シリーズ「大量データと知見の架け橋」

緒言・編集担当 佐藤行人（東北大学 東北メディカル・メガバンク機構）

本シリーズは、進化学研究に有用と思われるゲノム／オミクス情報の解析ツールやソフトウェアを紹介することを目指し、前々号（Vol .15, No.2 July, 2014）より開始しました。第2回目となる今号では、次世代シークエンサーデータのクオリティ評価とデータクリーニングを行うためのソフトウェアSUGARを紹介します。SUGARの原著論文を発表した私、佐藤行人と長﨑正朗が執筆しました。

データの精査は、あらゆる研究で基礎的かつ重要なステップです。とりわけ次世代シークエンサーの活用では、生み出される塩基配列データが大規模（例えば全ゲノムレベル）になるため、コンピュータと適切なソフトウェアの力を借りて、生データのクオリティ評価を行うことが必須となります。本稿はとくにIllumina社のプラットフォーム（MiSeqなど）に焦点を絞り、クオリティの低いデータが発生する機序とその対処方法、ソフトウェアSUGARを用いたデータクリーニングについて概説します。今回の内容は、進化多様性研究の分野においても、遺伝的多様性や生物種多様性の推定などを高精度に行う際に、おそらく役立つものと思われます。皆様の研究のご参考になれば幸いです。

第2回 ソフトウェアSUGARを用いた大量シークエンスデータのクオリティ評価とエラー除去

佐藤行人・長﨑正朗
（東北大学 東北メディカル・メガバンク機構 ゲノム解析部門）

1. はじめに

次世代シークエンサー（NGS; next-generation sequencer）で得られた大量の塩基配列データから意義ある知見を得るためには、生データのクオリティ評価とクリーニングを適切に行うことが大切である。データを精査することの重要さは、もちろん塩基配列決定に限らず、あらゆる実験・研究に対して言える。しかしながら、特にNGSのデータは大量であるため（例：HiSeq 2500では1ランで数千億塩基対）、その細部までを人の目で確認することは、もはや不可能だ。そこで、コンピュータとソフトウェアによる図表化・可視化の力を借りながら、適切なクオリティ管理を行う必要が生じる。

NGSデータのクオリティ管理を行うこと、その具体的意義も明白であろう。誤った塩基配列データからは、誤った塩基置換やアッセンブリー配列しか推定されない。目的が全ゲノム・リシークエンスであれば、SNPや遺伝的多様度の誤推定をもたらすし、目的がメタゲノムやメタバーコーディングであれば、生物種や分類群の誤同定、さらには多様性の誤った評価をもたらす。厄介なのは、多数の正解に紛れて少数のエラーが混入した状態のデータである。実は、そのような『およそ高クオリティ』という状態のデータを扱わざるを得ないのが、多くの場合の現実である。なぜなら、NGSも決して完璧な機械ではないため、出力データの一部に、原理的ならびに偶発的な要因によるエラーが混じることがあるからである。そのため、何らかの根拠や手段に基づいて、エラーと判断されるデータを取り除く（フィルタリングする）ことが、NGSデータに基づく解析の信頼度、ひいては研究の質を左右する。

そこで本稿では、NGSデータのクオリティ管理について概説し、とくに筆者らが開発したエラー評価と除去のためのソフトウェアSUGAR[1]に焦点を当てて解説する。SUGARを用いることによって、より高精度な配列解析、遺伝的多様性推定、分子進化解析、メタゲノム／環境DNA解析などが可能になると期待される。SUGARは現在のところ、Illumina社のプラットフォーム（HiSeq, MiSeqなど）で配列決定されたデータの処理に対応する。そのため、以後の話題は主としてHiSeqやMiSeqマシンに絞られる。Pacific Biosciences社とLife Technologies社のプラットフォーム（それぞれPacBio RSIIおよびIon Torrent）については、適用できる内容は一部あるものの、基本的には本稿の対象外となることご容赦頂きたい。これらのクオリティ管理については、他の文献・書籍等を参照されたい[2]。

2. Illuminaプラットフォームにおけるシークエンシング・エラー

Illumina社のNGS（MiSeq, HiSeqなど）における配列決定のエラーは、(1) 原理的なエラーと (2) 偶発的なエラーの2つに大別される。前者は、マシンの配列決定原理に因る技術的な限界から生じる必然的なエラーであり、下で述べるように既存のツール／ソフトウェアを活用した定型的な対処が可能だ。一方、後者のエラーは、実験者の技術等に依らずある程度の確率で起きてしまう偶発的な性質のものを指す。フローセル流路（配列決定の基盤）への気泡や微少ゴミの混入、小規模なイメージングエラー（ハレーションやピンボケなど）が主なものだ。従来、後者の偶発的エラーについては、定型的に対処するソフトウェア／ツールが無かった。それを可能にしたのが筆者らの開発したSUGARである。まずこのセクションでは、Illuminaマシンにおける原理的エラーと偶発的エラーについてそれぞれ概説し、SUGAR開発の意図と意義について理解して頂く。

2.1. 原理的エラー

Illumina社のマシンにおける原理的な要因で生じる配列決定エラーとしては、主に次の3種類が知られる[3]。一つめは『位相のずれ』である。Illuminaマシンでは、フローセルと呼ばれるガラス担体上でクローナルなDNA断片をPCR増幅し、これを鋳型としたDNA合成反応を行う。その際に取り込まれるヌクレオシド三リン酸（dATP, dTTP, dGTP, dCTP）の種類を蛍光検出することで、配列を読み取る。この時、クローナルなDNA断片群が全て同期して蛍光を発するなら問題は無い。しかし実際には、ごく一部のDNAで蛍光反応が正常に起こらず、周囲のDNAとズレた発光サイクルにシフトしてしまう現象が起こる。例えば、読み取るDNA配列が“ATG…”だとした場合、大多数のDNAは1塩基目にA、2塩基目にT、3塩基目にGの蛍光を発するが、ごく一部のものでは2塩基目にA、3塩基目にTが発光したり（位相の遅れ）、または1塩基目にT、2塩基目にGが発光したりする（位相の早まり）。このようにサイクルがズレてしまうDNAは全体のごく一部であるものの、配列決定プロセスが進むほど累積していくことが知られる[3]。結果として、解読したいDNA配列の末端に近くなるほど、解読結果のブレが増大してデータのクオリティが下がる。

原理的エラーの2つめは『蛍光体の不分離』である。各塩基の読み取りに用いる蛍光体は、配列決定のサイクルごとに緩衝液でウォッシュアウトされる。しかし、このプロセスを完璧に行うことも現実には困難であり、ごく一部の蛍光体が洗い流されずに残ってしまう。Sheikh and Erlich[3]の例示によると、4種類の塩基（A, T, G, C）に対応する蛍光体のうち、Tを標識するものが比較的洗い流されにくい。すると、解読しているDNA配列の末端部分に近くなるほど、Tに対応する蛍光体の残留度が増し、他の塩基（A, G, C）の蛍光に対して干渉を起こす[3]。そのため、配列末端に近くなるほど、やはりデータのクオリティが低下する。最後、3つめの原理的エラーは『DNAクラスターおよびシグナルの減衰』である。これは、読み取りサイクルごとに行われる上述のウォッシュによって、蛍光体だけでなく配列決定の鋳型であるDNA断片も少しずつ洗い流されてしまうことで起こる。これにより、塩基読み取りのための蛍光シグナルそのものが、次第に弱くなる。この現象も、解読する配列の末端に近くなるほど影響が顕在化する[3]。

これら3種類の原理的エラーで重要なことは、そのいずれもが累積的な現象だということだ。つまり、これらのエラーは原因と機序こそ互いに異なるものの、要するに配列末端部分のクオリティが下がるという結果が

共通する（**図1A**）。そのため、これらの原理的エラーに対しては、比較的シンプルかつ定型的な対処が可能となる。代表的な対処法は、クオリティの低い配列末端部位をデータから一括削除することである（**図1B**）。設定した閾値（例：Phred score＜20）よりもクオリティ値の低い塩基が現れたら、その箇所より先をフィルターするプログラムが出回っており、SolexaQA ソフトウェアパッケージ[4]に含まれるDynamicTrim.plなどが代表的である。

もう1つの対処法は、ペアドエンド・シークエンシングのforward 側配列とreverse側配列の間で、オーバーラップする領域をアッセンブルすることだ（**図1C**）。この方法が適用できるケースは、配列決定のターゲットDNAが比較的短く、両配列間でオーバーラップする領域が存在する場合に限られる（目安として 10 bp 以上[5]）。この方法では、クオリティが概して低下する配列末端部位について、2回の配列決定とそのクオリティ値に基づくコンセンサスを取ることが出来るため、アッセンブルされた配列では全長に渡って高品質なデータが得られる利点がある。特定遺伝子の変異解析や、16S rRNA、mtDNA COI部分配列の決定などでは、この「ペアドエンドリード・アッセンブリー」の手法が採れるようPCRプライマーを設計することで、信頼度の高いデータ取得が可能になるだろう。ペアドエンドリード・アッセンブリーを実行するソフトウェアとしては、FLASh[5]やPEAR[6]などがよく用いられる。後者は、分子系統解析でおなじみのRAxMLを開発したAlexandros Stamatakisの研究グループによるものである。

2.2. 偶発的エラー

Illuminaマシンでは、次に挙げる偶発的なエラーによっても、フローセルの一部で配列読み取りの精度が低下することがある。偶発的エラーには、気泡や微小ゴミの混入、フローセルのヒビ、結露や試薬成分の析出などによるフローセルの曇り、イメージングの不具合（ハレーション、ピンボケなど）がある[7]。もちろん、これらのエラーが頻繁に起こるマシンについては、初期不良または保守対象としてメーカーが対応してくれる。また、通常に運転できているマシンでは、上記のような偶発的エラーが頻繁には入らないということであり、入ったとしても一般にはデータのごく一部が影響を受けるに過ぎない。ただし、配列決定の目的に依っては、これら気泡混入などで生じる局所的な低クオリティ・データの影響を看過できない場合がある。例えば、サンプルDNA中に低頻度で存在する体細胞性突然変異を探索する目的や、シークエンス深度（同じDNA配列を何回繰り返して読むか）が低い状態で一塩基多型（SNP）やその遺伝子型を推定する目的などである。

気泡混入などが厄介なのは、実験の改善による対応や定型的な対処を行うことが、比較的困難な点にある。これら偶発的エラーは、実験者の技術やマシンの個体差には関係無く発生する側面があるからだ。実際、筆者らが所属する機構で行った1,000人ゲノム配列決定においても、たまに発生する気泡混入等について一定の傾向をつかむことは出来なかった。加えて、例に挙げた気泡などだけではなく、原因がよく判明していな

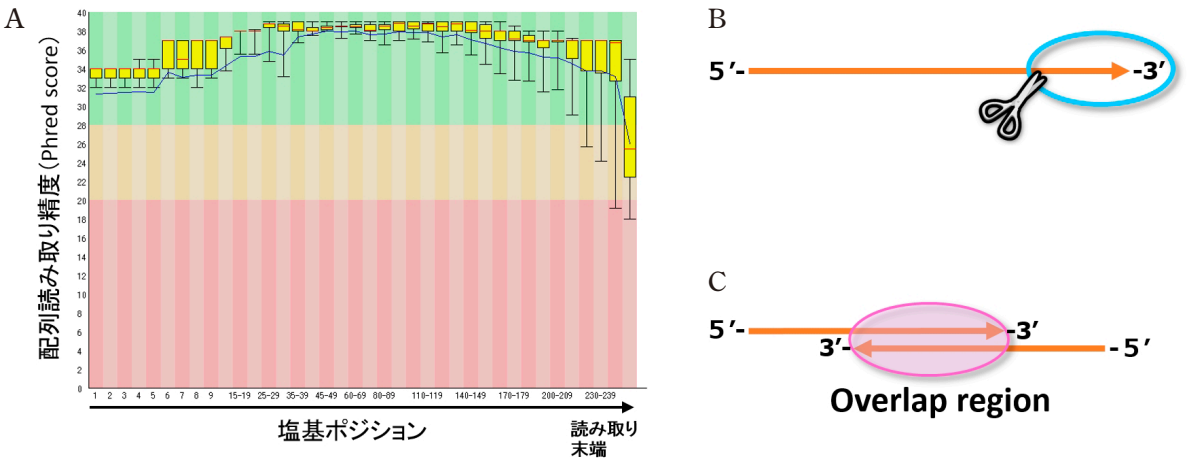

図1　原理的な要因によるクオリティ低下と定型的対処。（A）配列末端のクオリティ低下の例。（B）末端部位除去および（C）ペアドエンドリード・アッセンブリーによる対処の概念図。

い局所的エラーもある[7]。その1つ1つを追求することは、もはや研究の本分から離れてしまう。そこで、*ad hoc*な姿勢になることは承知しつつも、適切なソフトウェアを開発して偶発的エラーの有無を監視・検出し、結果に応じて、適宜局所的なデータクリーニングを実施する必要があると考えた。そのようなソフトウェアの開発は、集団ゲノミクスなどの大プロジェクトにおいてデータの質をルーチン的に揃えることに寄与する。また、個々の研究者が、取得した貴重な配列データを有効活用する上でも役立つだろう。

3. SUGARソフトウェアの開発

上記の背景により筆者らは、東北メディカル・メガバンクプロジェクトで行う1,000人ゲノム配列決定でのデータ解析パイプラインの一部として、気泡混入などの偶発的エラーに対処するソフトウェアSUGARを開発した[1]。SUGARの前にも既存のソフトウェアが複数発表されていたものの（**表1**）、気泡混入などに効果的に対応するためには、幾つかの新機能が必要だったからだ。

必要な機能の1つ目として、気泡や微小ヒビ等の局所的シグナルを的確に捉えるための高精細な可視化がある。Illuminaプラットフォームでの配列決定反応はフローセル上で行われるが、フローセルは「タイル」と呼ばれる単位に区切られて画像としてキャプチャーされ、配列が決定される。例えばHiSeq 2500 Rapid Runモードのフローセルでは、1レーンが64個のタイルに分割される。配列決定が完了すると、タイルの番号とその中での座標が、その位置で決定されたDNA配列の名前に書き込まれた状態で結果（fastqファイル）が出力される。つまり、各DNA配列がフローセル上のどの位置で読み取られたかを、配列名から分析できるようになっている。その情報を利用することで、フローセル上でのデータクオリティ分布を可視化できる（**図2**）。

フローセル上のデータクオリティ分布を可視化する既存ソフトウェアにHTQC[9]とSolexaQA[4]があるが、これらは気泡や微小ヒビを捉えられるほどの解像度を持たない（**図2A**）。両ソフトウェアともタイルを可視化の最小単位としているが、一方で気泡混入などの偶発的エラーは、タイルよりも微細なスケールで起きることが少なくないからだ。そこでSUGARは、タイルをさらに100分割した「サブタイル」(10x10格子）として可視化できるよう設計した。その結果、混入した気泡や、気泡がウォッシュプロセスにより押し流される様子、何らかの原因で配列が出力されなかった微小区画などを、簡便に発見できるようになった（**図2B**）。クオリティ分布の可視化は幾つかの基準、すなわち低クオリティ配列の含有割合（**図2B**）、配列出力密度（**図2C**）、平均クオリティ値（**図2D**）、およびマッピング・クオリティ値（**図2E**；BAM/SAMファイル[10]を読み込んだ場合）で行うことが出来る。とくにBAM/SAMファイルの分析機能は、偶発的エラーが配列のアライメントに齟齬を生じさせているかどうかを評価する上で重要である。なお、設定によりさらに高精細な可視化（タイルあたり20x20以上）も可能だが、使用メモリー量は増加する。筆者らの経験では、10x10格子の解像度で実用上十分だと思われる。

必須となるもう1つの機能は、ヒトゲノム・リシークエンシングで必要な規模のデータを、最小限の計算およびメモリーコストで処理出来ることだ。既存ソフトウェアのHTQC[9]とSolexaQA[4]は、Illuminaの従来機種であるGAIIxおよびHiSeq1000が主流であった当時に発表されたものであるためか、東北メディカル・メガバンク機構で採用したHiSeq 2500のフルデータを処理することは出来なかった。当機構ではヒトゲノム配

表1 SUGARと他のデータクオリティ管理ソフトウェアの機能比較

	FastQC	HTQC	SolexaQA	SUGAR
データクオリティ可視化	○	○	○	○
タイルレベルのクオリティ分布視覚化	○	○	○	○
高精細なクオリティ分布視覚化	×	×	×	○
HiSeq2500 フルデータの処理	○	×	×	○
低クオリティデータ除去	×	○	○	○
高精細な低クオリティデータ除去	×	×	×	○

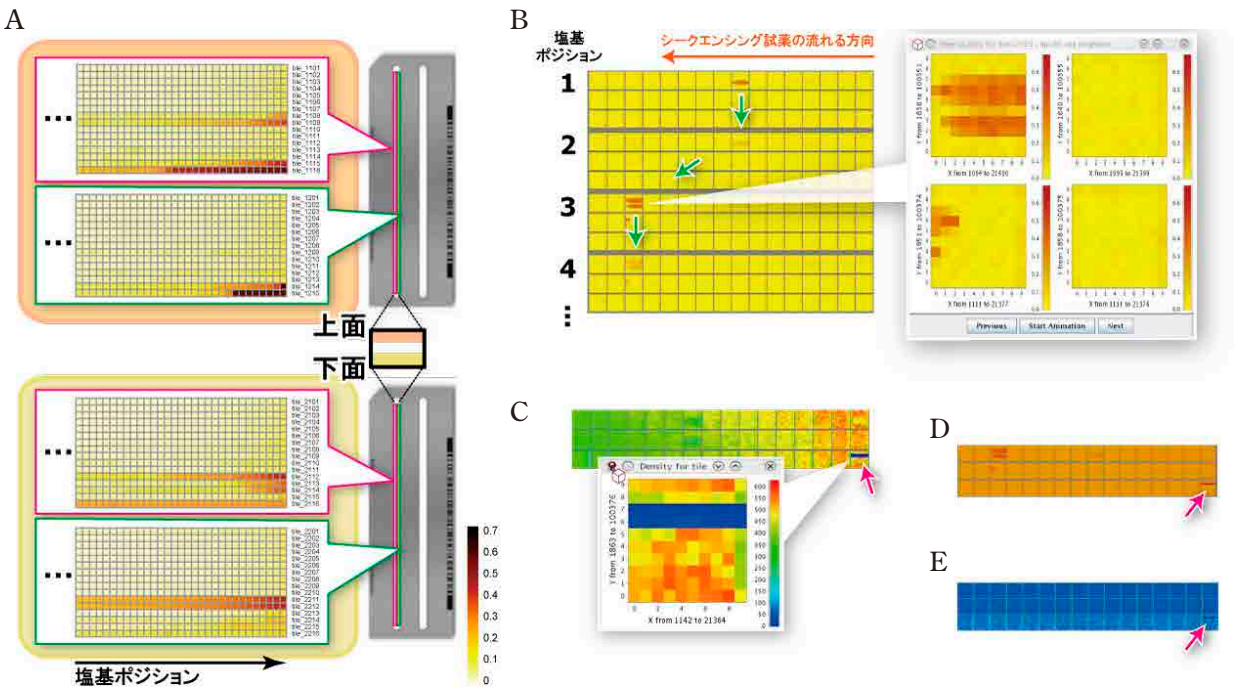

図2 フローセル上のデータクオリティ分布の可視化の例。(A) SolexaQAによるタイルレベル解像度での可視化。カラーはPhred スコア30以下の塩基が占める割合を表す。(B) SUGARによる10x10サブタイルレベルでの可視化。気泡と思われる低クオリティスポットがシークエンシングサイクル毎に流されていく様子が分かる。(C) 出力配列密度、(D) 平均クオリティスコア、および (E) マッピングクオリティスコア分布の可視化。これらにより出力データの局所的な低下スポットが発見され (赤矢印)、これがマッピング解析の結果に影響したことも分かる

列を30xの深度 (＞90 Gb；900億塩基対以上) で取得するが、その規模のデータを処理すると、上記2つのソフトウェアはメモリーのオーバーフローを起こしクラッシュしてしまう (Xeon CPU E5-2640 2.50 GHz 1,500コアと＞500 GBメモリーのクラスターマシン環境)。そのような現象を防ぐために、両ソフトウェアにはデータのダウンサンプリング機能が備えられており、部分的なデータ解析によって**図2A**に示したようなクオリティ分布の評価のみは行うことが出来る。しかし、ダウンサンプリングしたデータの解析結果を元に、オリジナルのフルデータについて的確なクリーニング処理を行うことは通常できない。そこでSUGARは、処理が高速なJavaプラットフォームで開発を行い、1処理あたり最大250 MBのメモリー使用量でHiSeq 2500のフルデータを処理できるよう設計した (**表1**)。フローセル上のクオリティ分布を高精細に行い、かつ動物のゲノム・リシークエンシング規模のデータを処理することが出来るクオリティ管理ソフトウェアは、現在のところSUGARのみである。これにより、偶発的エラーの分析と、その結果に基づくデータクリーニングが実施できるようになった。以降から、SUGARの具体的な使用方法とその効果について解説したい。

4. SUGAR の使用と効果

当ソフトウェアは、コンピュータマウスを用いたグラフィカルな操作 (GUIモード) と、コマンドライン入力に基づいた操作 (CUIモード) のどちらからも使用できる。OSに依存せず、version 6 以降のJava Runtime Environmentがインストールされた環境で動作可能だ。プログラム本体はGitHubのプロジェクトホームページ (https://github.com/biomedinfo/sugar) で公開・配布されており、操作マニュアルはGUI版のヘルプメニューから閲覧することが出来る。ここではGUI版の操作を中心に、使用方法とその効果を概説したい。

4.1. SUGARの入手と実行

入手は上記ホームページの右側にある「Download ZIP」から行う。ダウンロードされるファイル「sugar-master.zip」を解凍し、生成されたディレクトリ内にある「Sugar_v1.zip」をさらに解凍すると、プログラム本体とテストファイルが現れる。なおディレクトリ「Sugar_v1-src」は、ライセンス上公開が義務付けられてい

るソースコードを格納している。プログラム本体のファイルは「Sugar.jar」であり、Windows環境ではダブルクリックすることにより起動する。コマンドラインでは下記コマンドにより起動する。

```
$ cd [path/to/your/]sugar-master/Sugar_v1.0
$ java -jar Sugar.jar
```

SUGARのGUI版が起動したら、メニューバーから「Help -> Contents」を閲覧することで、設定や実行の方法を参照できる。詳細についてはそちらに譲り、本稿ではクオリティ分布の可視化とデータクリーニングの操作の典型例を示したい。

4.2. クオリティ可視化とデータクリーニング

ファイルをロードし分析が完了すると、**図3A**のような高精細ヒートマップによるクオリティ分布の可視化が完了する。可視化の結果はhtmlレポートとして保存することができる。SUGARは、シークエンサーが出力する配列ファイル（fastqまたはfastq.gz）だけでなく、配列データをリファレンスゲノム配列などにアライメントした後のファイル（BAM/SAM）も分析できる。このアライメントファイルの分析機能により、配列決定のクオリティ値だけでなくマッピングクオリティ（MapQ）についても、フローセル上でどのように分布して

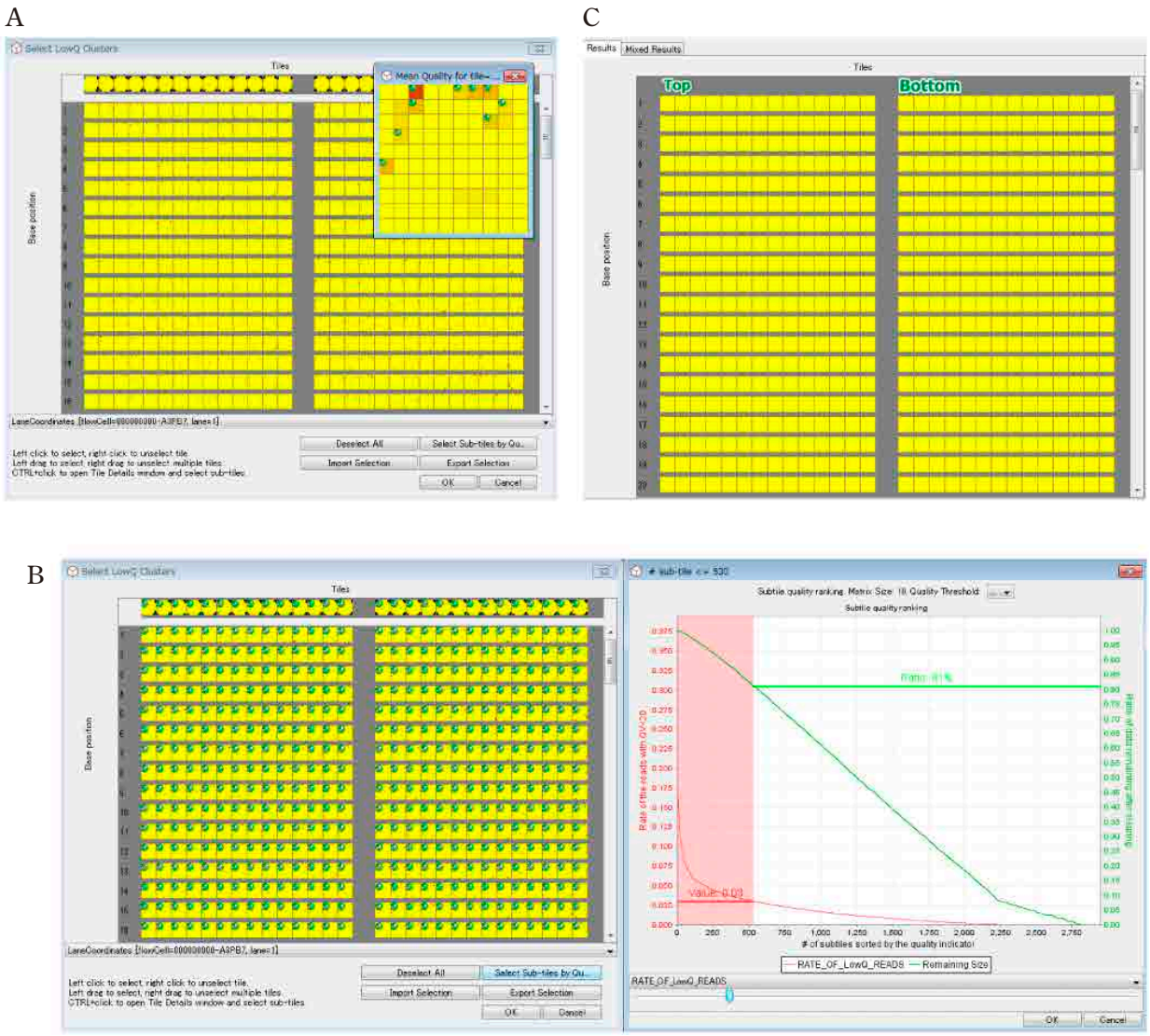

図3　SUGARによるクオリティ評価とクリーニング作業。（A）サブタイル・レベルでの低クオリティ・スポット可視化とマニュアル指定によるクリーニング作業の例、（B）クオリティ指標と残存データ予測量のラインングラフに基づく自動クリーニング設定作業例、（C）クリーニング後の状態

いるかを可視化できる。この機能を利用することで、偶発的エラーに起因する低クオリティ・スポットが、実際に高次解析（リファレンスゲノムへのマッピング）に影響を及ぼしたかどうかを評価できる。このように複数の基準によるクオリティ評価の結果を比較検討できるソフトウェアは、現在のところSUGARのみである。

図3Aのヒートマップ、または同・右上に表示されている拡大ヒートマップをマウスでクリックすることで、フローセル上に散在する局所的な低クオリティ・スポット（赤茶色のグラデーションで表示）を選択できる。選択したスポットについて、(1) そのスポットで決定された配列（リード）をすべて削除する、または、(2) そのベースポジションの塩基を全て「N」に変換する、というデータクリーニングを実行できる。このようなマウスクリックによるマニュアル操作だけでなく、SUGARには自動クリーニング機能も実装されている。その条件設定は、**図3B**右側に示したライングラフを用いて行う。このグラフは、データクリーニングで特定のクオリティ閾値を採用した時に（**図3B**赤ライン；例えばサブタイル中の平均リードクオリティ値をPhred score >35.0 にする、など）、クリーニングされた後の残存データ量がどのくらいになるのかをあらかじめ分析し表示するものである（**図3B**緑ライン；オリジナルデータに対する割合%）。画面下部のスライダーを用いて、望みのクリーニング閾値とデータ残存量の釣り合いが取れるポイントを探して適用すれば、該当するサブタイルが自動選択された状態でクリーニングが実行される。**図3C**に、自動クリーニング後のデータを再分析した結果を示した。なおSUGARは、ペアドエンドリードのデータ処理にも対応している。

4.3. データクリーニングの効果

次に、SUGARによる局所的なエラースポット除去の効果について例示したい。まず、視覚的な例として**図4**を示す。上が自動クリーニングの適用前、下が適用後である。この**図4**では、当機構で行ったヒトゲノムシークエンシングの結果のうち、ある1人の方の配列データについて、ミトコンドリアゲノムのリファレンス

図4　SUGARによる自動データクリーニングの結果例

配列にマッピングされたものの一部を示した。2015年2月時点で未発表（論文審査中）のデータであるため、ここでは具体的な数値を示すことが出来ずに恐縮であるが、クリーニング前（上のパネル）にはアライメント中に散在していた塩基置換が、クリーニング後（下のパネル）に相当数取り除かれていることが見て取れるだろう。つまり、局所的・偶発的なエラーが、確かにリファレンス配列との齟齬の一因となっていることが分かる。クリーニング後のデータを解析に用いることで、真に存在しうるヘテロプラスミーや核内ミトコンドリア様配列（NUMT; nuclear mitochondrial DNA）を探索するための統計的扱いや解析が、より容易になると思われる。

もう1つの例として、データベース上の公開データを用いたテスト解析から、具体的な数値の向上を示したい。使用したデータは、国際HapMapプロジェクトのヒト検体NA12878のものであり、全ゲノム配列をHiSeq2500で決定したものだ（SRP021027；100 bpペアドエンド・シークエンシング）。そのフルデータについて、SUGARのライングラフ機能（**図3B**）を活用し、平均リードクオリティがPhred score 22未満のサブタイルを削除した（＞70％のデータが残存）。ヒト・リファレンスゲノム配列にアライメント解析したところ（Bowtie2[11]を使用）、アライメント結果のクオリティ指標である平均MapQスコアが、クリーニング前は29.5であった一方、クリーニング後は30.1に上昇した[1]。僅かな上昇だと思われるかもしれないが、データの母数は数10億以上にのぼる。そのため、削除された配列（平均MapQ値は25.0であった）による塩基置換の誤推定と、それによるMapQ値の低下は、研究の目的によっては無視できないものである。またもう1つの重要な点は、上記のデータクリーニングがマッピング結果（MapQ値）に基づいて行われたのではなく、あくまで配列のクオリティ値に基づいて検出された低クオリティスポットを対象に、行われたことである。それでもアライメント結果のMapQ値が上昇したということは、データに含まれる偶発的エラーが、確かに解析上の影響を持つことを示す。クオリティが一般に高いことが多い公開データだけでなく、クオリティが低いこともある個別のデータに対して、SUGARによる偶発的エラーのクリーニングを行うことの効果は、自ずと明らかであろう。

4.4. コマンドライン／バッチ処理での使用

SUGARはコマンドラインからも動作するので、バッチ処理によるパイプライン解析に組み込むことも可能である。その場合は、コマンドによりレポート出力やデータクリーニングについての各種設定を行う。使用できるコマンドは、GitHub上のプロジェクトホームページwiki（https://github.com/biomedinfo/sugar/wiki）の記載か、または下記のヘルプコマンドから参照することが出来る。

```
$ java -jar Sugar.jar --help
```

バッチ処理は、他の多くのプログラムやソフトウェアと同様に、下の例のようなシェルスクリプト等を用意して実施する。計算環境によっては、javaとシェルスクリプトのメモリー制限に関するオプションを追加して実行する必要がある（詳細は東京大学医科学研究所ヒトゲノム解析センターのホームページ等の解説を参考されたい：https://supcom.hgc.jp/japanese/utili_info/manual/faq.html#2829）。

```
#!/bin/bash

IN_DIR="../data"
OUT_DIR="../SUGAR_out"
SUGAR="../../tools/Sugar_v1.0/Sugar.jar"

mkdir $OUT_DIR
```

```
cat $IN_DIR/*R1*fastq > $IN_DIR/R1_merged.fastq
cat $IN_DIR/*R2*fastq > $IN_DIR/R2_merged.fastq

date +"%Y/%m/%d %H:%M:%S"
echo -e "Starting SUGAR analysis\n"
java -jar $SUGAR -f fastq $IN_DIR/*merged.fastq -o $OUT_DIR/
date +"%Y/%m/%d %H:%M:%S"
echo -e "Analysis finished\n"
```

```
$ nohup ./sugar.sh
```

5. まとめ

本稿では、Illumina社のNGSプラットフォームにおける原理的および偶発的な配列読み取りエラーの要因について概説し、さらに、後者の偶発的エラーに対処するために筆者らが開発したクオリティ管理ソフトウェアSUGARについて紹介した。すでに述べた通り、SUGARによる偶発的エラーへの対処は、データの中に少量含まれる塩基配列の誤推定を取り除く上で有効である。進化多様性研究の分野ならば、例えばメタバーコーディング手法に基づく微生物や動植物の多様性解析、また、サンプルに少量含まれるウィルス配列変異の検出やその分子進化解析などに役立つだろう。周到な研究デザインや実験により得たデータから、さらに高精度の解析を実現する上で、SUGARが役立てば幸いである。

6. 謝辞

SUGARソフトウェアのバグ修正、追加プログラミングおよびデータ処理作業にご尽力頂いた研究室の小野彰氏とアディーエミ・アディフェミ氏に御礼申し上げます。また、当ソフトウェアで使用したアイコンおよびロゴをデザインして頂いた畑中俊哉氏に感謝申し上げます。

引用文献

[1] Sato Y, et al. SUGAR: graphical user interface-based data refiner for high-throughput DNA sequencing. *BMC Genomics*. 2014, **15**: 664.
[2] 二階堂愛（編）．実験医学別冊『次世代シークエンス解析スタンダード　NGSのポテンシャルを活かしきるWET&DRY』．羊土社，2014，404 pp.
[3] Sheikh MA, Erlich Y. Base-Calling for Bioinformaticians. In Rodríguez-Ezpeleta N, Hackenberg M, Aransay AM. (eds) Bioinformatics for High Throughput Sequenc-ing. Springer, New York, 2012, pp. 67–84.
[4] Cox MP, et al. SolexaQA: At-a-glance quality assessment of Illumina second-generation sequencing data. *BMC Bioinformatics*. 2010, **11**: 485.
[5] Magoc T, Salzberg S. FLASH: Fast length adjustment of short reads to improve genome assemblies. *Bioinformatics*. 2011, **27**: 2957-2963.
[6] Zhang J, et al. PEAR: a fast and accurate Illumina Paired-End reAd mergeR. *Bioinformatics*. 2014, 30: 614-620.
[7] Dolan PC, Denver DR. TileQC: a system for tile-based quality control of Solexa data. *BMC Bioinformatics*. 2008, **9**: 250.
[8] Andrews S. FastQC: a quality control tool for high throughput sequence data. 2010. http://www.bioinformatics.babraham.ac.uk/projects/fastqc/
[9] Yang X, et al. HTQC: a fast quality control toolkit for Illumina sequencing data. *BMC Bioinformatics*. 2013, **14**: 33.
[10] Li H, et al. The Sequence alignment/map (SAM) format and SAMtools. *Bioinformatics*. 2009, **25**: 2078-2079.
[11] Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. *Nature Methods*. 2012, **9**: 357-359.