Japanese Journal of Lactic Acid Bacteria

解　説

次世代シーケンサーデータの解析手法 第19回
R Markdown

牧野 磨音[1]、清水 謙多郎[1, 2, 3]、門田 幸二[1, 2, 3*]

[1] 東京大学 大学院 農学生命科学研究科
[2] 東京大学 大学院 情報学環・学際情報学府
[3] 東京大学 微生物科学イノベーション連携研究機構

マークダウン（Markdown）は、HTMLを手軽に生成するために開発された軽量マークアップ言語である。R Markdownは、Rのコマンドと Markdownを組み合わせたものであり、近年主流となっているRの実行手段である。多くのユーザにとっては見慣れないファイルの拡張子（.Rmd）かもしれないが、アグリバイオインフォマティクス教育研究プログラムの中でも利用されている。本稿ではまず、R Markdownの基本的な利用法として、チャンクの概念やHTMLの生成などを述べる。次に、前回作成したRスクリプトファイルの内容をR Markdown化し、相違点について述べる。最後に、個々のRコマンドや実行結果として得られるオブジェクトについて解説する。ウェブサイト（Rで）塩基配列解析のサブ（URL: http://www.iu.a.u-tokyo.ac.jp/kadota/r_seq2.html）中のウェブ資料（以下、W）を併用してほしい。

Key words：R Markdown, normalization, RNA-seq, RStudio

はじめに

今回も、前回に引き続き**管理者としてRStudioを起動し、デスクトップ上に作成したhogeフォルダを作業ディレクトリにした状態**からスタートする（W01）。R Markdownの使用感は、ウェブブラウザでのPython実行環境であるGoogle Colaboratory（以下、Colab）やJupyter Notebookとよく似ている。それゆえ、これらを知っている読者は、前回の内容よりも今回のR Markdownのほうがわかりやすいかもしれない。また、Colabなどを使ったことがない（が興味はある）読者も、R Markdownを学ぶことによって、利用上のハードルが低くなることが期待される。

マークダウンという言葉に馴染みがない読者は、HTMLの正式名称（HyperText Markup Language）に含まれるマークアップという言葉と対比させるとよいかもしれない。HTMLファイルをテキストエディタで眺めると、例えば一番大きな見出しは <h1></h1>、イタリックは のタグで囲まれていることがわかる。このようなタグを利用して見出しなどの全体的な構造を表現するのがマークアップである。

マークダウンは、マークアップの表記法を簡単にしたものであり、軽量マークアップ言語というカテゴリに属する。マークダウンの記法で書かれた文書は、HTML（やPowerPointやWord）に簡単に変換することができる。もちろんマークダウン特有の記法を覚える必要はあるものの、GitHubやQiitaなどのウェブサービスでも広く普及している。例えば我々が提供しているTCC-GUI[1)]やMBCdeg[2)]のGitHubサイトにアクセスすると、README.mdというマークダウンファイルの中身が表示される。

R Markdownを利用する一番のメリットは、Rコードだけでなくその実行結果も含めてHTMLで保存できる点にある。例えば、前回[3)]のRスクリプトファイル（JSLAB18.R）は、実行に必要な情報しか含んでいない。我々の経験上、このスクリプトを数年後に再度実行すると、何らかの不具合が発生することが予想される。しかし、HTMLで

*To whom correspondence should be addressed.
Phone ：+81-3-5841-8155
Fax ：+81-3-5841-1136
E-mail ：koji.kadota@gmail.com

保持しておくと、たとえ数年後に作成当時のコードがうまく動かなくとも、当時どのようなコマンドでどのような結果が得られていたかがわかる。

パッケージやソフトウェアの依存関係

RStudio では、様々なファイル形式を変換するためのソフトウェアである pandoc を内部的に用いて、R Markdown の HTML などへの変換を実現している。RStudio 上で pandoc を実際に利用しているのは、knitr [4] という R パッケージである。R Markdown 本体に相当するのが rmarkdown [5] という R パッケージであり、rmarkdown 自体は knitr の機能を内部的に用いている（つまり依存関係がある）。このような場合、rmarkdown のみをインストール対象として指定すれば、依存関係にあるパッケージ（この場合は knitr）も自動的にインストールされる（W02）。なお、**管理者として RStudio を起動**するよう強調していたのは、一般ユーザとしてパッケージのインストールを行った場合に、一部の PC 環境で不具合が生じるためである。

Pandoc 自体は独立したソフトウェアであるため、インストールも独立して行わねばならない。バージョンアップも独自に行われるため、本稿作成時のバージョン（ver. 2.18）と読者が実際に試すバージョンは異なりうる。そして依存される側（pandoc）は、依存する側（knitr）には配慮しきれないのが現実である。今回インストールした rmarkdown ver. 2.14 は、要件として R 本体が ver. 3.0 以上、pandoc が ver. 1.14 以上だと記されている（W03）。この要件自体は、本稿執筆時点の最新版（2022 年 5 月末時点で R ver. 4.2.0 および pandoc ver. 2.18）との比較から、全く厳しいものではないことがわかる。つまり、インストールで躓くことはほぼない。

R パッケージ同士の依存関係で実際に躓く例としては、R ver. 4.0.5 での TCC [6] が挙げられる。TCC は内部的に DESeq2 [7] を用いているが、DESeq2 がさらに内部的に用いている locfit の要件が R ver. 4.1.0 以上となっているためである。これは 2022 年初頭に発生していたエラーであり、バイオインフォマティクス関係の Q&A サイトである Biostar [8] でも 2022 年 4 月末にこのエラーに関するやりとりが行われている。前回 [3] の冒頭部分で「R ver. 4.1.0 以前のもので不具合が生じた場合は、最新版をインストールして再度試してほしい。」と書いたが、その理由は TCC のインストールエラーを想定したものである。

R Markdown の基本的な利用法（W04）

図 1 は、R Markdown ファイルの新規作成例である。RStudio 上では、「①File →②New File →③R Markdown」という手順で行う。④最初にこのドキュメントの Title（デ

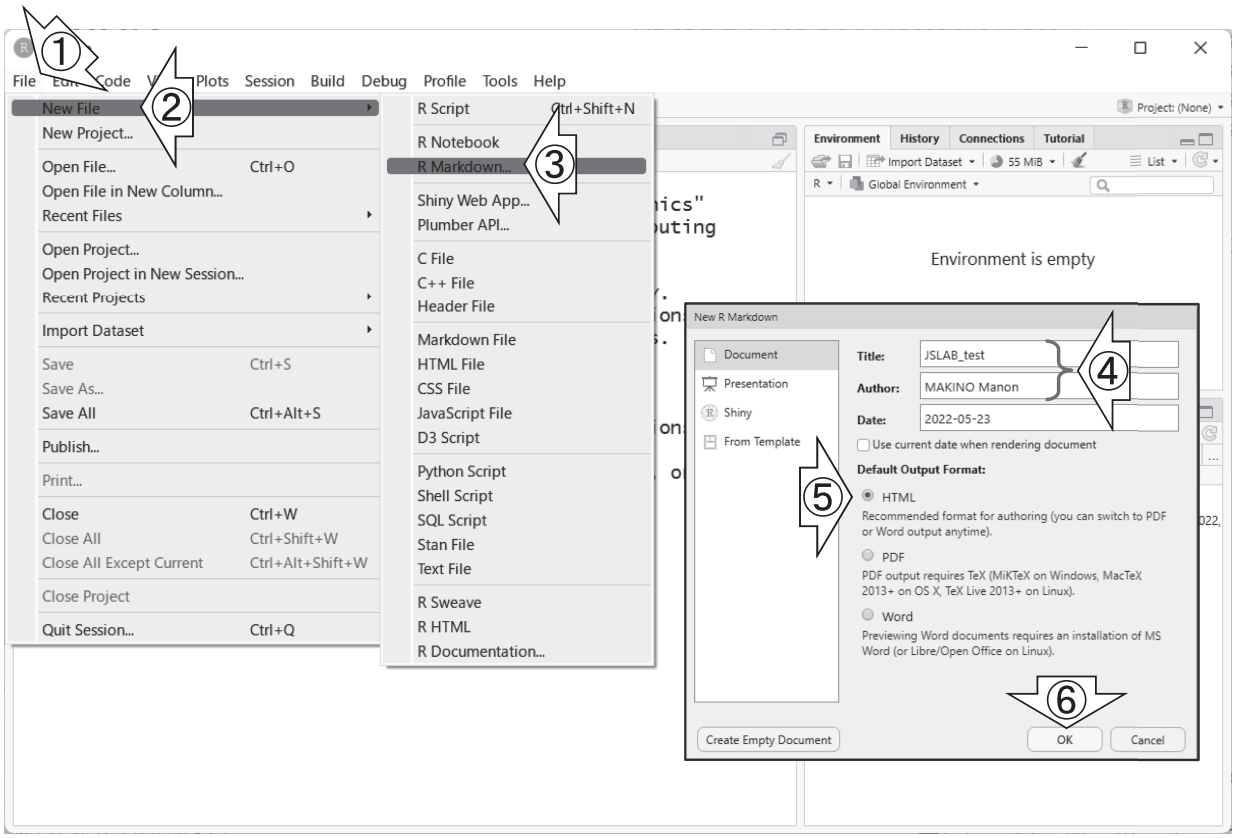


図 1. R Markdown の新規作成

フォルトは Untitled）と Author（デフォルトは空欄）を入力するように促される。R Markdown ファイルの変換は HTML と PDF と Word が選択可能であるが、基本的にデフォルトのままでよい（⑤HTML で⑥OK ボタンを押す）。

R Markdown は、R のコードと Markdown 形式の文書を組み合わせたものである。R コードは、チャンク（chunk）と呼ばれるブロック単位で実行される。例えば、第 18 回の**図 3** では、「複数のパッケージのインストールを行う 2 ～ 9 行目のコマンドを選択範囲として反転させたのち、Run ボタンを押して実行」した。マウス操作の場合は誤った選択範囲で実行してしまうことがあるが、これらの**コマンド群を最初から 1 つのブロックとして取り扱えるようにするのがチャンク**である。このやり方は Colab や Jupyter Notebook でも採用されている。

図 2 は、**図 1** で新規作成した R Markdown の 8 行目以降の記述内容を一旦削除したのち、8 行目にカーソルがある状態で①および②をクリックして新規 R チャンクを挿入し終えた状態である。③作成された新規 R チャンクは 8 ～ 10 行目の部分に相当し、8 行目が開始行、10 行目が終了行である。9 行目の部分が実際に実行させたいコードを書き込んでいく部分である。8 行目に {r} という記述があるが、これは 9 行目の記述内容が R コードだと認識させるためのおまじないのようなものだと解釈すればよい。④で Python が選択肢にあることからも予想できるように、RStudio は R だけでなく Python も実行することができる。これらを区別できるようにするため、チャンクの開始行（この場合は 8 行目）で明示的に宣言しているのだと解釈すればよい。

図 3 は、①**図 2** で作成した新規 R チャンク内に簡単な数値計算を行うコマンド（10 ～ 11 行目）を書き込んで実行（②の部分をクリック）し、③チャンク外に Markdown を 2 行分追加した R Markdown である。④test19.Rmd という名前で保存したのち、⑤Knit（ニットと読む）ボタンを押すと（これを**レンダリング**という）、⑥test19.html が生成されると同時に、その中身も表示される（中央下部の枠内）。③の 7 行目が⑦に対応することからもわかるように、**行頭の # は Markdown では見出し**を意味する。

HTML ファイルの表示結果には、著者や日付、R スクリプトとその実行結果が含まれる。③チャンク外に記載する Markdown として⑧のような**結果の考察なども書きこむことができるため、これで 1 つのレポートとして完結させることができる**。教員側（レポートを評価する側）も、R スクリプトファイルと実行結果を別々に提出されるよりハンドリングが楽である。これがレポート作成と絡めて R Markdown がよく紹介されるゆえんである。なお、第 18 回の R スクリプトファイルでは、# はコメントを意味していた。しかし R Markdown では、チャンク外の文書は基本的に全てコメントと同様の扱いになるため、

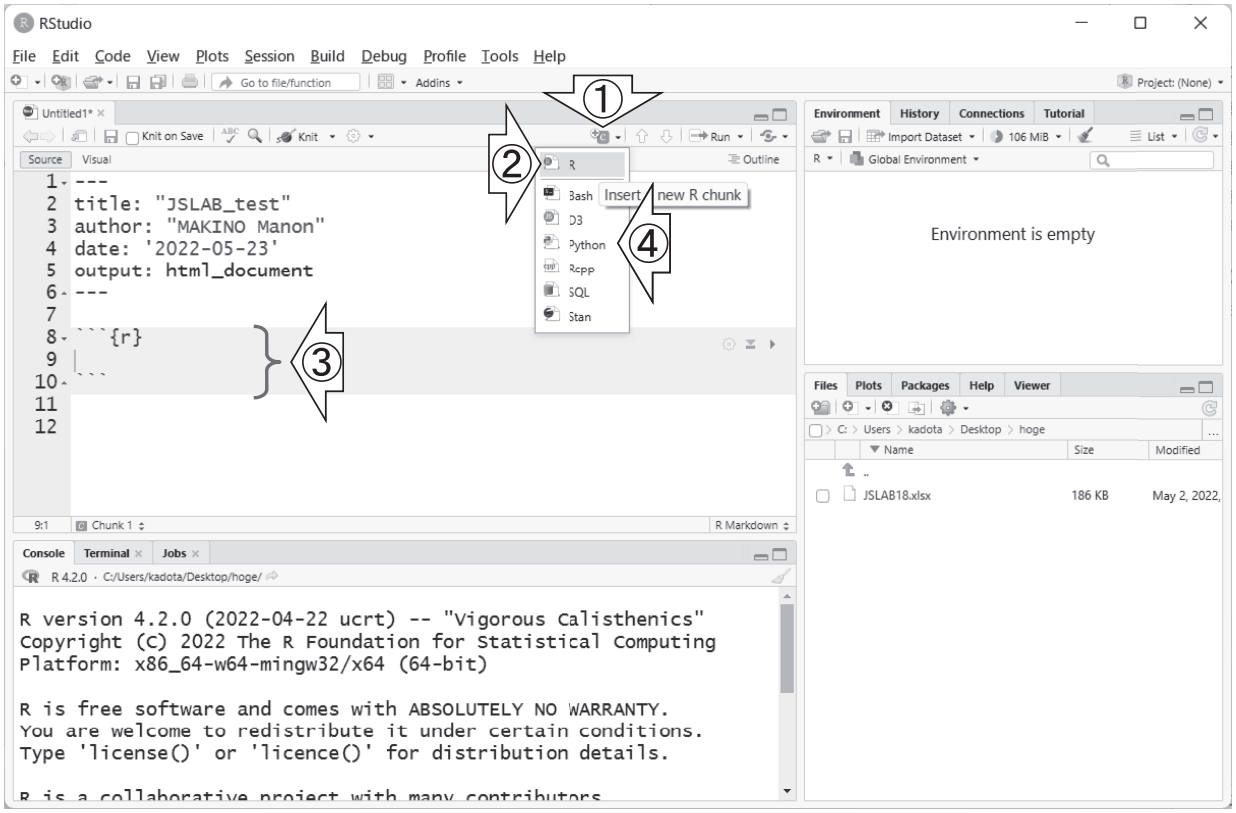


図 2. 新規 R チャンク挿入後の状態

Markdown 中の # の有無は、見出しかそうでないかの違いを表す。

JSLAB18.R の一部を Rmd 化（JSLAB18.Rmd の作成）

より実践的な R Markdown を作成すべく、ここでは第18回で実行した「パッケージのロード」と「サンプルのクラスタリング」に焦点を絞って解説する（W05）。まず前者については、R チャンク内にどのパッケージをロードするかが明記されている。また、パッケージロード時に表示されるメッセージが HTML レポートに含まれると、全体として冗長になってしまうという側面もある。R Markdown では、いくつかの R チャンクオプションが用意されている。デフォルトでは R チャンク内のコードと実行結果が全て HTML レポートに反映されるが、例えば include＝FALSE というオプションをつけることで、当該 R チャンク内のコードと実行結果を HTML レポートから除外することができる。

他の R チャンクオプションとしては、「実行結果は含めるがチャンク内のコードは含めない」場合は echo＝FALSE、「実行時にチャンク内を R コードとして実行しない（評価しない）」場合は eval＝FALSE とすることで、当該チャンク全体を制御することができる。さらに細かい設定として、R チャンク内の一部のみを制御することもできる。例えば当該チャンク内に 4 行分のコードが書かれていて、「実行結果は全て含めるが 1 行目と 4 行目のみコードは含めない」場合は、echo＝2:3 とすればよい。これは、コード部分の 1 行目が FALSE、2 ～ 3 行目が TRUE、そして 4 行目が FALSE という echo オプションの指定だと解釈してもよい。

第 18 回のスクリプトファイル（JSLAB18.R）では、「サンプルのクラスタリング」を 3 ステップ（データの読み込み、クラスタリング本番、作図）に分けている。Markdown の見出しの観点では、これらは最上位のレベル 1 より下のレベル 2 に相当する。HTML ファイルでは文字の大きさが 1 段階小さくなるのみであるが、見出しのレベル分けは目次（Table of Contents；TOC）の構成に直結するため、注意深く行ったほうがよい。

R Markdown では、Rmd ファイル上部の **YAML ヘッダ**と呼ばれる部分で「toc: TRUE」とすれば、目次を表示させることができる（W06）。デフォルトはウェブページの冒頭部分であるが、さらに「toc_float: TRUE」を追加することで、ページの左側にサイドメニューとして表示させることができる。**図 4** は、これまでの内容を反映した①JSLAB18.Rmd から②JSLAB18.html を生成した結果である。この Rmd ファイルの場合は、YAML ヘッダは 1 ～ 9 行目に相当する。③サイドメニューの目次は、**図 3** の 5 行目を**図 4** の 5 ～ 8 行目のようにすれば追加できる。R Markdown は、通常の Markdown に「YAML ヘッダと R

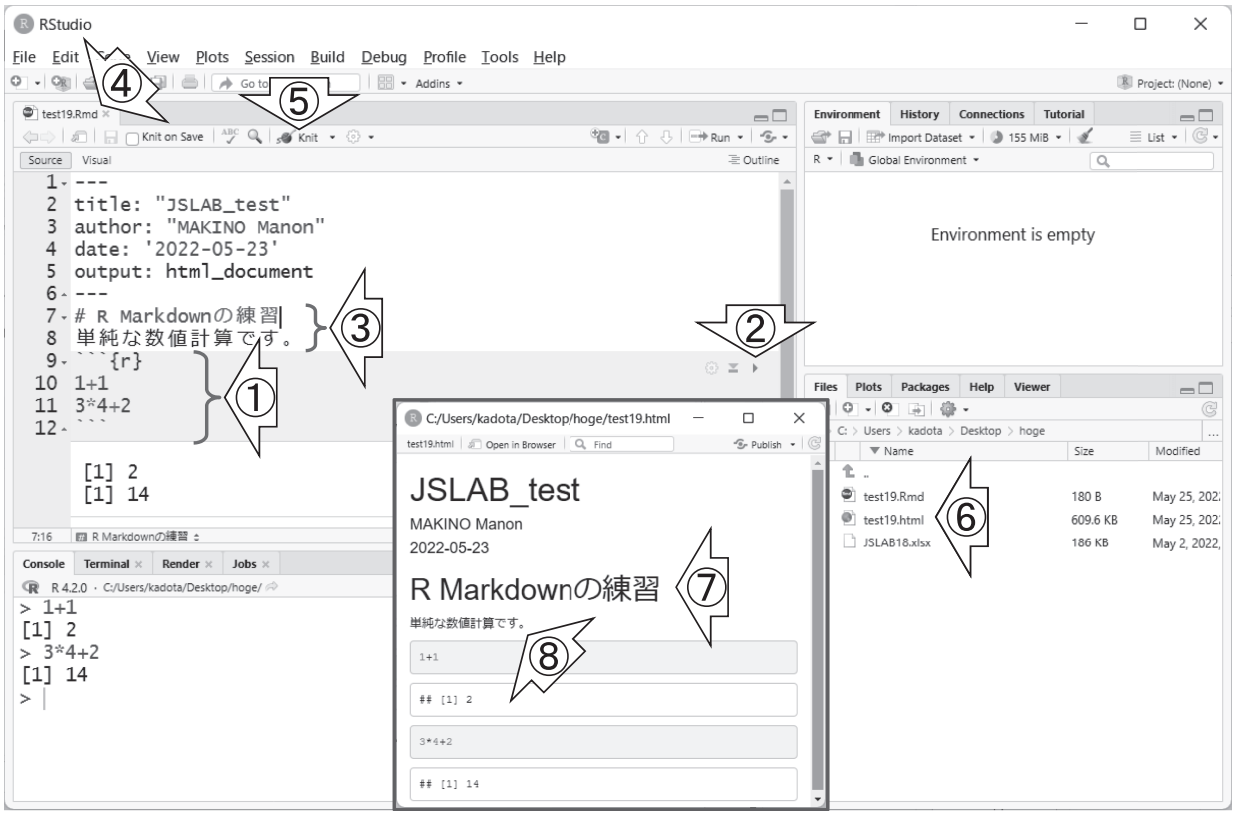


図 3. 単純な数値計算を含む R Markdown および HTML レポート生成例

チャンク」を組み込んだものという理解でもよい。

JSLAB18.Rの全部をRmd化（JSLAB19.Rmdの作成）

さらに実践的なR Markdownとして、JSLAB18.Rの内容を全てRmd化したものがJSLAB19.Rmdである。これは、前節で作成したJSLAB18.Rmdの内容に加えて、遺伝子クラスタリング用パッケージであるMBCluster.Seq[9]を発現変動遺伝子（DEG）検出に転用したMBCdeg[2]の実行まで含めたものである。JSLAB18.Rmdとの共通部分には、多少変更を加えている。例えば、パッケージのロード部分のRチャンクは、JSLAB19.Rmdでもinclude＝FALSEオプションをつけているのでHTMLレポートには反映されない。このため、JSLAB18.Rmdの10行目に存在していた「レベル1の見出し」は削除した（W07）。

JSLAB18.Rmdのサンプルのクラスタリング部分では、見出しのレベルの違いとHTMLレポート中の目次の見え方の違いがわかるように、レベル2の見出しをつけた3つのチャンクに分けて実行した（W08）。実際には、この程度の分量のものは1つのチャンクにまとめて実行する。このRチャンクでは、オプションとして「fig.width＝5, fig.height＝4」を指定している。この数値の単位はインチであり、クラスタリング結果の樹形図をplot関数で描画する際に利用している。

JSLAB19.Rmd中の29行目以降が、MBCdeg2（TCCパッケージ中の頑健なRNA-seqデータ正規化法DEGES[10]をMBCdegに組み込んだ方法）を実行する部分である。全部で約100行からなるため、MBCdeg2全体でレベル1の見出しとし、それをさらに計8つに細分化して説明すべくレベル2の見出しとした。こうすることで、Rエディタ上で任意の見出しやチャンクにジャンプすることができる（W09）。実用上も、特定の場所に任意のチャンクやMarkdownを挿入したりするため、何行目から何行目といった表現はせず、見出し名などで場所を特定するのが一般的である。

ファイルの読込とsubsetting（W10）

ここからは、MBCdeg2実行におけるレベル2の見出し（Rチャンク）単位で解説する。**図5**は、ファイルの読込とsubsetting（サブセットを得る作業のこと）を行うチャンクの実行結果である。①JSLAB19.Rmd中の、②のチャンクでは、JSLAB18.xlsx[11]を読み込んだ結果をdata_allというオブジェクトに格納している。読み込んだ直後の③data_allは、2,949遺伝子×9サンプルの数値行列である（第18回の図5）。但し、単純化のために「酸ストレス長期暴露群（pH4.5_24h）vs. 対照群（pH7_CCG）」の2群間比較を行いたいので、元の9列分のデータから、4～9列目のサブセッ

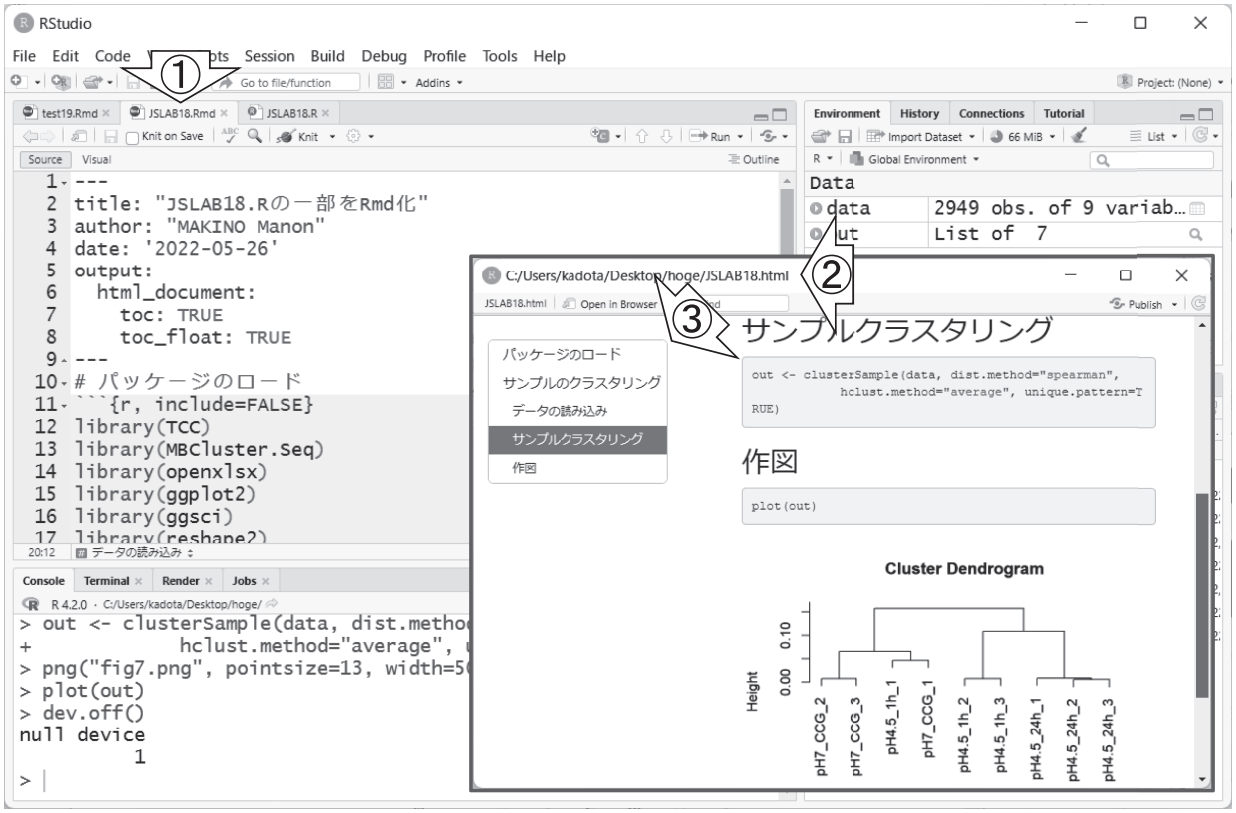


図4. JSLAB18.Rの一部をRmd化しHTMLを生成した結果

トを抽出した結果を data というオブジェクトに格納している。

この種の行列データの抽出・加工・操作を行うモダンな手段は、tidyverse と呼ばれる R パッケージ群の利用である。実際我々も、TCC のウェブアプリ版である TCC-GUI [1] に、(R Markdown はもちろんのこと) tidyverse を構成するパッケージ群 (dplyr, tidyr, and plotly) を利用した行列データの操作やインタラクティブな描画を実装している。しかし、本稿であまり沢山の事柄を盛り込みすぎても理解が追いつかないため、このチャンクではベーシックな subsetting を行っている。

34 行目では、2,949 行×9 列からなる data_all オブジェクトの中から、4 ～ 9 列目のみ抜き出した情報を data という名前で保存している。任意の列を抽出したい場合は、④コンマ (,) の右側に抽出したい列情報を指定する。同様に、任意の行を抽出したい場合は、④コンマの左側に抽出したい行情報を指定する。目的の 2 群のサブセットを抽出できているかどうかは、⑤environment タブ上で見えている⑥data オブジェクトを眺めることで確認できる。

群ラベル情報の作成と TCC 正規化 (W11)

群ラベルとは、2,949 遺伝子×6 サンプルの数値行列である data オブジェクト中のどのサンプルがどの群に属するかを示す情報のことである。バイオインフォマティクス**業界ではクラスラベルと呼ぶのが一般的**であるが、クラスという表現が難解だと感じる読者に配慮して、ここでは群ラベルとした。TCC 正規化とは、ここでは TCC パッケージが提供する頑健な正規化法である DEGES のことを指す。実用上は、「TCC パッケージ (ver. *X.Y.Z*) のデフォルトオプションを用いて実行した。」のように書けば十分であるため、本稿ではアルゴリズムの詳細は省く。

図 6 は、群ラベル情報の作成と TCC 正規化を行うチャンクの実行結果である。①群ラベル情報の作成は 39 行目に相当し、酸ストレス長期暴露群 (pH4.5_24h) に 1、対照群 (pH7_CCG) に 2 というラベルを割り当てている。このチャンクの目的は、MBCluster.Seq の入力として与えるための**サイズファクター**と呼ばれるデータ正規化に用いる情報の取得である。但し、TCC や edgeR [12] は**正規化係数**と呼ばれる値で取り扱うのが基本形である (43 行目の norm.factors)。それゆえ、得られた正規化係数をサンプルごとの総カウント数 (44 行目の colSums (data) に相当) に掛けて**有効ライブラリサイズ** (44 行目の ef.libsizes) とよばれるものをまず作成し (44 行目全体)、それを有効ライブラリサイズの平均値で割ることによってサイズファクター (45 行目の size.factors) 情報を得ている。

サイズファクターという概念は、edgeR と双璧をなすもう 1 つの有名な R パッケージである DESeq2 が使って

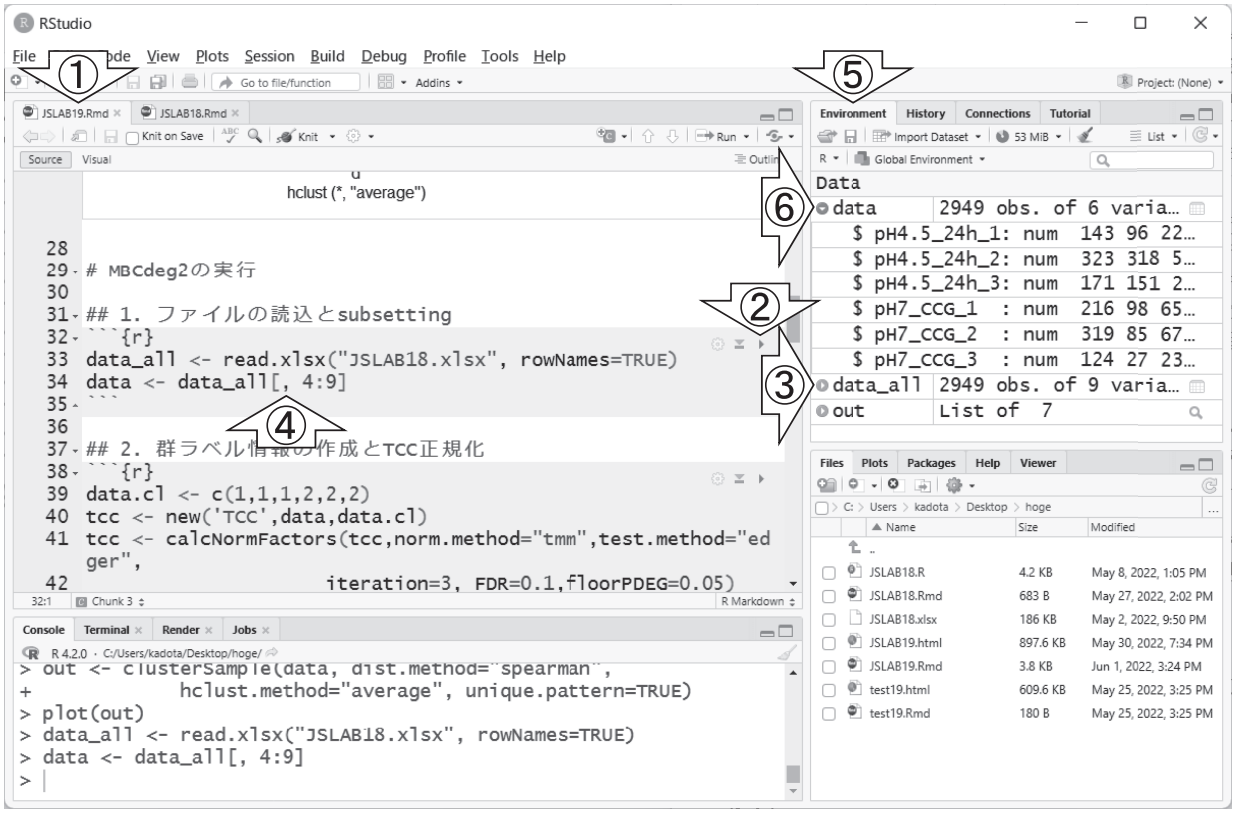


図 5. ファイルの読込と subsetting (チャンク 3) の実行結果

いるものであり、MBCluster.Seq もそれに準じている。**正規化係数とサイズファクターは、ともにデータ正規化時に利用される情報という意味で似た概念**である。しかし、②43 ～ 45 行目で全体的な関係性が明確に示されているように、**両者が別物**であるという点は正しく認識しておかねばならない。

MBCluster.Seq 実行の共通部分（W12）

ここからは、MBCluster.Seq パッケージ内での解析となる。ここでは、①RNASeq.Data という関数を用いて、②data オブジェクト、③群ラベル情報である data.cl オブジェクト、そして先ほど算出したサイズファクター情報である④size.factors オブジェクトの対数を入力として、⑤hoge というオブジェクトを作成している（**図 7**）。hoge の中身は、MBCluster.Seq 内でその後の解析に利用していくための特有の情報格納形式となっている。入力として与えた以上の情報にはなりようがないため、必要に迫られたときのみ内部構造を理解しにいくというスタンスでよいだろう。

MBCluster.Seq は、似た発現パターンを示す K 個のグループに遺伝子を分類する目的で利用するのが基本形である。内部的には K-means クラスタリングという乱数を発生させるアルゴリズムを採用しているため、試行ごとに結果が異なりうる。⑥ 50 行目の set.seed は同じ乱数を発生させたい場合に利用する関数である。括弧の中の数値は、発生させる乱数種の番号だという理解でよい。たまたま今年が 2022 年なので 2022 という数値にしているが、999 や 27 など任意の整数であればなんでもよい。

MBCluster.Seq 本番（W13）

図 8 は、K = 3 での MBCluster.Seq の実行結果である。① 57 ～ 58 行目で、K の値と「遺伝子ごとの各クラスタへの属しやすさを表す事後確率の数値行列」を格納するための出力ファイル名（MBCdeg_K3.xlsx）情報を与えている。59 ～ 62 行目が MBCluster.Seq 実行のメイン部分であり、cls というオブジェクトに実行結果を格納している。63 ～ 66 行目では、cls の中から事後確率情報部分のみを PP という名前で抽出し（63 行目）、その列名を変更して（64 行目）、その他の出力したい情報を合わせたものを（65 行目）、XLSX 形式ファイルに書き出している（66 行目）。

もう 1 つの主要な結果である「クラスタ中心の発現パターン（代表パターン）」情報は、67 行目の実行結果として②Console 画面上に表示される（但し②の画面上では隠れている）。ここでは③で当該部分のみ独立に示しているように、計 3 つの代表パターン（3 行 ×2 列の数値データ）情報が得られる。1 列目と 2 列目の数値の絶対値は同じであり、2 列目の列名（つまり 2）が「対照群（pH7_CCG）」

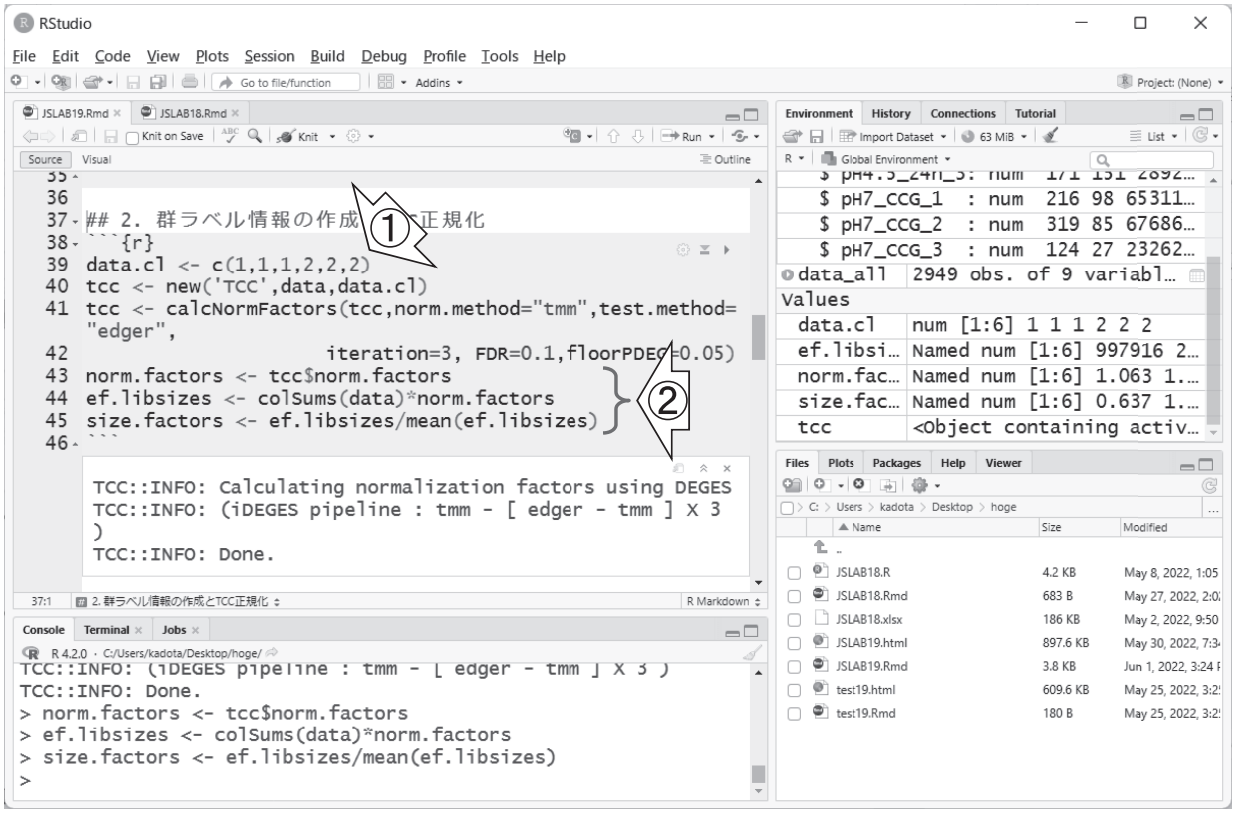


図 6. 群ラベル情報の作成と TCC 正規化（チャンク 4）の実行結果

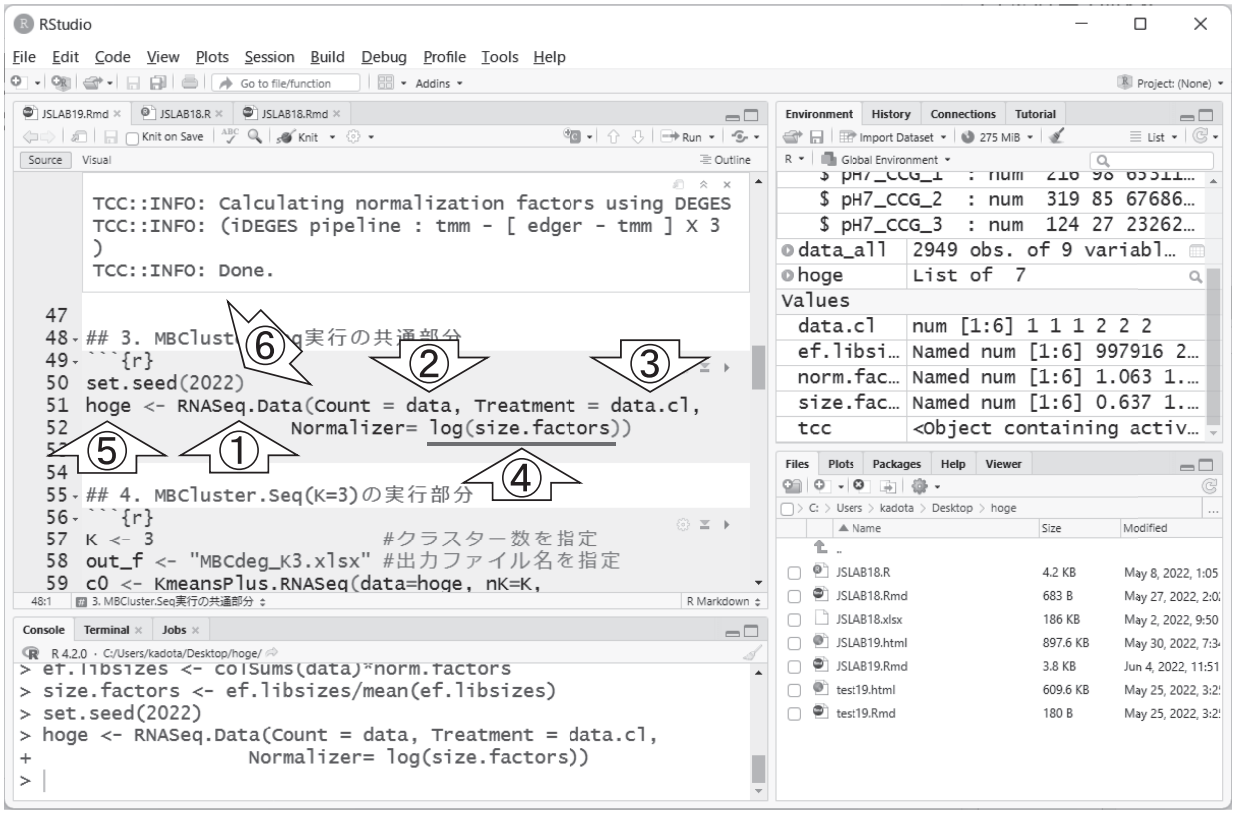


図 7. MBCluster.Seq 実行の共通部分（チャンク 5）の実行結果

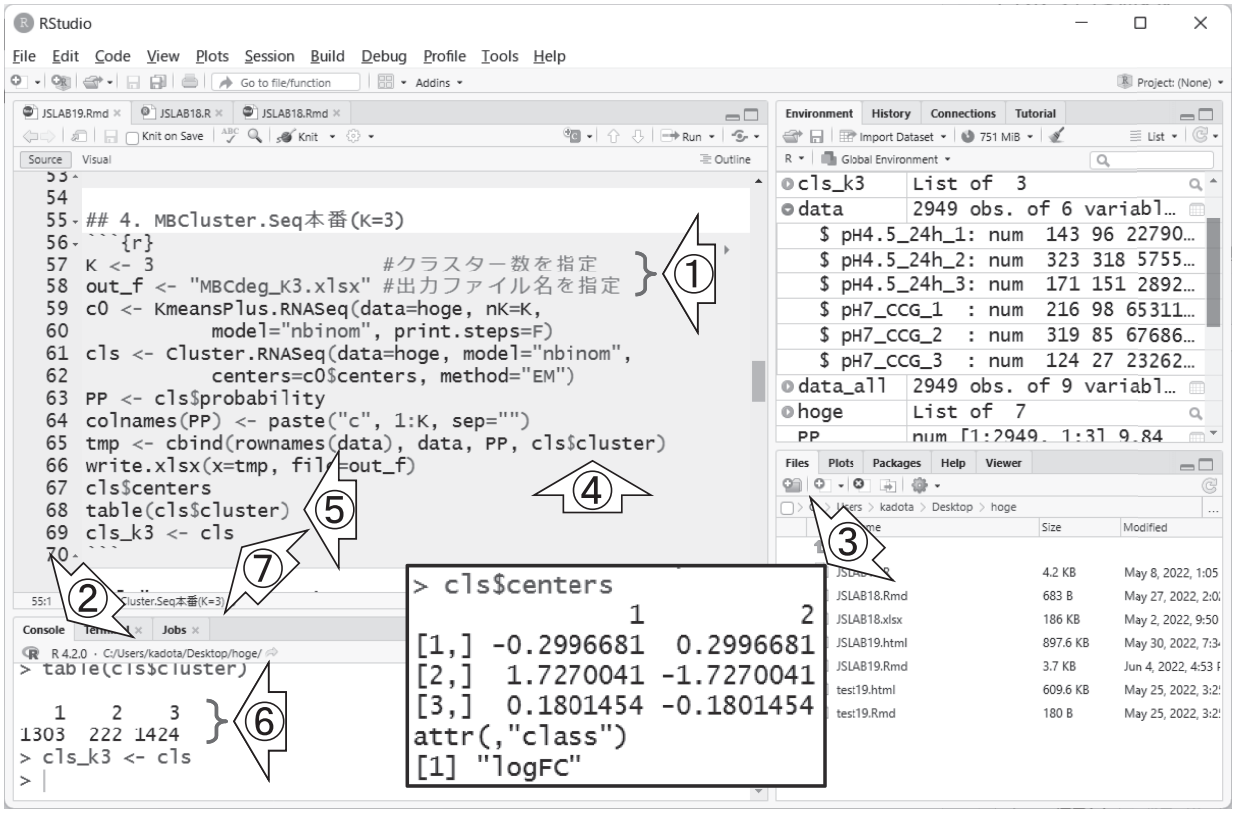


図 8. *K*=3 での MBCluster.Seq（チャンク 6）の実行結果

に対して割り当てたラベル情報である。行数は指定したクラスタ数Kに対応し、例えば1行目がクラスタ1（c1）の代表パターンである。

2列目を基準として考えると、1行目（c1）の代表パターンは、pH7_CCG群がpH4.5_24h群に比べて$e^{2\times(0.2996681)}$＝1.82091倍高発現だと解釈する。同様にして、c2はpH7_CCG群がpH4.5_24h群に比べて$e^{2\times(-1.7270041)}$＝0.03161865倍高発現（31.626910倍低発現）、そしてc3は$e^{2\times(-0.1801454)}$＝0.6974735倍高発現（1.433746倍低発現）だと解釈する。このような計算をする理由は、MBCluster.Seqが遺伝子ごとのカウント値をその平均値で割り、さらにその対数をとった状態で取り扱っているためである。

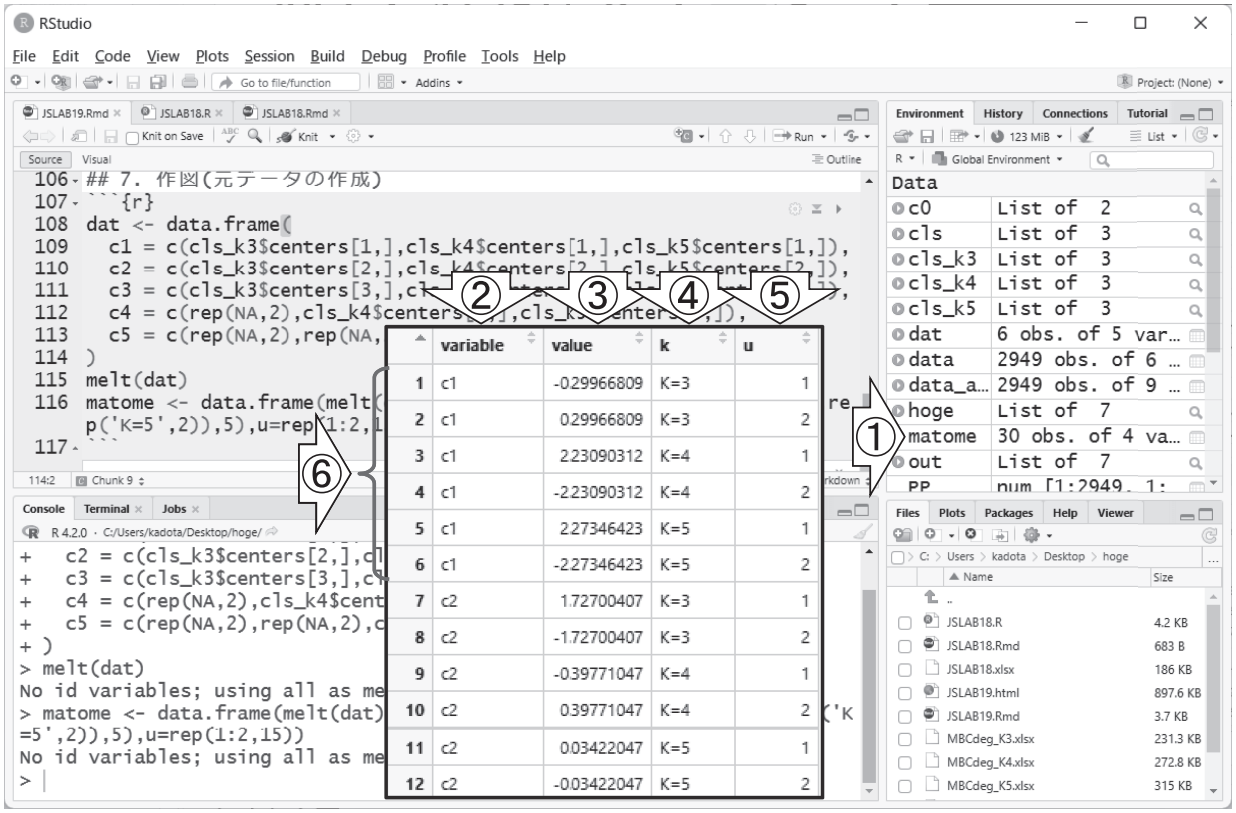
図8の③で見えている数値行列において、1列目と2列目の数値の絶対値が同じ理由は、平均値で割った値をそれぞれの群に割り振っているからである。それゆえ、例えばc1のクラスタ中心の群間の発現変動の度合いとして示した$e^{2\times(0.2996681)}$が難解だと感じた読者は、$e^{|-0.2996681|+|0.2996681|}$と読み替えてもよい。なお、これらの実際の発現パターンは、第18回の図8に示されているものと同じである。最も変動の少ないnon-DEGクラスタは、視覚的にもc3と判定できる。

MBCluster.Seqのような発現パターン分類の解析結果では、通常どのクラスタにいくつの遺伝子が割り振られたかが論じられる。この議論の基礎となる情報は、出力ファイル中の一番右側の列に示されている「どの遺伝子がどのクラスタに割り振られたかという数値ベクトル」であり、この元情報が**図8**の④で見えているcls$clusterというオブジェクトである。Rでは、このようなベクトルを入力として実行し、⑤ベクトル中の要素の種類ごとにその出現回数を返すtableという関数が用意されている。⑥で見えているものが実行結果であり、例えばc3に属する遺伝子数は1,424個である。

K＝4および5のMBCluster.Seqの実行結果についても、K＝3と同様の手順で実行することができる（W14）。重要な点は、**図8**の59～68行目に相当する部分が共通して利用されているということである。①K値と出力ファイル名、そして⑦主要な実行結果であるclsオブジェクトをそれぞれのK値由来であることがわかるように変更することで、コードの大部分を使いまわすことができる。もちろんこれらをより効率的なコードにすることは可能であるが、まずはこのような書き方からスタートしていくとよいかもしれない。

作図（元データの作成）（W15）

図9は、K＝3～5で得られたクラスタごとの代表パターン情報をまとめたオブジェクトmatomeを作成するチャンクの実行結果である。この①matomeオブジェクトは、30行×4列の行列データであり、ggplot2というパッケー



	variable	value	k	u
1	c1	-0.29966809	K=3	1
2	c1	0.29966809	K=3	2
3	c1	2.23090312	K=4	1
4	c1	-2.23090312	K=4	2
5	c1	2.27346423	K=5	1
6	c1	-2.27346423	K=5	2
7	c2	1.72700407	K=3	1
8	c2	-1.72700407	K=3	2
9	c2	-0.39771047	K=4	1
10	c2	0.39771047	K=4	2
11	c2	0.03422047	K=5	1
12	c2	-0.03422047	K=5	2

図9.「作図（元データの作成）」チャンクの実行結果

ジを用いて描画する際の入力として用いられる。中央下部の黒枠内が matome オブジェクトの最初の 12 行分に相当する情報である。② 1 列目（variable 列）はクラスタ番号情報であり、クラスタあたり 6 行分×計 5 クラスタ（K=5 が最大値なので c1, c2, …, c5）なので、30 行分の情報となる。③ 2 列目（value 列）は代表パターンの情報であるが、④ 3 列目（k 列）と⑤ 4 列目（u 列）の情報を合わせると理解しやすいであろう。

例えば、⑥ 1 ～ 6 行目は、④K=3 ～ 5 の MBCluster.Seq 実行結果の、②クラスタ 1（c1）の③代表パターン情報が格納されている。④からも想像できるが、1 ～ 2 行目が K=3 の結果、3 ～ 4 行目が K=4 の結果である。⑤は群ラベル情報であり（**図 6**）、1 が pH4.5_24h 群、2 が pH7_CCG 群である。重要な点は、フォーマットの詳細を理解するというより、「確かにこれだけの情報が含まれていれば、クラスタごと、K 値ごとの代表パターンを描画できるはず」と納得することであろう。

作図（本番）（W16）

図 10 は、クラスタごと、K 値ごとの代表パターンを描画するチャンクの実行結果である。121 行目で matome オブジェクトを入力として与え、ggplot 関数を実行しているのがわかる。①右下の黒枠が得られる図である。3 行×5 列からなり、②列がクラスタ番号、③行が K 値である。行と列を入れ替えたい（行列の転置をしたい）場合は、④「k ~ variable」を「variable ~ k」とすればよい。122 ～ 128 行目の記述の仕方が独特だと感じるかもしれないが、例えば 123 ～ 127 行目を削除した状態でも実行可能であるため、いろいろと試してみるとよいだろう。

おわりに

本稿では、R Markdown の基礎的な事柄からスタートし、前回の R スクリプトファイルの Rmd 化、そして前回より踏み込んだ形で具体的な R コードの解説を行った。マークダウン・チャンク・レンダリングなど聞きなれない用語に最初は戸惑うが、W04 以降を参考にしながら進めていけばすぐに慣れるであろう。

謝　辞

本内容の一部は、JSPS 科研費 21K12120 の助成を受けたものです。

利益相反（COI）

牧野磨音、清水謙多郎、門田幸二：本論文発表の内容に関連して開示すべき COI 状態はない。

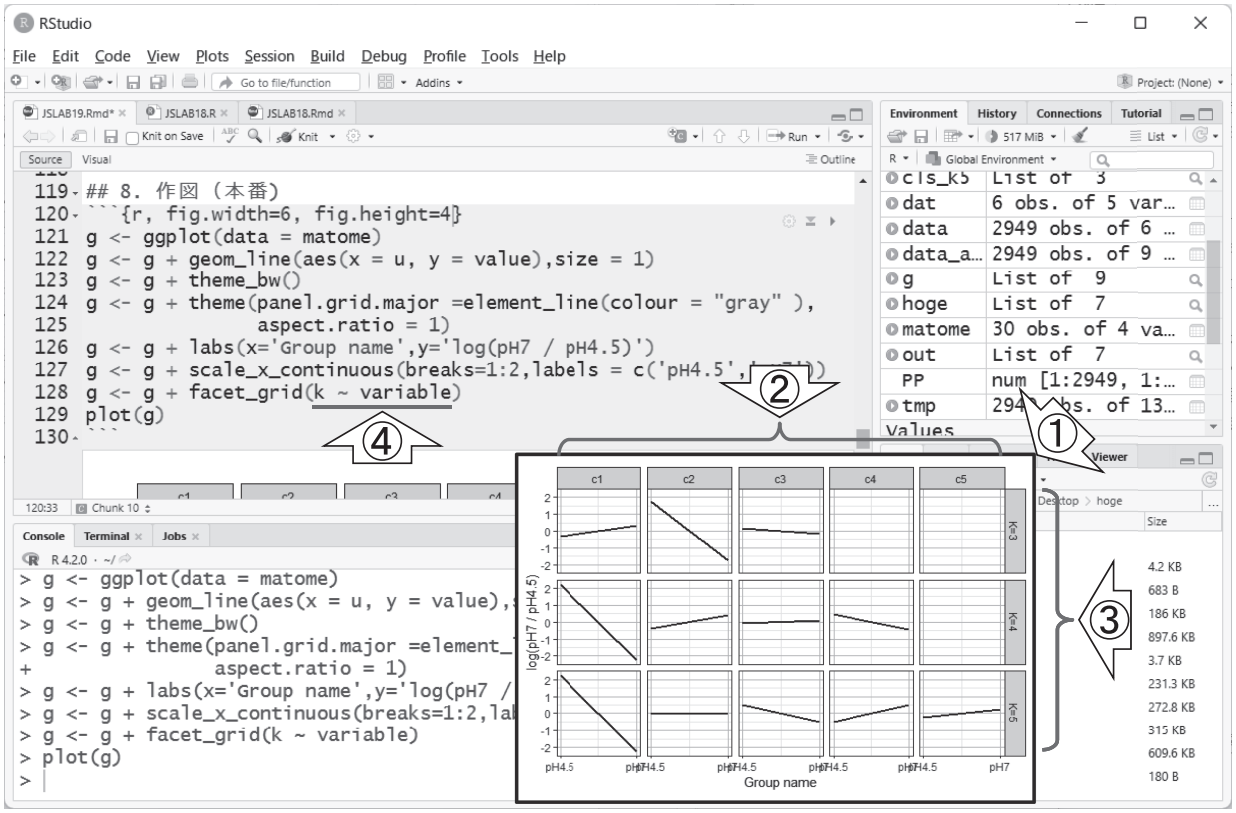


図 10.「作図（本番）」の実行結果

参 考 文 献

1) Su W, Sun J, Shimizu K, Kadota K (2019) TCC-GUI: a Shiny-based application for differential expression analysis of RNA-Seq count data. BMC Res Notes **12**: 133
2) Osabe T, Shimizu K, Kadota K (2021) Differential expression analysis using a model-based gene clustering algorithm for RNA-seq data. BMC Bioinformatics **22**: 511.
3) 牧野磨音，清水謙多郎，門田幸二（2022）次世代シーケンサーデータの解析手法：第 18 回遺伝子発現データのクラスタリング，日本乳酸菌学会誌 **33**: 87-94.
4) Xie Y. (2022) knitr: A General-Purpose Package for Dynamic Report Generation in R. R package version 1.39, https://yihui.org/knitr/.
5) Allaire J, Xie Y, McPherson J, Luraschi J, Ushey K, et al. (2022) rmarkdown: Dynamic Documents for R. R package version 2.14, https://github.com/rstudio/rmarkdown.
6) Sun J, Nishiyama T, Shimizu K, Kadota K (2013) TCC: an R package for comparing tag count data with robust normalization strategies. BMC Bioinformatics **14**: 219
7) Love MI, Huber W, Anders S (2014) Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biol **15**: 550.
8) Parnell LD, Lindenbaum P, Shameer K, Dall'Olio GM, Swan DC, et al. (2011) BioStar: an online question & answer resource for the bioinformatics community. PLoS Comput Biol. **7**: e1002216.
9) Si Y, Liu P, Li P, Brutnell TP (2014) Model-based clustering for RNA-seq data. Bioinformatics **30**: 197-205.
10) Kadota K, Nishiyama T, Shimizu K (2012) A normalization strategy for comparing tag count data. Algorithms Mol Biol **7**: 5.
11) Bang M, Yong CC, Ko HJ, Choi IG, Oh S (2018) Transcriptional Response and Enhanced Intestinal Adhesion Ability of Lactobacillus rhamnosus GG after Acid Stress. J Microbiol Biotechnol **28**: 1604-13.
12) Robinson MD, McCarthy DJ, Smyth GK (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics **26**: 139-40.

Methods for analyzing next-generation sequencing data 19. R Markdown.

Manon Makino[1], Kentaro Shimizu[1, 2, 3], Koji Kadota[1, 2, 3]

[1] Graduate School of Agricultural and Life Sciences, The University of Tokyo.
[2] Interfaculty Initiative in Information Studies, The University of Tokyo.
[3] Collaborative Research Institute for Innovative Microbiology, The University of Tokyo.

Abstract

R Markdown is a recent mainstream way of executing R, consisting of R commands and a unique notation called Markdown. Its file extension is Rmd, which may not be familiar with many researchers, but is also used in our graduate educational and research program. We first describe the basic usage of R Markdown, including the concept of chunks and HTML generation. We next convert the contents of the previously created R script file into R Markdown and discuss the differences. Finally, we describe individual R commands and objects resulting from their execution in detail. Supplementary materials are available online at https://www.iu.a.u-tokyo.ac.jp/kadota/r_seq2.html.