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3回 Linux 環境構築から NGS データ取得まで

孫 建強[1]、三浦 文[2]、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

Bio-Linux は、多くの次世代シーケンサー（以下、NGS）解析用プログラムが予め組み込まれた Linux 環境の1つである。Bio-Linux は、普段利用するホスト OS（Windows や Macintosh）ではなく、仮想マシンを導入してゲスト OS 上に構築するのが一般的である。連載第3回は、ホスト OS 内にゲスト OS（Bio-Linux 8）が存在する感覚を掴めるように、スクリーンショットを例に仮想マシンの概念から説明する。ホスト－ゲスト間でのデータのやりとり（共有フォルダの設定）や Bio-Linux 8 の基本的な使い方を述べ、Linux 環境下での乳酸菌ゲノムデータ取得を行う。また、チェックサム、ファイルの解凍と圧縮、コマンドマニュアル、リダイレクトとパイプなど、NGS 解析を効率的に行う上で有用なテクニックを述べる。公共データベース（DB）中の乳酸菌 NGS データを概観し、日米欧三極の DB の特徴や注意点を述べるとともに、*Lactobacillus casei* 12A のトランスクリプトーム（RNA-seq）データ取得を行う。利用可能なハードディスク（HDD）容量の制約など、個々のユーザの置かれた PC 環境下でも、Linux コマンドを組み合わせることで高速かつ効率的に目的を達成することができる。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, Bioinformatics, Linux commands, Virtual machine

はじめに

連載第3回の本文は、ホスト OS が Windows 7（以下、Win）環境で動作確認した結果を中心に述べる。ホスト OS が Macintosh の OS X Yosemite（以下、Mac）環境では現象を再現できない事柄もあるため、第3回のウェブ資料は、Win 用と Mac 用の両方を用意した。本文中の流れとの整合性を保つため、Mac 用は多少説明の順番の前後や内容の重複がある点をご容赦願いたい。Win と Mac の Linux 環境の違いは、それほど大きなものではない。しかし、Mac ユーザの Linux 初心者にとって、Win との違いに起因するトラブルに何度も遭遇し、その都度、それに対処しなければならないのはストレスがたまるだけである。そのため、今回のウェブ資料作成にあたっては、Mac ユーザに最大限の配慮を行っている。第3回の本稿を読み始める前に、Win と Mac の両ユーザともにウェブ資料（以下、W）に一通り目を通しておくことを勧める。本文中で触れない番号も存在するが、想定されるミスやその対処法など、ノウハウも含めている。例えば W10-2（ウェブ資料の 10-2 ページ）は、W10-1 までのウェブ資料の内容を実行し理解しているという前提で記載しているので注意していただきたい。

*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

Bio-Linuxの導入と起動

連載第2回[1]では、WinとMacのコマンドライン環境を紹介した。また、Linux環境構築の1つとして、VirtualBoxおよびBio-Linux 8[2]の詳細なインストール手順をウェブ上で示した。第3回は、VirtualBoxを用いて構築した仮想マシン環境にBio-Linux 8がインストールされているという前提のもとで話を進める。第1回[3]でも述べたように、Bio-Linux 8はLinuxの一種であるUbuntuをベースにしてカスタマイズされたOSである。NGS解析用プログラムを含む様々な解析ソフトが予め組み込まれたデータ解析環境であるため、基本的なLinuxコマンドだけでなく、NGSデータ取得後のクオリティコントロールを行う代表的なプログラムであるFastQCなどがBio-Linux 8のターミナルで利用可能である。

（Rで）塩基配列解析中の手順に従ってBio-Linux 8のインストールを行うと、仮想化ソフトVirtualBox内に最大で150 GBのHDD容量、2048 MBのメモリ、OSがUbuntu（64 bit）の「BioLinux8」という名前の仮想マシン環境が構築される［W1］。もちろん、ゲストOSで確保可能なHDDやメモリ容量はホストOS（通常はWinまたはMac）のスペック次第であるため、ダウンロードするNGSデータのファイルサイズなどを考慮し、ユーザのPC環境に応じて変更してもよい。しかし、後述する*L. casei* 12A株のNGSデータ（SRR616268）が解凍前で約15GB、解凍後のFASTQ形式で約80 GBに達するため、それよりも少ないHDD容量しか確保できない場合にはダウンロードすらままならない点に注意が必要である[4]。

「BioLinux8」は、Bio-Linux 8がインストールされた「biolinux（またはbielinux）」というPC名の仮想マシン（ゲストOS）として起動される。この仮想マシンに、例えばユーザ名iuでログインしたとする。この手順およびログイン後の初期画面は、ホストOSの違いによらず基本的に同じである［W2］。しかし、PC（ホストOS）の中にPC（ゲストOS）が存在するという概念がなかなか理解しづらい読者もいるだろう。この状況を限定的にではあるが確認する1つのやり方は、画面の取り込み（キャプチャ）である。W2-4は、コンピュータの画面に表示されている内容を、キーボードのPrint Screenボタン（ノートPCの場合はPrtScキーなど；Macはcommand + shift + 3）を押してクリップボードにコピーしたイメージ（スクリーンショット）である。一見ごく当たり前のことを述べているだけのように思うかもしれないが、このスクリーンショットを撮ることができるのは、BioLinux8がアクティブでない場合に限られる［W3］。

一方、BioLinux8ウィンドウ（ゲストOS）がアクティブな状態で画面の取り込みを行おうとPrint Screenボタンを押すと、W2-4とは違った結果となる［W4］。W4-2中のスクリーンショットは、ゲストOSのPC全画面分の内容である。つまり、ゲストOSであるBio-Linux 8というLinux環境の全てに相当するものだと解釈すればよい。それゆえ画面サイズを最大化すれば、WinやMacといったホストOSの種類に関係なく、見栄えも含めて同じBio-Linux 8というLinux環境で作業を行うことができる［W4-3］。1つの独立したPC環境を構築できることが、仮想マシン（バーチャルマシン、VM）と呼ばれる所以である。

Bio-Linux（ゲストOS）とホストOS間のやりとり

ゲストOSのPC全画面分の内容に相当する画像［W4-2］は、Copy to Clipboardボタンを押した後に、ホストOSのプレゼンテーションソフト（Microsoft PowerPoint）に貼りつけて作成したものである［W5-1］。しかし、仮想マシン環境のデフォルト設定（クリップボードの共有が無効）では、残念ながらこの作業はできない。ゲストOSとホストOS間で双方向（bidirectional）にクリップボード情報を共有するように設定を変更する必要がある［W5-2］。
もちろんゲストOS内のみで一通りの作業を行うことは可能である。例えば先程のBioLinux8（ゲストOS）がアクティブな状態で画面の取り込みを行った結果を、そのままゲストOS内に保存することができる［W6-1］。保存先フォルダを開いて画像ファイルを開くこともできる［W6-2］。しかし実際問題として、使い慣れないゲストOSのGUI環境での作業は苦行である。例えば、画像表示ソフトImage Viewerでファイルを開いていたとしても、作業中のフォルダに隠れて見えないことがある。使い慣れたホストOS上であれば、画像表示ソフトのウィンドウをアクティブにすればよいと瞬時に判断できるが、使い慣れないゲストOS上では、ソフトの名前もよくわからない上、何が起こっているのかもよくわからない場合が多い。たとえ状況がわかったとしても、対策がよくわからずストレスを感じることもあるだろう［W6-3］。

Bio-Linux 8では、他にもウェブブラウザ（Firefox）、文章作成ソフト（LibreOffice Writer）、表計算ソフト（LibreOffice Calc）、プレゼンテーションソフト（LibreOffice Impress）などを利用することができる［W7］。これらのソフトの使い方を最初から覚えるのも1つの手段ではあるが、Linux（ゲストOS）環境でなくてもできる作業は、使い慣れたホストOS上で行いたいと考えるのが一般的であろう。仮想化ソフトVirtualBox（ver. 4.3.18）の機能として、ゲストOS中のファイルをホストOSにドラッグ＆ドロップ（以下、D&D）することはできないものの、ホストOSからゲストOSへのファイル移動は可能である［W8］。また、VirtualBoxウィンドウのメニュー画面経由で（ゲストOSの）スクリーンショットを撮れば、pngファイルの保存先がホストOSとなる［W8-8］。

一部の読者は、設定画面でD&Dを「双方向」にしてい

るにもかかわらず、なぜゲストからホストへのファイルのやりとりができないのか不便に思うのではないだろうか［W5-2; W8-1; W9-1］。これは、仮想マシンという閉ざされた環境とホストOSとの間で画面操作がシームレスに行えないことによるもので、ウェブ検索などで他の手段を見つけ出すのが最も現実的な解決策である。本稿で提示する有効な解決策は、ホストOSとゲストOSで同じフォルダの中身が見られるように「共有フォルダの設定」を行うことである［W9］。W9-2では、ホストOSのデスクトップ上にshareフォルダを新規作成し、ホストOSのVirtualBox側で共有フォルダとして追加している。次に、ゲストOSのデスクトップ上にmac_shareフォルダを新規作成して、このゲストOS上のmac_shareフォルダとホストOS上のshareフォルダを共有フォルダとして認識させている。この認識させる作業をマウント（mount）といい、「shareフォルダをmac_shareフォルダにマウントする」という表現がなされる。なかなか理解しづらい概念かもしれないが、ここでは同一のフォルダを異なるOS間で共有するために必要な作業だと思えばよい。

ここで著者が経験した不便と思われる事柄、およびその対処法を2つ挙げておく。1つ目の事柄は、マウスカーソルの位置に関するものである。D&Dでホスト OSからゲストOSへのファイルのやりとりが可能であることは先に述べた。しかし著者は、ドロップ（ファイルを置く）先のゲストOSのウィンドウ画面上で、マウスカーソルが駐車禁止マークのような形に変わってドロップできない現象に遭遇したことがある。当初は原因がよくわからず、ゲストOSの再起動を行ったのち再度トライするとなぜかできた経験がある。結論として提示する合理的な対処法は、ゲストOSのウィンドウ内でマウスカーソルの位置を変えていき、マウスカーソルがドロップできるマークに変わる場所を見つけることである［W9-5］。

2つ目の事柄は、ファイル名に関するものである。共有フォルダの設定ができた後、著者が最初に試したのは、ゲストOS上にあるファイルを共有フォルダに置いてホストOS（Win）で見られるかどうかを確認することであった［W9-2］。ウェブ資料の手順通りに作業を行っていくと、W6-1でPicturesフォルダに保存したPNGファイル（ゲストOSのPC画面のスクリーンショット；ファイル名はScreenshot from 2014-11-14 12:19:47.png）がゲストOS環境下で作成された唯一のファイルとなる。このファイルを共有フォルダ（ゲストOSのデスクトップ上にあるmac_shareフォルダ）に置こうとすると、Protocol errorという何をどう対処すればいいのか見当もつかないエラーメッセージに遭遇する。著者らも現象の把握および対処法の確認に数時間程度かかったが、結論としては共有フォルダの設定手順には何の問題もなく、Bio-Linux 8上で自動生成されるファイル名が原因であった。連載第2回でファイル名の常識・非常識について述べたが、上記ファイル名の中にも著者の感覚では非常識な「スペースやコロン（:）」が含まれている。W9-2では、ファイル名をhoge.pngに変更すれば共有フォルダmac_shareに置けることを示しているが、最低限ファイル中のコロンをなくせばいいようである。例えば、Screenshot from 2014-11-14 12_19_47.pngのようにコロンをアンダースコア（_）にするとエラーが出ない。

ではなぜこのような不都合が生じやすいファイル名を自動生成するようになっているのか？これはおそらく単純にBio-Linux上での利便性を犠牲にして視認性を重視したためであろう。Macの場合は、スクリーンショット（command + shift +3）をとると、同様の名前のpngファイルがデスクトップ上に自動生成される。この際、ファイル名中の時刻（12:19:47）に相当する部分がコロン（:）ではなくドット（.）になっていることに気づくであろう。WinとMacはファイル名にコロンという記号を使わないが、Bio-Linuxは許容するという思想の違いなのであろう。しかし、三者三様に異なる改行コードの違いや、WinとMac間での新規フォルダ作成時にデフォルトで付与されるアクセス権限の違い［W9-2-6］など、OS間をまたぐ作業は通常の利用以上に幅広い知識を要求される。重要なのは、ユーザが対処法を知っておくことである。

ここまで、Bio-Linux 8の基本的な使い方、およびホスト⇔ゲスト間のファイルのやりとりについて述べた。慣れないうちは、Linux環境でしかできない必要最小限の解析をゲストOS（Bio-Linux 8）上で行い、それ以外の作業をホストOS上で行うという方針でもいいだろう。しかし、OSまたはPC間でのギガバイト（GB）レベルのファイルのやりとりは、一般に通信速度の問題や上述の改行コード問題など様々な困難に直面することが多い。理想的には、覚悟を決めてLinuxの世界に飛び込み、できるだけ多くの作業を同じOS（Bio-Linux 8）上で行うことである。また、数MB程度のファイルであればホストからゲストOSにD&Dするのが効率的だと思われるが、数十GBにもおよぶファイルは共有フォルダに置いたほうがいいだろう。D&Dだと同じファイルがホストとゲスト両方に存在することになり、HDD容量がすぐに枯渇するからである。

データ取得とチェックサム

連載第2回で行った作業をLinux（Bio-Linux 8）環境でおさらいする。前回は、ホストOSのデスクトップ上にhogeフォルダを作成し、その中にEnsembl[5-6] Bacteriaのウェブサイトからダウンロードした乳酸菌ゲノム（*Lactobacillus casei* 12A株）配列のFASTA形式ファイル（以下、FASTAファイル）を置き、コンティグ数をgrepコマンドで計数した。ここでは、コマンドライン環境で一般に行われるwgetコマンド（だぶるげっと、と読む）を用いたファイル取得、sumコマンドを用いたチェッ

クサム（check sum；ダウンロードが正しく行われたかどうかを確認）、gzip コマンドを用いた圧縮ファイルの解凍などを含めて解説する。

ゲスト OS である Bio-Linux 8 のターミナル起動後の状態から、① ls, ② cd Desktop, ③ ls, ④ mkdir hoge, ⑤ cd hoge, ⑥ ls, ⑦ pwd, ⑧ whoami を打ち込んだ一連の結果を**図 1** に示す。若干見栄えが異なるものの、第 2 回原稿中の図 2 や図 3（Mac の画面上のコマンド実行結果）とかなり似ていることがわかる。一番の違いは、⑥の ls コマンド実行結果として FASTA ファイルの名前が表示されないことである。ホスト OS とゲスト OS は独立した操作環境であるため、たとえ「ホスト OS – デスクトップ – hoge」中に FASTA ファイルがあったとしても、それは「ゲスト OS – デスクトップ – hoge」とは無関係だという当たり前のことを述べているに過ぎない。

「ホスト OS – デスクトップ – hoge」中の FASTA ファイルを「ゲスト OS – デスクトップ – hoge」に置くやり方は、D&D［W10-3］と共有フォルダの利用［W10-5］の 2 通りある。これは、予めウェブブラウザ（Internet Explorer や Safari など）を利用して、FASTA ファイルを目的の URL まで移動して gzip 圧縮ファイルをダウンロードおよび解凍したのち、「ホスト OS – デスクトップ – hoge」に置いたものが存在するという前提にたつ。これ以外のやり方もいくつか存在するが、Linux 環境で作業する場合には、wget コマンドを利用して直接「ホスト OS – デスクトップ – hoge」にダウンロードするのが一般的である。

wget コマンドの基本的な利用法は、「wget *URL*」である。ここで、URL のところにはダウンロードしたいファイルの URL 情報を入力する。*Lactobacillus casei* 12A 株の場合には、「ftp://ftp.ensemblgenomes.org/pub/bacteria/release-22/fasta/bacteria_15_collection/lactobacillus_casei_12a/dna/Lactobacillus_casei_12a.GCA_000309565.1.22.dna.toplevel.fa.gz」となる。ls コマンドの「-la」オプションなど既にいくつかの Linux コマンド実行時にも利用しているが、wget コマンドにもいくつかの便利なオプションが存在する。例えば「-c」オプションをつけておくと、ダウンロードが途中で止まってしまった場合でも、そこからレジュームする（再開する）ことができる。「wget *URL*」でタイムアウトしてしまった場合などに「wget -c *URL*」でもう一度実行するというのが基本的な使い方ではあるが、最初から「-c」をつけておいても問題はない［W11-1］。数 GB ～数十 GB の NGS データをダウンロードする際にぜひ利用してほしい機能である。

ダウンロードするファイルのバージョンにも注意してほしい。上記 URL からも分かるように、このゲノムデータは Ensembl Genomes release 22（2014 年 4 月）のものであるが、2014 年 11 月には release 24 となっている。バージョンアップには、新規ゲノムの追加なども含まれるであろうが、既存ゲノム配列にも多少の変更がなされているかもしれない。**（R で）塩基配列解析**では、説明やデータ取得時の利便性および統一性を優先しているものの、特段の事情がない限り最新リリースのゲノムを利用したほうがよいだろう。

想定外に早いダウンロード所要時間（およびそれに起因する小さいファイルサイズ）、あるいはエラーなくダウンロードできたように思われるものの、なぜか「ファイルが壊れています」という類のメッセージに遭遇するなど、ダウンロード後に何らかのトラブルに遭遇した経験は少なからずあるだろう。疑問に思いつつも 2 回連続で失敗する確率は低いので、大抵の場合再度トライするとうまくいく。うまくいったかどうかの大まかな判断基準は、データ提供側（この場合 Ensembl）のウェブサイトで表示されているファイルサイズと、実際にダウンロードしたファイルサイズの一致であろう［W11-2］。しかし、最も確実かつ一般的な一致の確認の手段は、サイズではなくチェックサムである。

チェックサム（check sum）とは、ダウンロードしたファイルが提供元と同一かどうかをチェックする方法の一種であり、ファイル中のデータから算出されるものである。詳細は省くが、ユーザが行う作業は「ダウンロードしたファイルに対してコマンドを 1 つ実行して得られた値」と

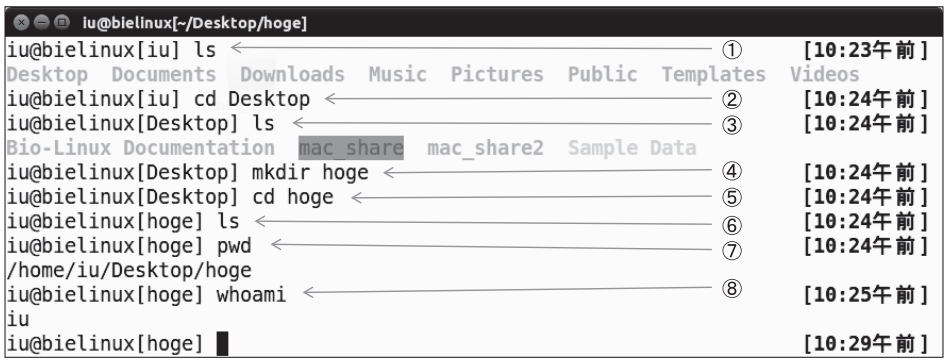

図 1．Bio-Linux 8 のターミナル画面。PC 名は bielinux、ユーザ名は iu。ゲスト OS のデスクトップ上に hoge フォルダを作成する④の mkdir コマンドを含め、計 8 つのコマンドを実行している。

「ファイル提供元が提示している値」が同じかどうかを判定するだけである。それゆえ、実用上「ファイルサイズを自分で計算した結果」と「ウェブ上に記載されているファイルサイズ」の一致を目視確認する作業と似たようなものだという理解で差支えない。ただし複数の算出方式があるため、方法ごとにコマンド名が異なる点に注意が必要である。NGS 解析分野においては、sum コマンドで計算される「チェックサム」と、md5sum コマンドで計算される「MD5 チェックサム」の 2 つを知っておけば十分であろう［W12］。乳酸菌ゲノムデータ提供元である Ensembl は、sum コマンド実行結果のチェックサムを提供している。また、R 経由で FASTQ ファイルをダウンロードする際には、リポジトリ側から md5sum 情報が提供されている。しかし著者らが知る限りにおいて、NGS データリポジトリである DDBJ SRA（以下、DRA），EMBL-EBI ENA（以下、ENA），NCBI SRA（以下、SRA）は、いずれのチェックサムも表示していない。これはおそらくデータ登録者に MD5 チェックサム情報の登録を推奨してはいるものの、義務付けてはいないためであろう。

ファイルの解凍、圧縮、概観

ダウンロードしたファイルは、gzip 形式の圧縮ファイル（.gz）として hoge ディレクトリ中に存在する。gz ファイルの解凍は、gzip コマンドに「-d」（decompress）オプションをつけて実行すればよい。実行結果として、拡張子 .gz がとれた FASTA ファイル（.fa）が得られる。逆に、gzip 圧縮ファイルにしたい場合は「-d」オプションをつけずに gzip コマンドを実行すればよい［W13］。ENA が提供している NGS データは、gzip 圧縮した FASTQ ファイルである。DRA は、bzip2 圧縮（拡張子が .bz2 または .bzip2）した FASTQ ファイル、および拡張子 .sra がついた sra 形式ファイル（以下、sra ファイル）を提供している。SRA は､sra ファイルのみを提供している。sra ファイルは、それ自体が圧縮ファイルである。NGS データに特化したものであるため、圧縮率が非常に高いのが特徴である。NCBI が提供する SRA Toolkit というプログラム群をインストールし、その中の fastq-dump というコマンドを用いれば、sra ファイルから FASTQ ファイルを得ることができる。しかし、.gz や .bz2 のように通常の Linux コマンドで気軽に解凍すれば FASTQ ファイルが得られるわけではなく、bzip2 圧縮ファイルのほうが sra ファイルに比べてファイルサイズが小さく、NGS データ解析時の入力ファイルの業界標準は FASTQ ファイルである。それゆえ、高圧縮率を誇り FASTQ 以外の情報も保持するものの、あえて sra ファイルを取扱う意義は著者らには見いだせない。DRA から bzip2 圧縮ファイルをダウンロードし、bunzip2 で解凍して FASTQ ファイルを取扱うのがスマートであろう。

データ取得および解凍後の次のステップは、ファイル内容の概観である。数千万～数億リードにもおよぶ NGS データファイルは、Win や Mac のテキストエディタで軽快に眺めることが困難である。たとえファイルサイズが 3MB 弱の乳酸菌ゲノムであっても、その行数は数万行になる。それゆえ、grep コマンドを用いたコンティグ数の確認（連載第 2 回の図 4）や、以下に述べる head、tail、more、less コマンドなどを用いたファイルの一部の視認で代用する場合が多い。ここではファイル名を短縮して 1 コマンド 1 行にすることを目的（視認性向上）として、解凍した乳酸菌ゲノムのファイル名を mv コマンドで変更し、genome.fa として取り扱う［W14-1］。

head コマンドは、ファイルの先頭の数行を画面上に表示するコマンドである。一方、tail コマンドはファイルの最後の数行を表示するコマンドである。ファイルの頭（head）と尻尾（tail）なので直観的である。2 つのコマンドともに、デフォルトでは 10 行分が表示されるが、「-n」オプションを利用することで任意の行数を表示可能である［W14-2］。具体的な目的としては、対象のファイルが、利用したい NGS データ解析プログラムの入力形式と同じになっているかどうかの確認が挙げられる。head コマンドで、入力ファイルの最初のほうが同じ形式になっているかどうかの確認はよく行われる。また、最終行の最後の文字の後に改行コードが入っていないために警告メッセージが出ることもある。その場合に tail コマンドでの確認が行われる。

ファイルの全内容を一気に表示させることもできる。数千万行以上になる FASTQ ファイルに用いることはないが、バクテリアゲノムなど数万行程度のファイルであれば、まず wc コマンドでファイルの行数を大まかに把握（数万行なのか数十万行なのかといった程度）しておく。そして、ファイルの全内容を表示する cat コマンドを実行する［W14-4］。著者らは、cat コマンド実行中は、なにか変なところがないかどうか（N が延々と続く領域や、description 行以外で文字数が異なる箇所があるかどうかなど）を確認する視点をもって表示画面を眺める。大丈夫そうだと判断したら「CTRL キー + C キー」を押して実行中のコマンドを中断して次の作業へと進む［W14-5］。他には more や less コマンドなどもよく利用される。特に less コマンドは大規模なファイルでも快適に眺めることができる。more コマンドと同様に「スペースキー」で画面表示されているページから次ページに進むことができるほか、矢印キーでページの上下スクロールもできる。「G キー」を打ち込んでファイル最終ページや、「g キー」を打ち込んでファイル先頭ページへの移動ができる。簡単な文字列検索もできる。例えば「GCCCTTATGA」の文字列を検索したい場合は、先頭にスラッシュ（/）をつけて「/GCCCTTATGA」で検索できる［W14-6］。このような作業は、時間を度外視すれば Microsoft Word など通常の

テキストエディタでももちろん可能であるが、ギガバイトレベルのファイルを取扱う際に快適さを実感できるであろう。

Linux コマンドマニュアルとテキストエディタ

一般に、1 つの Linux コマンドには複数のオプションが存在する。例えば「wc genome.fa」実行結果は、genome.fa の行数、単語数、バイト数を返すだけであるが、「-l」（はいふんえる、と読む；l は line の意味）、「-w」（w は word）、「-c」を付けることでそれぞれの結果のみを返すことができる［W15-1］。他にも「wc -L genome.fa」を実行すれば、デフォルト（つまりオプションなし）で得られる情報以外の出力結果が返される。コマンドごとに利用可能なオプションは、検索エンジン以外にも「*コマンド名* --help」で見ることができる［W15-2］。コマンドによっては「-h」や「--h」も同様な効果をもつ。「man *コマンド名*」とすれば、より詳細な情報を含むマニュアルページを開くことができる［W15-3］。使い方は less コマンドとほぼ同じである。最初のうちは、挙動がよく分かっている wc や ls を man コマンドで眺め、記述形式に慣れておくとよい。

本連載の原稿は、Microsoft Word で作成している。Word に限らず、この種の文章作成ソフトは非常に高機能であり、多少のタイプミスや引用符（クォーテーションマーク）も自動修正してくれる（オートコレクト機能）ので便利である。また、**（R で）塩基配列解析**で公開している本連載の PDF ファイルのように、Linux コマンドをコピー＆ペースト（以下、C&P）で実行すれば、コマンド入力の手間やタイプミスを減らすことができる。このやり方の大きな利点は、たとえ「ls」が「えるえす」なのか「いちえす」なのかわからなくても、コピーさえできればよいという点である。しかし、PDF ファイル中の「grep “>” genome.fa」の C&P がうまくいかないことからもわかるように、PDF, Word, PowerPoint などのファイル中の文字の C&P は、多くの場合 Linux との相性がよくない［W16-1］。正解は「grep ">" genome.fa」である［W16-2］。開始記号と終了記号の形が異なるダブルクォーテーションマークは間違いであり、記号の形が同じほうを用いなければならない。

「""」が「“”」に自動変換されるのは、慣れれば見た目上区別がつきやすいのでまだマシである。「-（ハイフン）」が「–（ダッシュ）」に自動変換されるのは、バイオインフォ中上級者でも比較的厄介な事例である。具体的には、「grep -c ">" genome.fa」は正しく、「grep –c ">" genome.fa」は間違いである［W16-3］。コード作成者が気づかないうちに自動修正されてしまうために、この種の間違いは著者らも無意識に犯している。重要なのは、エンドユーザ側がこのような事例を認識し、対処法を身につけておくことである。

第 2 回で述べたように、バイオインフォマティシャンの多くは、vi や emacs などのテキストエディタを利用して作業を行っている。これらのエディタは、Linux 上での利用を想定したものであるため、上記のオートコレクト問題とは無縁である。また、指定した行番号への移動、特定の行の C&P など、あらゆる編集作業がキーボード操作のみでできる。vi か emacs のいずれかを使いこなせるようになるのが理想であるが、初心者でも直観的に利用可能な gedit からのスタートが現実的かもしれない［W17-1］。著者らは普段、これらのエディタを用いて NGS 解析を行うための一連のコマンドを記したファイル（JSLAB3_code.txt）を作成している［W17-2］。特に wget コマンドの部分は URL 情報がそのまま保存されるので、Ensembl のどのバージョンのデータをダウンロードしたのかまで辿ることができる。これはバイオインフォマティシャンの実験ノートのようなものであるため、データ解析結果の再現性向上にも資する。

リダイレクトとパイプ

第 2 回で触れたリダイレクトやパイプについていくつかの実例を述べる。リダイレクトは、NGS データの解析結果など何かを実行した結果をファイルに保存する目的で主に用いる。例えば「grep ">" genome.fa」は、genome.fa 中の > を含む行（実質的に description 行）を画面上に表示するコードであるが、「grep ">" genome.fa > headers.txt」とすることで実行結果を headers.txt というファイルに保存することができる。引き続いて「wc headers.txt」を実行すれば、headers.txt の行数（つまりこの場合はコンティグ数）を知ることができる。実は「grep -c ">" genome.fa」と同じだが、grep の -c オプションを知らなくても「オプションなしの grep と wc」の組み合わせでコンティグ数を調べることができる［W18-1］。どちらが正解ということはなく、最低限の目的を短時間で達成できればよいだろう。

他のリダイレクト利用例は、サブセットの作成である。生の NGS データは数千万～数億リードからなるが、少なくとも著者らは全データをいきなり入力として NGS 解析の本番（全データの実行）を行うことはない。全リード数の 1/100 ～ 1/10 程度のサブセットを作成し、本番前のプレ解析を行う。head コマンドとリダイレクトを利用してサブセットを作成し、動作確認および全データでの実行時間の目安を立てる。

ただし、NGS データ解析の場合は出力結果のファイルサイズも尋常ではないため、むやみに headers.txt のような中間ファイルを作る習慣は控えたほうがよい。例えば、自分で中間ファイル名を tmp.txt や hoge.txt などに固定して、一通りの解析が終わったらすぐに削除するといったようなことは、多くのバイオインフォマティシャンが実際に

行っていると思われる。サブセットの解析結果ファイルについても同様である。ときどきディスク使用量を df や du コマンドで調べ、効果的に不要なファイルを削除する習慣を身につけておくとよいだろう［W18］。コマンド入力作業はどうみても正しいが、なぜか出力ファイル生成段階でパーミッション（書き込み権限）違反以外の見たこともないようなエラーメッセージに遭遇したときには、HDD 容量に起因する場合が多い。

パイプ（|）は、複数のコマンド同士を組み合わせて利用する際に、中間ファイルの生成を避ける目的で用いる。「grep ">" genome.fa > headers.txt」に引き続いて「wc headers.txt」を行う作業は、パイプを用いたほうがすっきりとする好例である。目的は grep と wc を組み合わせてコンティグ数を調べることであり、中間ファイルに相当する headers.txt の生成ではない。2 つのコマンドはパイプを用いることで「grep ">" genome.fa | wc」のようにまとめることができる。パイプ（pipe）は、文字通り「（左のものを右に流す）配管」であり、左のコマンド（grep ">" genome.fa）の実行結果を右のコマンド（wc）の入力として「引き渡す」機能を果たす［W19］。

NGS 解析現場では様々なファイル形式（FASTQ, FASTA, SAM, BAM, BED 形式など）が存在し、プログラムごとに対応可能な形式が限定されている場合が多い。そのため、BAM から BED のようなファイル形式の変換がよく行われる。パイプは、中間ファイルを生成することなく最終目的のファイル形式のものを得たいときなどに利用される。また、解析プログラム実行時に「入力ファイルの 1005 行目の読み込み時にエラーが発生しました」のようなメッセージが出る場合がある。原因を追究するために該当行周辺を眺める際にも、head と tail をパイプでつなぐことでファイルの両端以外の部分を表示して眺めることができる［W19-3］。具体的には、head で最初の 1010 行分を取り出した結果をパイプで tail の入力として受け渡し、最後の 10 行分を表示させるのである。そうすることで、目的行（この場合 1005 行目）周辺の 1001 ～ 1010 行目をピンポイントで表示できる。

Linux での解析では、オートコレクトや HDD 容量のようなミス以外にも、プログラムが動作せず原因究明に時間がかかる様々な問題に遭遇する可能性がある。オリジナルのコードがパイプを含んでいれば、中間ファイルを生成させて問題が起こっている箇所を特定すべく、意図的にパイプをなくすことも行われる。

公共 DB で乳酸菌 NGS データを眺める

公共 NGS データは、3 大 DB である DRA, ENA, SRA をくまなく探すことを推奨する。検索は、データ総量がおそらく最も多い SRA が最初であろう。例えば「Lactobacillus casei」で検索し、どの NGS 機器（プラットフォームという表現もなされる）でどのようなデータ（ゲノムまたはトランスクリプトーム）が公開されているかの全体像を俯瞰する。ただし、（ユーザごとにストレスを感じるポイントは異なるが）SRA は sra 形式のみでしかデータを公開していないこと、SRX, DRX, ERX という単位で検索結果が示されていて、より大きな単位である SRP, DRP, ERP レベルでの総数が把握しづらい（2014 年 12 月 10 日調べ）。そのため、SRA での検索結果で得た情報をもとにして、DRA 上でも検索することを勧める。DRA にも目的の NGS データが存在すれば、FASTQ 形式の bzip2 圧縮ファイルをダウンロードするか、ENA で FASTQ 形式の gzip 圧縮ファイルをダウンロードするとよいだろう［W20］。

「SRX や SRP の最初の S」は SRA の頭文字である。同様に「DRX や DRP の最初の D」は DRA の頭文字である。S, D, E の違いは、単純に最初にどの DB に登録されたかという程度の理解でよいだろう。3 大 DB 間で統一的な検索結果にならないことからもわかるように、桁違いのデータ量のためかデータの同期は遅れがちなようである。また、2012 年頃までは ID の種類が DRP, DRA, DRX, DRR のようにそれほど多くなかったように記憶しているが、2014 年 12 月現在、PRJDB, SAMD, DRS など新たな ID がつけられるようになっている。DDBJ による解説記事を参考にして、定期的に情報を更新するとよい[7]。ENA は、ID の対応関係や全体像が表形式で俯瞰できるようになっており便利である。著者らは以下のように使い分けて利用している。ただし DRA は、sra 形式と FASTQ 形式の両方を提供しているからか、FASTQ ファイルの提供が若干遅れがちである（例：http://trace.ddbj.nig.ac.jp/DRASearch/study?acc=ERP004457；2014 年 12 月 25 日調べ）。その場合は次善策として ENA を利用するとよい。

SRA：最初の検索時に利用し、目的のデータセットを同定
ENA：目的のデータセットの全体像（ID の対応関係）を俯瞰
DRA：bzip2 圧縮 FASTQ ファイルをダウンロード

「Lactobacillus casei」での検索結果を概観する（2014 年 12 月 10 日調べ）。DRA では、Illumina HiSeq 2000 という NGS 機器で取得されたゲノムデータ（ERP001475 と ERP004457）が得られる［W20-3］。ENA では、Ion Torrent PGM という NGS 機器で取得されたゲノムデータ（DRP000852）が得られる［W20-2］。SRA では、上記データに加えて Illumina MiSeq という NGS 機器で取得されたゲノムデータ（SRP034739）、Illumina HiSeq 2000 で取得された *Lactobacillus casei* A2-362 のトランスクリプトームデータ（SRP017154）と *L. casei* 12A のトランスクリプトームデータ（SRP017156）などが得られる［W20-1］。ただし、検索する場所にも気をつけた方がよい。先程の DRA は Organism という場所での検索結果を述べたも

のであるが、Keyword という場所で検索すると大幅にヒット数が増える［W20-4］。Keyword 欄での検索結果を眺めていくと、SRA の検索結果で認められた *L. casei* 12A のトランスクリプトームデータ（SRP017156）が DRA 上に存在することがわかる。DRA に限らず、このような事例はよく見られる。ただのノウハウではあるものの、検索手段次第で得られる結果が大きく変わるので注意されたい。

乳酸菌 RNA-seq データ取得

Bio-Linux 上で、wget コマンドを用いて *L. casei* 12A の RNA-seq データ（SRP017156）を取得する。ENA で眺めた全体像から、この SRP ID には 2 つの Run accession 番号（SRR616268 と SRR616269）が付随していることがわかる［W21-3］。SRP017156 に対する原著論文、Abstract、および Description がないため詳細は不明であるが、Experiment accession 番号（SRX204226 と SRX204227）の記述内容から、ある程度このサンプルの情報を読み取ることがわかる。つまり、このデータは RNA 鎖の方向性を考慮（stranded）しており、Illumina HiSeq 2000 で取得した増幅 cDNA のペアエンド（paired-end）のリードである［W21-1］。

ペアエンドというのは、断片化した cDNA の 5' 側と 3' 側の両末端からシークエンスするやり方である。対比語であるシングルエンド（single-end）は、片側のみをシークエンスするやり方である。読み取った塩基配列は、リード（read）と呼ばれる。かつては読み取れる配列の長さが短かった（50 塩基未満）ためショートリードと呼ばれていたが、Illumina 社の HiSeq シリーズを含む比較的最近の NGS 機器は 100 塩基以上読めるようになっているため、ショートという形容詞はなくなりつつある。ペアエンドでシークエンスする主な利点は、リードが同じ断片由来であるため、ゲノム配列にマップすると同じ染色体上にマッピングされるはずである。トランスクリプトーム配列にマップする場合には、リードペアが同じ転写物上にマップされるはずであり、マップされたリードペア間の距離が極端に離れることはない。トランスクリプトーム配列がない場合でも、また、アセンブルプログラムを実行することで転写物配列を再構築することができる。最近の動向や詳細については、NGS 関連の文献を参照されたい。

FASTQ ファイルのダウンロードに話を戻す。DRA からペアエンドリードデータを取得する場合、5' 側と 3' 側の 2 つのファイル（例：SRR616268_1.fastq.bz2 と SRR616268_2.fastq.bz2）に分割されているそれぞれの URL 情報を得ておく必要がある。［W21-2］。著者らは通常、データセットごとに 1 つのディレクトリを作成し、そのディレクトリ内で一通りの解析を行う。ここでは、ホームディレクトリ（/home/iu）から見られる Documents に移動したのち、srp017156 というディレクトリを mkdir で作成し、/home/iu/Documents/srp017156 で作業を行う［W22-1］。もちろん、読者自身が分かる場所に好きな名前のディレクトリを作成して作業してもよい。ここでは SRP017156 の一部（SRR616268 のペアエンドデータ）のみダウンロードして取り扱うが、基本的には wget コマンドで bzip2 圧縮ファイルをダウンロードしたのち、bunzip2 コマンド（または bzip2 -d）で解凍するなどして FASTQ ファイルを用意する［W22］。

ゲスト OS に 100GB 程度以上の HDD 容量を確保したユーザは、ここまでの作業を無事に実行できたと思われる。その一方で、PC 本体の HDD 容量などの物理的な制約により 50GB 程度しか確保しなかったユーザは、ファイル解凍時に HDD 容量に関するエラーメッセージが出て失敗しているはずである。仮に SRP017156 のデータ全てをダウンロードしようとすると bzip2 圧縮ファイル状態でも約 30GB に達する（SRR616268 が 15GB、SRR616269 が約 14GB）。一般に、解凍後のファイルサイズは解凍前の数倍になる。実際、SRR616268 の 2 ファイル解凍前後で 15GB から 80GB 程度に膨れ上がっていることからもわかるように、自分が解析したいデータセットが自分のマシンスペックで取り扱えるものかどうかの目途をつけておくことが大事である［W22-4］。本連載は、ノート PC（数百 GB 程度の HDD 容量）で解析を行うユーザ環境を想定している。著者らの推奨戦略は、処理速度は遅いものの高圧縮率を誇る bzip2 圧縮 FASTQ ファイルを提供する DRA のフル活用である。圧縮率は劣るものの処理速度が速い gzip 圧縮 FASTQ ファイルを提供する ENA の利用も一理あるが、DRA はファイルサイズもウェブサイト上に表示しているので、ダウンロードに要する時間や必要な HDD 容量の目途を立てやすい［W21］。

wc コマンドは、リード数を知る目的で利用される［W23］。FASTQ 形式ファイルは、1 リードの情報を 4 行で記述するという決まりがある。それゆえ、wc コマンドで得た行数を 4 で割った値がリード数となる。つまり SRR616268 は、539,023,984 行 /4 = 134,755,996 リードからなる。必要最小限という意味では、SRR616268_1.fastq または SRR616268_2.fastq の片側のみのリード数を調べればよいが、著者らは通常両方を調べる。もしリード数が異なっていれば、「何かがおかしい」と考え、これまで行ってきた作業の検証を行う。この場合はリード数が同じであることから、2 つのファイルサイズ（41GB と 38GB）が異なる理由づけとして、両末端間でのリード長が 41:38 くらいの違いがあるのだろうと想像する。仮に両末端のリード長が同じなら、大抵の場合ファイルサイズも同じだからである。尚、FASTQ ファイルの 1 行目は「@」で始まり、リードの名前や長さなどの description 情報が記載される。2 行目はリードの塩基配列である。3 行目は「+」で始まり、ほとんどの場合何も記載されていないか 1 行目と同じ内容が記載されている。そして、4 行目は 1 文字表記のクオリ

ティスコアが記載されている。詳細はウェブ検索や拙書などを参考にされたい[4]。

次に、head と tail コマンドで FASTQ ファイルの最初と最後の数行を眺める。概ね画面上で一望できる程度の行数にする。ペアエンドデータの場合は、同じリードのシリアル番号（SRR616268.7 や SRR616268.20 など）が表示されているかどうかをチェックする［W24］。DRA の FAQ にもあるが、DRA から得られる FASTQ ファイル中のリード数（=134,755,996）は、DRA のウェブ上で表示されている数およびオリジナルの sra ファイル中の数（=135,073,834）よりも若干少ない［W24-3］。これは、DRA が fastq-dump コマンドを用いて sra ファイルから FASTQ ファイルを生成する際に用いるオプションのうち、主に N が 10 個以上連続するリードを除去する -E オプションの効果によるものである。FASTQ ファイル中の最初のリード（SRR616268.7）と、DRA のウェブ上に表示されている N を多く含むリード（例：SRR616268.1 から SRR616268.6 まで）を見比べれば、DRA が採用しているフィルタリングの合理性が理解できるであろう。

効率的な NGS 解析のために

NGS データ解析を遂行するうえで、解析環境の構築やある程度の HDD 容量の確保は重要なポイントである。そのため、50GB 程度の HDD 容量しか確保できなかった一部の読者は、bzip2 圧縮状態で合計 15GB に達する SRR616268 の 2 つの FASTQ ファイルの解凍段階で断念したかもしれない。しかし、Linux コマンドやパイプを駆使することで、自分が使える範囲の解析環境でもかなりの作業ができる。例えば、約 7GB の *.fastq.bz2 ファイルから約 40GB の *.fastq ファイルを一旦作成しないと wc コマンドで行数を調べることができない、というわけではない。この場合、bzip2 コマンド実行時に、解凍を意味する -d オプションの他に、元の *.fastq.bz2 ファイルを残したまま解凍した結果を標準出力（画面上に出力）する -c オプションをつければよい。これだけだと約 5.4 億行分の FASTQ ファイルの情報が画面上に出力されてしまうので、パイプで wc コマンドの入力として受け渡すのである。こうすることで、約 40GB の *.fastq ファイル作成を回避しつつ、圧縮ファイルの内部情報を自在に取り出すことができる［W25］。

数十 GB、数億行レベルの FASTQ ファイルに対する Linux コマンドの実行は、十数分から数十分以上かかる。解析プログラムは、ものにもよるが数時間から数日というオーダーである。それゆえ、解析したいファイルの処理時間が数時間程度で終わるのか、それとも数週間を要するものなのかを、ファイルサイズを含め大まかに見積もっておくことが大切である。全リード数の 1/100 ～ 1/10 程度のサブセットを作成して動作確認を兼ねたプレ解析を行うが、このときに総リード数が約 1.35 億程度であることを DRA のウェブサイトで事前に把握しておけば、wc コマンド実行の必要はない。そして、例えば 100 万リード（1 リードで 4 行なので 4,000,000 行分）のサブセットであれば非圧縮状態でも 0.5GB 程度に収まるという大まかな見積もりを行ったのち、複数の Linux コマンド、リダイレクト、およびパイプを組み合わせた「bzip2 -d -c SRR616268_1.fastq.bz2 | head -n 4000000 > subset_1.fastq」のようなコードを実行するのである［W25-2］。そうすれば、指定したリード数からなるサブセットのファイルサイズ分しか HDD 容量が増加せず、十数秒足らずで処理が終わる。中間ファイルである SRR616268_1.fastq を一旦作成する手順［W22-5］に比べて、圧倒的に有利なことがわかる。

第 3 回は、Bio-Linux の導入や各種環境設定、Linux コマンドの復習から拡充を経て、乳酸菌 NGS データの取得までを述べた。バイオインフォマティクスの真骨頂は時間の短縮である。DRA の情報を有効利用し、HDD 容量を意識しながら戦略を立て、複数のコマンドとオプションを組み合わせれば、解析の幅が一気に広がる。できないことができるようになるといっても過言ではない。次回以降は、シェルスクリプト、OS 間での改行コード問題への対処、プログラムのインストールなどの Technical Tips を織り交ぜながら、NGS データのクオリティコントロール、RNA-seq リードの乳酸菌ゲノムへのマッピング、数値解析へと進んでいく予定である。

謝　辞

本連載の一部は、独立行政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。

参 考 文 献

1） 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.
2） 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.
3） 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.
4） 門田幸二（2014）1.3 RNA-seq，p.8–27，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.
5） Flicek P, Amode MR, Barrell D, Beal K, Billis K, et al. (2014) Ensembl 2014. Nucleic Acids Res., **42**, D749–D755.
6） 坊農秀雅（2014）第Ⅰ部 第2章 1. Ensembl，p.65–67，実験医学増刊 今日から使える！ データベース・ウェブツール，内藤雄樹 編，羊土社，東京.
7） 児玉悠一，真島淳，高木利久，中村保一（2014）NGS データを公共データベースへ登録する，p.387–395，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，二階堂愛 編，羊土社，東京.

Methods for analyzing next-generation sequencing data III. From setting a Linux environment to manipulating Lactobacillus RNA-seq data

Jianqiang Sun[1], Aya Miura[2], 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

There are many prerequisites for NGS data analysis on Linux system. We describe basic usage and commands on Linux environment. This includes the setting of Bio-Linux as a virtual machine, shared folders, and understanding of hardware requirements for NGS analysis. We show how to get a Lactobacillus genome sequences on Ensembl database and transcriptome data (bzipped FASTQ files) on DDBJ SRA. Many tasks on NGS analysis can effectively be performed by combining some Linux command line facilities, such as pipe and redirection.