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5回 アセンブル、マッピング、そしてQC

孫 建強[1]、清水 謙多郎[1,2]、門田 幸二[2*]

東京大学大学院農学生命科学研究科
[1]応用生命工学専攻
[2]アグリバイオインフォマティクス教育研究ユニット

次世代シーケンサー（以下、NGS）データの解析は、大まかに①データ取得、②クオリティコントロール（以下、QC）、③アセンブルやマッピング、④数値解析の4つのステップに分けられる。連載第5回は、アセンブルやマッピングを紹介しつつ、QCの重要性に焦点を当てる。第4回でインストールしたQCプログラムFaQCs（ver. 1.34）実行、およびFastQC（ver. 0.11.3）でのアダプター / プライマー配列除去確認から始める。そして、アセンブルやマッピングの試行を通じて、QCで除去し切れていない、（本来トリムすべき）末端部分を発見した事例を紹介する。ウェブサイト（R で）塩基配列解析（URL: http://www.iu.a.u-tokyo.ac.jp/~kadota/r_seq.html）中に本連載をまとめた項目（URL: http://www.iu.a.u-tokyo.ac.jp/~kadota/r_seq.html#about_book_JSLAB）が存在する。ウェブ資料（以下、W）や関連ウェブサイトなどのリンク先を効率的に活用してほしい。

Key words：NGS, assembly, mapping, quality control

はじめに

連載第1回では「できるだけRで解説する」と宣言していたが、事実上撤回している。これは、2014年9月に2週間かけて行われたNGS速習コース講習会において、予想に反し多くの受講生がLinux環境構築を自力で行えた事実を目の当たりにしたのが大きい[1)]。Rについては、すでに豊富な情報を（R で）塩基配列解析や拙書[2)]などで提供している。このため、連載開始当初は夢物語だと思っていた「Linux環境でNGS解析を自在に行う」ための詳細かつ丁寧な自習用教材提供に第2回以降の内容を切り替えた。

第4回[3)]では、Bio-Linux[4)]にプレインストールされているプログラムの利用、および各種プログラムのインストール手順を解説した。これらの内容は、主に速習コース受講生の要望を反映させたものである。第4回ウェブ資料中の共有フォルダ設定については、2015年7-8月に開催されたNGSハンズオン講習会期間の前半は正常動作していたが、後半ごろから設定がリセットされるという不具合に遭遇した。このため、該当部分の記載内容を2015年8月に変更したので注意されたい。今後も不具合が生じれば、できる限り柔軟にウェブ資料やウェブサイト上で修正を行っていきたいと考えている。第5回も、多少のミスや勘違いを恐れずに最新プログラムをできるだけ紹介する。読者からも積極的にバグレポートやコメントをいただければ幸いである。

連載である以上、前回までの内容との整合性は重要である。しかし、第3回でダウンロードしたbzip2圧縮状態で計14GBにもなる*Lactobacillus casei* 12AのpairedendRNA-seqデータ（SRR616268；各134,755,996リード）ファイルをスタート地点とするのは、ダウンロードすらままならなかった一定数の読者にとっては理不尽であろう。また、ノートPCレベルの仮想環境では、この規模の全データを取り扱うのは困難である。それゆえ、第5回では100

*To whom correspondence should be addressed.
Phone ：+81-3-5841-2395
Fax ：+81-3-5841-1136
E-mail ：kadota@bi.a.u-tokyo.ac.jp

万リードからなるサブセットの gzip 圧縮 FASTQ ファイル（SRR616268sub_1.fastq.gz と SRR616268sub_2.fastq.gz）のみを出発点とする。また、クオリティチェック用プログラム FastQC（ver. 0.11.3； 第 4 回の W9）、およびアダプター除去兼クオリティフィルタリング用プログラム FaQCs（ver. 1.34； 第 4 回の W17）[5] をインストール済みという前提で話を進める。もちろんオリジナルの約 1.35 億リードからなるファイルを残すかどうかは自由である。

ゲノムアセンブリ周辺

一般に NGS データには、アダプターやプライマー配列など、解析サンプル由来以外の塩基配列が含まれている。アセンブルやマッピング結果に大きく影響するため、クオリティコントロール（QC）の一環としてのこれらの正確な除去は、NGS 解析における最も重要なステップの 1 つである。ゲノム用とトランスクリプトーム用、NGS 機器の種類や試薬などによっても QC 戦略は異なる。初期の戦略は、FastQC [6] 実行結果を眺めながら、QC 用の基本プログラム群から構成される FASTX-Toolkit [7] を用いてクオリティフィルタリングやトリミングを行い、その結果をまた FastQC を実行して眺めるという作業が行われていた。

ゲノムアセンブル時に重要となるのは、シーケンスエラーを含むリードの除去である。ショートリード時代によく行われたのは、*k*-mer 出現頻度に基づくフィルタリングである。おそらく Quake というエラー同定および補正プログラムの原著論文[8] が初出である。約 3GB からなるヒトゲノム配列決定時に、その 10 倍程度（約 30GB）読んでアセンブルされたのは有名な話である。生物種によってゲノムサイズは異なるため、任意の生物種のゲノムサイズを X とすることで、10X などと表現できる。これがいわゆるカバレッジ（coverage）と呼ばれるものである。読めるリード長が 100 塩基程度未満の頃の NGS データの場合は、ゲノムサイズの 100 倍程度（つまり 100X）読まないとアセンブルが困難であった[9]。ゲノムの場合は、どの領域でも概ね coverage が一定している。それゆえ、NGS リードの長さ *L* よりも短い、任意の長さ *k* の連続塩基（これがいわゆる *k*-mer と呼ばれるもの）で考えた場合、シーケンスエラーを含む *k*-mer は想定 coverage よりも非常に少ない出現回数となる。つまり、極端に低い出現頻度をもつ *k*-mer 由来リードを除くことで、シーケンスエラー由来リードのフィルタリングが達成されるのである。

乳酸菌を含むバクテリアのゲノムアセンブリは、第 3 世代 NGS 機器の代表格である PacBio RS II か Illumina MiSeq データの利用が主流になりつつあるようである。最近報告された約 2.3Mbp の *L. hokkaidonensis* LOOC260^{T} は、上記 2 種類の NGS データを組み合わせることで 1 本の環状染色体（と 2 本の環状プラスミド）を得ている[10]。PacBio データは、平均 4 kbp の長さからなる 163,376 リード（正確にはサブリード）を入力として HGAP [11] でアセンブルを行い、7 コンティグ（総塩基数 2,400,586 bp）を得ている。250 bp の paired-end MiSeq データは、2×2,971,310 リードを入力として Platanus [12] ver. 1.2（デフォルト設定）でアセンブルされている。MiSeq アセンブル結果によって得られた 53 コンティグ（総塩基数 2,359,642 bp；300 bp 未満の配列を除く）、および PacBio の結果を合わせることで完全なゲノム配列を得られたようである。この論文中でも行われているように、アセンブリ結果の評価は、得られたゲノム配列をリファレンス配列として用い、NGS リードのマッピング結果を眺めて検証するのが一般的である。Viewer は、第 4 回の最後にインストールした Integrative Genomics Viewer（IGV）[13] がよく利用される。

ウェブベースで手軽に利用できるバクテリア用の解析パイプラインも存在する。連載第 1 回でも触れた Galaxy ベースのものとしては、Orione [14] というウェブツールが提供されている。Orione の枠組みで、リードの QC、*de novo* assembly、CISA [15] による scaffolding やアセンブリ後の解析、Prokka [16] によるアノテーションまで一通りの解析が可能である。

FaQCs（ver. 1.34）による QC

乳酸菌 RNA-seq データ（SRR616268）の 100 万リードからなるサブセットの FastQC クオリティチェック結果を眺めると、用いた NGS 機器（Illumina HiSeq 2000）由来のアダプター（TruSeq Adapter）やプライマー（Illumina Single End PCR Primer 1）配列が含まれていることがわかる。これらは一般にリードの両端に存在し、クオリティスコアによるフィルタリングで自動的に除去されるわけではない（クオリティとは無関係）ため、専用のトリミングプログラムを適用しなければならない。これまでに多くのプログラムが開発されており、例えば Skewer [17] の Table 1 のように、原著論文の表などで他のプログラムとの比較がなされている場合が多い。最近開発されたものであれば、通常は paired-end データ、複数のアダプター配列の同時除去、圧縮ファイルへの対応などができている。これは単純に、査読者の立場になって考えた場合、世界の潮流に乗り遅れたプログラムの投稿論文は推薦しないからである。もちろん昔からあるプログラムであっても、定期的にバージョンアップされており、目的を達成できるものであれば基本的に何を使ってもいいだろう。

第 4 回でインストールした FaQCs [5] は、精度云々というよりは、インストールが比較的難しいプログラムの一例として取り上げたものである。しかし、最新の部類に入るだけのことはあり、実行時に -adapter オプションをつけるだけで、Illumina のアダプターやプライマーを自動的に除去してくれる（**図 1；[W1-1]**）。実際に除去できたかどうかは、FaQCs 実行後（トリム後）のファイル（QC.1.trimmed.fastq と QC.2.trimmed.fastq） を 入 力

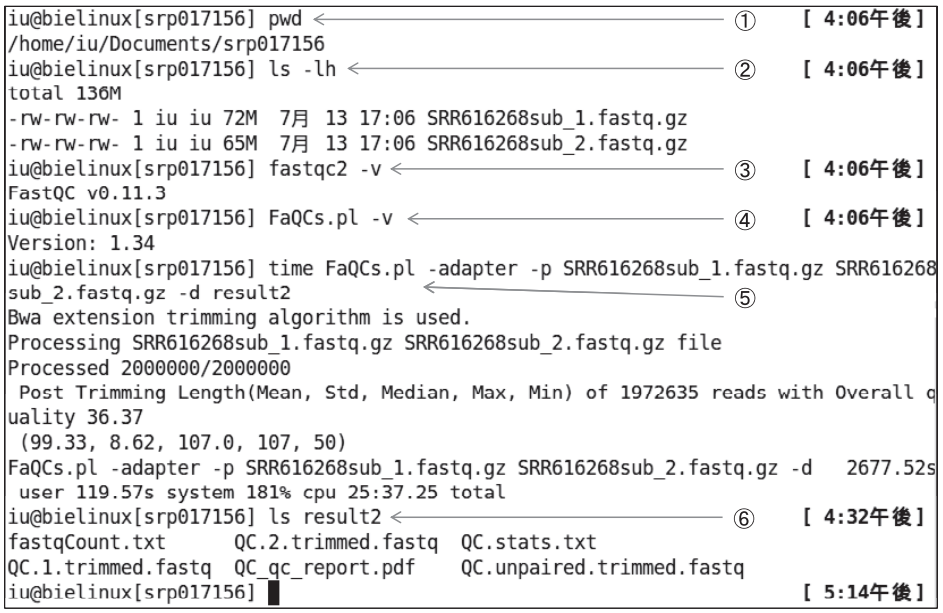

図 1．第 5 回の初期状態と FaQCs の実行。
① pwd でカレントディレクトリを表示。②「ls –lh」で作業ディレクトリ中のファイルを表示。③ fastqc2 コマンドのバージョン情報を表示。④ FaQCs.pl のバージョン情報を表示。⑤ FaQCs.pl を「-adapter」オプションをつけて実行。Paired-end の 2 つのファイルを同時に入力として与えている。result2 というディレクトリを作成してそこに出力ファイルを保存するようにしている。Paried-end の 100 万リードの実行に約 25 分かかっていることがわかる。⑥「ls result2」で result2 ディレクトリ中のファイルを表示。確かに *.trimmed.fastq ファイルが作成されていることがわかる。

として、FastQC によるクオリティチェックを行えばよい［W1-2］。著者らは、FastQC 実行結果ファイルの項目（Overrepresented sequences）を眺めて、トリム前に見えていた既知のアダプターやプライマー配列が、トリム後に正しく見えなくなっていることを確認して安心している［W1-3］。

このデータに関して結論からいえば、forward 側の 107 bp のリードファイル（SRR616268sub_1.fastq.gz → QC.1.trimmed.fastq）のうち、100-107 塩基付近に乳酸菌に由来しないものがトリムしきれずに多く残っている。これは、アセンブルやマッピングがうまくできない、という実害を被ることでわかる。計算時間がかかるため、できるだけ QC 段階で問題解決するという方針もあろう。しかし、やってみてはじめてわかることもある。以降の内容は、著者らが実際に行ったことを問題解決に至る思考回路とともに述べる。大まかに述べると、Rockhopper2 [18] によるトランスクリプトームアセンブリ、QuasR [19] による乳酸菌ゲノムへのマッピング、そして QC 再実行である。

トランスクリプトームアセンブリ

ゲノムのアセンブリは、断片化されたゲノム配列由来リードをつなぎ合わせて、元のゲノム配列を再構築する作業である。この再構築に相当する英語がアセンブリ（assembly）であり、再構築を行うプログラムをアセンブラ（assembler）という。デノボ（*de novo*）という言葉が同時に用いられることが多いが、これは「最初から」とか「一から」という意味である。このため、リードのみを入力として（つまり他の情報を一切利用せずに）アセンブルする際には、*de novo* assembly という表現がなされる。トランスクリプトームアセンブリとは、アセンブル対象がゲノムではなく解析サンプル中で発現している全転写物（トランスクリプトーム）の場合を指す。RNA-seq データのみを入力として一からアセンブルする場合は、*de novo* transcriptome assembly などと呼ばれる。

Multiple-k [20] や Trans-ABySS [21] などの初期のトランスクリプトーム用アセンブラは、ゲノム用を内部的に用いていた。詳細は省くが、上述の *k*-mer の *k* の値（正の整数）を大きくすればするほど、得られるコンティグは長くなり、高発現のものに偏る傾向にある [22]。*k* の値は、アセンブル時の「のりしろ」に相当するものである。パリンドロームを避けるべく、通常は奇数が採用される [23]。*k* の値を小さくするほど、低発現転写物を拾いあげることが原理的には可能であるが、得られるコンティグは短くなり（断片化）、似た配列からなるコンティグが多く得られる傾向（重複）にある。このためこれらのプログラムは、複数の *k* の値を用いて独立にゲノム用アセンブラを適用し、できるだけ多くの転写物配列をコンティグとして得ることに主眼を置いていた。それゆえ、コンティグ集合からいかに重複をとり除くかが課題であった。

おそらく現在もっとも頻用されているトランスクリプトーム用アセンブラは、Trinity [24] である。Trinity は、トランスクリプトーム専用としてデザインされた最初のプ

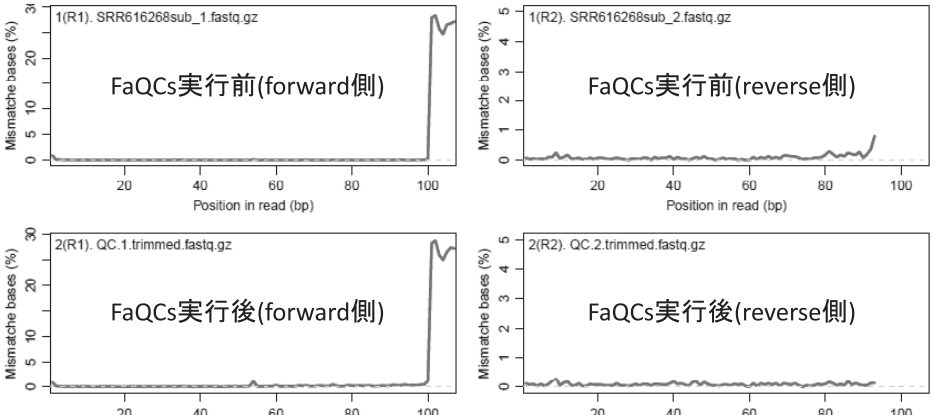

図 2. QuasR を用いたマッピング結果レポートの一部。
マッピングは、FaQCs 実行前後の 2 サンプルに対して行った。FaQCs 実行の有無に関係なく、forward 側リードの 100–107 bp 付近のミスマッチ率が極端に高いことがわかる。

ログラムであること、*k*=25 という単一の *k*-mer のみで幅広い範囲の発現レベルからなる転写物配列の再構築に成功したという原著論文の内容だったこと、インストールが簡単であったことなどが主な理由であろう。しかし、単一より複数の *k*-mer を用いたほうが、少なくとも原理的には、幅広い発現レベルからなるトランスクリプトームをより広範囲に捕捉できる。また、アセンブリの評価基準は、精度以外に使用メモリや計算時間も一定のウェイトを占める。Trinity 以外にも、新規プログラムは継続的に提案されている。例えば、複数 *k*-mer 戦略（multiple-*k* strategy）を採用したプログラムとして、比較的最近提案された Bridger [25] なども試してみるといいかもしれない。

一般に、Trinity を含む *de novo* transcriptome assembler 出力結果をそのまま利用することはない。得られたコンティグ間の塩基配列の類似度を調べると、非常に似た配列のものが一定の割合で含まれる。似たもの同士はなるべく 1 つの配列にまとめたいので、クラスタリングなどが行われる。つまり、アセンブリ後（post-assembly）に行う重要なステップは、重複の除去（redundancy check）である。このステップでは、CD-HIT [26] がよく用いられる。最近では、coding potential や機能ドメイン予測などを行ってより確からしいものを推測する IFRAT [27] なども提案されている。

この分野における *de novo* assembly の事実上の対比語は、reference-based assembly である。ゲノム配列などリファレンスとして利用可能な配列がある場合は、無理に *de novo* assembly をやる必要はない。基本戦略は、リファレンス配列への RNA-seq リードのマッピングである。「マップされた領域 = RNA が転写された領域」となるので、マップされたリードの和集合領域が転写領域である。過去に転写が報告されていない領域で、その領域の coverage が非常に高い場合には確度の高い新発見であろう。逆に、coverage が低い領域は、偶然マップされたリードからなる偽陽性の領域かもしれない。これらの結果は、マッピング時に用いるパラメータ、exon-intron 構造をもつ高等生物の場合はジャンクションリードのマッピング精度などによっても変わりうる。reference-based assembly 中の assembly は、実質的には 1 つの遺伝子領域から複数の転写物（transcripts または splice variants）が生成されうる高等生物のデータで、shared exon 上にマップされたリードの分配や転写物配列の再構築の意味で使われているのであろう。Cufflinks [28] という非常に有名なプログラムは、このカテゴリに属する。高精度なゲノム配列がリファレンスとしてあれば、我々が知識として持っている exon-intron 境界の GT-AG 則との一致度、paired-end の場合はマップされたリードペア間の距離や向きなどの情報を利用することができる。

バクテリア専用のアセンブラも少数ながら存在する。reference-based assembler として Rockhopper [29] や TruHmm[30] が、そして *de novo* assembler の Rockhopper2 [18] が挙げられる。次節では、Rockhopper2 のインストールから、乳酸菌 paired-end RNA-seq データ（SRR616268）のアセンブリまでの一通りの手順を紹介する。

De novo transcriptome assembly（Rockhopper2 ver. 2.0.3）

Rockhopper2 は、Java というプログラミング言語で記述されている。そのため、Rockhopper2 の prerequisite（前もって必要な事柄）は、推奨バージョン以上の Java がインストールされているかどうかの確認である。Rockhopper2 のダウンロードページ上部では、System Requirements（Java ver. 1.6、2GB 以上のメモリ；2015 年 9 月 3 日現在）として記載されている［W2-1］。Bio-Linux 8 には、連載第 4 回の FastQC（ver. 0.11.3）インストール時に Java ver. 1.7.0_55 が入っていることを確認済みで

ある。著者らのPC環境で改めて確認すると、ver. 1.7.0_79となっていた［W2-2］。この違いは、おそらく本連載用以外にもBio-Linux 8を利用していることに起因する。つまり、何らかの拍子にアップデートされたと考えるのが自然である。いずれにせよ推奨のver. 1.6以上であることに変わりはないので、この程度の細かな違いは気にも留めない。

ダウンロード後に得られる実行ファイルはRockhopper.jarである。Javaの場合は、jarという拡張子のついたファイルが得られ、これが実行ファイルになる。つまり、基本的にJavaファイルのダウンロード完了がインストール完了を意味する［W2-3］。これはWindows版（Rockhopper.exe）やMacintosh版（Rockhopper.dmg）についても同じである。Bio-Linux 8では、GUI版とコマンドライン版の両方が利用可能であり、基本的に指示された通りのコマンドを打てばよい［W2-4］。バックグラウンドジョブ（nohupと&の付加）やプロセス管理（psとkill）は、特に遺伝研スパコンなどの大型計算機にセキュアシェル（secure shell; SSH）経由でログインして解析する際に利用すると思われる。このため、GUI版の起動説明（java -Xmx1200m -jar Rockhopper.jar）と絡めて、これらの基本的な利用法を示した［W3］。

コマンドライン版の実行コマンド（java -Xmx1200m -cp Rockhopper.jar Rockhopper）も、GUI版と似ている［W4-1］。「-Xmx1200m」は、最大メモリを1200MB分確保するという意味である。「-cp」は、クラスパス（classpath）を意味し、「-classpath」と書いてもよい。これは、「パスを通す」ことと本質的に同じ概念である。しかしながら、第4回（W9-5；W15-5；W18-3）で示したような「sudo ln -s /home/iu/Downloads/Rockhopper.jar /usr/local/bin」を実行してもRockhopper.jarのタブ補完がうまくいくようになるだけである。この作業では、コマンドライン版をうまく実行できない。RockhopperのEXAMPLE EXECUTIONは「java Rockhopper <options> …」となっているが、「java Rockhopper」でエラーが出ないようにするには、クラスパスを正しい手順で設定する必要がある［W4-4］。Java特有の概念であること、Rockhopper中で説明されている2つのコマンド（java -Xmx1200m -cp Rockhopper.jar Rockhopperと java Rockhopper <options> …）間に乖離があることから難解な印象を与えるが、クラスパスの設定自体は簡単である。著者らの環境では、「/home/iu/Downloads/Rockhopper.jar」がRockhopper.jarの絶対パスである。この場合は、「export CLASSPATH=/home/iu/Downloads/Rockhopper.jar」と設定することで、EXAMPLE EXECUTIONで示されている「java Rockhopper <options> …」がどのディレクトリ上からも利用可能となる。

FaQCs実行後のpaired-end FASTQファイル（QC.1.trimmed.fastqとQC.2.trimmed.fastq）があるディレクトリ［W1-1］に移動して、*de novo* assemblyを行う。Paired-endの場合は、2つのファイルを%で連結して［W4-2］、「java Rockhopper QC.1.trimmed.fastq%QC.2.trimmed.fastq」のようにすればよい［W5-1］。著者らは、メモリ不足に起因するエラーに遭遇したため、「-Xmx2000m」オプションを追加して最大メモリを2GBに引き上げることで、プログラムを最後まで実行させることができた［W5-2］。但し、ターミナルの出力画面（Total number of assembled transcripts: 0）でも示されているように、アセンブルされた転写物は1つもなかったことがわかる。この原因は、前述のようにforward側の107 bpのリードファイル（QC.1.trimmed.fastq）にある。特に、100-107塩基付近に乳酸菌に由来しないもの（以下、$f_{100-107}$）がトリムしきれずに多く残っているためである。ただし、これは乳酸菌ゲノム配列にQuasR[19]を用いてリードのマッピングを行った結果（後述）を眺めることで後に判明したことである。

アセンブル結果のみを眺めていた当時は、single-endのみで実行した結果よりもpaired-endの結果のほうが悪いという、理解に苦しむ現象に苦悩していた［W6］。具体的には、①forward側ファイル（QC.1.trimmed.fastq）のsingle-endアセンブル結果が1 transcripts（107 bp）、②reverse側ファイル（QC.2.trimmed.fastq）のsingle-endアセンブル結果が423 transcripts（平均437 bp）、そして③paired-endのアセンブル結果が0 transcriptsであった（Rockhopper2 ver. 2.0.3）。

Rの基本的な利用法とパッケージのインストール

Bio-Linux 8にはR[31]がプレインストールされている。著者らの環境では、2015年4月にリリースされたR（ver. 3.2.0）が利用可能である。Biostringsなどいくつかの代表的なパッケージもプレインストールされているものの、マッピングからカウントデータ取得まで行えるQuasRを含む比較的最近のパッケージは、インストールから行う必要がある。ここでは、ゲストOS（Bio-Linux8）上でのRの基本的な利用法とQuasRパッケージのインストール法を示す。ホストOS（WindowsやMacintosh）上でのR本体および各種パッケージのインストールや基本的な利用法については、ウェブサイト（**Rで）塩基配列解析**中の該当項目や拙書[2]などを参照されたい。

Rの起動と終了は、「R」と「q ()」と打てばよい［W7-1］。これがわかれば、基本的な見栄えはホストOS上でのR GUI版と同じであるため、R経験者は心穏やかであろう。但し、パッケージのインストール時は、書き込み権限に起因するエラーを避けるため、通常は「sudo R」として管理者（root）権限でRを起動する［W7-4］。QuasRはBioconductor[32]から提供されている。ウェブサイト上で示されている手順通りに、①source関数でbiocLite.Rをネットワーク経由で読み込んだのち、②インストールしたいパッケージ名（この場合はQuasR）を指定して、

biocLite 関数を実行すればよい［W7-7］。もう 1 つのリポジトリである、CRAN から配布されているパッケージのインストールも、Bioconductor で示されている手順と同じやり方で可能である。CRAN、Bioconductor、パッケージと R 本体との関係については、連載第 1 回[33] で述べた。R 起動後は、pwd，ls，cd などの Linux コマンドを利用することはできない［W8］。getwd()，list.files()，setwd() などの対応する R コマンドで対処してもよいが、R を起動する場所や入力ファイルの絶対パス指定（後述）をうまく利用すれば、R 独特の世界にそれほど深く入り込むことなく解析を終えられるだろう。

R でゲノム解析（Linux 版）

第 1 回の最後の項目（R でゲノム解析）では、*L. casei* 12A ゲノムの gzip 圧縮 FASTA 形式ファイル（Lactobacillus_casei_12a.GCA_000309565.1.22.dna.toplevel.fa.gz）をダウンロードした。そして、解凍後の FASTA 形式ファイルを入力として、ホスト OS（Windows）上の R GUI 版で、原著論文[34] の記載内容と同じ結果が得られることを示した。もちろんこの作業は、ゲスト OS（Bio-Linux 8）内で完結させることができる。ダウンロードと解凍は wget と gunzip コマンド［W9-1］、ゲスト OS 付属のウェブブラウザ Firefox を用いて、一連の R コードをコピー & ペースト（以下、C&P）すればよい［W9-5］。第 1 回の**図 2** と同じ結果が得られていることがわかるであろう［W9-6］。

この例のように、一連のコードが数十行になる場合、毎回コードを全選択して C&P するのは面倒である。第 4 回では、一連のコマンド群を含むファイルを読み込んで実行するシェルスクリプトの基本的な利用法を述べた。R の場合は、一連のコードを保存したファイル（JSLAB5_1.R）を用意しておき、それを source 関数で読み込むことで同様の目的を達成することができる［W10-1］。このやり方は、R を一旦起動し「大なり記号（>）」のプロンプトが出ている状態で行うというものである。これは「対話モード」での作業に相当し、ホスト OS 上の R GUI 版で行う通常のやり方と同じである。Linux のコマンド入力待ち状態で「R」と打つと R の世界（対話モード）に入るが、R の世界に入ることなく実質的に Linux コマンドの一部のような感覚で利用することもできる。それが「バッチモード」と呼ばれるものである。最も簡単な例は、R のバージョンを調べる目的で利用する「R --version」であろう［W10-4］。「R --version」実行後は、R 終了時に必要な「q ()」を打ち込むことなく、通常の Linux コマンド入力待ち状態に戻っていることがわかる。

バッチモードで R スクリプトファイル JSLAB5_1.R を実行する最小限のコマンドは、「R --vanilla < JSLAB5_1.R」である［W10-5］。但し、一般的には「R --vanilla --slave < JSLAB5_1.R」のように、--slave オプションも同時に用いられる［W12-1］。ウェブサイト（**R で）塩基配列解析**中の多くの項目は、必要な入力ファイルが作業ディレクトリ中にあるという前提で記述されているので、この基本形を踏襲すればよい。発展形として、例えば入力ファイルを絶対パスで指定することで、作業ディレクトリ上にない入力ファイルを読み込むこともできる［W12-3］。ここでは、第 1 回当時と同じ Ensembl[35] Bacteria（Release 22）のゲノムファイルを用いて解析結果の再現性（28 コンティグ；2,885,619 bp）を重視した。しかし、最新版は Release 28（2015 年 9 月 9 日現在）であり、1 本の環状染色体（2,907,892 bp）となっている。よほどのことがないかぎり、最新版を利用したほうがいいだろう［W13］。

マッピング（R ver. 3.2.0；QuasR ver. 1.8.4）

QuasR[19] は、フィルタリングやアダプター除去を含む QC、マッピング、カウントデータ取得まで行う守備範囲の広い R パッケージである。R の講習会や大学院講義でも数年前から取り上げているため、比較的多くの読者がこのパッケージの存在を知っているかもしれない。Linux 環境で通常利用するマッピングプログラムではないが、本稿で取り上げたのは下記理由による：

①ホスト OS（Windows や Macintosh）上での利用を想定して記述されているウェブサイト（**R で）塩基配列解析**を Linux 環境で利用するための橋渡し。QuasR は多くの項目で利用されている。
②著者らが実際に Windows 環境で乳酸菌 RNA-seq データの *de novo* transcriptome assembly でうまくいかなかった理由を突き止められたのが、後述する QuasR 出力結果の Mismatche bases という項目だった。
③第 5 回は、これまでに述べてこなかった事柄を織り交ぜつつ、著者らが実体験し、実際にとった行動、原因究明に至る思考回路の伝授が主目的。

ここでは、*L. casei* 12A ゲノム（Release 28）への RNA-seq リードのマッピングを行う。乳酸菌は、遺伝子と転写物が 1:1 対応である。つまり、イントロンがないため、TopHat2[36] のような計算コストのかかる spliced aligner（複数エクソン間をまたぐジャンクションリードにも対応したマッピングプログラム；splice-aware aligner などとも呼ばれる）をわざわざ使う必要がない。QuasR（ver. 1.8.4）は、spliced aligner（内部的に SpliceMap[37] を利用）と unspliced aligner（内部的に Bowtie[38] を利用）の両方の機能を持っており、デフォルトでは unspliced aligner（basic aligner などとも呼ばれる）が実行される。もちろん single-end と paired-end の両方に対応しており、マップする側の FASTQ ファイルが gzip 圧縮状態のままでもよい［W14-1］。ただし、マップされる側のリファレンス配列は非圧縮状態でなければならない［W14-5］。

QuasR を実行するためには、当然ながらマップする側とされる側の 2 つの情報を与える必要がある。マップする側の情報は、リストファイルとして与える仕様になっており、複数サンプルのマッピングを一度に実行可能である。[W14-2]。ここでは、QC 前（SRR616268sub_1.fastq.gz と SRR616268sub_2.fastq.gz）と FaQCs による QC 後（QC.1.trimmed.fastq.gz と QC.2.trimmed.fastq.gz）の計 4 つのファイル名情報を含むリストファイル（ファイル名：JSLAB5_4.txt）を入力として与えている。マップされる側のリファレンス配列（ファイル名：Lactobacillus_casei_12a.GCA_000309565.2.28.dna.toplevel.fa）は、必ずしもマップする側と同じディレクトリ上にある必要はない。QuasR 実行用スクリプトファイル（ファイル名：JSLAB5_5.R）では、リファレンス配列を絶対パスで示している [W14-4]。スクリプトファイルの実行は、最低限「R --vanilla < JSLAB5_5.R」だけでよく、著者らの環境では約 15 分で終了した [W14-5]。

QuasR 実行結果の PDF レポートを眺めると、QC 前のデータは forward と reverse 合わせて 200 万リード（total=2e+06）のうち、約 0.4% しかマップされなかったことがわかる [W15-6]。QC 後のデータは、ごくわずかにマップ率が改善されたものの、計（3,908,808/4）×2 = 1,954,404 リード（total=1.95e+06）のうち約 0.5% でありほぼ誤差範囲である。著者らは、**図 2** に示す「リードのポジションごとのミスマッチ塩基の割合」を眺めることで、*de novo* アセンブルやマッピングが不調に終わった主因を理解した [W15-7]。つまり、前述の $f_{100-107}$ がトリムしきれずに多く残っていたということである。この図は、おそらく数少ないマップされたリードのうち、ミスマッチがあった塩基のポジション分布を示しているのだろう。このデータの場合、reverse 側はマップされたものの、forward 側がマップされなかったリードがほとんどだったと思われる。両方マップされたリードペアのみを出力する仕様のために、99% 以上のリード（ペア）がマップされなかったという結果になったのだろう。reverse 側のみ良好で paired-end の場合に *de novo* アセンブルがうまくいかなかったのも [W6]、$f_{100-107}$ に阻まれてペアでアセンブルされたリードが 1 つもなかったのだと解釈すればよい。Rockhopper2 は、（個人的には違和感があるが）ペアのリード同士がつながったものだけを出力する方針なのだろう。

対策（QC）

アセンブルやマッピングを改善する最も効果的な手段は、$f_{100-107}$ をトリムすることである。これは、上記マッピング結果までを眺め、そうすればいいだろうと思い、改善することを確認した上で述べている。トリミングの 1 つの手段は、R の Biostrings パッケージの利用である [W16-1]。このパッケージは Bio-Linux 8 にプレインストールされているため、QuasR のようにパッケージのインストールから行う必要はない。「R でゲノム解析（Linux 版）」の節で利用したコードをよく見ると、library（Biostrings）として Biostrings パッケージのロードを行っていることに気づくであろう [W9]。Biostrings パッケージが提供する readDNAStringSet という関数のおかげで、FASTA 形式ファイルの読み込みを行うことができるのである。他の手段としては、（こちらがより一般的ではあるが）FASTX-toolkit（ver. 0.0.14）で提供されている fastx_trimmer の利用が挙げられる [W16-2]。プログラムの本質的な部分にはバグがないことを、著者らも確認済みである。gzip 圧縮ファイル状態での入力に対応していないため、gunzip 実行結果をパイプで流す必要があるものの、一連のコマンドを一塊のものとみなして実行すれば何の問題もない。最近は多機能なプログラムが多いが、今回のような他に一切余計なことをしてもらいたくない場合には、今でもこの種の単機能なプログラムが利用される。

$f_{100-107}$ トリム後で FaQCs 実行前の paired-end データで再度行った Rockhopper2 によるアセンブル結果は、794 transcripts（平均 565 bp）であった。これは、トリム前で FaQCs 実行後の paired-end 結果（0 transcripts）[W5-2] はもちろんのこと、トリム前で FaQCs 実行前（424 transcripts；平均 436 bp）[W17-6] および実行後（423 transcripts；平均 437 bp）[W6-4] の reverse 側のみの single-end 結果と比べても明らかに改善されていると言ってよいだろう。QuasR によるマッピング結果についても、トリム後にマップされたリードの割合は 34.6% であり、トリム前（0.4%）と比べて劇的に改善されていることを確認済みである [W18-6]。

おわりに

第 5 回は、乳酸菌 RNA-seq データの QC において、比較的新しい QC 用プログラム（FaQCs）でもトリムしきれていない領域が存在し、それらが *de novo* アセンブルやゲノムへのマッピング時に決定的な悪影響を及ぼしうることを示した。また、Java プログラム（Rockhopper2）のクラスパス設定とその利用、R パッケージ（QuasR）のインストール法とバッチモードでの効率的な利用法を紹介しつつ、ウェブページ（**R で）塩基配列解析**中の多くの項目を Linux 環境で活用するためのノウハウを示した。著者らの本職は数値解析であり、アセンブルやマッピングなどを通常業務とする配列解析屋ではない。そのため、今回取り扱った乳酸菌データが運悪く解析が難しかったのか、それとも比較的一般的な事象なのかは不明である。第 1 回でも述べた Galaxy [39] や DDBJ パイプライン [40] などを使えば、今回遭遇した $f_{100-107}$ 問題に気づくことすらなく、よりよい解析ができたかもしれない。著者らの知る限り、このデータの原著論文は未だ公開されていない。submitter らの研究グループは、もしかしたら今回我々が発見した $f_{100-107}$ 問題にまだ気づいておらず、データ解析で苦悩しているのか

もしれない。

話の展開上本文中では省略したが、結論としては $f_{100-107}$ 問題に QC 段階で気づくことはできる［W15-5］。具体的には、--nogroup オプションをつけて FastQC を実行した結果を眺めればよい。特に Kmer Contents の項目は、ゲノムアセンブリのところでも述べた *k*-mer（ver. 0.11.3 のデフォルトは *k*=7）の出現頻度をリードのポジションごとに調べ、出現頻度の期待値に比べて実測値が極端に多い上位の *k*-mer とその位置をリストアップしたものである。また、--nogroup は「長いリードの場合に 10 番目以降のポジションを一定幅でグループ化する（デフォルト）」機能をオフにするオプションである［W19-1］。著者らは、--nogroup オプションの有無によって Kmer Contents 項目の結果までが異なることを最近まで知らなかった。つまり、--nogroup オプションをつけずにデフォルトで実行した FastQC の結果（Kmer Contents や Per base sequence content 項目）を眺めていたがために、$f_{100-107}$ 問題に気づけなかったのである［W19-4］。第 6 回は、アセンブルプログラム Velvet をオプションつきでインストールすることで指定可能な数値範囲を変更できること、複数の異なる *k*-mer で実行した乳酸菌ゲノムアセンブル結果の違いなどを紹介する予定である。

謝　辞

本連載の一部は、国立研究開発法人科学技術振興機構バイオサイエンスデータベースセンター（NBDC）との共同研究の成果によるものです。乳酸菌 *Lactobacillus hokkaidonensis* LOOC260^T ゲノム配列決定部分については、原著論文著者（遠野雅徳氏、谷澤靖洋氏、神沼英里氏、中村保一氏、有田正規氏）より詳細情報をいただきました。

参 考 文 献

1) 門田幸二（2015）平成 26 年度 バイオインフォマティクス人材育成カリキュラム（次世代シークエンサ）速習コース 実施報告書，http://biosciencedbc.jp/gadget/human/h26_ngs_report.pdf
2) 門田幸二（2014）シリーズ Useful R 第 7 巻 トランスクリプトーム解析，金明哲 編，共立出版，東京.
3) 孫建強，湯敏，清水謙多郎，門田幸二（2015）次世代シーケンサーデータの解析手法：第 4 回クオリティコントロールとプログラムのインストール．日本乳酸菌学会誌 **26**：124-132.
4) Field D, Tiwari B, Booth T, Houten S, Swan D, et al. (2006) Open software for biologists: from famine to feast. Nat Biotechnol **24**: 801-803.
5) Lo CC, Chain PS. (2014) Rapid evaluation and quality control of next generation sequencing data with FaQCs. BMC Bioinformatics **15**: 366.
6) Andrews S. (2015) FastQC a quality control tool for high throughput sequence data, http://www.bioinformatics.babraham.ac.uk/projects/fastqc/
7) Lab H. (2010) FASTX-Toolkit, http://hannonlab.cshl.edu/fastx_toolkit/
8) Kelley DR, Schatz MC, Salzberg SL. (2010) Quake: quality-aware detection and correction of sequencing errors. Genome Biol **11**: R116.
9) Gnerre S, Maccallum I, Przybylski D, Ribeiro FJ, Burton JN, et al. (2011) High-quality draft assemblies of mammalian genomes from massively parallel sequence data. Proc Natl Acad Sci USA **108**: 1513-1518.
10) Tanizawa Y, Tohno M, Kaminuma E, Nakamura Y, Arita M. (2015) Complete genome sequence and analysis of *Lactobacillus hokkaidonensis* LOOC260^T, a psychrotrophic lactic acid bacterium isolated from silage. BMC Genomics **16**: 240.
11) Chin CS, Alexander DH, Marks P, Klammer AA, Drake J. (2013) Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. Nat Methods **10**: 563-569.
12) Kajitani R, Toshimoto K, Noguchi H, Toyoda A, Ogura Y, et al. (2014) Efficient de novo assembly of highly heterozygous genomes from whole-genome shotgun short reads. Genome Res **24**: 1384-1395.
13) Thorvaldsdóttir H, Robinson JT, Mesirov JP. (2013) Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. Brief Bioinform. **14**: 178-192.
14) Cuccuru G, Orsini M, Pinna A, Sbardellati A, Soranzo N, et al. (2014) Orione, a web-based framework for NGS analysis in microbiology. Bioinformatics **30**: 1928-1929.
15) Lin SH, Liao YC. (2013) CISA: contig integrator for sequence assembly of bacterial genomes. PLoS One 8: e60843.
16) Seemann T. (2014) Prokka: rapid prokaryotic genome annotation. Bioinformatics **30**: 2068-2069.
17) Jiang H, Lei R, Ding SW, Zhu S. (2014) Skewer: a fast and accurate adapter trimmer for next-generation sequencing paired-end reads. BMC Bioinformatics **15**: 182.
18) Tjaden B. (2015) De novo assembly of bacterial transcriptomes from RNA-seq data. Genome Biol **16**: 1.
19) Gaidatzis D, Lerch A, Hahne F, Stadler MB. (2015) QuasR: quantification and annotation of short reads in R. Bioinformatics **31**: 1130-1132.
20) Surget-Groba Y, Montoya-Burgos JI. (2010) Optimization of de novo transcriptome assembly from next-generation sequencing data. Genome Res **20**:1432-1440.
21) Robertson G, Schein J, Chiu R, Corbett R, Field M, et al. (2010) De novo assembly and analysis of RNA-seq data. Nat Methods **7**: 909-912.
22) Gibbons JG, Janson EM, Hittinger CT, Johnston M, Abbot P, et al., (2009) Benchmarking next-generation transcriptome sequencing for functional and evolutionary genomics. Mol Biol Evol **26**: 2731-2744.
23) Miller JR, Koren S, Sutton G. (2010) Assembly algorithms for next-generation sequencing data. Genomics **95**: 315-327.
24) Grabherr MG, Haas BJ, Yassour M, Levin JZ, Thompson DA, et al. (2011) Full-length transcriptome assembly from RNA-Seq data without a reference genome. Nat Biotechnol **29**: 644-652.
25) Chang Z, Li G, Liu J, Zhang Y, Ashby C, et al. (2015) Bridger: a new framework for de novo transcriptome assembly using RNA-seq data. Genome Biol **16**: 30.
26) Fu L, Niu B, Zhu Z, Wu S, Li W. (2012) CD-HIT: accelerated for clustering the next-generation sequencing data. Bioinformatics **28**: 3150-3152.
27) Mbandi SK, Hesse U, van Heusden P, Christoffels A. (2015) Inferring bona fide transfrags in RNA-Seq derived-transcriptome assemblies of non-model organisms. BMC Bioinformatics **16**: 58.
28) Trapnell C, Williams BA, Pertea G, Mortazavi A, Kwan G, et al. (2010) Transcript assembly and quantification by

RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. Nat Biotechnol **28**: 511-515.
29) McClure R, Balasubramanian D, Sun Y, Bobrovskyy M, Sumby P, et al. (2013) Computational analysis of bacterial RNA-Seq data. Nucleic Acids Res **41**: e140.
30) Li S, Dong X, Su Z. (2013) Directional RNA-seq reveals highly complex condition-dependent transcriptomes in *E. coli* K12 through accurate full-length transcripts assembling. BMC Genomics **14**: 520.
31) R Core Team (2015) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.
32) Huber W, Carey VJ, Gentleman R, Anders S, Carlson M, et al. (2015) Orchestrating high-throughput genomic analysis with Bioconductor. Nat Methods **12**: 115-121.
33) 門田幸二，孫建強，湯敏，西岡輔，清水謙多郎，(2014) 次世代シーケンサーデータの解析手法：第1回イントロダクション．日本乳酸菌学会誌 **25**：87-94.
34) Broadbent JR, Neeno-Eckwall EC, Stahl B, Tandee K, Cai H, et al. (2012) Analysis of the *Lactobacillus casei* supragenome and its influence in species evolution and lifestyle adaptation. BMC Genomics **13**: 533.
35) Cunningham F, Amode MR, Barrell D, Beal K, Billis K, et al. (2015) Ensembl 2015. Nucleic Acids Res **43**: D662-669.
36) Kim D, Pertea G, Trapnell C, Pimentel H, Kelley R, et al. (2013) TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. Genome Biol **14**: R36.
37) Au KF, Jiang H, Lin L, Xing Y, Wong WH. (2010) Detection of splice junctions from paired-end RNA-seq data by SpliceMap. Nucleic Acids Res **38**: 4570-4578.
38) Langmead B, Trapnell C, Pop M, Salzberg SL. (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol **10**: R25.
39) Goecks J, Nekrutenko A, Taylor J; Galaxy Team. (2010) Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biol **11**: R86.
40) Nagasaki H, Mochizuki T, Kodama Y, Saruhashi S, Morizaki S, et al. (2013) DDBJ read annotation pipeline: a cloud computing-based pipeline for high-throughput analysis of next-generation sequencing data. DNA Res **20**: 383-390.

Methods for analyzing next-generation sequencing data V. assembly, mapping, and quality control

Jianqiang Sun[1], Kentaro Shimizu[1, 2], and Koji Kadota[2]

[1]Department of Biotechnology, [2]Agricultural Bioinformatics Research Unit, Graduate School of Agricultural and Life Sciences, The University of Tokyo.

Abstract

RNA-seq differential expression analysis workflow generally consists of four steps: (i) retrieving data, (ii) quality control (QC), (iii) de novo assembling and/or read mapping, and (iv) statistical analysis. We explain the third step with a recent QC program FaQCs (ver. 1.34). We introduce de novo transcriptome assembly by Rockhopper2 (ver. 2.0.3; a Java program) and mapping for a *Lactobacillus* genome by QuasR (ver. 1.8.4; an R/Bioconductor program). We demonstrate the importance of QC.