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8回 アセンブリ後の解析

谷澤 靖洋[1]、神沼 英里[1*]、中村 保一[1]、遠野 雅徳[2]、寺田 朋子[3]、清水 謙多郎[3]、門田 幸二[3*]

[1] 国立遺伝学研究所生命情報研究センター
[2] 農業・食品産業技術総合研究機構 畜産研究部門
[3] 東京大学大学院農学生命科学研究科

de novo ゲノムアセンブリ結果から、概要・完全配列（draft and complete genome sequences）にする作業は、基本的な塩基配列解析用プログラムの活用や自作、プログラム実行結果の検証や合理的な解釈など、ウェットとドライ両面の幅広い知識とスキル、そして精神力を要する。第8回は、PacBio データの *de novo* ゲノムアセンブリの後処理として、特に染色体ゲノムに相当する長いコンティグの検証作業を解説する。具体的には、DFAST による乳酸菌に特化したアノテーション、BLAST の実行と可視化、環状染色体の完成、Illumina データのマッピングによる検証と修正について述べる。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を効率的に活用してほしい。

Key words：genome assembly, completion, mapping, blast

はじめに

第8回は、乳酸菌（*Lactobacillus hokkaidonensis* LOOC 260^{T}）ゲノム配列決定論文[1)]のデータ解析部分を引き続き解説する。前回までに PacBio データ（DRR054113）の *de novo* ゲノムアセンブリ結果として、4つのコンティグが得られた[2-5)]。配列の長い順（2,289,497 bp, 86,892 bp, 45,853 bp, and 11,372 bp）に sequence1 から sequence4 として取り扱い、そのうち配列の両末端部分の重複を確認[6-7)]した sequence3 および sequence2 については、環状のプラスミド配列であろうと判断されている。今回は、残りの sequence1 および sequence4 についての検証を中心に行い、完全ゲノム配列（complete genome sequences）の一歩手前までの工程を紹介する。

第7回までの原稿やウェブ資料を理解済みであるという前提で話を進めるが、第7回を含む各回終了時点の ova ファイルを引き続き提供している。これまで実習を実際にはやっていない（エアーハンズオン）読者は、第6回[8)] W1-3 で示した「共有フォルダ設定情報を含む ova ファイルからのインストール手順」を参考にして Bio-Linux 環境構築にチャレンジしてほしい。本連載の枠組み以外にも、Bio-Linux 環境構築後の基本的な作業や共有フォルダの概念などを理解することを目的とした初心者向け（学部3年生程度）の教材を提供している［**W1**］。これらを足がかりとしてもよいだろう。

第7回の後半は、共有フォルダ上で作業を行った。作業自体はゲスト OS 側で行ったが、共有フォルダ内のファイルの実体はゲスト OS（Bio-Linux）の「~/Desktop/mac_share」ではなく、ホスト OS（Windows or Macintosh）の「~/Desktop/share」にある点に注意してもらいたい。

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

これは、第7回終了時点のovaファイル中には、共有フォルダ内の情報は含まれていないことを意味する。また、ホストOSの「~/Desktop/share」フォルダの中身が、第7回終了時点と異なるヒトも一定数見込まれる。これらの事情を踏まえ、*de novo* アセンブリ実行結果ファイル（result.zip）のダウンロード（第7回W9-3）以降の作業を、ゲストOS（Bio-Linux）の「~/Desktop」上で行う手順として再提示した[**W2**]。これにより、次項（ゲノムアノテーション）で用いるLH_hgap.faの存在場所は、上記作業を改めて行ったヒトは「~/Desktop/result」に、そうでないヒトは「~/Desktop/mac_share/result」になる。これ以外の違いについても、各自の事情に応じて適宜読み替えてもらいたい。シェルスクリプトの基本形についても示したので、余裕のあるヒトはスキルアップに努めてほしい［**W3**］。

ゲノムアノテーション

ゲノムのアノテーション（genome annotation）とは、ゲノム上のどの位置にどのような遺伝子がコードされているかなどを調べ、注釈づけを行う作業である。バクテリアの自動アノテーションに関しては、MiGAP[9]やRAST[10-12]などの様々なウェブサービスが提供されており、手軽に実行可能である。今回は、乳酸菌に特化したアノテーションパイプラインDFAST（DDBJ Fast Annotation and Submission Tool）[13]を用いる。本ウェブサービスは、連載第5回[14]でも紹介したアノテーションパイプラインProkka[15]をベースとして、乳酸菌（主に*Lactobacillus*属および*Pediococcus*属）用に整備された参照データベースを組み合わせたものである。また、アノテーションだけでなく、DDBJ[16]への塩基配列登録支援を行うこともできる（もちろん登録は任意）のが特徴である。典型的なゲノムサイズ（数MB程度）の乳酸菌であれば、5分ほどで結果が返される。ここでは、アセンブリ結果の検証の一環として、アセンブリ結果ファイル（LH_hgap.fa）を入力としてDFASTを実行する［**W4**］。位置づけとしては、予備的なアノテーションである。

オプションとして、Job Titleには好きな名称をつけられ、DDBJ Pipeline実行時と同じくジョブ完了通知をメールで受け取ることもできる。“Minimum Contig Length”オプションは、設定値以下の短い断片配列を除くためのものである。今回の入力配列は、全てデフォルトの200塩基以上なので影響はない。属・種名などのオプションは、デフォルトのままで構わない。アノテーション結果に影響を与えることはなく、後で変更することも可能だからである［**W4-3**］。主なアノテーション結果は、入力ファイル中の配列の順番通りに、「どの配列（コンティグ）中のどの座標上にどんな遺伝子が存在するか」という情報である。ウェブ上のFeaturesタブ経由で見られる情報以外に、拡張子が.gbkのGenbank形式やGFF3形式ファイルがダウンロード可能である。また、CDSやRNA配列のFASTA形式ファイルなども提供されている［**W4-5**］。

アノテーションされた遺伝子を概観する。大まかな指針として、hypothetical proteinは既知のアミノ酸に対して相同性が認められなかったものや機能未知のものであるため現段階では無視してよい。また、transposaseなどトランスポゾン（動く遺伝子；転移因子）関連のものは、自己複製的にゲノム中に複製されていくため比較的多く見られる。数十～数百コピー以上見られることも特別なことではないため、基本的に気にしなくてよい。アセンブリ結果の検証に資する部分について列挙する。まずsequence2（86,892 bp）とsequence3（45,853 bp）については、プラスミドの複製に関連する遺伝子（plasmid replication protein）［**W4-7**］や、接合伝達に関わる遺伝子（conjugal transfer protein）が見られる［**W4-8**］。これらの結果は、第7回までで予想していた通り、この2つの配列が（環状の）プラスミドであることを支持している。

sequence1（2,289,497 bp）については、最初の30,000塩基あたりまでにprophage proteinなどファージ関連遺伝子が見られる［**W4-6**］。また、sequence4（11,372 bp）については、4,912番目から5,699番目の領域にtransposaseがコードされている。この領域については後で議論するが、アノテーション結果の概観レベルでは見逃してもよい。後述するBLAST[17]の実行結果と合わせて総合的に判断する視点が重要である。

BLASTの実行と可視化

第7回では、sequence3同士を例として配列内の両末端部分の重複をドットプロットで大まかに調べ（第7回W14-2）、BLASTで重複領域の詳細なアラインメントを行った。ドットプロットについては、sequence3よりも2倍程度長いsequence2同士についてもdotter[18]を実行可能であり、両末端部分の重複が見いだせる［**W5-1**］。sequence4同士のドットプロットは、［1, 500 bp］と［750, 1350 bp］付近の領域が似ているものの、環状を示唆する結果は得られなかった［**W5-2**］。約2.3Mbと最も長いsequence1同士は、ゲストOS（Bio-Linux）への割り当てメモリが2GBの著者らのPC環境では、dotterを実行できなかった（数分程度では描画できなかった）［**W5-3**］。

通常は、sequence1 vs. sequence1のような同一配列間の比較以外にも、sequence1 vs. sequence4のような異なる配列間の比較も行い、配列類似領域を探索する。合計4配列しかないこのアセンブリ結果の場合はそれほどの作業量でもないが、ペアワイズアラインメントだと一般に組合せ数が膨大になる。BLASTはこのような局面でも威力を発揮する。例えば、query側の配列としてsequence1を、データベース（DB）側の配列としてLH_hgap.faを指定すれば、（自分自身を含む）4通りの比較に相当する［**W6**］。

但し、sequence1 は非常に長いため、デフォルト出力形式の BLAST 実行結果ファイル（sequence1_blast.txt）を眺めて全体像を把握するのは困難である。

もちろん、BOV[19] や BLASTGrabber[20]［**W7-1**］など、BLAST 実行結果の全体像を把握するための可視化ソフトウェア（ビューワ；viewer）は存在する。ここでは、BlastViewer を利用する［**W7-2**］。BlastViewer は Windows 用と Macintosh 用のみが提供されているため、ホスト OS 上でインストールして利用する。XML 形式の BLAST 実行結果ファイル（sequence1_blast.xml）しか受け付けないが、DB 側の配列ごとにヒット数（配列類似領域数；HSP 数）が示されているなど、全体的な操作感がよい［**W7-4**］。例えば、sequence1 に対するヒット数が 1,347 個、sequence4 が 2 個、sequence3 と sequence2 がそれぞれ 1 個であったことがわかる。また、ヒット数やスコア分布の全体像を眺めることで、sequence1 にいくつかの重複領域が存在することや、11,372 bp からなる sequence4 の大部分の領域が sequence1 と類似していることなどがわかる［**W8-1**］。

sequence4 は sequence1 の一部

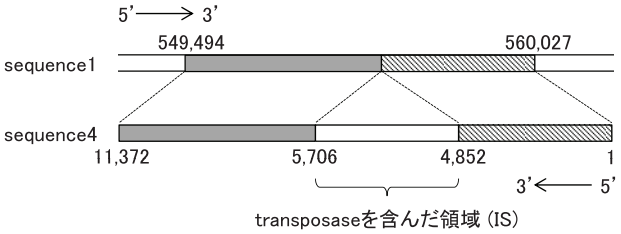
BlastViewer で sequence4（11,372 bp）に対する sequence1（2,289,497 bp）のヒット領域を眺める［**W8**］。スコアの高いほう（Score＝10,320）の 1 つめの HSP（以下、HSP1）は、sequence1 の［549494, 555171 bp］と、sequence4 の［11372, 5706 bp］の領域から形成されていた［**W8-3**］。また、スコアの低いほう（Score＝8,907）の 2 つめの HSP（以下、HSP2）は、sequence1 の領域［555167, 560027 bp］と、sequence4 の領域［4852, 1 bp］から形成されていた［**W8-4**］。いずれの HSP も、sequence1 が Plus（+）鎖、sequence4 が Minus（-）鎖でアラインメントされていた。これは、sequence1 の一続きの領域［549494, 560027 bp］と、領域［4853, 5705 bp］を除く sequence4 の全長がほぼ一致していることを意味する（**図 1a**；W8-5）。

アラインメントされなかった sequence4 の領域［4853, 5705 bp］に対するアノテーション結果を眺めると、transposase がコードされていたことがわかる（**図 1b**；W8-5）。これは、該当領域が挿入配列（insertion sequence; IS）であることを示唆する。これはおそらく、乳酸菌の培養途中で一部の細胞に IS の挿入が起こったためであろう。結果として、シークエンスされた細胞集団の中に IS を含むものと含まないものが混在することになり、sequence4 が独立したコンティグとして出力されたものと思われる。sequence4 は全体的にクオリティスコアが低く（第 7 回 W11-7）、また、後述の Illumina によるシーケンス結果には当該部分が確認できなかったことから、IS が挿入された細胞の存在比率は高くないと考え、sequence4 は除外した。

sequence1 はプロファージ領域を含む環状染色体

BlastViewer で sequence1（2,289,497 bp）に対する

(a)



(b)

sequence4	4133..4753	CDS	mannose-specific PTS system IID component
sequence4	4781..4873	CDS	mannose-specific PTS system IIA component
sequence4	4912..5220	CDS	transposase
sequence4	5361..5699	CDS	transposase
sequence4	5711..6058	CDS	mannose-specific PTS system IIA component
sequence4	6114..6596	CDS	mannose/fructose/sorbose-specific PTS system IID component

図 1. sequence4 は sequence1 の一部
（a）BLAST 実行結果の模式図。sequence1 の一続きの領域［549494, 560027 bp］と、領域［4853, 5705 bp］を除く sequence4 の全長がほぼ一致していることがわかる。（b）DFAST によるアノテーション結果の一部抜粋。sequence4 の領域［4853, 5705 bp］には、transposase がコードされている。

sequence1（2,289,497 bp）のヒット領域を眺める［**W9**］。最もスコアの高いHSP（HSP1）はsequence1同士の全長が100%一致のものであるため、2番目のHSP（HSP2）以降のアラインメントが精査対象となる。総ヒット数（1,347個）が非常に多いため、ここではスコアが4,000以上という条件を満たすHSP1-33の一致領域をまとめた［**W9-2**］。ゲノム配列決定論文[1]中では、結論として下記の計4領域が議論の対象となっているが、どこまでのHSPを眺め、どのように解釈して結論づけるかは、利用可能な情報を徹底的に調べ、試行錯誤しながら整理していく以外にはないだろう。

・HSP4（HSP5）の一致領域①［1, 5860 bp］と①'［37329, 43187 bp］
・HSP8（HSP9）の一致領域②［5839, 11509 bp］と②'［2283820, 2289497 bp］

上記以外のHSPのアノテーションとして、HSP10, 14, 22, 24, 25, 28の領域には、ribosomal RNAがコードされていた。また、残りのHSP2, 6, 12, 16, 18, 20, 30, 32は、［2057850, 2065197 bp］にかけての反復構造を含んだ領域の中に全て含まれ、この中には3つの遺伝子（adhesion exoprotein, mucus-binding protein, and hypothetical protein）がコードされていた［**W9-3**］。遺伝子名からは細胞接着に関わる表層タンパクと推察され、ここに見られた反復構造が何らかの働きを持っているのかもしれない。いずれもsequence1末端から10,000塩基以上離れた領域であることから、アセンブリ結果の検証という点では無関係であろう。

sequence1末端付近の重複は、HSP8の一致領域（②と②'）に相当する。これにHSP4の一致領域（①と①'）を含めた模式図を示す（**図2a**；W10-1）。もし①の領域がなければ、sequence2や3と同様に「両末端の重複 → 環状コンティグ」と結論づけられる。現在利用可能な他の情報として、アノテーション結果の領域［1, 37328 bp］を再度眺める。主な視点は、「なぜ①が存在するのか？、②および（②と①'の間の）領域［11510, 37328 bp］にはどんな遺伝子がコードされているのか？」である。我々は当時、領域［1, 43187 bp］にファージ関連遺伝子が多くコードされている事実を突き止め［**W10-2**］、領域［5839, 43187 bp］がプロファージ領域であろうと予想した。次に、ファージの挿入・切出し機構[21]を調べ、実際の染色体構造（**図2b**；W10-3）や、プロファージ領域が染色体から切り出されて環状化した状態（**図2c**；W10-4）の予想を立て、PCRによる確認やプロファージの他の特徴を満たすかどうかについても検証した［**W10-5**］。ここまでの結果を踏まえ、①の領域は環状ファージDNAがシークエンスされた結果として生じたものと考えた［**W10-6**］。つまり、実際の染色体上には存在しないということである。

改めて強調しておきたい点は、**配列相同性とアノテーションの併用**の重要性である。BLAST結果だけではわからないこともある。しかし、どのような遺伝子がコードされているかなど、アノテーション情報と合わせて総合的に判断すればわかる場合もある。また、シークエンス対象**サンプルは均一な集団ではない**点も胆に銘じておかねばならない。実際、今回の乳酸菌サンプル中には、**図2b**で示すようなプロファージ領域を含む環状染色体だけでなく、**図2c**で示すような（i）プロファージ領域が切り出されてできた環状ファージDNAや、（ii）プロファージ領域が切り出されてなくなった残りの環状染色体も含まれていた。このあたりが、本稿の冒頭で述べた幅広い知識や合理的な解釈に相当する。

乳酸菌ゲノム概要配列の作成

ここまで*de novo*アセンブリ結果ファイル（LH_hgap.fa）を入力として、アノテーションや配列相同性検索を行った。得られた方針は下記の通りである：

sequence1：環状染色体（**図2**）。主に①や②'の重複部分を除けばよい。ここでは、W10-4のアラインメントを参

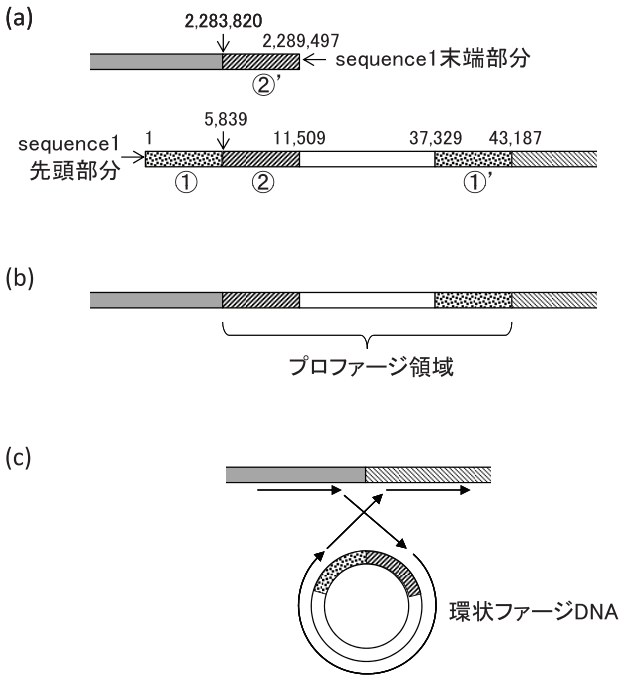


図2. sequence1の構造。
（a）両末端付近の模式図。BLAST実行結果のHSP8（②と②'）が両末端付近で一致（環状であることを示唆）している。HSP4（①と①'）の領域も示されている。（b）実際の染色体構造。アノテーション結果と合わせ、②から①'の範囲［5839, 43187 bp］がプロファージ領域であると予想した。（c）ファージの機構。ファージが染色体に組み込まれる（integration）場合は、矢印の方向に沿って行われる。①の領域は、環状化したファージDNAがシークエンスされた結果として生じたものであり、実際の染色体上には存在しないと結論づけた［**W10**］。

考にして、領域［5839, 2283819 bp］を抽出する。
sequence2：環状プラスミド［**W5-1**］。BLAST で重複領域のアラインメントをとり、領域［2641, 84270 bp］を抽出する［**W11-1**］。
sequence3：環状プラスミド。領域［2450, 43422 bp］を抽出する（第 7 回 W16）。
sequence4：sequence1 の一部なので却下（**図 1**）。

この方針に従って重複除去を行い、概要配列（LH_draft.fa）を作成する［**W11-2**］。どのレベルを「概要（ドラフト）」と呼ぶかはヒトそれぞれであるが、少なくとも重複除去でおしまいではないため、ここでは重複除去後の状態を概要と呼ぶ。

概要配列への MiSeq データのマッピング

乳酸菌ゲノム配列決定の原著論文[1]では、概要配列（LH_draft.fa）の元となった PacBio データ（DRR054113）以外に、paired-end Illumina MiSeq データ（DRR024501）も存在する。ここでは、第 6 回 W5-4 で得られた（forward 側と reverse 側それぞれ）297,633 リードからなる MiSeq データを用いてマッピングを行う［**W12-1**］。目的は、（MiSeq リードのマップ率云々ではなく）概要配列の検証およびエラー補正である。大まかには、マッピング結果の BAM 形式ファイルをもとに、リファレンスの**概要配列と異なる部分のみ**を抽出した VCF（Variant Call Format）と呼ばれる形式のファイルを作成・援用し、可視化ソフトで確認しながら概要配列を補正するという流れになる。

代表的なマッピングプログラムである BWA[22] は、Bio-Linux[6] にプレインストールされているため、マッピング結果ファイルまでは容易に得ることができる［**W12-2**］。しかし DDBJ Pipeline[4] 上で実行すれば、VCF ファイルまで一気に得ることができるので便利である。ここでは、マップされる側のリファレンス配列（LH_draft.fa）とマップする側の Illumina MiSeq データファイル（QC.1.trimmed.fastq.gz と QC.2.trimmed.fastq.gz）を DDBJ Pipeline にアップロードして BWA を実行する。

多くの読者は、DDBJ Pipeline のアカウント作成（第 6 回 W13）、および（Windows 用の FTP クライアントである WinSCP というソフトウェアを用いた）MiSeq データファイルのアップロードおよび登録（第 6 回 W14）までは完了済みであろう。本稿では、新たに Cyberduck という FTP ソフトのインストールと DDBJ Pipeline への接続手順を示した［**W13**］。Cyberduck は、Windows 版と Macintosh 版の両方が提供されている。諸事情により MiSeq データのアップロードに失敗していたユーザは、是非 Cyberduck で再挑戦してみてほしい。尚、リファレンス配列は FTP 経由ではなく HTTP 経由（DDBJ Pipeline の画面上から）のアップロードとなる。

DDBJ Pipeline 上での BWA 実行時のオプションは、基本的にデフォルト設定のままでよい［**W14-5**］。補足として、オプション画面の Step3（ユニーク化処理）は、ペアのリードがともにリファレンス配列の 1 か所にのみマップされたリード（uniquely mapped read; unique mapper）を残し、それ以外のリードを除去している。これは、反復配列のような領域が存在すると、1 つのリードが複数個所にマップされて解析結果の解釈が難しくなるからである。変異解析では、これらのリードは除外して行う場合が多い。

BWA（ver. 0.6.1-r104）実行結果として、297,633 リード中 281,303 個（94.513%）がマップされた［**W15-2**］。リファレンス配列（LH_draft.fa）のゲノムサイズは 2,400,584 bp であるが、そのうち 2,400,552 bp 分がマップされたリードで覆われていた。つまり被覆率（coverage）は、2,400,552 / 2,400,584 = 99.99867% である。また、マップされたリードの総塩基数（130,565,653 bp）をマップされたリードで覆われている領域（2,400,552 bp）で割れば、平均してどれだけの厚み（depth）でマップされているかがわかる。この場合は、depth = 54.390 である。

DDBJ Pipeline 上で BWA を実行すると、マッピング結果の標準形式[23] である SAM（Sequence Alignment/Map の略；拡張子が .sam）および BAM（SAM のバイナリ版；拡張子が .bam）ファイルが生成される。また、リファレンスの概要配列と異なる部分のみを抽出した VCF ファイルも生成される。可視化ソフトの入力として用いるのは、BAM ファイル、BAM のインデックスファイル（後述）、VCF ファイル、そしてリファレンス配列ファイルである。ここでは、説明用の 2 つの SAM ファイルを含め、以下に示す計 5 ファイルを共有フォルダにダウンロードしておく［**W15-3**］：

- 項目「BWA：SAMPE」の out.sam.zip
- 項目「Uniquify SAM（Remove Multiple Hits）」の uniq out.sam.zip
- 項目「Sort BAM File［For Unique SAM］」の out2.bam.zip
- 項目「Create BAM Index File［For Unique SAM］」の out2.bam.bai.zip
- 項目「Filter BCF and Convert to VCF File」の out-unique.var.flt.vcf.zip

SAM/BAM ファイル

ダウンロードした 2 つの SAM ファイルの関係について述べる。項目「BWA：SAMPE」の out.sam.zip（116.3 MB）は、マッピング結果の大元のファイルに相当する。この中から、ユニーク化処理（上述のオプションの Step3）を行って複数個所にマップされたリードやマップされなかったリードを除去したものが、項目「Uniquify

SAM（Remove Multiple Hits）」の uniqout.sam.zip（96.3 MB）である。out.sam.zip はマップされなかったリード情報も含むが、マップ率（94.513%）から大まかに 116.3×0.94513＝110 MB 程度分の情報がマップされたリードに関するものだと推測可能である。ユニーク化処理後でも 96.3 MB であったことから、マップされた全リード（約 110 MB）のうち、ユニークにマップされたリードペアが（96.3 / 110＝）87.5% 程度を占めていると解釈できる。

out.sam（out.sam.zip 解凍後のファイル；371,881,212 bytes；約 355MB）を用いて、SAM 形式[23] を解説する。まず、out.sam の行数は 595,270 であり［**W16-2**］、マップする側の paired-end リードの総数（297,633 リード×2＝595,266）よりも 4 だけ多い。この余分な 4 行は、SAM ファイルのヘッダー部分に相当する。ヘッダー部分には、マップされる側のリファレンス配列の情報（最初の 3 行分）、および用いたマッピングプログラムの情報（4 行目）が記載されている［**W16-4**］。ヘッダー部分の行数はリファレンスの配列数にも依存する。今回用いた LH_draft.fa は、配列数が全部で 3 個（chromosome, plasmid1, and plasmid2）であったことを思い出せば納得できるだろう［**W11-2**］。

ヘッダー行を除いた残り（out.sam の場合は 5 行目以降）がリードごとのマッピング結果情報である。入力は paired-end データなので、2 行で 1 つのペアのマッピング結果を表している。SAM ファイルは、マップする側のリードの塩基配列、クオリティ値、リファレンス配列へのマップの有無、マップされた領域の座標やミスマッチの位置など、1 行の中にリードごとの全情報を書き込んでいるため非常に横長である。このような場合でも、less コマンドの N や S オプションを駆使することで全体像を把握することが可能である［**W16-6**］。マップされなかったリード（unmapped reads）は、3 列目にアスタリスク（＊）がつけられている。リファレンス中には存在しない塩基配列情報を含むため、マップされなかったリードのみを抽出して別の解析を行うこともある［**W16-7**］。

SAM ファイルはタブ区切りテキスト形式であるため、Excel で眺めてもよい［**W17**］。例えば、1 番目のリード（ID が DRR024501.1）は、forward 側・reverse 側ともに 251 塩基の長さで［**W17-2**］、chromosome 上の 565,592 番目（forward 側）および 565,460 番目（reverse 側）の位置が開始点としてマップされ［**W17-3**］、リード間の距離（インサート長）が 383 であることがわかる。この値は、マップされた位置の差（565,592－565,460＝132）に、リード長（＝251）を加えた値として計算されている［**W17-4**］。このデータの場合は、インサート長が 350-400 bp 程度に分布していることがわかる。インサート長が極端に異なる数値を示している箇所は、ミスアセンブリの可能性を検討することになる。尚、uniqout.sam.zip は、out.sam の 12 列目に含まれる XT というタグを目印として、ユニークにマップされたリードペアを抽出したものである［**W17-5**］。BWA には、3 つのアルゴリズム（BWA-backtrack, BWA-SW, and BWA-MEM）が実装されており、DDBJ Pipeline のデフォルトは BWA-backtrack（ver. 0.6.1-r104; 2016 年 9 月 12 日現在）である。XT タグは BWA-backtrack 利用時のみ出力される。BWA-SW や BWA-MEM 利用時には XT タグが出力されないため、これらの方法ではユニーク化処理を行うことができない。

SAM/BAM ファイルは、マップされた位置順にソートし直すのが一般的である。第一義的な理由は、多くの可視化ソフトがソート後の BAM ファイルを入力の前提としているためであるが、ソートした結果を眺めることでインサート長が大きく異なる箇所を探しやすいというメリットもある。項目「Sort BAM File［For Unique SAM］」の out2.bam.zip がソート後の BAM ファイルに相当し、一般に「ソート前の SAM ファイル → ソート前の BAM ファイル → ソート後の BAM ファイル」の流れで作成される。out2.bam はバイナリファイルであり直接眺めることはできないが、samtools view コマンドで（実質的に SAM ファイルに変換したうえでそれを less で）見ることも可能である［**W17-6**］。尚、BAM インデックスファイルは、可視化ソフト上で検索を高速に行うためのおまじないのようなものであり、エンドユーザが記述内容を理解する必要はない。項目「Create BAM Index File［For Unique SAM］」の out2.bam.bai に相当する。通常、インデックスファイルはソート済み BAM ファイル（out2.bam）と同じディレクトリ内に置いて使用する。

VCF ファイル

VCF は、マッピング結果からリファレンスの**概要配列と異なる部分のみ**を抽出したファイルであり、Variant Call Format の略である。エンドユーザにとっては DDBJ Pipeline 上で BWA を実行した結果として得られるように見えるが、実際には BWA 実行結果をもとに VCFtools[24] というソフトを使用して内部的に計算処理を行うことで得られている［**W15-3**］。ダウンロード済みの out-unique.var.flt.vcf は、55 行からなる［**W18-1**］。# から始まる最初の 27 行がヘッダー部分であり、残りの 22 行分が検出された変異情報である。VCF は 1 つの変異を 1 行で表現するため、PacBio のアセンブリ結果から得られた概要配列（LH_draft.fa）中の変異は 22 箇所ということになる。VCF ファイル中に含まれる genotype には、0/1 と 1/1 の 2 種類が存在する。ここで、0 は概要配列側（VCF ファイル中の REF 列）のアリル、1 は変異側（ALT 列）のアリルである。0/1 は概要側と変異側の両方が存在する（2 倍体生物における）ヘテロ接合体（heterozygote）を、1/1 は変異塩基が大多数を占めるホモ接合体（homozygote）を意味する。したがって、後者の 1/1 となっていた、計 5

箇所が概要側の有力な修正すべき箇所ということになる［**W18-4**］。尚、0/0（概要側のみの塩基からなるホモ接合体）は、マップされたリード中の該当塩基部分がミスマッチや indel を含まないことを意味する。つまり、マップされたリード中の該当塩基部分が概要配列と同じである。一般に興味の対象外であるため、通常 0/0 は存在しない。

専門用語に不慣れなヒト向けの説明としては、マップされた Illumina リードがリファレンスの概要配列側と異なる箇所を示しているのが 0/1 や 1/1 である。Illumina リードがマップされた特定の箇所で、リファレンス（REF）の塩基と同じ塩基のものも一定数あるが、異なる（ALT）塩基も一定数ある場合は 0/1 と表現される。このような箇所は、概要側も PacBio のアセンブリ結果として一定の信頼度を持って決定されたものであるため、単純に「変異側の塩基であったリード数が概要側の塩基であったリード数よりも 1 つでも多ければ変異側の塩基を採用」というわけにはいかない。その一方で、概要側と異なる変異側の塩基であったリード数が圧倒的多数派を占めると判断された 1/1 の箇所は、出力リード数およびクオリティスコアが「Illumina >> PacBio」であることも鑑み、次項で述べる可視化ソフトを用いた確認の後、変異側の塩基を採用する決断を下す。

Viewer による変異箇所の確認

リファレンス配列へのマッピング結果をローカル環境で可視化する代表的なソフトウェア（Viewer）としては、Integrative Genomics Viewer [25] や Tablet [26] が挙げられる。ここでの目的は、リファレンス配列（LH_draft.fa）、BAM ファイル（out2.bam）、そして VCF ファイル（out-unique.var.flt.vcf）を読み込み、検出された 22 箇所の変異を目視確認することである。本稿では、Tablet のインストールからファイル読み込みの基本手順を示し［**W19**］、plasmid2 上の 2,569 番目の欠失変異（1/1；反映すべきと判断；W20-1）、plasmid1 上の 66,609 番目の点変異（0/1；あえて反映しなくてもよいと判断；W21-2）、chromosome 上の 1,240,248 番目の挿入変異（1/1；反映すべきと判断；W22-3）、chromosome 上の 2,058,145 番目の点変異（1/1；C を T に変更すべきと判断；W22-4）などを示した。

合計 22 個の変異箇所を目視確認した結果として、我々は計 5 箇所（全て 1/1）の変異を概要配列に対して反映させることにした［**W24-1**］。このうち 4 箇所は、同一塩基が複数個連続したホモポリマー（homopolymer）領域の挿入・欠失変異であった。ホモポリマー部分のエラーについては、他の PacBio データのアセンブリ結果でも議論がなされており [27]、妥当な結果といえよう［**W25**］。

変異の反映

変異の反映を手作業で行う場合は、他の箇所に影響を及ぼさないように気をつけて行う。特に反映の順番には気を付けたほうがよい。以下で示すように、配列ごとに位置を表す数値が大きいものから順番に行えば、位置のずれに悩まされることはない。

① chromosome 上の 2,058,145 番目の C を T に変更
② chromosome 上の 1,455,552 番目の CAAAA を CAAAA に変更
③ chromosome 上の 1,240,248 番目の GCCCC を GCCCCC に変更

① plasmid2 上の 37,805 番目の TAAAAA を TAAAA に変更
② plasmid2 上の 2,569 番目の CAAAAA を CAAAA に変更

連載第 2 回 [28] で紹介した高機能エディタである vi や emacs を用いれば、簡単に目的の箇所（例：chromosome 上の 2,058,145 番目の塩基）にアクセスして作業を行うことができる。基本的な Linux コマンド（head, tail, cut, grep -o など）を駆使し、文字列置換のみで変異の反映を行うことも原理的にはおそらく可能である［**W26**］。しかし現実には、ユニークヒットとなる比較的短い領域を見つけるのは想像以上に難しいため、上記の基本テクニックを併用して目的位置周辺の文字列で複数個所のヒットを許容しつつ検索し、ホスト OS 上で利用可能な高機能エディタ（Windows の場合は EmEditor など）を用いて変異の反映を行う［**W27**］。反映後のファイルサイズ（LH_draft2.fa; 2,400,619 bytes）は、1 塩基の置換、2 塩基分の欠失、そして 2 塩基分の挿入であるため、反映前（LH_draft.fa）と同じである［**W28**］。

おわりに

第 8 回は、PacBio データ（DRR054113）の *de novo* ゲノムアセンブリ結果ファイル（LH_hgap.fa）をスタート地点として、DFAST によるゲノムアノテーション結果と BLAST 実行結果を合わせた、環状染色体の構築を含む概要配列（LH_draft.fa）の作成を行った。次に、得られた概要配列に対して Illumina MiSeq データのマッピングを行い、出力ファイル形式（SAM/BAM と VCF）の解説、Tablet による可視化（変異状況の確認）および変異の反映までを述べた。本稿では、DDBJ Pipeline 上で BWA を実行し、得られた変異の反映を（それほど数が多くなかったこともあり）手作業で行ったが、Pilon [29] のような Illumina データと概要配列を入力として自動でエラー補正

を行う専用のプログラムをまずは実行してみてもいいかもしれない。次回は、複製開始点の同定、（*dnaA* 遺伝子の開始コドンが１番目の塩基となるように）環状染色体の回転、DFAST によるゲノムアノテーションの再実行などを解説する予定である。

謝　辞

本連載の一部は、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（JST-NBDC）、および情報・システム研究機構 国立遺伝学研究所（遺伝研）との共同研究（2012-2088, 2013-2070）の成果によるものです。また、JSPS 科研費 JP25712032, JP15K06919 の助成を受けたものです。DDBJ Pipeline における計算処理は、遺伝研が有するスーパーコンピュータシステムを利用して行われています。

参　考　文　献

1) 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.
2) Flusberg BA1, Webster DR, Lee JH, Travers KJ, Olivares EC, et al. (2010) Direct detection of DNA methylation during single-molecule, real-time sequencing. Nat Methods **7**: 461-465.
3) Chin CS, Alexander DH, Marks P, Klammer AA, Drake J, et al. (2013) Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. Nat Methods **10**: 563-569.
4) 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.
5) 谷澤靖洋，神沼英里，中村保一，遠野雅徳、大崎研、清水謙多郎，門田幸二（2016）次世代シーケンサーデータの解析手法：第７回ロングリードアセンブリ．日本乳酸菌学会誌 **27**: 101-110.
6) 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.
7) Maizel JV Jr, Lenk RP. (1981) Enhanced graphic matrix analysis of nucleic acid and protein sequences. Proc Natl Acad Sci U S A. **78**: 7665-7669.
8) 谷澤靖洋，神沼英里，中村保一，清水謙多郎，門田幸二（2016）次世代シーケンサーデータの解析手法：第６回ゲノムアセンブリ．日本乳酸菌学会誌 **27**: 41-52.
9) Sugawara H, Ohyama A, Mori H, Kurokawa K. (2009) Microbial Genome Annotation Pipeline (MiGAP) for diverse users. 20th Int. Conf. Genome Informatics (Kanagawa, Japan) S-001: 1-2.
10) Aziz RK, Bartels D, Best AA, DeJongh M, Disz T, et al. (2008) The RAST Server: rapid annotations using subsystems technology. BMC Genomics **9**: 75.
11) Overbeek R, Olson R, Pusch GD, Olsen GJ, Davis JJ, et al. (2014) The SEED and the Rapid Annotation of microbial genomes using Subsystems Technology (RAST). Nucleic Acids Res **42**: D206-214.
12) Brettin T, Davis JJ, Disz T, Edwards RA, Gerdes S, et al. (2015) RASTtk: a modular and extensible implementation of the RAST algorithm for building custom annotation pipelines and annotating batches of genomes. Sci Rep **5**: 8365.
13) Tanizawa Y, Fujisawa T, Kaminuma E, Nakamura Y., Arita M. (2016) DFAST and DAGA: Web-based integrated genome annotation tools and resources. Biosci Microbiota Food Health **35**: in press.
14) 孫建強，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５回アセンブル、マッピング、そしてQC．日本乳酸菌学会誌 **26**: 193-201.
15) Seemann T. (2014) Prokka: rapid prokaryotic genome annotation. Bioinformatics **30**: 2068-2069.
16) Mashima J, Kodama Y, Kosuge T, Fujisawa T, Katayama T, et al. (2016) DNA data bank of Japan (DDBJ) progress report. Nucleic Acids Res **44**: D51-D57.
17) Zhang Z, Schwartz S, Wagner L, Miller W. (2000) A greedy algorithm for aligning DNA sequences. J Comput Biol **7**: 203-214.
18) Sonnhammer EL, Durbin R. (1995) A dot-matrix program with dynamic threshold control suited for genomic DNA and protein sequence analysis. Gene **167**: GC1-10.
19) Gollapudi R, Revanna KV, Hemmerich C, Schaack S, Dong Q. (2008) BOV--a web-based BLAST output visualization tool. BMC Genomics **9**: 414.
20) Neumann RS, Kumar S, Haverkamp TH, Shalchian-Tabrizi K. (2014) BLASTGrabber: a bioinformatic tool for visualization, analysis and sequence selection of massive BLAST data. BMC Bioinformatics **15**: 128.
21) Ehrmann MA, Angelov A, Picozzi C, Foschino R, Vogel RF. (2013) The genome of the *Lactobacillus sanfranciscensis* temperate phage EV3. BMC Res Notes **6**: 514.
22) Li H, Durbin R. (2010) Fast and accurate long-read alignment with Burrows-Wheeler transform. Bioinformatics **26**: 589-595.
23) Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, et al. (2009) The Sequence Alignment/Map format and SAMtools. Bioinformatics **25**: 2078-2079.
24) Danecek P, Auton A, Abecasis G, Albers CA, Banks E, et al. (2011) The variant call format and VCFtools. Bioinformatics **27**: 2156-2158.
25) Thorvaldsdóttir H, Robinson JT, Mesirov JP. (2013) Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. Brief Bioinform **14**: 178-192.
26) Milne I, Stephen G, Bayer M, Cock PJ, Pritchard L, et al. (2013) Using Tablet for visual exploration of second-generation sequencing data. Brief Bioinform **14**: 193-202.
27) Sakai H, Naito K, Ogiso-Tanaka E, Takahashi Y, Iseki K, et al. (2015) The power of single molecule real-time sequencing technology in the *de novo* assembly of a eukaryotic genome. Sci Rep **5**: 16780.
28) 孫建強，湯敏，西岡輔，清水謙多郎，門田幸二（2014）次世代シーケンサーデータの解析手法：第２回 GUI 環境からコマンドライン環境へ．日本乳酸菌学会誌 **25**: 166-174.
29) Walker BJ, Abeel T, Shea T, Priest M, Abouelliel A, et al. (2014) Pilon: an integrated tool for comprehensive microbial variant detection and genome assembly improvement. PLoS One **9**: e112963.

Methods for analyzing next-generation sequencing data VIII. Post-assembly analysis

Yasuhiro Tanizawa[1], Eli Kaminuma[1], Yasukazu Nakamura[1], Masanori Tohno[2], Tomoko Terada[3], Kentaro Shimizu[3], and Koji Kadota[3]

[1]Center for Information Biology, National Institute of Genetics.
[2]Institute of Livestock and Grassland Science, National Agriculture and Food Research Organization.
[3]Graduate School of Agricultural and Life Sciences, The University of Tokyo.

Abstract

Genome finishing still remains a laborious work that includes various validation processes requiring both wet and dry knowledge and consideration, although long-read sequencers such as PacBio RSII have largely contributed to lighten the burden. We here introduce a procedure of post-assemble validation in which draft contigs are circularized into complete chromosome or plasmid sequences. We also describe the DFAST annotation web service specialized for lactic acid bacteria, visualization of BLAST alignments, and correction of contigs by mapping Illumina reads. Supplementary materials are available at our web site, http://www.iu.a.u-tokyo.ac.jp/~kadota/r_seq.html#about_book_JSLAB.