Japanese Journal of Lactic Acid Bacteria

総　説

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

孫 建強[1]、湯 敏[1]、西岡 輔[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

実験系研究者が普段利用する Windows や Macintosh の PC 環境は GUI 環境と呼ばれ、マウス操作などで直観的な利用が可能である。一方、バイオインフォマティクス系研究者が普段利用するデータ解析環境は、コマンドライン環境である。次世代シーケンサー（以下、NGS）データ解析用プログラムの多くは Linux 上で動作し、コマンドライン環境での利用を想定している。それゆえ、コマンドライン環境に慣れ、基本的なコマンドを使いこなせるようになることが NGS データを自在に解析するための前提条件といえる。連載第 2 回では、Windows および Macintosh 付属のコマンドライン環境（コマンドプロンプトおよびターミナル）での基本的な操作を GUI 環境と対比させながら示し、最低限必要だと思われるコマンドや概念を解説する。前回同様、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）中に本連載で述べるリンク先を掲載してあるので効率的に活用してほしい。

Key words：NGS, Command prompt, Terminal, Bioinformatics, commands

GUI 環境とコマンドライン環境

NGS 解析を行う上で理想的なデータ解析環境は、「京」や遺伝研スパコン[1)]である。特に遺伝研スパコンは、頻繁にメンテナンスがあるものの NGS 解析に適した構成となっており[2)]、ある程度 Linux に慣れたエンドユーザが比較的気軽に利用可能なスーパーコンピュータである。ただし、他のユーザに迷惑がかからない程度の Linux の使い方は知っておく必要がある。第 1 回でも述べたように、覚えるのは大変であるが Linux コマンドを駆使することで効率的な NGS データ解析が可能となる。Macintosh（以下、Mac）ユーザの場合は、「ターミナル」を起動することで Linux コマンドを利用可能である。ターミナルは、「アプリケーション」-「ユーティリティ」中に存在する。Windows（以下、Win）で同様な役割を果たすものとしては、「コマンドプロンプト」が挙げられる。コマンドプロンプトは、「スタート」-「全てのプログラム」-「アクセサリ」中に存在する。

多くの実験系研究者は、普段マウスやタッチパッド（ノート PC のキーボード手前にある指でカーソルを動かすもの）を利用して PC 作業を行っている。Win や Mac の GUI 環境での作業の具体例として、第 1 回の最後に入力として用いた *Lactobacillus casei* 12A 株の FASTA 形式ファイル（"Lactobacillus_casei_12a.GCA_000309565.1.22.dna.toplevel.fa"）をデスクトップ上の hoge フォルダ内に置く作業を解説する。Ensembl のウェブサイトから一時的な保存場所としてデスクトップにダウンロードした直後は、拡張子 .gz がついた gzip 形式の圧縮ファイルとしてデスクトップに存在する。著者の Win 環境では、Lhaplus というフリーの圧縮 / 解凍ソフトウェアがインストールされている。圧縮ファイルをマウスでダブルクリックする

*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

ことで、拡張子 .fa がついた目的の FASTA 形式ファイル（以下、FASTA ファイル）を得ることができる。ただし、FASTA ファイルは hoge フォルダに自動で入ってくれるわけではないので、ユーザは FASTA ファイルを hoge フォルダにドラッグアンドドロップ（マウスでつかんで hoge フォルダ上で離す作業のこと）することで上記目的を達成する。

ユーザが Win などの PC との情報のやりとりをするための接点のことをユーザインタフェース（User Interface; UI）という。ユーザは普段、hoge フォルダに FASTA ファイルが置かれていく様子を、マウスを動かしその挙動をディスプレイ上で見ながら作業している。この方式を GUI（Graphical UI）という。普段 Win や Mac で行っていることは GUI での作業である。一方、Mac のターミナルや Win のコマンドプロンプト上で行う作業は、基本的にキーボード経由での文字の打ち込みである。それゆえこの方式を CUI（Character UI または Console UI）または CLI（Command Line Interface）という。日本では CUI という表現がよく使われるようであるが、GUI 環境の対比的な用語はおそらくコマンドライン環境である。GUI 環境での作業に慣れているユーザにとっては、コマンドライン環境に最初は戸惑うかもしれない。コマンド（呪文）を知らなければ何も作業をすることができないからである。しかし、NGS データ解析を自在に行う上で Linux を使いこなせるにこしたことはない。そのための第一歩は、コマンドライン環境に慣れ、基本的なコマンド群を使いこなせるようになることである。

Win のコマンドプロンプト

デスクトップ上の hoge フォルダ中に上記 FASTA ファイルが 1 つだけ存在するという前提のもと、コマンドライン環境でそのファイルを見にいく作業を解説する。ここではユーザ数の多い Win のコマンドプロンプトを用いてコマンドライン環境を説明するが、後述する Mac のターミナルも基本的な概念は同じである。共通するコマンドもあるため、Mac ユーザもスキップせずに一通り読んでほしい。

ユーザ名 kadota でログインしている状態でのコマンドプロンプト起動直後は、「C:¥Users¥kadota>」となっていることがわかる（**図 1**）。もしユーザ名が dokusha なら、起動直後は「C:¥Users¥dokusha>」のように見えているであろう。フォントの問題でバックスラッシュ（\）と円マーク（¥）の違いはあるものの、基本的には見栄えだけの問題であり、文字コードとしては同じなので気にする必要はない。「C:¥User¥kadota>」の C:¥Users¥kadota は、現在の作業ディレクトリが「C ドライブ – ユーザー – kadota」という場所であることを意味する。フォルダとディレクトリは、正確には異なる概念のものであるが、実用上は同じものを指すという理解で差支えない。読者は、フォルダやディレクトリのことを単純に「場所」と読み替えればよい。

NGS 解析に限らず、コマンドライン環境では自分が現在どこで作業をしているかを正確に把握しておくことが重要である。現在の作業ディレクトリは、**図 1** を眺めるとユーザ kadota の直下であることがわかるが、作業ディレクトリを表示するコマンド cd も存在する［ウェブ資料 1］。通常、コマンドプロンプト起動直後の作業ディレクトリのことを「ホームディレクトリ」と呼ぶ。この場合、「C ドライブの、Users、kadota がホームディレクトリ」という言い方をする。cd コマンドは「ここはどこ？」の結果であるが、「私は誰？（Who am I?）」に対応する whoami コマンドも存在する［ウェブ資料 2］。この結果は「kadota-pc¥kadota」となっており、このノート PC の名前は kadota-pc、ログインユーザ名は kadota と読みとる。

作業ディレクトリに何があるかを調べたい場合には、dir コマンドを実行する。これは、GUI 環境において「C ドライブ – ユーザー – kadota」のフォルダを開いて何があるかを眺める作業と同じである。この PC の日本語 GUI 環境では、カタカナの「デスクトップ」フォルダが見られる［ウェブ資料 3］。その一方で、dir 実行結果のコマンドライン環境では、英語の「Desktop」になっていることが分かる［ウェブ資料 4］。

作業ディレクトリの移動は、cd コマンドを実行する。作業ディレクトリにて dir コマンド実行結果［ウェブ資料 4］で見られるフォルダに移動したい場合には、「cd フォルダ名」とすればよい。ここで、cd とフォルダ名の間に

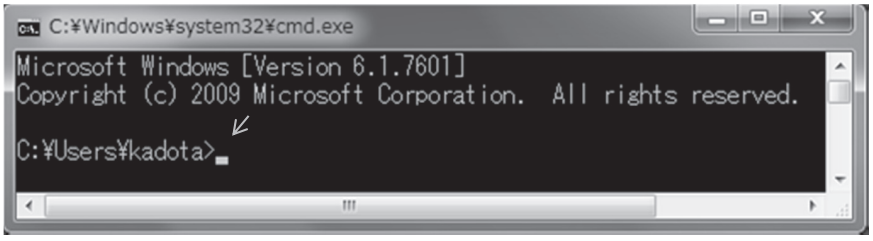

図 1. Windows 7 でのコマンドプロンプト起動直後の初期画面。コマンドプロンプトは、上記画像のコマンドライン環境そのものを指す以外に、灰色矢印で示した白い点滅カーソルの左側の「大なり記号（>）」のことも指す。

は半角スペースが存在する。ウェブ資料5は、① cd で作業ディレクトリを確認、②「cd ですくとっぷ」と打ち込んでエラーとなり、③「cd デスクトップ」でもエラーとなることを確認し、④「cd Desktop」と打ち込んで作業ディレクトリが変更されたことをコマンドプロンプトの位置でも認識し、⑤ cd で念のため確認をした、という一連のコードおよび実行結果である。なぜひらがなやカタカナではだめで英語の Desktop としなければならないのか？それはコマンドライン環境での dir 実行結果が英語の「Desktop」になっているからである。ではなぜ GUI 環境では日本語で見られるのか？それは Windows のシステム言語が日本語となっているからである。仮にシステム言語を英語に変更すれば GUI 環境でも英語で表示される。

ここで、当初の目的（コマンドライン環境で hoge フォルダ中の FASTA ファイルの存在確認）を再確認する。現在の作業ディレクトリはデスクトップである。多少冗長ではあるが、① dir で hoge フォルダがあることを確認、②「cd hoge」で作業ディレクトリの変更、③ dir で目的の FASTA ファイルが見られることを確認、という手順になる［ウェブ資料6］。Win や Mac の GUI 環境での作業は、デスクトップ上の hoge フォルダをダブルクリックし、視点をそこに合わせて FASTA ファイルが確かに存在することをファイル名から認識することに相当する［ウェブ資料7］。おそらくほとんど全ての読者は PC についての基本的な知識や技能（コンピュータリテラシー）が備わっているため、「デスクトップにある hoge フォルダをダブルクリック」というだけで必要な作業を実行できるであろう。たとえメールソフトやウェブブラウザを画面いっぱい起動していたとしても、無意識にマウスポインタをタスクバー上に移動させ、そこで右クリックして「デスクトップを表示」を選んでデスクトップに存在する hoge フォルダが見られる状態にするであろう。「Win キー + D」というショートカットキーを知っているユーザもいるかもしれない。Mac の場合は、同様な機能として「Fn + F11」を利用可能である。

PC 初心者にとっては、たとえ hoge フォルダを視認できたとしても、マウスポインタを hoge フォルダ上に移動し、マウスの左側のボタンをすばやく2回押す（クリックする）という一連の操作説明の用語は意味不明であろう。マウス、ポインタ、クリック、ドラッグ、フォルダなど、GUI 環境で普段仕事をしているユーザも、無意識のうちに初心者にとっては意味不明な用語を駆使して PC 上で作業を行っている。ここまで用いたコマンド群（cd, whoami, dir）のコマンドライン環境での実行ですら、GUI 環境でしか作業していなかったユーザにとっては苦行かもしれない。しかし、最初は入力スピードが遅いキータイピング練習のようなものであり、そのうち指が慣れてくるので是非チャレンジしてほしい。尚、「作業ディレクトリ」という呼び名は、Linux 環境ではあまり一般的ではない。現在のディレクトリという意味で、「カレントディレクトリ（current directory）」と呼ばれることが多い。「作業フォルダ」や「ワーキングディレクトリ」という用語も散見されるが、いずれも同じ意味合いで用いられる。

Mac のターミナル（基礎編 1）

次に、さきほどと同じくデスクトップ上の hoge フォルダ中に FASTA ファイルが1つだけ存在するという条件下で、Mac 環境でそのファイルを見にいく手順を解説する。Mac のターミナルも、基本的な見栄えは Win と同じである。ターミナル起動直後は、ユーザ名 kadota のホームディレクトリ（~）がカレントディレクトリである［ウェブ資料8］。Win のコマンドプロンプト「>」に相当するものが「$」であり、カーソルが点滅してコマンド入力待ち状態となっている点も Win と同じである。しかし、同じ作業を実行する上でも、Win と Mac ではコマンド名が異なる場合がある点に注意が必要である。

結論を先に述べると、Mac のコマンドを覚えるほうが有意義である。NGS データを効率的に解析できる環境は Linux であり、Mac のターミナルで用いるコマンドは Linux コマンドそのものだからである。例えば、カレントディレクトリ表示コマンドは、Win が cd であるのに対して、Mac（つまり Linux）では pwd である［ウェブ資料9］。コマンド名が異なっている理由は、おおまかには OS や規格の違いによる。Win は MS-DOS（Microsoft Disk Operating System）の流れを汲んでいる。MS-DOS のことを「えむえすどす」と一続きで発音してもピンとこないかもしれないが、マイクロソフト社が昔開発・販売していた OS の略であると解釈できれば納得するであろう。1995年に販売された Windows 95 という OS より前は、コマンドライン環境での作業が主流であった。主にその時代に用いられていたコマンド群が、今でもコマンドプロンプトを起動すれば利用できるのである。

Mac でターミナルが標準搭載されるようになったのは、2001年に販売された現行 OS である OS X（おーえす てん、と読む）以降である。この OS は、Linux と同じく UNIX（ゆにっくす、と読む）と呼ばれる OS をベースとしている。そのため、UNIX システムで利用可能なコマンド群の多くは、Linux および OS X でも継承されている。Linux が UNIX 系システムや UNIX の一種と表現されるのはそのためである。UNIX や UNIX/Linux などという用語も散見されるが、本連載では Linux で統一している。

Mac（Linux）でカレントディレクトリの表示を行う pwd コマンド実行結果は「/Users/kadota」であることがわかる［ウェブ資料9］。Win のコマンドプロンプトでカレントディレクトリの表示を行う cd 実行結果と若干異なり、Mac では C ドライブに相当する記述「C:」が消えていることがわかる。しかし、C ドライブという概念は Win

特有のものであるため気にする必要はない。Win の cd コマンドは、i）カレントディレクトリの表示と ii）ディレクトリの変更の 2 つの役割を果たしていた。それに対して Mac の cd コマンドは、ii）のディレクトリの変更のみの役割を果たす。Mac のターミナル上で「cd」のみを打ち込む作業は、「ホームディレクトリへの移動」を意味する。ターミナル起動直後は、カレントディレクトリがホームディレクトリである。それゆえ、ウェブ資料 9 で cd 実行後に何も起こっていないように見えるのは、ホームディレクトリにいる状態でホームディレクトリへの移動を行ったことに相当し、見かけ上ディレクトリの移動が行われていないからである。「私は誰？」コマンドは、Win と同じく Mac でも whoami である［ウェブ資料 10］。Win の結果は「PC 名 ¥ ユーザ名」であったが、Mac の結果は「ユーザ名」のみが表示されている。この場合は「kadota」である。このように、たとえ同じコマンド名であっても出力結果が若干異なる場合もあるので注意されたい。

カレントディレクトリに何があるかを表示する ls コマンド実行結果は、Win の dir コマンド実行結果よりもシンプルである［ウェブ資料 11］。ls の l は、数字の 1（いち）ではなくアルファベットの l（える）なので注意されたい。つまり「えるえす」である。著者らも全ての Linux コマンドを知っているわけではないが、経験上 Linux コマンドは小文字のアルファベットで構成されるという理解で差支えない。カレントディレクトリがホームディレクトリでの ls コマンド実行結果に話を戻す。Desktop, Documents, Downloads などは Win の dir 実行結果でも見られ、ホームディレクトリの基本構成は OS 間でそれほど違わないことがわかる。

Win や Mac の GUI 環境では、フォルダ上部のオプション変更によって、更新日時、ファイルサイズや種類を表示させることができる。更新日時でソートさせるテクニックは、フォルダ内のファイル数が多い場合に頻用しているであろう。Win の dir 実行結果は、更新日時やディレクトリかどうか（<DIR> の有無）の詳細情報がデフォルトで表示される。それに対して Mac の ls 実行結果は、ディレクトリやファイルの名前のみである。もちろんこれは ls のデフォルト実行結果にすぎず、「ls -l」のように ls コマンドに引き続いて指定する様々なオプションを駆使すれば、表示形式を自在に変更可能である［ウェブ資料 12］。このオプションつきコマンドは「えるえす（すぺーす）はいふんえる」と読む。上記の l（える）オプションは、詳細情報を表示するためのオプションであり、long の頭文字の意味が込められている。メインのコマンド（この場合 ls）を打ち、スペースを入れてからハイフンで始まる任意のオプション（この場合 -l）を指定する記述形式は、Linux 環境で動作する多くの NGS 解析用プログラムや Linux コマンドを実行する上での基本である。

ls コマンドの他のオプション使用例を紹介する。a オプションは、隠しファイルを含む全てのファイルを表示したいときに利用する［ウェブ資料 13］。主に、「.」から始まるファイルを表示する目的で使用される。この .（どっと）から始まるファイルには環境設定ファイルなどが含まれる。Win や Mac の通常利用環境下では特に意識する必要はないが、NGS 解析を Linux 環境下で本格的に始める場合には環境設定ファイルの中身を変更するテクニックを身につけておいたほうがよい。ls コマンドや a オプションは、環境設定ファイルをコマンドライン環境で視認する最低限のリテラシーである。

バイオインフォマティクス分野の常識・非常識

コマンドライン環境初心者がよく犯すミスは、適切な場所への半角スペースの入れ忘れである。例えば、「ls -a」と打っているつもりで「ls-a」と打つと、ls と -a の間に半角スペースが適切に挿入されていないため、ls-a というコマンドとして認識される。当然のことながら「そのようなコマンドはない（command not found）」といわれる［ウェブ資料 13］。この程度であれば明確にエラーと認識できるので対処のしようはある。コマンドライン環境では、半角スペースは明確な意味を持つ場合が多い。それゆえ、NGS データ解析に限らず、解析したいファイルを入力として与える際にファイル名の中にスペースを入れるのは、コマンドライン環境中心のバイオインフォマティクスの世界では非常識である。他に「全角文字」や「!, “, #, $ などの英数字以外の文字」も忌避される。一般に多数のファイルが 1 つのディレクトリ内に存在する場合、意味を持たせたファイル名にすることが多い。例えば、group1_rep1.fa, group1_rep2.fa, group2_rep1.fa, group2_rep2.fa のような具合である。この場合、上記のようなブラックリストを眺めるのではなく、使っても大丈夫という経験に基づくホワイトリストを利用するほうが手っ取り早い。例えば、著者らは主に「xxx_yy_zzzz_001.fa」のような英数字とアンダースコア（_）の組合せを利用する。

ファイルの拡張子にも気をつけたほうがよい。上記 FASTA ファイルの拡張子は .fa であったが、.fasta でもよい。拡張子が .fa か .fasta であれば、常識的な範囲で他はなんでもよいという意味合いで＊.fa や＊.fasta という表現もなされる。塩基ごとのクオリティ情報を含む FASTQ 形式ファイル（以下、FASTQ ファイル）の拡張子は、＊.fq または＊.fastq が一般的である。もちろんそれ以外の拡張子でも受け入れてくれる NGS 解析用プログラムは存在するかもしれない。例えば、オプションで FASTQ 形式だということを宣言さえしておけば、実際の拡張子は .txt でも .doc でも .pdf でも受け入れてくれるかもしれない。しかし著者らは、＊.doc や＊.pdf での動作確認（ブラックリスト作成）には関与しない。無難な＊.fq や＊.fastq を素直に利用する。

NGS 解析を効率的に行うスキルを身につける上で、バイオインフォマティクス分野の常識・非常識の感覚を知ることは重要である。現役バイオインフォマティシャンの多くは、体系的なバイオインフォマティクス教育を受けていない。研究室の先輩から得られる情報は、「こうすればうまくいった」、「私はこのプログラムを使っている」、「これは普通やらない」という経験談がほとんどである。そして、C でも Perl でも R でもなんでもよいが、ある特定の目的を達成するためのプログラムを毎回最初から作っているバイオインフォマティシャンは、おそらく皆無である。比較的簡単なプログラムであっても、先輩からもらったものや自分の過去のプログラム群の中から目的に近いもの（例：template.pl）を選びだし、それをコピーして新たなファイル名（例：template_jibun.pl）として一旦保存する。その後、多くのコマンドライン環境で利用可能な vi（ぶいあい、と読む）や Emacs（いーまっくす、と読む）などの高機能エディタを駆使し必要最小限の箇所を変更することで、新たな別のプログラムとしての機能を持たせるのである。ときどき最低限必要な Linux コマンドについての問い合わせを受けるが、回答は難しい。例えば、上記のようなファイルのコピーを行う cp コマンドなどは基本中の基本だと思っているが、そのコマンドを使うことなく特定の NGS 解析を行うことは可能だからである。また、一口に NGS 解析とはいっても、ゲノム解析、トランスクリプトーム解析、エクソーム解析など応用分野は多岐にわたる。例えば、著者らにとって未知の領域である NGS を用いたメタゲノム解析分野においては、どのような Linux 系スキルが必要とされるかについてのコメントは難しい。また、バイオインフォマティクス初級、中級、上級といったカテゴリー分けも、あってないようなものである。初級とはこうあるべきという基準も人それぞれであり、結局は所属するボスとの相性がほとんど全てである。

今日では、多くの研究者がツイッターやブログなどで NGS 解析を効率的に行うための自分の経験談やノウハウを公開している。バイオインフォマティクス分野のユーティリティープレイヤーとして生き残る重要な資質は、むやみにヒトに聞かずに自分で問題解決を試みる姿勢であろう。ウェブ検索を行う（通称、ググる）ことで、特定の目的を達成するための様々なやり方を入手することができる。それゆえ、「この目的を達成するためにはどうすればいいですか？」のような一から教えてもらう姿勢は疎まれる場合が多い。一通り自分で調べて得た解析手順についての見解や、ウェブ検索でも解決できなかった問題を QA サイト（ライフサイエンス QA、SEQanswers[3]、Biostar[4]）などで聞くのが本来あるべき姿であろう。

現役のバイオインフォマティシャンは、自力での Linux 環境構築ができるなどコマンドライン環境での作業が苦も無くできる資質を持っていたか、あるいはその苦行を乗り越えたヒトが多い。そのためかどうかは不明であるが、Linux コマンド名やコマンドライン環境に慣れたヒトにしかわからない専門用語を無意識に使って会話をする傾向にある。これが、専門外のヒトがよく漏らす「バイオインフォ系のヒトは何をしゃべっているのか分からない」という感想を引き出すのである。すでにいくつかの Linux コマンドやコマンドライン環境特有の単語が出ているが、コマンドライン環境に慣れれば会話がだんだん成立するようになる。また、これらの用語やコマンドを知っていれば、これまで解読不可能であったウェブ上の多くの NGS 関連情報を活かすことが可能となる。ここまで、身近な Win と Mac 上で GUI 環境と対比させながらコマンドライン環境を体験し、NGS 解析を行う上で最低限必要だと思われる Linux コマンド利用の基本形、および情報収集に対する基本的な心構えを述べた。引き続き基本コマンドの解説を行うが、これらを土台として、パスの概念、任意のプログラムのインストール、シェル、外部サーバへのアクセスなど、ノート PC での本格的な Linux 環境上での作業へと移行し、自前サーバ構築やスパコン上での作業へとステップアップしてゆく。

Mac のターミナル（基礎編 2）

基礎編 1 の最初に述べた、本来の目的であるデスクトップ上の hoge フォルダ中に存在する FASTA ファイルを視認するやり方に話を戻す。Desktop の下の階層に hoge が存在するということが既知であれば、ホームディレクトリがカレントディレクトリの状況下でも「ls Desktop/hoge」と打ち込むことで目的の FASTA ファイルを視認することができる［ウェブ資料 14］。ターミナル起動後の状態から① pwd, ② cd, ③ whoami, ④ ls, ⑤ ls-a, ⑥ ls Desktop, ⑦ls Desktop/hoge を打ち込んだ一連の結果を**図 2** に示す。手元に Mac のない Win ユーザも実際の操作のイメージがつかめるであろう。

ここで、Tips（秘訣や裏技、的な意味）を 3 つ紹介する。落穂拾い的な位置づけではあるものの、PC 上での作業効率を上げるものであり、この段階から使いこなせたほうがよい。1 つ目は Tab キー（以下、Tab）による補完機能の利用であり、通称「タブ補完」と呼ばれるものである。これまで出てきた程度のコマンドであれば、コマンドを見ながら手で打ち込むことは容易である。一般にディレクトリ名やファイル名が長くなるほど打ち間違う確率は上昇するが、タブ補完を利用すればタイプミスを大幅に減らすことができる。コマンドライン環境でこの機能を利用していないバイオインフォマティシャンはおそらく一人もいない。異次元の速さでキー入力しているのではなく、単純にタブ補完を利用して効率的に作業を行っているだけである。
タブ補完の基本的な利用法は、目的のディレクトリ名やファイル名の最初の数文字を打ち込んでから Tab を押すだけである。例えば、**図 2** の 6 番目のコマンド「ls

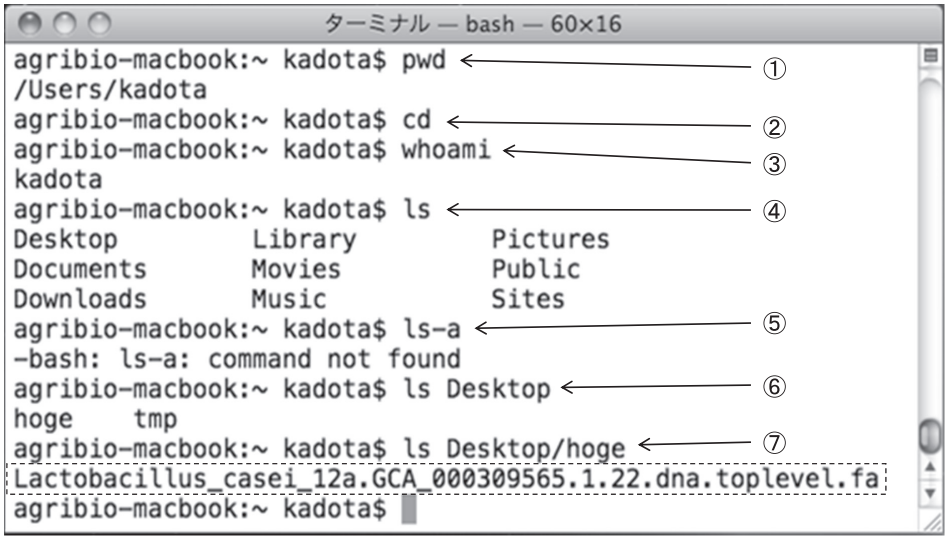

図 2. MacBook のターミナル画面。PC 名は agribio-macbook、ユーザ名は kadota。計 7 つのコマンドを実行した結果を示している。

Desktop」は、「ls De」まで打ち込んでから Tab を押すと「ls Desktop/」と残りの文字を自動的に補完してくれるので、後はリターンキー（Enter キー）を押すだけでよい［ウェブ資料 15］。もちろん「ls Des」や「ls Desk」後に Tab を押すのでもよいが、キー入力の労力を減らすことを目的としたタブ補完の趣旨にむやみに反することはないだろう。次に、「ls D」後に Tab を押してみよう。コマンド実行結果に相当する部分に「Desktop、Documents、Downloads」が表示されており、一見タブ補完がうまく機能していないように見える。しかし、**図 2** の④「ls」のみの実行結果からわかるように、大文字の D から始まるものは上記 3 つのみである。つまり、コマンド実行結果に相当する部分に表示されているものは、D から始まる候補をリストアップするとともに、次に何か目的の名前をタブ補完できる最低限のキー入力を促しているのである。今は Desktop の中身を表示させることを目的としているので、他の 2 つの D から始まる候補と区別する最小限の文字は「De」である。それゆえ最初に述べたように「ls De」後に Tab を押せばよいのである。同様にして、Documents の中身を表示させたい場合には、「ls Doc」まで打ち込めば最後までタブ補完してくれる。「ls Do」後に Tab では、Do から始まる 2 つの候補（つまり Documents と Downloads）が表示される［ウェブ資料 15］。

2 つ目の Tips は、「上下左右の矢印キー」の利用である。通常キーボードの右下のほうに配置されている矢印キーの「上向き矢印」を 1 回押すと、1 つ前に打ち込んだコマンドが表示される。例えば、2 つ前に打ち込んだコマンドが「ls Desktop/」であり、今入力したいコマンドが「ls Desktop/hoge」なら、「上向き矢印」を 2 回押せばよい。そうすると「ls Desktop/」が現れるので「hoge」を追加入力するだけでよい。もちろん「ls Desktop/h」の状態からのタブ補完でもよい。3 つ目の Tips は、「ヒストリー機能」の利用である。これは「上下左右の矢印キー」利用の補完的な位置づけという理解でもよい。「何回前だったか忘れたが確か前に打ったコマンド」を探す場合には、「上向き矢印」を何回か押してゆき、行き過ぎれば「下向き矢印」で戻ることで目的のコマンドを探すのが基本である。しかし、常に 1 コマンド分しか表示されないのを不便に感じるときもあろう。ヒストリー機能の実体である history コマンドを実行すると、直近のコマンド数十個を表示してくれるので、その中から目的のコマンドを選択実行することができる［ウェブ資料 16–17］。

一部の読者は、このような Tips は無駄であり NGS 解析に必要最小限の情報のみ教えてもらえばよいと思うかもしれない。しかし NGS 解析用の多くのプログラムは多数のオプションからなり、Linux の「リダイレクト」という出力先を変更する機能や、「パイプ」というコマンド同士を組み合わせる機能を併用するのが一般的である。例えば、今自分がどこのディレクトリ上で作業をしていて、どこにあるファイルを入力として用い、どのディレクトリ上にあるプログラムをどういうオプションを用いて実行し、結果ファイルをどういう形式でどのディレクトリ上にどういう名前で保存するかなどの情報を一行のコマンドで記述する。当然ながら、どこか 1 か所でもミスがあるとエラーが出ることが多い。1 つの実行コマンドではあるものの、長ければターミナル上では複数行にわたって折り返し表示されるコマンド中のたった一か所を修正するだけのために、「上下左右の矢印キー」を利用しないのは労力および時間の無駄である。もちろん結果に本質的な影響を与えない出力ファイル名の打ち間違い程度であれば、（特に計算に時

間がかかる場合には）普通はもう一度実行することはせず、mv というコマンドを用いてファイル名の変更で対応する。基本的な利用法は、スペースが 2 か所にある「mv 変更前のファイル名 変更後のファイル名」であるが、通常は mv のあとのファイル名入力の際にタブ補完を利用する。本連載の想定読者は、全く試行錯誤やミスをせずに一通りの NGS 解析ができる完璧な研究者ではない。著者らは、通常利用する Linux コマンド群や Tips の積み重ねこそが、最小限の労力で自在に NGS 解析を行う一番の近道というスタンスである。

Mac のターミナル（基礎編 3）

排他的なものも存在するが、一般に Linux コマンドのオプションは組合せることができる。例えば、ディレクトリやファイルの情報を表示する ls コマンドは、全ファイルを表示する a オプションと詳細を表示する l オプションを「ls -la」として同時に利用可能である［ウェブ資料 18］。「ls -l」実行結果のウェブ資料 12 と比較することで、a オプションの有無による違いがよくわかる。尚、オプションの順番は特に気にする必要はない。例えば、「ls -al」や「ls -a -l」のいずれでもよい［ウェブ資料 19］。

ここまでは、カレントディレクトリがホームディレクトリの状態で、Desktop 上の hoge フォルダ中の FASTA ファイルを「ls Desktop/hoge」で表示させる作業を行った。しかし著者らは通常、ls コマンドのオプション（引数ともいう）としてディレクトリ名を指定することはせず、カレントディレクトリの変更を行う cd コマンドを利用して目的のディレクトリ（この場合 hoge）まで移動する。前述のように、Mac のターミナル上で「cd」のみを打ち込む作業は、ホームディレクトリへの移動を意味する［ウェブ資料 9］。cd コマンド実行時に、引数として実在するディレクトリ名を指定することで、任意のアクセス可能なディレクトリに移動することができる（**図 3**）。例えば、①カレントディレクトリがホームディレクトリ（つまり /Users/kadota）の状態で、「cd De」まで打ったのち Tab を押すことで②「cd Desktop/」となる。③ pwd で確認すると、確かに Desktop にディレクトリ変更できていることがわかる（つまり /Users/kadota/Desktop）。④ ls 実行結果は、**図 2** の⑥と同じである。⑤「cd hoge/」実行後に⑥ pwd で確認すると、確かに hoge にディレクトリ変更できていることがわかる（つまり /Users/kadota/Desktop/hoge）。⑦ ls の結果は、**図 2** の⑦と同じである。この ls, cd, pwd という一連のコマンド入力は、慣れてくると無意識に利用するようになるだろう。

前述の意味を持たせたファイル名の好例がこの FASTA ファイルである。異様に長いと思うかもしれないが、ファイル名を見れば乳酸菌ゲノムであること（Lactobacillus_casei_12a）、Ensembl Genomes のバージョンなどの情報（GCA_000309565.1.22）、マスクされていない DNA 配列であること（dna；マスクされているものは dna_rm となっている）、Toplevel のものであること（toplevel）などが分かる。mv コマンドおよびタブ補完機能を用いて効率的に genome.fa などにファイル名変更することはできるものの、むやみにファイル名変更しないほうがいいかもしれない。［ウェブ資料 20］。

Linux でゲノム解析

この FASTA ファイル（乳酸菌ゲノム配列）は、28 個のコンティグからなる[5]。連載第 1 回では、R で FASTA ファイルの読み込みからコンティグ数計測結果を含む各種解析が行えることを示した[6]。Mac のターミナルでも、grep コマンドを用いることで 28 という数値を得ることができる（**図 4**）。FASTA 形式ファイルは、”>” から始まる 1 行の description 行と、2 行目以降に塩基配列（やアミノ酸配列）を 1 文字表記で記述したものである。FASTQ 形式と同様、NGS 解析分野においても比較的よく用いら

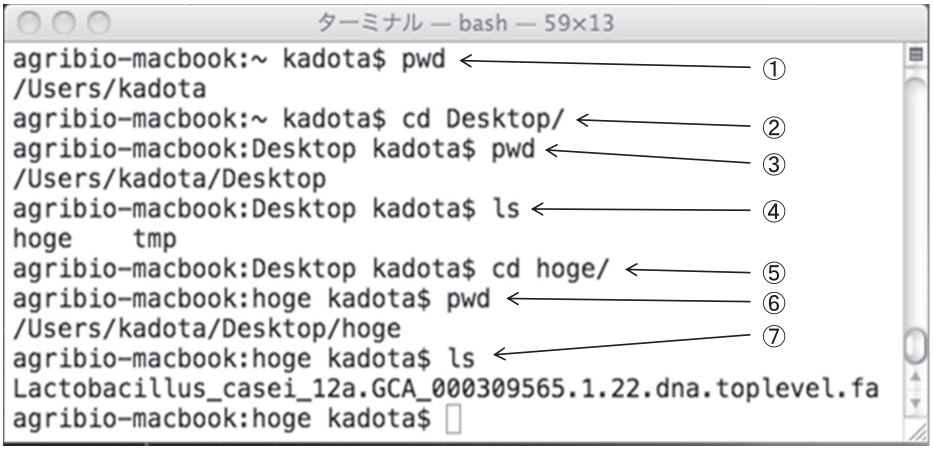

図 3. cd, pwd, ls を用いて目的の FASTA ファイルを視認する基本的な手順

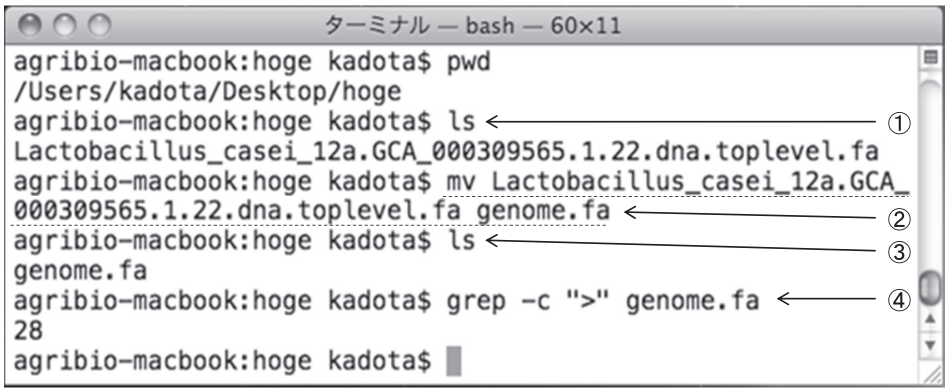

図 4. ①～③は mv コマンドで genome.fa にファイル名を変更し結果を確認する一連の手順。④は grep コマンドを用いて genome.fa 中の ">" を含む行数を出力している。

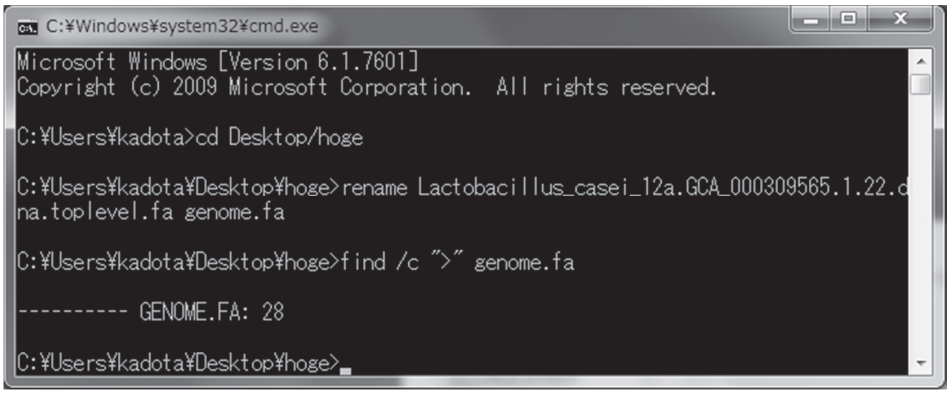

図 5. コマンドプロンプトでの作業例。① cd コマンドを用いた作業ディレクトリの変更、② rename コマンドを用いたファイル名の変更、③ find コマンドを用いた ">" を含む行数のカウント。

れるファイル形式である。grep は、文字列検索コマンドである。FASTA 形式ファイルである genome.fa を読み込んで、">" から始まる description 行を抽出し、出力を行数表示にする -c オプションを与えているため、結果が 28（行）と返される。行数表示の -c オプションをつけなかった場合、つまり「grep ">" genome.fa」を実行すると、description 情報がそのまま表示される［ウェブ資料 21］。grep は応用範囲の極めて広い非常に強力なコマンドである。例えば、アノテーションファイルから特定の染色体名を含む行のみを抽出したり、grep 出力結果をパイプでつないで他のコマンドで処理する作業が実際に行われる。また最近では、RNA-seq データからのキメラ転写物の同定に用いられるなど、grep コマンドの NGS 解析での有効性も報告されている[7]。もちろんここで示した程度のことであれば、Win のコマンドプロンプト上でも実現可能である（**図 5**；ウェブ資料 22-24）。例えば、Linux の grep に相当する Win のコマンドは find である。しかし、Linux にも find コマンドは存在し、（文字列検索ではなく）ファイルやディレクトリの検索機能をもつ。NGS 解析は、Win のコマンドプロンプト環境ではなく Linux 環境で行うのが一般的である。バイオインフォ業界では、文字列検索することを「grep する」といい、通常「find する」はファイルやディレクトリ検索のことを指す。つまり、一般にコマンド名で会話が成立するのは Linux コマンドのみである。

Bio-Linux の導入

連載第 2 回は、Win のコマンドプロンプトおよび Mac のターミナルという通常利用 PC 環境を用いて、GUI 環境とコマンドライン環境の見栄えの違い、および Win と Linux の基本コマンドを紹介した。Mac は Linux コマンドをすぐに使えるため、Linux 入門としてはおすすめである。しかし連載第 1 回でも述べたように、仮想ソフトをインストールして、様々な NGS 解析用プログラムが一通り組み込まれた Bio-Linux[8] を導入すれば、Win と Mac というホスト OS の違いによらず同じ Linux 環境で解析が可能である。2014 年 7 月末にリリースされた Bio-Linux 8 が最新版である（2014 年 10 月 15 日調べ）。連載第 3 回は、Bio-Linux 8（ゲスト OS）環境での解説を行う予定である。Win 用と Mac 用それぞれのインストール手順を掲載しているので、是非自力での Linux 環境構築にチャレンジしてほしい。

参 考 文 献

1) **Ogasawara, O., Mashima, J., Kodama, Y., Kaminuma, E., Nakamura, Y., Okubo, K., Takagi, T.:** DDBJ new system and service refactoring. *Nucleic Acids Res.,* **41**, D25-D29 (2013).
2) **小笠原理：VI プロトコール　データ解析と環境構築：**1　解析環境を導入する，p.323-331，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（2014）.
3) **Li, J.W., Schmieder, R., Ward, R.M., Delenick, J., Olivares, E.C., Mittelman, D.:** SEQanswers: an open access community for collaboratively decoding genomes. *Bioinformatics,* **28**, 1272-1273 (2012).
4) **Parnell, L.D., Lindenbaum, P., Shameer, K., Dall'Olio, G.M., Swan, D.C., Jensen, L.J., Cockell, S.J., Pedersen, B.S., Mangan, M.E., Miller, C.A., Albert, I.:** BioStar: an online question & answer resource for the bioinformatics community. *PLoS Comput. Biol.,* 7, e1002216 (2011).
5) **Broadbent, J.R., Neeno-Eckwall, E.C., Stahl, B., Tandee, K., Cai, H., Morovic, W., Horvath, P., Heidenreich, J., Perna, N.T., Barrangou, R., Steele, J.L.:** Analysis of the *Lactobacillus casei* supragenome and its influence in species evolution and lifestyle adaptation. *BMC Genomics,* 13, 533 (2012).
6) **門田幸二，孫建強，湯敏，西岡輔，清水謙多郎：**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（2014）.
7) **Panagopoulos, I., Gorunova, L., Bjerkehagen, B., Heim, S.:** The "grep" command but not FusionMap, FusionFinder or ChimeraScan captures the CIC-DUX4 fusion gene from whole transcriptome sequencing data on a small round cell tumor with t(4;19) (q35;q13). *PLoS ONE,* 9, e99439 (2014).
8) **Field, D., Tiwari, B., Booth, T., Houten, S., Swan, D., Bertrand, N., Thurston, M.:** Open software for biologists: from famine to feast. *Nat. Biotechnol.,* **24**, 801-803 (2006).

Methods for analyzing next-generation sequencing data II. From graphical user interface to command line interface

Jianqiang Sun[1], Min Tang[1], Tasuku Nishioka[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

Graphical user interface (GUI) is useful to perform general tasks. Analysis of next-generation sequencing (NGS) data is, however, non-trivial tasks. Many methods dedicated to NGS data have been implemented on Linux system that provides a command line interface (CLI) as an analysis environment. Therefore, it is desirable for researchers to analyze NGS data on Linux system. We here show the CLIs on Windows (i.e., Command prompts) and Macintosh (i.e., Terminal) systems in contrast with those GUIs. We also describe some basic commands such as “dir” and “ls.”