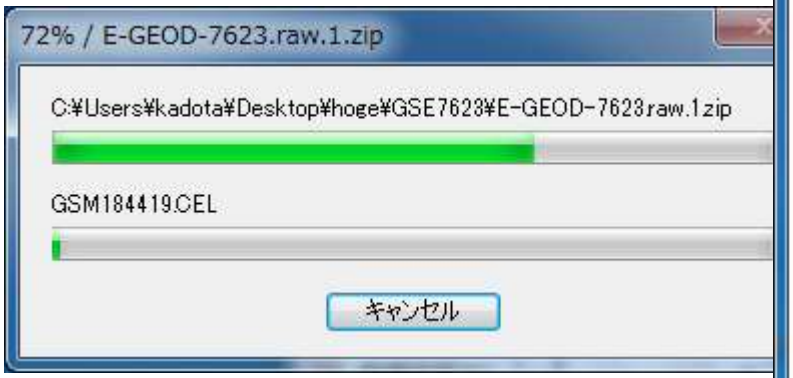
①コピペ実行中、コピペ実行後の状態。②4つのファイルがダウンロードされている。このうち、③zipファイル(の中身が目的物)を解凍

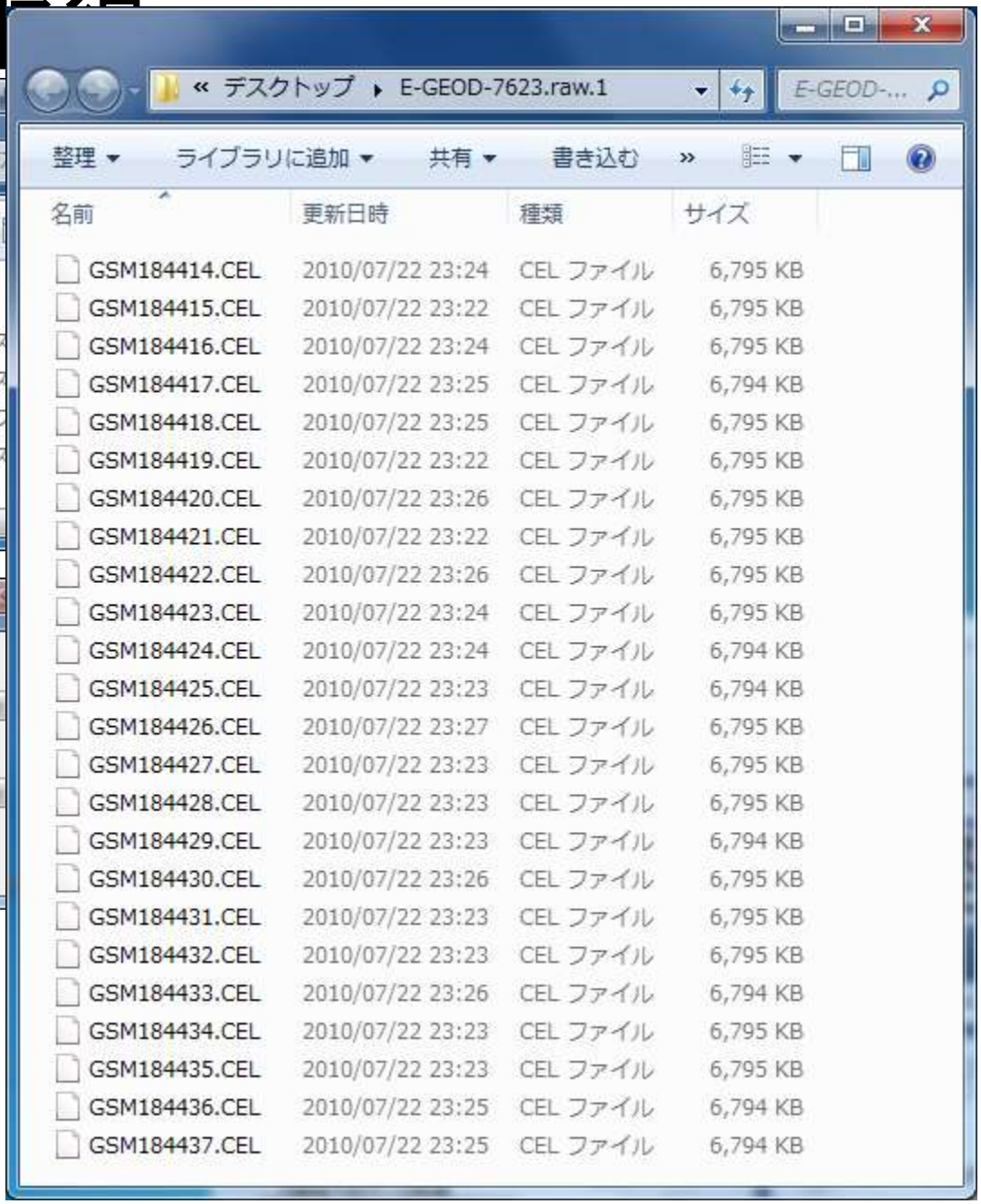


 参考 教科書p36- CELファイルのみからなるので便利

R経由で生データ取得







データ解析の全体像

ここまででプローブレベル数値データ（CELファイル）取得まで完了

	マイクロアレイ	RNA-seq
公共データ取得	GEO, ArrayExpress	GEO, ArrayExpress, NCBI SRA, EBI ENA, DDBJ SRA (DRA)
解析対象生物種	配列情報既知（アレイが提供されているもののみ）	モデル・非モデル問わず
生データ	プローブレベル数値データ	塩基配列（数億リード程度、数百塩基長）
		QC (Quality Control)：クオリティチェック、フィルタリング、トリミング
		アセンブリでトランスクリプトーム配列取得（マッピング時のリファレンスとしても利用）
		マッピング（bowtie2, TopHat2など）でSAM/BAMファイル取得
発現行列作成	前処理法(MAS5, RMAなど)適用後に遺伝子発現行列を得る	アノテーションファイルを利用してカウントデータ、配列長補正後のRPKM/FPKM、転写物レベルの発現情報など取得
発現変動遺伝子(DEG)同定	基本Rを利用(limma, SAM, Rank productsなど)	基本Rを利用(cuffdiff2, edgeR, DESeq2, TCCなど)
機能解析	GSEA, GSA, Cytoscapeなど	Rパッケージ SeqGSEAなどを利用。

データ解析の全体像

プローブレベル数値データ(CELファイル)を入力として、発現行列データを出力するのが①前処理(preprocessing)

	マイクロアレイ	RNA-seq
公共データ取得	GEO, ArrayExpress	GEO, ArrayExpress, NCBI SRA, EBI ENA, DDBJ SRA (DRA)
解析対象生物種	配列情報既知(アレイが提供されているもののみ)	モデル・非モデル問わず
生データ	プローブレベル数値データ	塩基配列(数億リード程度、数百塩基長)
	①	QC (Quality Control):クオリティチェック、フィルタリング、トリミング
		アセンブリでトランスクリプトーム配列取得(マッピング時のリファレンスとしても利用)
		マッピング(bowtie2, TopHat2など)でSAM/BAMファイル取得
発現行列作成	前処理法(MAS5, RMAなど)適用後に遺伝子発現行列を得る	アノテーションファイルを利用してカウントデータ、配列長補正後のRPKM/FPKM、転写物レベルの発現情報など取得
発現変動遺伝子(DEG)同定	基本Rを利用(limma, SAM, Rank productsなど)	基本Rを利用(cuffdiff2, edgeR, DESeq2, TCCなど)
機能解析	GSEA, GSA, Cytoscapeなど	Rパッケージ SeqGSEAなどを利用。

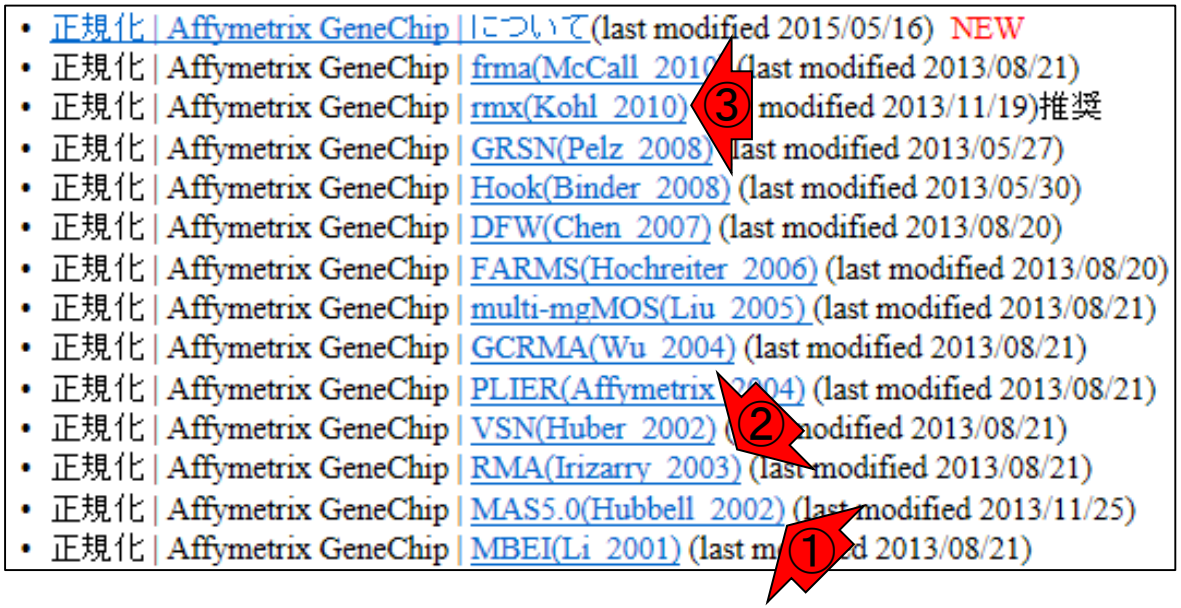
Contents

- トランスクリプトーム解析技術の原理や特徴
  - マイクロアレイとRNA-seq、遺伝子≠転写物
  - 様々な解析目的、トランスクリプトーム（転写物）配列取得
  - アノテーションファイルの読み込み（Rで転写物配列取得のイントロ）
  - Rで転写物配列取得（アノテーションファイルとゲノム情報ファイルから）
  - マイクロアレイの特徴
  - 発現データベース（DB）
- 発現DBからのプローブレベルデータ取得
  - Affymetrix GeneChip
  - R経由（教科書の § 2.2.1）
- Affymetrix GeneChipデータ前処理法を実行

①MAS5、②RMA、③RMXをやります

前処理法の違いを実感してみよう

- ①MAS5 (Hubbell et al., *Bioinformatics*, **18**: 1585–92, 2002)
  - 特徴：アレイごとに独立して前処理を実行（per-array basis）
  - 正規化：**グローバル正規化**
- ②RMA (Irizarry et al., *Biostatistics*, **4**: 249–64, 2003)
  - 特徴：読み込んだ複数サンプル（複数アレイ）の情報を用いて前処理を実行（multi-array basis）
  - 正規化：**quantile正規化**（プローブレベルデータに対して実行）
- ③RMX (Kohl et al., *BMC Bioinformatics*, **11**: 583, 2010)
  - 教科書中のRobLoxBioCと同じ方法



正規化 | Affymetrix GeneChip | RMA (Irizarry_2003)

Affymetrix chip (GeneChipTM)を用いて得られた*.CELファイルを元に、RMA(Irizarry et al., Biostatistics, 2003)アルゴリズムを用いてSummary scoreを算出。

「ファイル」-「ディレクトリの変

1. (CELファイルがあるディレ

```
out_f <- "hoge1.txt"

#必要なパッケージをロー
library(affy)

#データファイルの読み込
hoge <- ReadAffy()

#本番
eset <- rma(hoge)

#ファイルに保存
write.exprs(eset, fi
```

正規化 | Affymetrix GeneChip | MAS5.0 (Hubbell_2002)

Affymetrix chip (GeneChipTM)を用いて得られた*.CELファイルを元に、MAS5.0 (Hubbell et al., Bioinformatics, 2002)アルゴリズムを用いてSummary scoreを算出するやり方を示します。低発現領域で
のばらつきが大きいことが指摘
をすれば決して悪い方法ではな
アレイごとに独立して正規化を
の利点があります。

「ファイル」-「ディレクトリの変更

1. (CELファイルがあるディレク

```
out_f <- "hoge1.txt"

#必要なパッケージをロード
library(affy)

#データファイルの読み込み
hoge <- ReadAffy()

#本番
eset <- mas5(hoge)

#対数変換
summary(exprs(eset))
exprs(eset)[exprs(eset
summary(exprs(eset))
exprs(eset) <- log(exp

#ファイルに保存
write.exprs(eset, file
```

正規化 | Affymetrix GeneChip | rmx (Kohl_2010)

RobLoxBioCというRパッケージ中に実装されているrobust radius-minimax (rmx) estimatorというsummarization法です。論文中にMAS5の拡張版と書いてあるように、最後のステップのsummarization score計算時に(MAS5で採用されている)Tukey's biweightの代わりにrmx estimatorを利用しているところがポイントのようです。サンプルごとに独立して正規化を行っています。オリジナルはMASと同じくlog変換前のデータになっているので、robloxbioc関数をかけたあとに、自分で1以下の数値を1にした後にlog2変換したものを出力しています。

「ファイル」-「ディレクトリの変更」で適切なディレクトリに移動し以下をコピペ。

1. (CELファイルがあるディレクトリ上で)手元にあるCELファイルの読み込みから行う場合:

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

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

#データファイルの読み込み(*.CELファイル)
hoge <- ReadAffy()                            #*.CELファイルの読み込み

#本番
eset <- robloxbioc(hoge)                      #rmxを実行し、結果をesetに保存

#対数変換
summary(exprs(eset))                          #得られたesetの遺伝子発現行列のシグナル強度
exprs(eset)[exprs(eset) < 1] <- 1             #対数変換（log2）できるようにシグナル強度が
summary(exprs(eset))                          #上記処理後のシグナル強度分布を再び表示させ
exprs(eset) <- log(exprs(eset), 2)            #底を2として対数変換

#ファイルに保存
write.exprs(eset, file=out_f)                 #結果を指定したファイル名で保存
```

トップページへ

3つのコードの主な違いは、①パッケージ名部分と②前処理法の違いを表す関数名部分

ウェブページ中の例題は、①出力ファイル名が同じなので注意。②GSE7623_02samplesフォルダ中の2つの.CELファイルを入力として実行してみよう

正規化 | Affymetrix GeneChip | RMA (Irizarry_2003)

Affymetrix chip (GeneChipTM)を用いて得られた*.CELファイルを元に、RMA(Irizarr
2003)アルゴリズムを用いてSummary scoreを算出。
「ファイル」-「ディレクトリの変

1. (CELファイルがあるディレ

```
out_f <- "hoge1.txt"

#必要なパッケージをロー
library(affy)

#データファイルの読み込
hoge <- ReadAffy()

#本番
eset <- rma(hoge)

#ファイルに保存
write.exprs(eset, fi
```

正規化 | Affymetrix GeneChip | M

Affymetrix chip (GeneChipTM)を用いて得られた*.CELファイルを元に、MAS5.0 (Hubbell et al., nformatics, 2002)アルゴリズムを用いてSummary scoreを算出するやり方を示します。低発現領域で
らつきが大きいことが指摘
をすれば決して悪い方法ではな
アレイごとに独立して正規化を
の利点があります。
「ファイル」-「ディレクトリの変更

1. (CELファイルがあるディレク

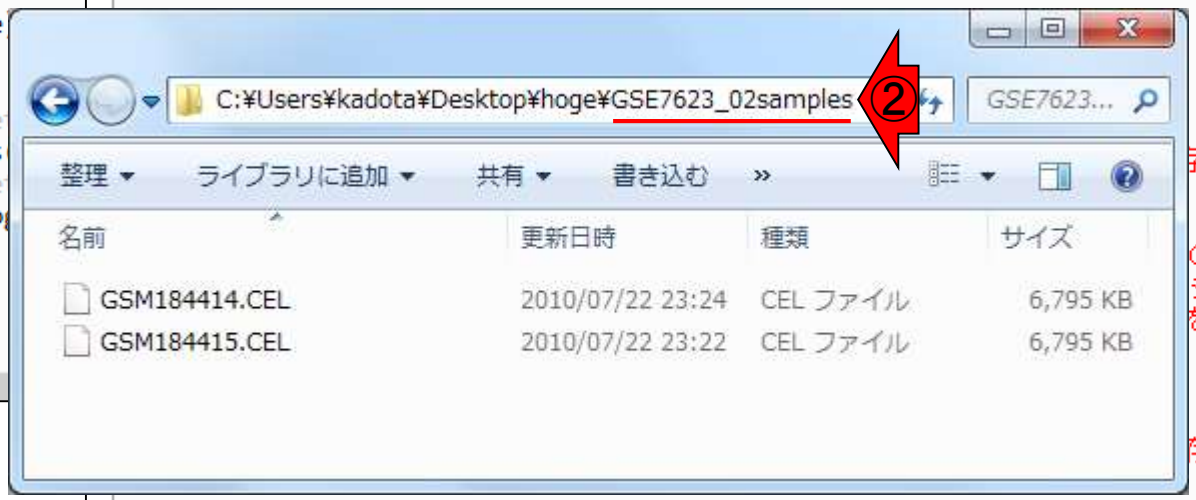
```
out_f <- "hoge1.txt"

#必要なパッケージをロード
library(affy)

#データファイルの読み込み
hoge <- ReadAffy()

#本番
eset <- mas5(hoge

#対数変換
summary(exprs(ese
exprs(eset)[exprs
summary(exprs(ese
exprs(eset) <- lo

#ファイルに保存
write.exprs(eset,
```

正規化 | Affymetrix GeneChip | rmx (Kohl_2010)

RobLoxBioCというRパッケージ中に実装されているrobust radius-minimax (rmx) estimatorというsummarization法です。論文中にMAS5の拡張版と書いてあるように、最後のステップのsummarization score計算時に(MAS5で採用されている)Tukey's biweightの代わりにrmx estimatorを利用しているところがポイントのようです。サンプルごとに独立して正規化を行っています。オリジナルはMASと同じくlog変換前のデータになっているので、robloxbioc関数をかけたあとに、自分で1以下の数値を1にした後にlog2変
したものを出力しています。
ファイル」-「ディレクトリの変更」で適切なディレクトリに移動し以下をコピペ。

1. (CELファイルがあるディレクトリ上で)手元にあるCELファイルの読み込みから行う場合:

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

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



トップページへ

私のやり方

メモ帳やワードパッドなどのテキストエディタを開いて、①出力ファイル名などを適宜変更した一連のコードをファイル(②rcode_preprocessing.txt)として保存しています。バイオインフォマティシャンの実験ノートに対応します

rcode_preprocessing.txt ②

```
###############
###  MAS5  ###
###############
out_f <- "data_mas.txt"  ①          #出力ファイル名を指定してout_fに格納
library(affy)                        #パッケージの読み込み
hoge <- ReadAffy()                   #*.CELファイルの読み込み
eset <- mas5(hoge)                   #MASを実行し、結果をesetに保存
summary(exprs(eset))                 #得られたesetの遺伝子発現行列のシグナル強度分布を表
exprs(eset)[exprs(eset) < 1] <- 1    #対数変換（log2）できるようにシグナル強度が1未満のも
summary(exprs(eset))                 #上記処理後のシグナル強度分布を再び表示させて確認
exprs(eset) <- log(exprs(eset), 2)   #底を2として対数変換
write.exprs(eset, file=out_f)        #結果を指定したファイル名で保存

###############
###  RMA  ###
###############
out_f <- "data_rma.txt"  ①           #出力ファイル名を指定してout_fに格納
library(affy)                        #パッケージの読み込み
hoge <- ReadAffy()                   #*.CELファイルの読み込み
eset <- rma(hoge)                    #RMAを実行し、結果をesetに保存
write.exprs(eset, file=out_f)        #結果を指定したファイル名で保存

############################
###  RMX (RobLoxBioC)  ###
############################
out_f <- "data_rob.txt"  ①           #出力ファイル名を指定してout_fに格納
library(RobLoxBioC)                  #パッケージの読み込み
hoge <- ReadAffy()                   #*.CELファイルの読み込み
eset <- robloxbioc(hoge)             #rmxを実行し、結果をesetに保存
summary(exprs(eset))                 #得られたesetの遺伝子発現行列のシグナル強度分布を表
exprs(eset)[exprs(eset) < 1] <- 1    #対数変換（log2）できるようにシグナル強度が1未満のも
summary(exprs(eset))                 #上記処理後のシグナル強度分布を再び表示させて確認
exprs(eset) <- log(exprs(eset), 2)   #底を2として対数変換
write.exprs(eset, file=out_f)        #結果を指定したファイル名で保存
```

私のやり方

①作業ディレクトリの変更と②.CELファイルが2つあることを確認し、左のコード全体をコピペで実行。挙動はR本体のバージョンによっても異なります。ここでは③R ver. 3.3.0の結果を示す。約8分

rcode_preprocessing.txt

```
###############
###  MAS5  ###
###############
out_f <- "data_mas.txt"                #出力ファイル名を
library(affy)                          #パッケージの
hoge <- ReadAffy()                     #*.CELファイル
eset <- mas5(hoge)                     #MASを実行し、結
summary(exprs(eset))                   #得られたesetの
exprs(eset)[exprs(eset) < 1] <- 1      #対数変換（log2）
summary(exprs(eset))                   #上記処理後のシク
exprs(eset) <- log(exprs(eset), 2)     #底を2として対数
write.exprs(eset, file=out_f)          #結果を指定した

###############
###  RMA  ###
###############
out_f <- "data_rma.txt"                #出力ファイル名を
library(affy)                          #パッケージの読み
hoge <- ReadAffy()                     #*.CELファイルの
eset <- rma(hoge)                      #RMAを実行し、結
write.exprs(eset, file=out_f)          #結果を指定した

############################
###  RMX (RobLoxBioC)  ###
############################
out_f <- "data_rob.txt"                #出力ファイル名
library(RobLoxBioC)                    #パッケージの
hoge <- ReadAffy()                     #*.CELファイル
eset <- robloxbioc(hoge)               #rmxを実行し、
summary(exprs(eset))                   #得られたeset
exprs(eset)[exprs(eset) < 1] <- 1      #対数変換（log2）
summary(exprs(eset))                   #上記処理後のシク
exprs(eset) <- log(exprs(eset), 2)     #底を2として対数
write.exprs(eset, file=out_f)          #結果を指定した
```

③

```
R Console

R version 3.3.0 (2016-05-03) -- "Supposedly Educati$
Copyright (C) 2016 The R Foundation for Statistical$
Platform: x86_64-w64-mingw32/x64 (64-bit)

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

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

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

> getwd()                                          ①
[1] "C:/Users/kadota/Desktop/hoge/GSE7623_02samples"
> list.files()                                     ②
[1] "GSM184414.CEL" "GSM184415.CEL"
>
```

途中経過(MAS5)

①rat2302cdf_2.18.0.zipというファイルを自動でダウンロードしている。②2.3MBだが意外と時間がかかる…

rcode_preprocessing.txt

```
###############
###  MAS5  ###
###############
out_f <- "data_mas.txt"                  #出力ファイル名を
library(affy)                            #パッケージの読み
hoge <- ReadAffy()                       #*.CELファイルの
eset <- mas5(hoge)                       #MASを実行し、結
summary(exprs(eset))                     #得られたesetの
exprs(eset)[exprs(eset) < 1] <- 1        #対数変換（log2）
summary(exprs(eset))                     #上記処理後のシ
exprs(eset) <- log(exprs(eset), 2)       #底を2として対数
write.exprs(eset, file=out_f)            #結果を指定した

###############
###  RMA  ###
###############
out_f <- "data_rma.txt"                  #出力ファイル名を
library(affy)                            #パッケージの読み
hoge <- ReadAffy()                       #*.CELファイルの
eset <- rma(hoge)                        #RMAを実行し、結
write.exprs(eset, file=out_f)            #結果を指定した

############################
###  RMX (RobLoxBioC)  ###
############################
out_f <- "data_rob.txt"                  #出力ファイル名を
library(RobLoxBioC)                      #パッケージの読み
hoge <- ReadAffy()                       #*.CELファイルの
eset <- robloxbioc(hoge)                 #rmxを実行し、結
summary(exprs(eset))                     #得られたesetの
exprs(eset)[exprs(eset) < 1] <- 1        #対数変換（log2）
summary(exprs(eset))                     #上記処理後のシ
exprs(eset) <- log(exprs(eset), 2)       #底を2として対数
write.exprs(eset, file=out_f)            #結果を指定した
```

R Console

```
> hoge <- ReadAffy()                     #*.CELファ$
> eset <- mas5(hoge)                     #MASを実行$
background correction: mas
PM/MM correction : mas
expression values: mas
background correcting...installing the source packa$

 URL 'https://bioconductor.org/packages/3.3/data/an$
Content type 'application/x-gzip' length 2372449 by$
downloaded 2.3 MB

* installing *source* package 'rat2302cdf' ...
** R
** data
** preparing package for lazy loading
 警告:  replacing previous import 'AnnotationDbi::t$
 警告:  replacing previous import 'AnnotationDbi::h$
** help
*** installing help indices
** building package indices
** testing if installed package can be loaded
*** arch - i386
 警告:  replacing previous import 'AnnotationDbi::h$
```

①

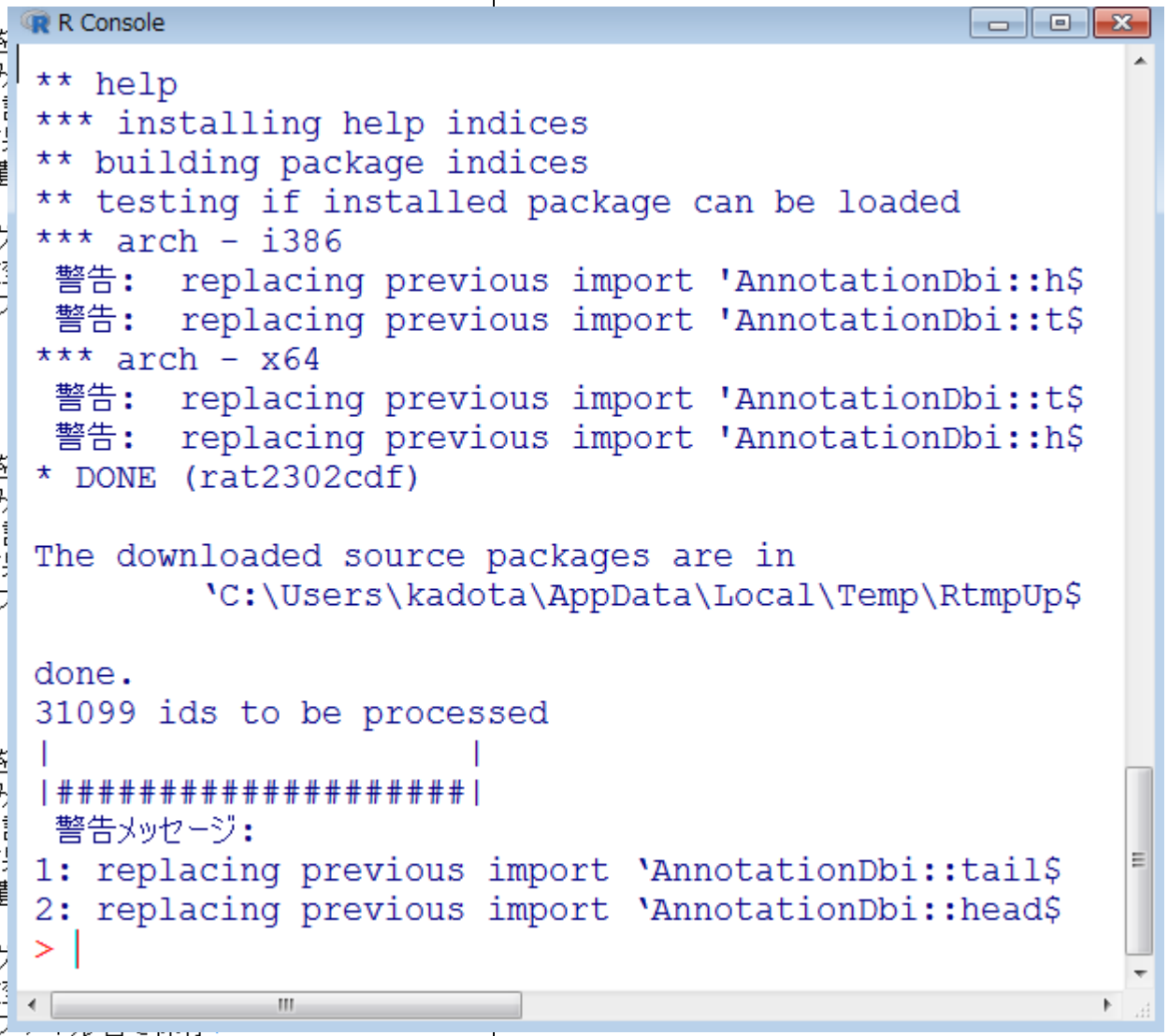
②

途中経過(MAS5)

赤枠のmas5関数実行後の状態。平成27年度の講義資料ではこのような警告メッセージは出ていなかった。Rのバージョンが異なると挙動も異なる

rcode_preprocessing.txt

```
###############
###  MAS5  ###
###############
out_f <- "data_mas.txt"                  #出力ファイル名を
library(affy)                            #パッケージの読み
hoge <- ReadAffy()                       #*.CELファイルの
eset <- mas5(hoge)                       #MASを実行し、結
summary(exprs(eset))                     #得られたesetの
exprs(eset)[exprs(eset) < 1] <- 1        #対数変換（log2）
summary(exprs(eset))                     #上記処理後のシ
exprs(eset) <- log(exprs(eset), 2)       #底を2として対数
write.exprs(eset, file=out_f)            #結果を指定した

###############
###  RMA  ###
###############
out_f <- "data_rma.txt"                  #出力ファイル名を
library(affy)                            #パッケージの読み
hoge <- ReadAffy()                       #*.CELファイルの
eset <- rma(hoge)                        #RMAを実行し、結
write.exprs(eset, file=out_f)            #結果を指定した

############################
###  RMX (RobLoxBioC)  ###
############################
out_f <- "data_rob.txt"                  #出力ファイル名を
library(RobLoxBioC)                      #パッケージの読み
hoge <- ReadAffy()                       #*.CELファイルの
eset <- robloxbioc(hoge)                 #rmxを実行し、結
summary(exprs(eset))                     #得られたesetの
exprs(eset)[exprs(eset) < 1] <- 1        #対数変換（log2）
summary(exprs(eset))                     #上記処理後のシ
exprs(eset) <- log(exprs(eset), 2)       #底を2として対数
write.exprs(eset, file=out_f)            #結果を指定した
```



途中経過(RMA)

①RMAは非常に早く終わります。それも流行った理由かも…。②RobLoxBioCのロード時に長いメッセージが延々と続きますが、特にエラーではなさそうなので、私は気にしていません

rcode_preprocessing.t

```
###############
###  MAS5  ###
###############
out_f <- "data_mas.txt"                    #出力ファイル名を
library(affy)                              #パッケージの読み
hoge <- ReadAffy()                         #*.CELファイルの
eset <- mas5(hoge)                         #MASを実行し、結
summary(exprs(eset))                       #得られたesetの
exprs(eset)[exprs(eset) < 1] <- 1          #対数変換（log2）
summary(exprs(eset))                       #上記処理後のシ
exprs(eset) <- log(exprs(eset), 2)         #底を2として対数
write.exprs(eset, file=out_f)              #結果を指定した

###############
###  RMA  ###
###############
out_f <- "data_rma.txt"                    #出力ファイル名を
library(affy)                              #パッケージの読み
hoge <- ReadAffy()                         #*.CELファイルの
eset <- rma(hoge)                          #RMAを実行し、結
write.exprs(eset, file=out_f)              #結果を指定した

############################
###  RMX (RobLoxBioC)  ###
############################
out_f <- "data_rob.txt"                    #出力ファイル名を
library(RobLoxBioC)                        #パッケージの読み
hoge <- ReadAffy()                         #*.CELファイルの
eset <- robloxbioc(hoge)                   #rmxを実行し、結
summary(exprs(eset))                       #得られたesetの
exprs(eset)[exprs(eset) < 1] <- 1          #対数変換（log2）
summary(exprs(eset))                       #上記処理後のシ
exprs(eset) <- log(exprs(eset), 2)         #底を2として対数
write.exprs(eset, file=out_f)              #結果を指定した
```

①

②

R Console

```
> ###############
> ###  RMA  ###
> ###############
> out_f <- "data_rma.txt"                  #出力ファ$
> library(affy)                            #パッケー$
> hoge <- ReadAffy()                       #*.CELファ$
> eset <- rma(hoge)                        #RMAを実行$
Background correcting
Normalizing
Calculating Expression
> write.exprs(eset, file=out_f)            #結果を指$
> ############################
> ###  RMX (RobLoxBioC)  ###
> ############################
> out_f <- "data_rob.txt"                  #出力ファ$
> library(RobLoxBioC)                      #パッケー$
 要求されたパッケージ RobLox ロード中です
 要求されたパッケージ distrMod をロード中です
 要求されたパッケージ distr をロード中です
 要求されたパッケージ startupmsg をロード中です
:startupmsg>  Utilities for Start-Up
:startupmsg>  Messages (version
:startupmsg>  0.9.3)
```

実行結果

うまく実行できれば、①list.files()で、
②3つの出力ファイルが見られるはず

rcode_preprocessing.txt

```
###############
###  MAS5  ###
###############
out_f <- "data_mas.txt"                #出力ファイル名を
library(affy)                          #パッケージの読み
hoge <- ReadAffy()                     #*.CELファイルの
eset <- mas5(hoge)                     #MASを実行し、結
summary(exprs(eset))                   #得られたesetの
exprs(eset)[exprs(eset) < 1] <- 1      #対数変換（log2）
summary(exprs(eset))                   #上記処理後のシ
exprs(eset) <- log(exprs(eset), 2)     #底を2として対数
write.exprs(eset, file=out_f)          #結果を指定した

###############
###  RMA  ###
###############
out_f <- "data_rma.txt"                #出力ファイル名を
library(affy)                          #パッケージの読み
hoge <- ReadAffy()                     #*.CELファイルの
eset <- rma(hoge)                      #RMAを実行し、結
write.exprs(eset, file=out_f)          #結果を指定した

############################
###  RMX (RobLoxBioC)  ###
############################
out_f <- "data_rob.txt"                #出力ファイル
library(RobLoxBioC)                    #パッケージの
hoge <- ReadAffy()                     #*.CELファイルの
eset <- robloxbioc(hoge)               #rmxを実行し、結
summary(exprs(eset))                   #得られたesetの
exprs(eset)[exprs(eset) < 1] <- 1      #対数変換（log2）
summary(exprs(eset))                   #上記処理後のシ
exprs(eset) <- log(exprs(eset), 2)     #底を2として対数
write.exprs(eset, file=out_f)          #結果を指定した
```

R Console

```
 Min.   :   32.04   Min.   :   32.08
 1st Qu.:   41.27   1st Qu.:   41.82
 Median :   60.17   Median :   63.05
 Mean   :  225.76   Mean   :  219.60
 3rd Qu.:  150.84   3rd Qu.:  150.68
 Max.   :12662.67   Max.   :13103.49
> exprs(eset)[exprs(eset) < 1] <- 1      #対数変換$
> summary(exprs(eset))                   #上記処理$
 GSM184414.CEL      GSM184415.CEL
 Min.   :   32.04   Min.   :   32.08
 1st Qu.:   41.27   1st Qu.:   41.82
 Median :   60.17   Median :   63.05
 Mean   :  225.76   Mean   :  219.60
 3rd Qu.:  150.84   3rd Qu.:  150.68
 Max.   :12662.67   Max.   :13103.49
> exprs(eset) <- log(exprs(eset), 2)     #底を2とし$
> write.exprs(eset, file=out_f)          #結果を指$
> list.files()
[1] "data_mas.txt"  "data_rma.txt"  "data_rob.txt"
[4] "GSM184414.CEL" "GSM184415.CEL"
> getwd()
[1] "C:/Users/kadota/Desktop/hoge/GSE7623_02samples"
>
```

①

②