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7回 ロングリードアセンブリ

谷澤 靖洋[1,2]、神沼 英里[2*]、中村 保一[2]、遠野 雅徳[3]、
大崎 研[4]、清水 謙多郎[5]、門田 幸二[5*]

[1]東京大学大学院新領域創成科学研究科
[2]国立遺伝学研究所生命情報研究センター
[3]農業・食品産業技術総合研究機構 畜産研究部門
[4]トミーデジタルバイオロジー株式会社
[5]東京大学大学院農学生命科学研究科

完全なゲノム配列の再現を目指す上で最も有効な手段は、ロングリードデータの利用である。第7回は、ロングリードの代表格であるPacBioデータを用いた、乳酸菌ゲノム配列決定について述べる。PacBioデータの概観、PacBioデータ用のアセンブラであるHGAP（Hierarchical Genome Assembly Process）の実行、得られたコンティグの概観・検証（クオリティスコア分布、ドットプロット、およびBLAST）、重複領域の除去（環状化）など、一連の手順を解説する。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、およびDDBJ Pipeline実行結果ファイルなどを効率的に活用してほしい。

Key words：genome assembly, long read, DDBJ Pipeline

PacBioのファイル形式とデータ解析の概要

第7回は、乳酸菌（*Lactobacillus hokkaidonensis* LOOC260^T）ゲノム配列決定論文[1]のPacBioデータを解析する。データ取得に用いられたPacBio RS IIは、SMRT cellsと呼ばれるセル単位でシーケンスを行う[2]［**W1-1**］。リード数を増やしたい場合はセル数を増やせばよく、乳酸菌論文[1]では4セル分のデータを得ている。公共データベース（以下、DB）上では、セルごとに異なるRun IDが割り振られる。4セル分の乳酸菌PacBioデータに対して、4つのDRR ID（DRR054113-054116）が割り振られているのはこのためである。尚、原著論文中に記載されているPacBioデータに相当するDRR024500（第6回の**W2-2**）は、第6回原稿執筆時に問題（後述）があったことが判明し、2016年1月末にDRR054113-054116に差し替えになっている[3]。2016年3月現在、公共DB上にDRR024500は存在しない。しかし、第6回で用いたIllumina MiSeqデータ（DRR024501）を足がかりにすれば、DRP002401［**W2-3**］やDRA002643［**W2-4**］のような上位階層のIDを経由して目的のPacBioデータ（DRR054113-054116）に辿り着くことができる［**W2-5**］。このデータの場合は、セルあたりのリード数が全て163,482個になっており、4セル分を合わせると163,482×4＝653,928リードになる。

PacBioデータの解析は、全体としてIlluminaに代表される他のNGSデータ解析とは異なるアプローチが必要である。PacBioの生データは、bax.h5という拡張子のついた特殊な形式のファイルで出力される。正確には、1セル分につき、3つのbax.h5ファイルと1つのbas.h5ファイ

*To whom correspondence should be addressed.
Phone ： +81-3-5841-2395
Fax ： +81-3-5841-1136
E-mail ： ekaminum@nig.ac.jp (for DDBJ Pipeline)
E-mail ： kadota@bi.a.u-tokyo.ac.jp (for the others)

ルが生成される。そして代表的なPacBio用*de novo*アセンブラ（正確にはアセンブリのためのパイプライン）であるHGAP[4]は、PacBioが提供するSMRT Analysisというソフトウェアの一部として提供されており、bax.h5ファイルのみを入力として受け付ける［**W2-6**］。その一方で、連載第3回[5]の「ファイルの解凍、圧縮、概観」でも述べた通り、公共DBからエンドユーザが取得可能な形式は、sraファイルとFASTQファイルの2種類のみである（2016年4月23日現在）。つまり現状では、HGAPプログラム実行に必要なbax.h5ファイルを公共DBから取得することはできない。

本稿では、1セル分（DRR054113）のbax.h5ファイルを公開しているが、ファイルサイズは約2.4GBに達する［**W2-7**］。4セル分全てだと10GB弱になるため、実質的にノートPCレベルでは解析不可能である。また、bax.h5ファイルは、テキストではなくバイナリ形式である。このため、Linuxコマンドの組み合わせで指定したリード数からなるサブセットを抽出し（第6回のW3）、再びbax.h5ファイルとして保存するような作業はおそらく誰も行っていない。HGAPは、DDBJ Pipeline[6]でも利用可能である。結論を先に述べると、前処理などを一切行わない生のbax.h5ファイルをDDBJ Pipelineにアップロードし、DDBJ Pipeline上でHGAPを実行するのが最も簡便である。理由は、①PacBioデータの*de novo*アセンブリには256GB程度のメモリを搭載したLinuxマシンが必要であること、②SMRT Analysisのインストールの難易度が高いこと、③解析プロトコルのステップを理解した上で自分でプログラムを動作させるのが大変であることなどが挙げられる。

PacBioデータの解析自体は、上述の推奨手順（DDBJ Pipelineにbax.h5ファイルをアップロードして実行）に従う限り、極めて簡単である。世間一般の評価同様、少なくとも乳酸菌を含むバクテリアのゲノム配列決定目的では、我々の経験上も申し分のない結果が得られる点を強調しておきたい。FastQC[7]によるクオリティチェックも基本的に不必要ではあるが、PacBioの特徴やHGAPアルゴリズム（計算手順）の理解にも役立つため、周辺情報とともに以下で述べる。

PacBioデータの概観 1

公共DBの1つであるDDBJ SRA（以下、DRA）[8]より、DRR054113のFASTQファイルをダウンロードし［**W3-1**］、FastQC（ver. 0.11.4）を実行する［**W3-2**］。リード数、配列長分布、クオリティスコア分布のいずれにおいても、これまで解説してきたIllumina HiSeq 2000のデータ（SRR616268; 連載第4回のW8）やIllumina MiSeqのデータ（DRR024501; 第6回のW4）と全般的に異なることがわかる［**W3-3**］。約2.4MBというファイルサイズからも想像できるように、リード数が非常に少ない（915個）。配列長は一定ではなくばらついている（923–8,076 bp）。特に読み始め（最初の1–3番目まで）のクオリティスコアの上下動が激しい。また、配列長分布を眺めてみると、全リードの約90%は配列長が4,000 bp以下であり、5,000 bp以上の長いリードは十数個程度しかないことが読み取れる［**W3-4**］。

実際の数値情報を調べたい場合には、例えば**（Rで）塩基配列解析**の項目「前処理|クオリティチェック|配列長分布を調べる」を参考にすればよい。bzip2圧縮ファイルからgzip圧縮ファイルを作成しているのは、FASTQファイルを読み込む際に用いるRパッケージShortRead[9]のreadFastq関数がbzip2圧縮ファイルに対応していないためである［**W3-5**］。ここでは、配列長ごとの出現頻度を計算するRスクリプトを実行し［**W3-6**］、5,000 bp以上のリード数が実際には16個だったことを確認した［**W3-7**］。

次に、DRAで表示されているリード数（163,482個）とFastQC実行結果として得られたリード数（915個）の極端な違いについて解説する。これは、大まかにはDRAで提供されているsraファイルとFASTQファイルのリード数の違いに起因する。異なるリード数に関する議論自体は、以前にIllumina HiSeqデータでも行ったが、ここまでの極端な違いではなかったことを思い出してほしい（第3回の**W24**）。つまり、sraファイルが135,073,834リード、FASTQファイルが134,755,996リードであり、1%以下の違いであった。DRA提供のPacBioのsraファイルは、生データファイル（3つのbax.h5ファイルと1つのbas.h5ファイル）を基に作成している。そして、（PacBioデータに限らず）DRA提供のFASTQファイルは、sraファイルを入力として作成している。全体として、PacBioの場合は「bax.h5 → sra → FASTQ」という不可逆的な流れで、sraおよびFASTQファイルが作成される。

DRR054113のsraファイルは約1.4GBであり、FASTQファイルの約2.3MBとは大きく異なる［**W3-8**］。これはbzip2圧縮ファイルのほうがsraファイルに比べてサイズが小さい[5]という一般的な特徴よりも、リード数の違いによる影響のほうが圧倒的に大きい。DRAで提供されているFASTQファイルのリード数（915個）が少ないのは、FASTQファイル作成時に用いるfastq-dumpプログラム実行時に、フィルタリングおよびトリミングのオプションをつけているためである［**W3-9**］。前述のDRR024500で登録されていたデータの問題点も、ファイル変換時のオプション指定に起因するものであった。フィルタリング・トリミングされていない状態のPacBioデータの特徴を眺めるべく、①fastq-dumpを含むSRA Toolkitのインストールおよび実行、②オリジナル（163,482個）に近いリード情報を含むFASTQファイルを入力としてFastQCを実行する。

SRA Toolkit（ver. 2.5.7）のインストールと利用

sra ファイルから FASTQ ファイルを作成する fastq-dump プログラムを利用すべく、NCBI が提供する SRA Toolkit のインストールを行う。本家サイトでは、OS ごとの tar.gz ファイルを wget でダウンロードし、tar コマンドで解凍するやり方が示されているが［**W4-1**］、ここでは第 4 回で紹介した apt-get コマンドを用いるやり方を示す。「sudo apt-get install *ソフトウェア名*」が基本的なやり方ではあるものの、まずはここで指定するソフトウェア名を把握しなければならない。基本戦略は apt-cache コマンドの利用である。「apt-cache search *キーワード*」として、ソフトウェア名の一部を構成するであろうと思われる部分文字列をキーワードとして指定すればよい。ここでは、「apt-cache -n search *SRA*」として、SRA というキーワードを含むソフトウェア名のみ（-n）リストアップさせ、目的のソフトウェア名が sra-toolkit であることを見出している［**W4-2**］。「sudo apt-get install *sra-toolkit*」でインストールを行うメリットは、パスを通すところまで行ってくれる点にある［**W4-4**］。実用上は、本家サイトのプログラム名を参考にしながら apt-cache で検索し、可能な限り apt-get を利用してインストールを行うのが効率的であろう。

DRA で表示されているリード数（163,482 個）に近い情報を含む FASTQ ファイルを得るためには、fastq-dump コマンドをデフォルトで実行すればよい［**W5-6**］。他の有用なオプションとしては、--gzip や --bzip2 が挙げられる［**W5-7**］。デフォルトの出力は非圧縮 FASTQ ファイルであるが、これらのオプションをつけることで gzip 圧縮や bzip2 圧縮ファイルとして出力させることができる。FastQC など、多くのプログラムは圧縮ファイルをそのまま入力として与えることができるため、ファイルサイズ削減の観点からも圧縮ファイルとして出力させておくとよい。残念ながら、今回の fastq-dump 実行結果は 163,380 リードであり、DRA で表示されている 163,482 リードと同じ結果は得られなかった［**W5-6; W2-5**］。また、DRA の FAQ に記載されている FASTQ 作成時のオプション通りに実行しても、DRA から直接ダウンロードした FASTQ ファイルのリード数（915 個）にはならなかった［**W5-2**］。これは用いたプログラムのバージョンの違いに起因するのかもしれないが、あくまでも sra → FASTQ ファイル作成時の話であり、PacBio データの推奨解析手順（DDBJ Pipeline に bax.h5 ファイルをアップロードして実行）とは無関係である。

PacBio データの概観 2

DRA で表示されているリード数（163,482 個）に近い 163,380 リードからなる FASTQ ファイルを得た主な理由は、このファイルを入力として FastQC を実行し、PacBio 生データの特徴をより正確に把握するためである。この FASTQ ファイル（163,380 リード）のクオリティスコアは 10 弱である［**W6-3**］。PacBio RS II の塩基配列決定精度は約 87%、つまりエラー率は 13%（$p_{err}=0.13$）である[10]。クオリティスコアは $-10\times\log_{10}(p_{err})$ として計算されるため[11]、エラー率 13% のときのスコアは、$-10\times\log_{10}(0.13)=8.86$ となる［**W6-4**］。DRA から直接ダウンロードした FASTQ ファイル（915 リード）の FastQC 実行結果は、PacBio の公式スペック以上のクオリティスコア（> 20）となっており、一言で言えばきれいすぎる［**W3-3**］。著者らは、公称値に近い 10 弱というスコアを見るほうがむしろ心穏やかである。

配列長は 116 ～ 28,874 bp の範囲となっており［**W6-3**］、全リード（163,380 個）の半数以上が 1,000 bp 未満、そしてごく少数のリードが 10,000 bp 以上である［**W6-5**］。具体的な数値情報は R で確認することができ、85,134 リード（52.1%）が 1,000 bp 未満であり、5,985 リード（3.66%）が 10,000 bp より長いという結果が得られる［**W6-6**］。しかしこれは本当の意味での生リード情報であり、アダプター配列を含んでいる。このこと自体は他の NGS テクノロジーと同じではあるが、リードの末端部分にアダプターを含む Illumina とは異なり、PacBio は SMRTbell と呼ばれる鉄アレイのような形のライブラリをテンプレートとしてシークエンスを行うため、リード内部にもアダプターを含む［**W6-7**］。このアダプター配列をトリムした後のリード（adapter-trimmed reads）はサブリード（subreads）と呼ばれる。

PacBio RS II は、セル（SMRT cell）あたり 150,000 個のウェルを搭載している。ウェルとは、1 分子（1 つの DNA 断片）が入る程度の小さい穴のようなものだと解釈すればよく、Zero-Mode Waveguides（ZMWs）とも呼ばれる。「150,000 ZMWs in PacBio RS II」などと表現されるのはこのためである。各ウェル上に 1 分子のテンプレートが入ってシークエンスされるため、理論上ウェル数の分だけしかリード数は得られない[12]。つまり、得られるリード数の理論値の上限は「150,000 リード（正確には 150,292 リード）/ セル」である。DRA で表示されているリード数（163,482 個；**W2-5**）や、fastq-dump 実行結果として得られたリード数（163,380 個；**W6-3**）は、明らかに理論値を超えている。この理由は、PacBio RS II のセルには、解析サンプルのリード情報を得るためのウェル以外に、コントロール用のウェルや、レーザーが正しく ZMW に照射されているかを確認するためのウェルが搭載されているためである。マイクロアレイにも、ポジティブコントロールやネガティブコントロール用のスポット（プローブ）が搭載されている。また、Illumina でシークエンスを行う際にも、PhiX というコントロール用サンプルが含められる。一般には、これらのコントロールデータの詳細情報までは記載されないと考えれば納得できるだろう。

乳酸菌ゲノム配列決定論文[1]では、クオリティスコア（Read Quality＝80）、および配列長（500 bp）でリードのフィルタリングを行った後、そのサブリードを入力としてHGAPによる*de novo*アセンブリが行われた。4セル分のPacBioデータの場合、フィルタリング前の（コントロールを除く）リード数は150,292×4＝601,168個となる。上記前処理（フィルタリングおよびアダプター除去）後に得られたサブリード数は163,376個であり、配列数としては約1/4に激減している［**W6-8**］。これは乳酸菌データに限った話ではなく、概ね一般的な減少度合いである。

DDBJ Pipeline（HGAPの実行）

1セル分（DRR054113）の3つのbax.h5ファイルをDDBJ Pipelineにアップロードして、HGAPを実行する一連の手順を示す。ファイルのアップロード自体は、第6回のW5で作成したpaired-end Illumina MiSeqデータ（DRR024501）のFTP経由での手順（第6回の**W14**）と同じである［**W7-2**］。登録（registration）作業は、bax.h5ファイルごとに独立して行う必要がある（2016年3月28日現在；**W7-9**）。非常に面倒ではあるが、近い将来一括登録が可能になるものと期待される。

HGAP法[4]のアセンブリ手順は、大まかには次のような流れで行われる：①最も長いサブリード群をシード（リファレンス配列）として用い、②シードに短いサブリードをマップし、③エラー補正を行ったpreassembled readsを作成し、④それらを用いてアセンブリの本番を行う。HGAP実行時に指定するパラメータは2つある。1つは推定ゲノムサイズ（GenomeSize）、そしてもう1つはシード（seeds）として用いるリードの最短配列長（Minimum Seed Length）である［**W8-2**］。本稿では、PacBioデータのみのが手元にある場合を想定して乳酸菌の平均的なゲノムサイズ（2.5MB; 2,500,000 bp）を与えたが、Illumina MiSeqデータから得られた推定ゲノムサイズ（約2.4MB; 2,400,000 bp）を与えてみてもいいだろう[3]［**W8-3**］。PacBioデータは、ランダムな位置でシークエンスエラーが起こる。そのため、②で短いリードをマップして、ポジションごとに出現する塩基の多数決ルールを適用することでエラー補正ができる。Xに相当する推定ゲノムサイズを与えることで、coverageの計算を行うことができる。

2つめのパラメータであるMinimum Seed Lengthは、基本的にデフォルトの6,000 bp、およびAutomatic Estimationでよい。PacBioの真骨頂は、リピート配列を超えうる長さのリードを得られる点にある。それゆえ、①で選択するシードは、安定的にアセンブルできる25Xを超える範囲で、できるだけ長いサブリードに限定するほうがよい。それを自動的に算出するのがAutomatic Estimationである。6,000 bpという値は、25Xという条件を満たす最短サブリード長が6,000 bp未満だった場合には6,000 bp以上の長さのサブリードをシードとして用いるという意味である。このデータの場合は、4セル分合わせて平均4,000 bpのサブリードが163,376個ある[1]。トータルの塩基数を乳酸菌の平均ゲノムサイズで割ると、4,000 bp×163,376個 / 2,500,000 bp＝261.4 Xとなり、25Xを大きく超える。本稿では1セル分しか用いなかったが、それでも261.4 / 4＝65.4Xのcoverageとなる。リード全体で65.4Xあれば、seedとなりうる十分に長いリードでも25X分確保でき、結果として4セルの場合と遜色のないコンティグ数（4 contigs）が得られたということなのであろう。HGAP法の実行には、PacBioデータの特徴、アルゴリズムの大まかな理解に加えて、解析サンプルの推定ゲノムサイズおよびcoverageを把握しておいたほうが結果の解釈も容易になるだろう。

尚、乳酸菌原著論文当時は、P4-C2試薬を用いて平均4,000 bpのサブリードが得られたが、現在の一般的な手順（P6-C4試薬、20,000 bpライブラリ、6時間movie）で行うと、8,000－10,000 bp程度の長さのサブリードが得られる。HGAPの推奨coverage（リード全体のcoverage）は、データにも依存するが概ね60-100Xである[13-14]。

HGAPの計算時間は、DDBJ Pipelineの実体である遺伝研スパコンの混み具合にもよるが、概ね半日～1日である［**W9-1**］。4セル分で実行した原著論文[1]の解析時は、HGAP（Protocol2; ver. 2.0.0）で行い、7 contigsを得た。今回1セル分のみで実行したHGAP（Protocol3; ver. 2.2.0）実行結果は、4 contigsであった［**W9-2**］。この違いがプログラムのバージョンによるものかセル数によるものかまでは調査していないが、Platanus[15]実行結果の52配列（300 bp未満の配列を除く；第6回の**W20-7**）に比べて、少なくとも配列数の点ではよい結果が得られている。尚、このデータの正解は、全て環状で3 contigs（1 chromosome＋2 plasmids）、2,400,586 bp（約2.4MB）である[1]。

HGAP実行結果の概観と前処理

DDBJ PipelineでHGAP（Protocol3; ver. 2.2.0）を実行した結果（result.zip）には、計4つのファイルが含まれる［**W9-4**］。エンドユーザが欲しい最終結果ファイルは、polished_assembly.fastaである。Linuxの基本コマンド［**W9-5**］やR［**W9-6**］を駆使して、このファイルの全体像を大まかに把握し、ある程度予想を立てる。具体的には、最も長い2,289,497 bpのコンティグは、乳酸菌の平均ゲノムサイズ（約2.5MB）に近いことから、染色体（chromosome）だろうと予想した。それ以外の3つのコンティグ（86,892 bp, 45,853 bp, and 11,372 bp）は、染色体の一部、プラスミド、ミスアセンブルのいずれかであろう。もしコンティグが環状であれば、プラスミドである可能性が高い。そこでまずは、コンティグごとに環状かどうかのチェックを行う。

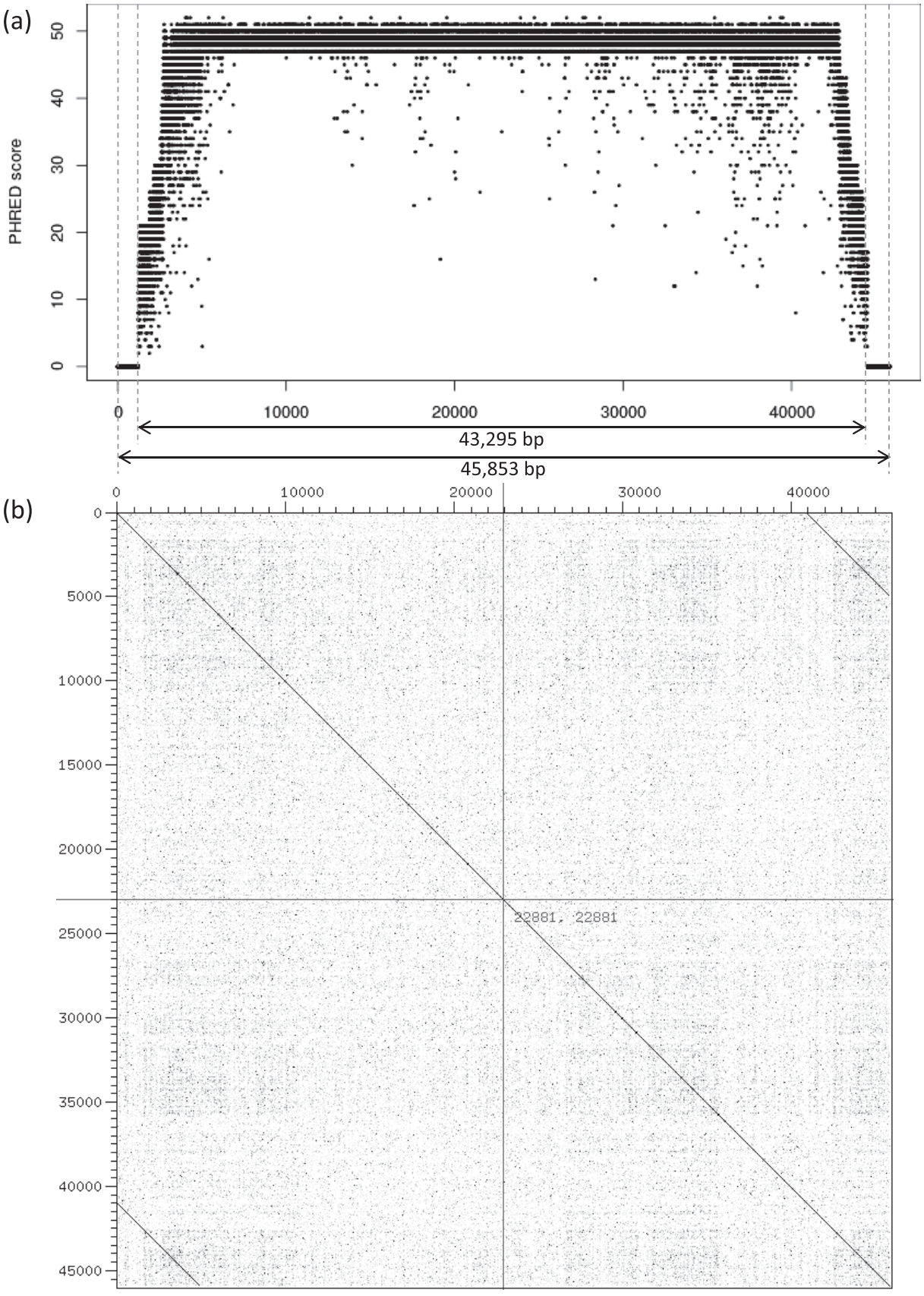

図 1. 45,853 bp からなるトリム前の sequence3 の特徴
（a）クオリティスコア分布。横軸：塩基ポジション、縦軸：PHRED スコア（高いほどよい）。最初の 1,223 bp までと最後の 1,335 bp がスコア 0。（b）ドットプロット。最初と最後の約 5,000 bp が重複している（類似している）ことが分かる。

コンティグ数が少なくコンティグごとに作業を行う場合は、multi-FASTA ファイルを分割し、コンティグ数分だけ single-FASTA ファイルを作成しておいたほうが効率的な場合もある。ここでは、ファイル分割手段として R を用いるやり方、および自作プログラム（fastaLengthFilter.py；第 6 回の W12）と Linux コマンドを組み合わせたやり方を示した［**W10**］。どちらが正解ということはなく、自分の感性に合う手段を用いればよい。

一般に、HGAP アセンブリ結果として得られるコンティグの末端部分のクオリティは、中央部分に比べて低い。HGAP 実行結果には、FASTA ファイル（polished_assembly.fasta）だけでなく FASTQ ファイル（polished_assembly.fastq）も含まれる。ここでは、FASTQ ファイルを入力として、コンティグごとのクオリティスコア分布を眺めておく。例えば 2 番目に短いコンティグ（45,853 bp; sequence3.fq）のスコア分布の場合（**図 1a; W11-9**）、最初の 1,223 bp までと最後の 1,335 bp が連続してスコア 0 になっており、中央の 1,224 bp から 44,518 bp までの計 43,295 bp（＝44,518－1,224＋1）がスコア 1 以上になっていると判断できる［W11-12］。これは、コンティグが環状であることを確認したあとに、どの部分をトリミングするかについての合理的な指針を与えるものでもある。

配列のドットプロット

ドットプロット（dot plot）は、比較したい 2 つの配列の類似度を視覚的に評価するために古くから用いられている描画手段である [16]。基本的には、比較する 2 つの配列を x 軸 y 軸にそれぞれ並べて、同一塩基部分をハイライトさせるだけである。ここでは、単純な塩基配列を用いたドットプロットの解釈の基礎を述べ、実際の環状コンティグ（またはゲノム）の実例を sequence3 で示す。Bio-Linux には dotter [17] というドットプロット用プログラムがプレインストールされているが［**W12-1**］、これは環状チェックの本番で用いる。まず seqinr [18] という R パッケージ中の dotPlot 関数を用いて、ドットプロットの基本形を示す。環状かどうかのチェックの場合は、同一の配列を並べて比較する。同一配列間のドットプロットの主な特徴は、必ず対角線上に位置する塩基が同じになるという点である［**W12-6**］。環状の場合は、コンティグの両末端の配列がほぼ同じになる。これは、ドットプロット上で対角線と平行の位置に、重複塩基数分だけの長さの一致部分がハイライトされることで判別できる［**W12-8**］。

例えば、ランダムな塩基配列 “ACTCGTCAGA” が真の環状ゲノムだと仮定すると、*de novo* アセンブリによって得られるコンティグは “ACTCGTCAGAACTC” のような感じとなる。この場合、灰色の 4 塩基分の重複がドットプロット上でハイライトされ（**図 2a**）、重複除去（塩基のトリミング）が環状化作業に相当する。末端塩基のトリミングの選択肢として、4 塩基重複の場合は計 5 通り存在するが、アスタリスク（＊）のついた両末端から概ね同数の塩基をトリミングするのが推奨である（**図 2b**）。その理由は、**図 1a** で示すようなコンティグ末端部分のクオリティスコア分布を眺めれば納得できるであろう。この単純な例の場合は、結論としてはどの選択肢を採用しても同じ結果になるものの、数千塩基におよぶ実際のプラスミドコンティグの重複部分が完全一致となる可能性はほぼゼロという現実的な問題に当てはめるとよい。

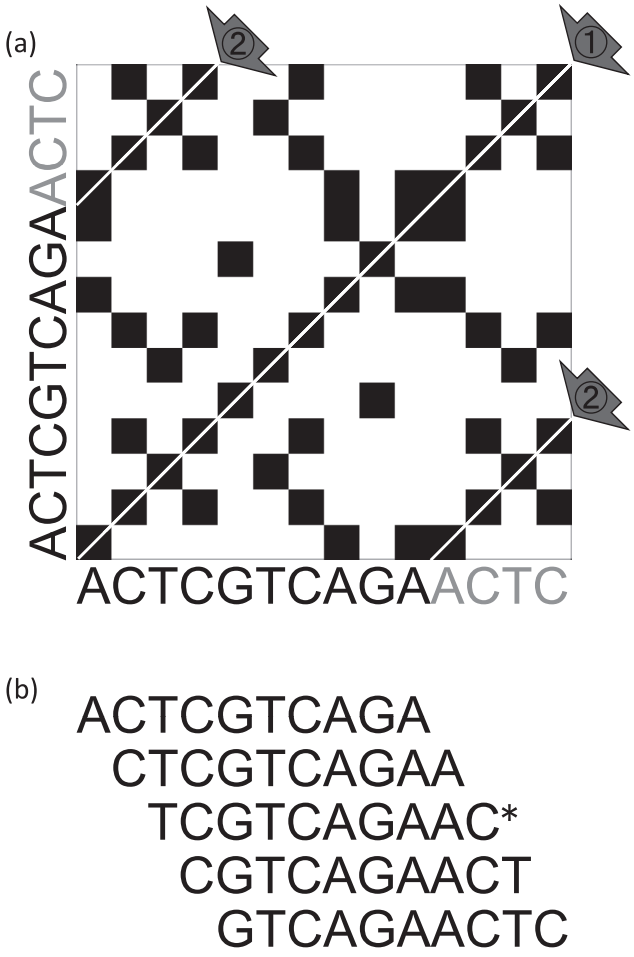

図 2.（a）仮想環状コンティグ配列 ACTCGTCAGAACTC のドットプロット。灰色で示した最後の 4 塩基が最初の 4 塩基と同じで、重複領域に相当する。（b）環状なので重複除去手段は計 5 通り。＊のついた前後 2 塩基づつトリムするやり方が推奨。

数万～数百万塩基におよぶ配列のドットプロットの作成は、seqinr パッケージの dotPlot 関数では現実的に難しいため［**W14-1**］、Bio-Linux にプレインストールされている dotter プログラムを用いる［**W12-1**］。HGAP アセンブリ結果として得られた計 4 コンティグのうち、2 番目に短いコンティグ（45,853 bp; sequence3.fa）が環状かどうかを確認したい場合は、「dotter sequence3.fa sequence3.fa」と打てばよい［**W14-2**］。ドットプロット全体を眺めると、対角線と平行の直線が配列末端部分に現れていることがわ

かる（**図 1b**）。大まかには、最初と最後の約 5,000 bp が重複しており、この重複除去がアセンブル後の後処理に相当する［**W14-4**］。もちろんより正確な一致領域を調べる必要はあり、著者らはこの目的のために BLAST[19] を用いる。

配列相同性検索（BLAST）

BLAST（Basic local alignment search tool）は、局所的なアラインメント（local alignment）を行うプログラムである。問い合わせ（query）配列と呼ばれる手元の塩基配列またはアミノ酸配列の特徴や性質を把握する目的で、GenBank などの公共塩基配列 DB に対して検索を行ったことがある人は多いだろう。プログラム内部では、query 配列と似たものがあるかどうかを DB 配列に対して検索し、指定した閾値を満たす query 側と DB 側の部分配列のアラインメント結果が出力される。ここでは、query 側および DB 側の配列を sequence3.fa として、（Bio-Linux にプレインストールされている）BLAST を実行する[20]。同一配列間の比較なので、トップヒット（top hit）は sequence3 の全長配列間で 100% 一致のアラインメントとなる。今詳細に調べたいアラインメント結果は、セカンドヒットの「最初と最後の約 5,000 bp の重複配列」である。これらの予想は、**図 1b** のドットプロットを事前に眺めておけば立てられる。ドットプロットは、コンティグの全体像の理解に役立つだけでなく、BLAST 実行結果の理解の助けにもなる。補足情報的な位置づけではあるが、なるべく併用するといいだろう。

BLAST の実行は、① DB 側配列の BLAST 用 DB への変換、および② query 配列の相同性検索の 2 ステップで完了する［**W15**］。具体的には、①では makeblastdb コマンドで DB 側配列である sequence3.fa を入力として、BLAST 用 DB（インデックスファイル）を作成する［**W15-1**］。②では、query 側と DB 側の配列の種類（塩基配列またはアミノ酸配列）や目的に応じて、以下に示す 5 つのプログラムを使い分ける：

blastn：query 側、DB 側がともに塩基配列。
blastp：query 側、DB 側がともにアミノ酸配列。
blastx：query 側は塩基配列、DB 側はアミノ酸配列。query 配列をアミノ酸配列に翻訳して検索。
tblastn：query 側はアミノ酸配列、DB 側は塩基配列。DB 配列をアミノ酸配列に翻訳して検索。
tblastx：query 側、DB 側がともに塩基配列。両方をアミノ酸配列に翻訳して検索。

今回は query 側と DB 側がともに塩基配列のため、最も一般的な blastn を利用する。計算自体はほぼ一瞬で終了し、指定した名前の BLAST 結果ファイル（sequence3_blast.txt）が作成される［**W15-2**］。BLAST 結果を眺めると、45,853 bp からなる sequence3.fa の［1, 4884 bp］と［40967, 45853 bp］の範囲が 99% 一致となっていることがわかる［**W15-9**］。重複除去の選択肢は、以下に示すように概ね 4,884 通り存在する：

・［4885, 45853 bp］を残す
　（最初の 4884 bp、および最後の 0 bp 分をトリム）
・［4884, 45852 bp］を残す
　（最初の 4883 bp、および最後の 1 bp 分をトリム）
・［4883, 45851 bp］を残す
　（最初の 4882 bp、および最後の 2 bp 分をトリム）
…
・［3, 40968 bp］を残す
　（最初の 2 bp、および最後の 4885 bp 分をトリム）
・［2, 40967 bp］を残す
　（最初の 1 bp、および最後の 4886 bp 分をトリム）
・［1, 40966 bp］を残す
　（最初の 0 bp、および最後の 4887 bp 分をトリム）

概ねとした理由は、［1, 4884 bp］と［40967, 45853 bp］の範囲のアラインメント結果には Gap が含まれており、この Gap の取り扱いに関する不確定要素があるためである。範囲と塩基数の関係や計算法を含めて混乱しがちなところではあるが、始端側の［1, 4884 bp］の塩基数は（4884 − 1 + 1）= 4884 bp と計算し、終端側の［40967, 45853 bp］の塩基数は（45853 − 40967 + 1）= 4887 bp と計算する。アラインメント結果の始端側と終端側の塩基数が異なるのは、結論としては Gap 数の違いに起因するためであり、気にしなくてよい。実際に我々が行う重複除去は、概ね両側から同数程度の塩基のトリムである。その理由は、アセンブリ結果として得られるコンティグの末端部分のクオリティは、中央部分に比べて低いためである（**図 1a; W11-9**）。重複塩基数が 4900 bp 程度であることを踏まえ、BLAST アラインメント結果のセカンドまたはサードヒットの 2400-2500 番目付近を眺め、Gap やミスマッチのない領域でトリミング領域を決定する［**W16-1**］。ここでは、tail と cut コマンドを組み合わせて［2450, 43422 bp］の範囲を抽出し、（43422 − 2450 + 1）= 40,973 bp の長さの環状コンティグ（sequence3_trimmed.fa）として出力した［**W16-3**］。

確　認

動作確認（この場合、トリミングが正しく行えたかどうかの確認）は、複数の手段で行うことを推奨する。BLAST のアラインメント結果からトリム後の塩基配列の最初と最後の部分がわかるので［**W16-2**］、トリム後の FASTA ファイル（sequence3_trimmed.fa）の最初と最後を眺めておくのは基本であろう［**W16-4**］。トリム後の

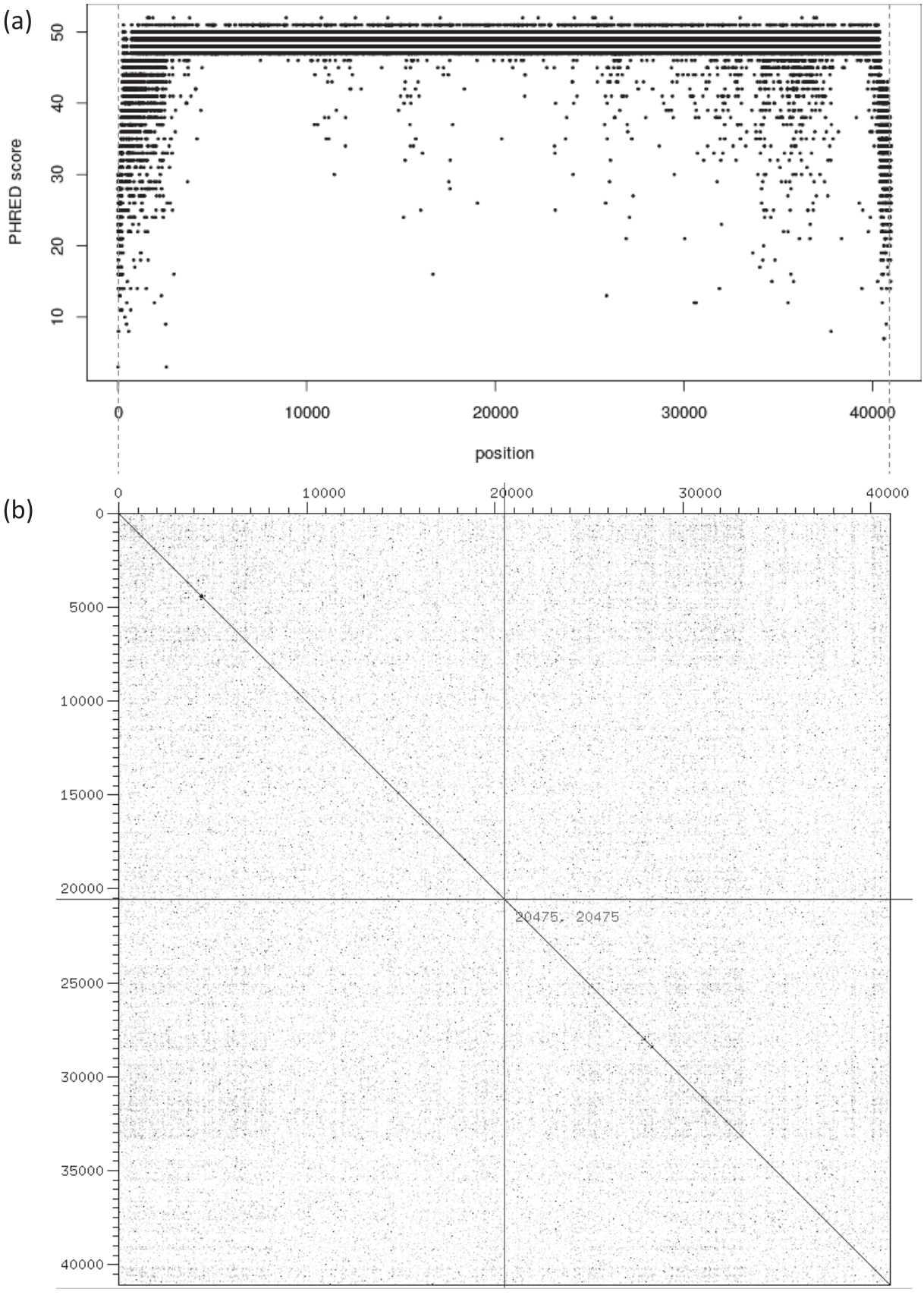

図 3. 40,973 bp からなるトリム後の sequence3 の特徴
（a）クオリティスコア分布。**図 1a** で見られた両端の低クオリティ領域がなくなっていることがわかる。(b)ドットプロット。**図 1b** で見られた右上と左下の対角線（配列類似）領域がなくなっていることがわかる。

FASTQ ファイル（sequence3_trimmed.fq）を作成して、トリム後のクオリティスコア分布（**図 3a; W17-2**）をトリム前のもの（**図 1a; W11-9**）と比較しておくのも重要であろう。FASTQ 形式ファイル（sequence3.fq）については、塩基配列情報からなる 2 行目とクオリティ情報からなる 4 行目についてのみ「cut -v 2450-43422」を実行すればよい［**W17-1**］。これは、FASTQ 形式の中身を理解していれば、FASTQ ファイルについても任意の領域を手持ちのスキルのみで抽出可能という例でもある。45,853 bp のコンティグから、最初の 2,449 bp および最後の（45,853 − 43,423 + 1）= 2,431 bp をトリムしているので、両側ともにほぼ同数の塩基をトリムしていることになる。この視点でトリム前後の分布を比較することで、クオリティの低い両末端をうまくトリムできていると判断することができる。トリム後の配列でドットプロットを作成し、重複領域が消えていることまで確認しておけば間違いないだろう［**W17-3**］。

本稿ではドットプロットの作成や BLAST の実行を Bio-Linux 上で行ったが、例えば NCBI の BLAST ウェブサービスを利用してもよい［**W18**］。今回重複除去して得られた 40,973 bp の長さの環状コンティグ（sequence3_trimmed.fa; **W16-3**）は、原著論文中では 40,971 bp のプラスミド（accession number: AP014682）に相当する。2 bp の違いは Gap 由来だと思われる。86,892 bp からなる sequence2 についても、sequence3 と同様の枠組みで重複除去を行えるので、ぜひチャレンジしてみてほしい。この sequence2 について重複除去した結果は、原著論文中では 81,630 bp のプラスミド（accession number: AP014681）に相当する。最も長い染色体候補の sequence1（2,289,497 bp）と最も短い sequence4（11,372 bp）の関係性については、アノテーション（遺伝子領域および機能予測）結果を含めた解釈が必要になるため、乳酸菌に特化したアノテーションツール DFAST（http://dfast.nig.ac.jp）を含め次回以降述べる予定である。尚、Illumina MiSeq データ（DRR024501）も利用可能なこの乳酸菌データの場合は、マッピング（マップする側が MiSeq データ、マップされる側が重複除去後の HGAP アセンブリ結果）を行うことで、より詳細なアセンブリ結果の評価を行うこともできる。

おわりに

第 7 回は、ロングリードの代表格である PacBio データの特徴を述べ、DDBJ Pipeline 上での HGAP アセンブリプログラムの実行、および後処理の一部を示した。現状では、公共 DB で PacBio の生データ（bax/bas.h5 形式ファイル）が提供されていない。また、多くの PacBio 用の解析プログラムは、FASTQ ファイルではなく生データファイルを入力とするため、一般の研究者が公共 PacBio データにアクセスして解析しづらい状況にあることは間違いない。将来的には、公共 DB からの PacBio 生データの提供や、FASTQ ファイルを入力とした PacBio 用プログラムの提供が本格化するかもしれない。本稿のデータ取得に用いた NGS 機器は、セルあたり 150,000 ZMWs の PacBio RS II System であったが、2015 年 10 月に 1,000,000 ZMWs の Sequel System がリリースされている。Sequel System が出力する生データは、BAM 形式ファイルである。SMRT Analysis 3.1 では、BAM 形式ファイルを取り扱うとともに、bax/bas.h5 → BAM コンバータも提供されている。アセンブラについては、今回は DDBJ Pipeline 上で実行可能な HGAP 法を利用したが、MHAP 法[21] や Falcon など新しいプログラムも続々と開発されている。環状化（重複除去）については、今回は概念の説明を含めて手作業で行ったが、Circlator[22] など自動的に行うプログラムも提案されている。

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（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) 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.
4) 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.
5) 孫 建強，三浦 文，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 3 回 Linux 環境構築から NGS データ取得まで．日本乳酸菌学会誌 **26**: 32-41.
6) 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.
7) Andrews S. (2015) FastQC a quality control tool for high throughput sequence data, http://www.bioinformatics.

babraham.ac.uk/projects/fastqc/
8) 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**: D51–57.
9) Morgan M, Anders S, Lawrence M, Aboyoun P, Pagès H, et al. (2009) ShortRead: a bioconductor package for input, quality assessment and exploration of high-throughput sequence data. **25**: 2607–2608.
10) Rhoads A, Au KF. (2015) PacBio Sequencing and Its Applications. Genomics Proteomics Bioinformatics **13**: 278–289.
11) 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.
12) パックマンの挑戦ーPacBio シークエンサーー (http://pacbiobrothers.blogspot.jp/2013/05/hgap.html)
13) Utturkar SM, Klingeman DM, Land ML, Schadt CW, Doktycz MJ, et al. (2014) Evaluation and validation of de novo and hybrid assembly techniques to derive high-quality genome sequences. Bioinformatics **30**: 2709–2716.
14) Liao YC, Lin SH, Lin HH. (2015) Completing bacterial genome assemblies: strategy and performance comparisons. Sci Rep **5**: 8747.
15) 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.
16) 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.
17) 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.
18) Charif D, Thioulouse J, Lobry JR, Perrière G. (2005) Online synonymous codon usage analyses with the ade4 and seqinR packages. Bioinformatics **21**: 545–547.
19) Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. (1990) Basic local alignment search tool. J Mol Biol **215**: 403–410.
20) Zhang Z, Schwartz S, Wagner L, Miller W. (2000) A greedy algorithm for aligning DNA sequences. J Comput Biol. **7**: 203–214.
21) Berlin K, Koren S, Chin CS, Drake JP, Landolin JM, et al. (2015) Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. Nat Biotechnol. **33**: 623–630.
22) Hunt M, Silva ND, Otto TD, Parkhill J, Keane JA, et all. (2015) Circlator: automated circularization of genome assemblies using long sequencing reads. Genome Biol. **16**: 294.

Methods for analyzing next-generation sequencing data VII. long-read assembly

Yasuhiro Tanizawa[1, 2], Eli Kaminuma[2], Yasukazu Nakamura[2], Masanori Tohno[3], Ken Osaki[4], Kentaro Shimizu[5], and Koji Kadota[5]

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

Abstract

Long-read sequencing represented by Pacific Biosciences' single-molecule real-time (SMRT) technology has been widely used for microbial genomes. We overview an analysis procedure of *Lactobacillus hokkaidonensis* LOOC260^T genome using the so-called "PacBio" data. We describe (i) the characteristics of PacBio data, (ii) genome assembly using the HGAP program provided in the web-based NGS analysis tool called "DDBJ Pipeline", (iii) the survey of assembled contigs such as quality scores and circularization. We also describe a typical strategy of finishing the circular contig (plasmid).