新型シーケンサー時代の超大規模ゲノム情報解析実習

長浜バイオ大学

本実習テキストは、「文部科学省のライフサイエンス分野の統合データベース整備事業」で、長浜バイオ大学が参画していた学部学生に対するアノテーター人材育成の実習用教材として、作成している。

ゲノム配列解読技術については、新型シーケンサーによって、大きな技術革命が起こっている。アメリカを中心にヒトゲノムを 1,000 ドルで読める時代を目指し、ゲノム配列解読技術の改良が鋭意行われている。現在でも一台の機械の一回の測定(これを 1 ランと呼ぶ)で、200G 塩基(ギガ、ヒトゲノムの 60 倍以上、20 億リード)を超えるゲノム断片配列が解読可能である。余談であるが、現在は(2010 年 11 月時点)、500 ドル程度用意できれば、10 日で自分自身のゲノムを解読することができる。一方、2003 年にヒトゲノムプロジェクトから発表されたヒトゲノム配列は、約１５年、数十億円をかけて、解読した成果である。金額と時間を見ただけでも非常に大幅な技術革新が行っていることが分かると思う。

このような超大量なゲノム断片配列を相手にするには、当然ながら情報処理技術が必要不可欠である。本実習では、バイオインフォマティクス初心者を対象に、新型シーケンサーより算出される配列データ解析実習を行うためのテキストである。今回は、大腸菌（*Escherichia coli* str. K-12 substr. MG1655）を対象に実習を行なう。

必要スキルとして、コマンドユーザーインターフェースである UNIX（Linux）の基本コマンドと perl プログラムの基礎を学んだ方を対象にする。UNIX（Linux）基本コマンドについては、「ライフサイエンス統合データベース」人材育成 AJACX 向教材ページの「アノテーター養成」ページにある「UNIX 再入門 1, 2」（http://motdb.dbcls.jp/?MotDB%2FNagahama-i-Bio）を参考にして下さい。また、Perl プログラムについては、初心者でも Linux 上で実習が行えるように、perl のサンプルプログラムを提供している。

また、本実習では、特別な計算機環境を必要とせずに実習を行えるように、全て 32bit、ならびに、低メモリ（2G byte 以上）環境でも実習可能にしている。

＜実習を行うためのマシン仕様・環境＞

１．ディスク空き容量 1.5Gbyte 以上、メモリ 2Gbyte 以上

２．OS が 32bit Linux で Perl 5.0 以上がインストールされていること

３．（オプション）Microsoft Office 2007 がインストールされている Windows マシン

１．新型シーケンサーの概要について

新型シーケンサーについて、簡単に説明する。現在、新型シーケンサーとして、販売されている機械は、ロシュ・ダイアグノスティック社が提供している Genome Sequencer FLX System （通称：454, http://www.roche-biochem.jp/catalog/index.php/category_33571.html）、 イルミナ社が提供している Hiseq2000 や Genome Analyzer IIx（通称：Solexa, http://www.illuminakk.co.jp/）、アプライドバイオシステムズ社が提供している SOLiD （http://www.appliedbiosystems.co.jp/）の３台が主流である。機器ごとに、DNA 増幅方法やシーケンス反応方法などが異なり、得られるゲノム配列は、件数、長さや精度など機器ごとに特色がある。表１に３台の新型シーケンサーの特徴をまとめた。新型シーケンサーについて、興味がある人は、各社の HP 上に日本語での解説があるので、そちらを参考にしてほしい。また、動画サイト YOUTUBE 上に分かりやすい動画あるので、そちらも参考にするとより理解できると思う。

動画サイトリンク

454：http://www.youtube.com/watch?v=bFNjxKHP8Jc

Solexa：http://www.youtube.com/watch?v=77r5p8IBwJk

SOLiD：http://www.youtube.com/watch?v=nlvyF8bFDwM

表１　新型シーケンサー比較

	GS FLX	*SOLiD*	*Hiseq2000 (solexa)*
1 リード長	500 bp	50 bp	35 or 50 or 100bp
1 ラン当たりのデータ量	400M – 600M	32 – 40G	150 – 200G (100bp)
解析手法	Pyrosequenceing	Sequencing by Ligation	Sequencing by Syntesis
アプリケーション	新規ゲノム解析 cDNA 解析 各種 PCR 産物解析 メタゲノム	ゲノム変異解析	ゲノム変異解析 ChiP 解析 Small RNA 解析

新型シーケンサーの特徴を比較してみると、１回の測定（1 ラン）あたりで解読できる配列件数が相当な数になっていることに気がつくが、得られる配列長は非常に短い。従って、目的に応じて、新型シーケンサーの種類を使い分けなければならない。例えば、新規生物種のゲノム配列を決めるためには、得られる配列長が長い GS FLX を用いた方が良く、ヒトの個人差をみるために、一塩基置換を測定する場合には、配列の精度が良い SOLiD が良いなどである。

２．実習で行う内容の概要

本実習で行う実習内容の簡単な解析概要を図１に示す。まず、新型シーケンサーによって解読されたゲノム断片配列を用いて、参照するゲノム（リファレンスゲノムともいう。本テキストでは、大腸菌を使用。）配列上に相同性検索プログラムによる貼り付け（マッピング）を行う。次に、得られたマッピング結果と遺伝子機能情報などのゲノム配列のアノテーション情報との比較を行ない、得られた結果から考えられる生物学的考察を行なう。この２点だけを実施するだけの非常にシンプルな実習であるとも言えるが、大量な件数の配列を解析することになる。

さぁ、それでは、実際の解析にはいっていこう。

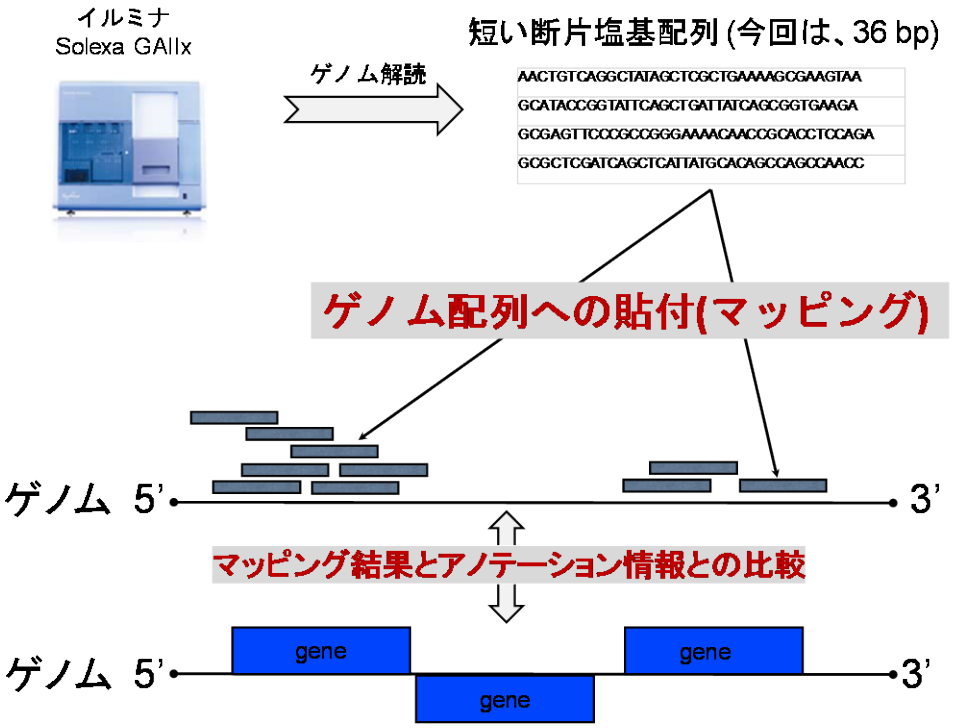


図１　本実習で行う解析の概要

３．配列データの取得

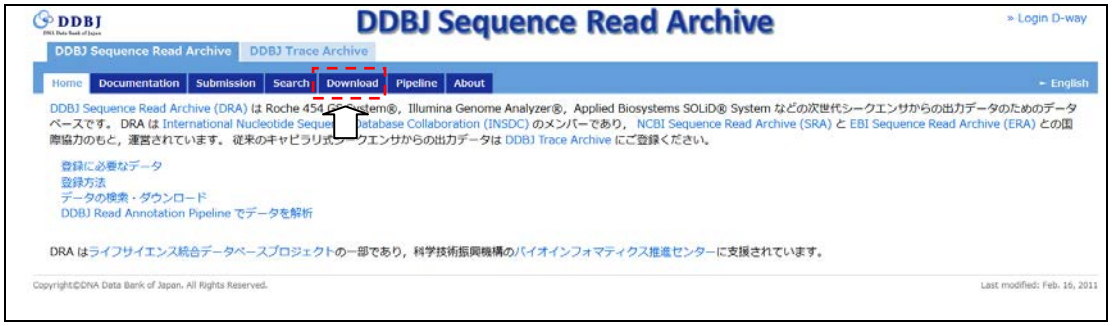
実習で使用する配列データを取得する。
（注意）このテキストでは、全て home ディレクトリ以下に作成した Annotation ディレクトリ以下で作業すると仮定して、説明を行っていく。

３－１．新型シーケンサー配列データのダウンロード

現在、国際塩基配列データベース（DDBJ/EMBL/GenBank）では、新型シーケンサー配列データの登録を受け付けており、Sequence Read Archive から公開を行っている。今回は、日本 DNA データバンク（DDBJ）の Sequence Read Archive である DRA（http://trace.ddbj.nig.ac.jp/dra/index.shtml, 画面 1）から新型シーケンサー配列データの取得を行う。

今回、対象とする新型シーケンサー配列データとして、大腸菌 *Escherichia coli* str. K-12 substr. MG1655 の再ゲノム解読（re-sequence）データであるアクセッション番号 SRA026422 の配列データを使用する。以下に、SRA026422 の取得方法を示す。

まず、DRA トップページへアクセスすると、画面１のようなトップ画面にアクセスできる。ここで、search タブ（画面１の赤枠）をクリックすると、登録配列データの検索画面にアクセスできる（画面２）。



（画面 1）DRA トップページ

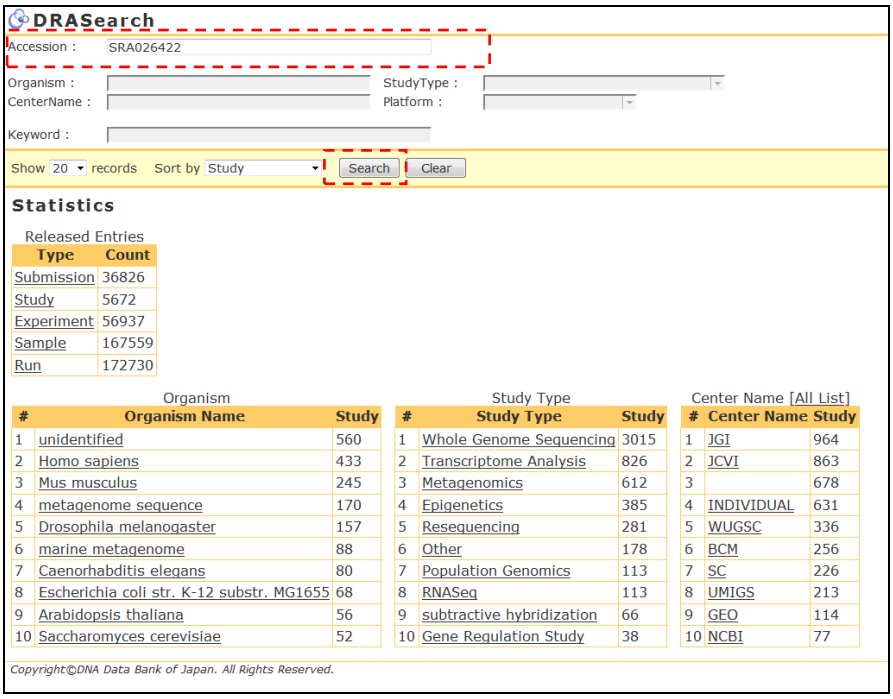
画面２の Accession に今回実習で使用するアクセッション番号「SRA026422」を入力し、検索を行う（画面３）。画面３左側にあるで「Navigation」中の「Run」列の【SRAlite】をクリックするとダウンロードが始まります。（取得ファイル名：SRR072749.lite.sra）

上述のようにブラウザを用いてファイルをダウンロードしてもよいが、UNIX コマンドではインターネット上で公開されているサイトから、複数ファイルを取得できる wget コマンドが用意されている。

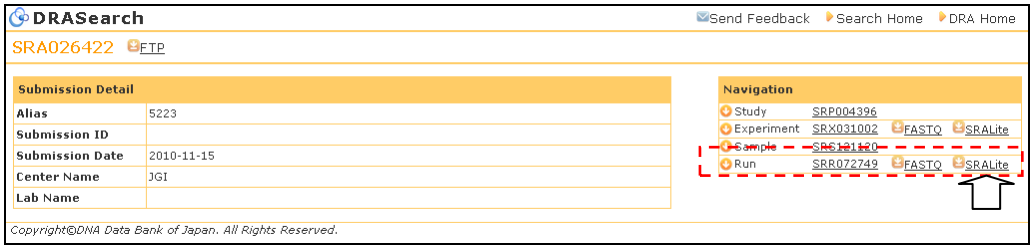
折角なので、wget コマンドでの配列取得を試みてみよう。

```
Wget ftp://ftp.ddbj.nig.ac.jp/ddbj_database/dra/sralite/ByExp/litesra/SRX/SRX031/SRX031002/SRR072749/SRR072749.lite.sra
```

配列の URL については、画面３の【SRA lite】にマウスポインタを移動後、右クリックすると、メニューの中に【リンクアドレスをコピー】があるので、それを選択すれば URL をコピー出来る。



（画面２）キーワード検索画面



（画面３）キーワード検索結果

３－２．取得した配列データの変換

取得した配列形式 SRALite は配列公開用に国際塩基配列データベースが定義した圧縮ファイル形式であり、配列データに変換を行う必要がある。新型シーケンサー配列データで使用される配列形式として FASTQ 形式がある。通常の FASTA 形式との違いを図２に示す。

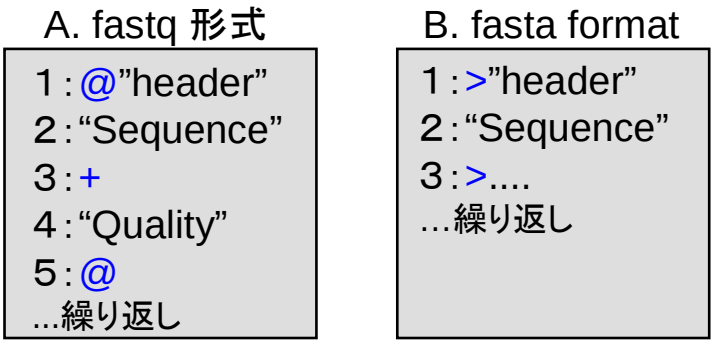


図２. FASTQ 形式(A)と FASTA 形式(B)。「左端の数字と”：”」は、説明のために加えた行数であり、実際のファイルには存在しない。定型文字を青字で示す。FASTQ 形式は、FASTA ファイルと比べ、3 行目と 4 行目で配列のクオリティー値を追記した形式である。

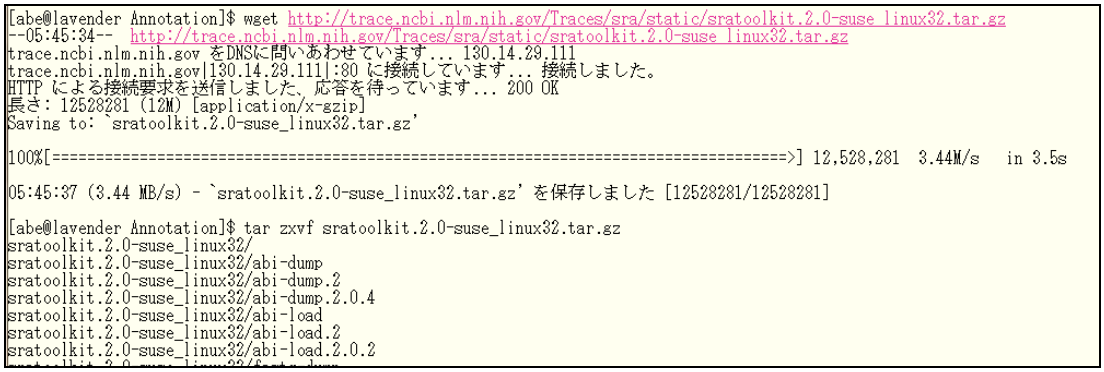
変換を行うために、NCBI Seqquence Read Archive（SRA）が提供している SRA Toolkit のダウンロードを行う。

●SRA Toolkit 提供サイト
http://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?cmd=show&f=software&m=software&s=software

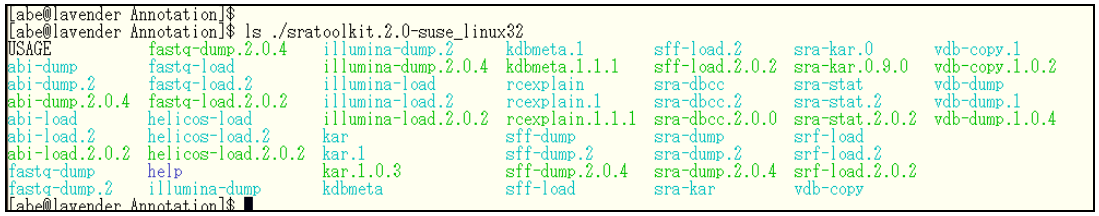
ダウンロード方法と解凍方法については、Linux を使用していることを前提に、wget と tar コマンドを使用した例を示す。
ソフトウェアはコンパイル済みファイルをダウンロードするが、はじめに書いたとおり、32bit 版でも実行可能とすることを目的に、ここでは、32bit 版のダウンロードを行う。使用する環境に応じて、64bit 版をダウンロードしても良い。

Wget
http://trace.ncbi.nlm.nih.gov/Traces/sra/static/sratoolkit.2.0-suse_linux32.tar.gz
tar zxvf sratoolkit.2.0-suse_linux32.tar.gz

実行後の画面については、画面４， ５も参考にすると良い。



（画面４）SRA Toolkit を wget で取得後、tar で解凍を実施



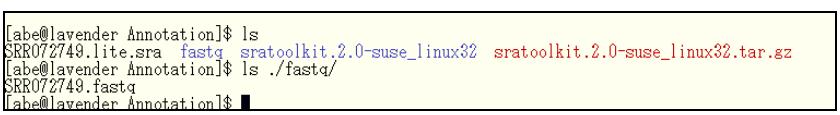
（画面５）解凍後、sratoolkit.2.0-suse_linux32 というディレクトリができ、以下のようなファイルが作成される。

次に、FASTQ ファイルへの変換を行う。使用するプログラムは、sratoolkit.2.0-suse_linux32 ディレクトリ以下の fastq-dump を使用する。

実行形式：./sratoolkit.2.0-suse_linux32/fastq-dump -A SRR072749 -D SRR072749.lite.sra -O fastq
なお、ここで使用しているオプションは、”-A”は配列名に付与される ID、”-D” 入力 SRA lite ファイル名、”-O”は出力先のディレクトリを指定、となる。

実行前後については、画面６、７を参考。

（画面６）fastq-dump 実行後の画面



（画面７）実行後、「fastq」ディレクトリ以下に「SRR072749.fastq」ファイルが作成される。

３－３．参照する大腸菌ゲノム配列情報の取得

国際塩基配列データベースで公開されている大腸菌 *Escherichia coli* str. K-12 substr. MG1655 のゲノム配列情報を取得する。今回は、NCBI より公開されている大腸菌ゲノム配列情報を取得する。
ゲノム配列が公開されている URL は以下の通りである。

```
ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Escherichia_coli_K_12_substr__MG1655_uid57779/
```

このディレクトリ以下に含まれている[NC_000913.fna]と[NC_000913.ptt]ファイルを取得する。ここで、NC_000913.fna は全ゲノム配列の FASTA 形式ファイル、NC_000913.ptt はタブ区切り形式の遺伝子アノテーションファイルである。
ブラウザを用いて、ファイルをダウンロードしてもよいが、UNIX コマンドでは、インターネット上で公開されているサイトより、複数ファイルの取得できる wget コマンドが用意されている。W3110_genome ディレクトリに移動後、以下のコマンドを入力し、ファイルを取得することができる。

```
wget ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Escherichia_coli_K_12_substr__MG1655_uid57779/NC_000913.fna
wget ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Escherichia_coli_K_12_substr__MG1655_uid57779/NC_000913.ptt
```

ここまでに取得したファイルの一覧を画面８に示す。

```
[abe@lavender Annotation]$ ls ./*
./NC_000913.fna  ./SRR072749.lite.sra
./NC_000913.ptt  ./sratoolkit.2.0-suse_linux32.tar.gz

./fastq:
SRR072749.fastq

./sratoolkit.2.0-suse_linux32:
USAGE             fastq-load.2.0.2     kar.1            sff-load.2      sra-stat.2
abi-dump          helicos-load         kar.1.0.3        sff-load.2.0.2  sra-stat.2.0.2
abi-dump.2        helicos-load.2       kdbmeta          sra-dbcc        srf-load
abi-dump.2.0.4    helicos-load.2.0.2   kdbmeta.1        sra-dbcc.2      srf-load.2
abi-load          help                 kdbmeta.1.1.1    sra-dbcc.2.0.0  srf-load.2.0.2
abi-load.2        illumina-dump        rcexplain        sra-dump        vdb-copy
abi-load.2.0.2    illumina-dump.2      rcexplain.1      sra-dump.2      vdb-copy.1
fastq-dump        illumina-dump.2.0.4  rcexplain.1.1.1  sra-dump.2.0.4  vdb-copy.1.0.2
fastq-dump.2      illumina-load        sff-dump         sra-kar         vdb-dump
fastq-dump.2.0.4  illumina-load.2      sff-dump.2       sra-kar.0       vdb-dump.1
fastq-load        illumina-load.2.0.2  sff-dump.2.0.4   sra-kar.0.9.0   vdb-dump.1.0.4
fastq-load.2      kar                  sff-load         sra-stat
[abe@lavender Annotation]$
```

（画面８）ここまでに取得したファイル一覧

４．Perl の復習：FASTQ ファイルに含まれる配列件数の取得

配列解析を行うには、各ソフトウェアが指定する入力ファイル形式に、ファイルの形式変換を行う必要がある。新型シーケンサー配列データのマッピングにおいても、３－２章で SRA lite 形式から FASTQ 形式へ新型シーケンサー配列データファイルの形式の変換を行った。ファイル形式の変換やそのファイルに含まれる配列件数のチェックなどは、データが少ない場合は人の手でもできるが、ゲノム解析用の大規模データになると非常に退屈な(苦痛な) 作業となり、間違いが起こりやすい。このような作業こそ、プログラムによって自動的に行うべきことである。ここでは、FASTQ ファイルに含まれる配列件数取得を例に、プログラム言語 perl の使用方法について復習する。

まず、Linux において Perl 言語で書いたプログラム（Perl プログラム）を実行する方法を解説する。そのために、簡単な Perl プログラムを作成しよう。*Annotation* ディレクトリ以下に、ex で、*vi* エディタを使って、以下の Perl プログラムを入力してほしい。なお、プログラムのファイル名は *hello.pl* とする。

プログラム名：*hello.pl*

```
1  #!/usr/bin/perl -w
2  use strict;
3
4  print "Hello!!¥n";
```

入力し終わったら、保存して *vi* エディタを終了し、同じ端末上で以下のコマンドを入力しよう。

$ chmod 744 hello.pl

これにより作成した Perl プログラムが実行可能となった。プログラムを作成したディレクトリで

$ ./hello.pl

と入力すると、

Hello!!

と出力されるはずである。ここで実行した *chmod* は、ファイルのアクセス権を変更するコマンドである。作成したばかりの *hello.pl* はただのテキストファイルであり、そのままでは実行できない。そこで、*chmod* で実行権限をつけることで、Perl プログラムとして実行できるようにしている。なお、今後は *chmod* により実行権限を与えることを明記しない。プログラムを作成したら各自で忘れずに実行してほしい。

次に、ファイルを読み込み、その中に書かれているデータを処理する簡単なプログラムを作成してみよう。*Annotation* ディレクトリにおいて、以下のプログラムを入力してほしい。このプログラムは、指定したファイルの中にある行の数を数えて表示するものである。作成したプログラムを、新型シーケンサーより出力された配列データファイルを対象に実行してみる。以下の画面９にプログラム *count.pl* とその実行例を示す。

```
[abe@lavender Annotation]$ cat -n count.pl
     1  #!/usr/bin/perl -w
     2  use strict;
     3
     4  my $line;
     5  my $count = 0;
     6  while ($line = <>) {
     7      $count = $count+1;
     8  }
     9  print "$count¥n";
    10
[abe@lavender Annotation]$ chmod 744 count.pl
[abe@lavender Annotation]$ ./count.pl ./fastq/SRR072749.fastq
25869236
[abe@lavender Annotation]$
```

（画面９）プログラム名：*count.pl* と実行例

画面９にあるように、SRR072749.fastq ファイルには、25869236 件あることがわかる。
では、*count.pl* を改変し、FASTQ 形式のファイルの中に含まれている配列数を数えて出力するプログラムを作成してみよう。FASTQ ファイルは、図 2B で示すように“@”で始まるコメント行と配列データからなるファイルであり、FASTQ ファイル中に含まれている配列件数を調べるには、“@”の数を数えると良い。*count.pl* のファイルを *cp* コマンドでコピーして *count_fastq.pl* を作成し、以下のようなプログラムに変更しよう。

```
[abe@lavender Annotation]$ cat -n count_fastq.pl
     1  #!/usr/bin/perl -w
     2  use strict;
     3
     4  my $line;
     5  my $count = 0;
     6  while ($line = <>) {
     7      if($line =~ /^¥@SRR/){
     8          $count = $count+1;
     9      }
    10  }
    11  print "The number of sequences is $count¥n";
    12
[abe@lavender Annotation]$ chmod 744 count_fastq.pl
[abe@lavender Annotation]$ ./count_fastq.pl ./fastq/SRR072749.fastq
The number of sequences is 6467309
[abe@lavender Annotation]$
```

（画面１１）プログラム名：*count_fastq.pl* と実行例

このプログラムで、7 行目に正規表現を使って、先頭文字が"@SRR"から始まる行かどうかを判定し、それが成り立つ場合は 8 行目でカウントを行なう。実行を行うと画面１０のように、SRR072749.fastq ファイルには、6467309 件あることがわかる。

課題１：ファイル件数のカウント

A) *count.pl* の機能と同等の機能を持つ UNIX（Linux）のコマンドを記述せよ。
B) *count_fastq.pl* の機能と同等の機能を持つ UNIX（Linux）のコマンドを記述せよ。

５．新型シーケンサー配列データのリファレンスゲノム上への貼り付け（マッピング）

新型シーケンサー配列データでは、特に Solexa や SOLiD の場合のように、30-50bp と非常に短い配列が大量に産出されることから、既にゲノム配列が解読されているゲノム （リファレンスゲノム） を対象に、遺伝子発現量や SNP（1 塩基多型）のような塩基配列の多様性などを調べる目的で、使用されることが主流である。そこで、重要となってくるのが、大量の新型シーケンサー配列データの参照するゲノム（リファレンスゲノム）配列上への貼り付け（マッピング）である。新型シーケンサー配列データがゲノム上のどの領域にマップされたかが分かれば、遺伝子機能などのアノテーション情報と組み合わせることによって、遺伝子発現量や、塩基配列やアミノ酸配列の違いなどの多様性を見ることが可能となる。その他のマッピングツールやアセンブラについては、
http://seqanswers.com/forums/showthread.php?t=43 を参照するとよい。本実習では、リファレンスゲノムへのマッピングツールとして、新型シーケンサー配列データ解析用に開発された SOAP （Short Oligonucleotide Alignment Package）を用いる。その他、本章では、各相同性検索ソフトの実行、ならびに、出力結果からのデータ解析用ファイルへの変換プログラムの作成を行う。

５－１．SOAP （Short Oligonucleotide Alignment Package）のインストールと実行

５－１－１．インストール方法

新型シーケンサーの登場に合わせて、開発された相同性解析ソフトウェアである。その名のとおり、短い配列に対し、高精度、かつ、高速な相同性解析が可能となっている。ソフトウェアは、開発グループである北京ゲノム研究所の HP（http://soap.genomics.org.cn/）よりダウンロード可能である。なお、最新版は SOAP2 であるが 64bit 環境のみで動作可能版しか提供されていないため、今回は 32bit 環境での実行を考慮し、SOAP1（ver. 1.11、以下 SOAP とする）を使用する。

SOAP（ver. 1.11）のソースファイルの格納場所：
http://soap.genomics.org.cn/soap1/soap_1.11.tar.gz

画面１２にソフトウェアのダウンロード方法と解凍方法を示す。

```
[abe@lavender Annotation]$ wget http://soap.genomics.org.cn/soap1/soap_1.11.tar.gz
--11:19:01--  http://soap.genomics.org.cn/soap1/soap_1.11.tar.gz
soap.genomics.org.cn をDNSに問いあわせています... 124.16.11.75
soap.genomics.org.cn|124.16.11.75|:80 に接続しています... 接続しました。
HTTP による接続要求を送信しました、応答を待っています... 200 OK
長さ: 50180 (49K) [application/x-gzip]
Saving to: `soap_1.11.tar.gz'

100%[==============================================>] 50,180      102K/s   in 0.5s

11:19:01 (102 KB/s) - `soap_1.11.tar.gz' を保存しました [50180/50180]

[abe@lavender Annotation]$ tar zxvf soap_1.11.tar.gz
soap_1.11/
soap_1.11/pairs.cpp
soap_1.11/main.cpp
soap_1.11/reads.cpp
soap_1.11/RELEASE.txt
soap_1.11/README.txt
soap_1.11/dbseq.h
soap_1.11/align.h
soap_1.11/utilities.cpp
soap_1.11/pairs.h
soap_1.11/param.cpp
soap_1.11/dealign.h
soap_1.11/align.cpp
soap_1.11/GPL_3.0.txt
soap_1.11/dealign.cpp
soap_1.11/utilities.h
soap_1.11/dealign.cpp.liyr
soap_1.11/makefile
soap_1.11/param.h
soap_1.11/soap_dealign.cpp
soap_1.11/reads.h
soap_1.11/dbseq.cpp
[abe@lavender Annotation]$
```

（画面１２）SOAP のダウンロードと解凍方法

SOAP のソースファイルの解凍後、*soap_1.11* というディレクトリが作成されるので、soap_1.11 ディレクトリに移動後、実行形式ファイルを作成するために、*make* を実行する。うまく動作が完了した後に、*soap_1.11* ディレクトリ以下に、*soap* という実行ファイルができる（画面１３）。インストールが上手く完了すれば、「*./soap*」と実行すれば、オプションヘルプが表示されるはずである（画面１４）。

```
[abe@lavender Annotation]$ cd soap_1.11
[abe@lavender soap_1.11]$ make
g++ -static -DMAXGAP=3 -DMAXHITS=10000 -DTHREAD -O3   -DDB_CHR -DREAD_60  -c align.cpp -o
align.chr.o
g++ -static -DMAXGAP=3 -DMAXHITS=10000 -DTHREAD -O3   -DDB_CHR -DREAD_60  -c dbseq.cpp -o
dbseq.chr.o
g++ -static -DMAXGAP=3 -DMAXHITS=10000 -DTHREAD -O3   -DDB_CHR -DREAD_60  -c main.cpp -o m
```

（画面１３）SOAP インストール方法

```
[abe@lavender soap_1.11]$ ls
GPL_3.0.txt     dbseq.h          makefile        param.short.o   soap_dealign.cpp
README.txt      dbseq.huge.o     pairs.chr.o     reads.chr.o     soap_dealign.o
RELEASE.txt     dbseq.short.o    pairs.contig.o  reads.contig.o  utilities.chr.o
align.chr.o     dealign.cpp      pairs.cpp       reads.cpp       utilities.contig.o
align.contig.o  dealign.cpp.liyr pairs.h         reads.h         utilities.cpp
align.cpp       dealign.h        pairs.huge.o    reads.huge.o    utilities.h
align.h         dealign.o        pairs.short.o   reads.short.o   utilities.huge.o
align.huge.o    main.chr.o       param.chr.o     soap            utilities.short.o
align.short.o   main.contig.o    param.contig.o  soap.contig
dbseq.chr.o     main.cpp         param.cpp       soap.huge
dbseq.contig.o  main.huge.o      param.h         soap.short
dbseq.cpp       main.short.o     param.huge.o    soap_dealign
[abe@lavender soap_1.11]$ ./soap
Usage:  soap [options]
       -a  <str>   query a file, *.fq or *.fa format
       -d  <str>   reference sequences file, *.fa format
       -o  <str>   output alignment file
       -s  <int>   seed size, default=10. [read>18,s=8; read>22,s=10, read>26, s=12]
       -v  <int>   maximum number of mismatches allowed on a read, <=5. default=2bp. For p
air-ended alignment, this version will allow either 0 or 2 mismatches.
       -g  <int>   maximum gap size allowed on a read, default=0bp
       -w  <int>   maximum number of equal best hits to count, smaller will be faster, <=1
0000
       -e  <int>   will not allow gap exist inside n-bp edge of a read, default=5bp
       -z  <char>  initial quality, default=@ [Illumina is using '@', Sanger Institute is
using '!']
       -c  <int>   how to trim low-quality at 3-end?
                   0:     don't trim;
                   1-10:  trim n-bps at 3-end for all reads;
                   11-20: trim first bp and (n-10)-bp at 3-end for all reads;
```

（画面１４）SOAP のイントールと SOAP の実行テスト結果

５－１－２．SOAP の実行 1

SOAP の実行方法は以下のとおりである。ここでは、*Annotation* ディレクトリで実行する場合でコマンド例を記載しているので、注意すること。

```
./soap_1.11/soap -a ./fastq/SRR072749.fastq –d ./ NC_000913.fna -o SRR072749.soap_result1
```

実行方法の詳細については、開発者らの HP も参照しよう。

http://soap.genomics.org.cn/soap1/

実行例を画面１５に示す。

```
[abe@lavender Annotation]$ ls
NC_000913.fna       count_fastq.pl    sratoolkit.2.0-suse_linux32
NC_000913.ptt       fastq             sratoolkit.2.0-suse_linux32.tar.gz
SRR072749.lite.sra  soap_1.11
count.pl            soap_1.11.tar.gz
[abe@lavender Annotation]$ ./soap_1.11/soap -a ./fastq/SRR072749.fastq -d ./NC_000913.fna
-o SRR072749.soap_result1
Start at:  Mon Apr 11 11:30:15 2011

Load in 1 db seqs, total size 4639675 bp. 0 secs passed
total_kmers: 1048576
Create seed table. 1 secs passed
Single read alignment:
Query: ./fastq/SRR072749.fastq  Reference: ./NC_000913.fna  Output: SRR072749.soap_result1
10000 reads finished. 1 secs passed
20000 reads finished. 1 secs passed
30000 reads finished. 2 secs passed
40000 reads finished. 2 secs passed
50000 reads finished. 2 secs passed
60000 reads finished. 2 secs passed
70000 reads finished. 2 secs passed
80000 reads finished. 2 secs passed
90000 reads finished. 2 secs passed
100000 reads finished. 3 secs passed
```

（中略）

```
6420000 reads finished. 113 secs passed
6430000 reads finished. 113 secs passed
6440000 reads finished. 114 secs passed
6450000 reads finished. 114 secs passed
6460000 reads finished. 114 secs passed
6467309 reads finished. 114 secs passed
Total number of aligned reads: 4550729 (70%)
Done.
Finished at Mon Apr 11 11:32:09 2011
Total time consumed:  114 secs
[abe@lavender Annotation]$
```

（画面１５）SOAP の実行方法と実行画面例

出力結果は以下のとおりとなる（画面１６）。

```
[abe@lavender Annotation]$ head -5 SRR072749.soap_result1 | cat -n
     1  SRR072749.1     GAAGGAGGTGTGAGATGGGAAGTGGATGGCATGAAT    IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
IIIIIIIII       1       a       36      +       gi|49175990|ref|NC_000913.2|    943242  0
     2  SRR072749.2     GCGTAACAATGTTAGTGATGTTTTGGTAGAAATTAC    IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
IIIIIIIII       1       a       36      +       gi|49175990|ref|NC_000913.2|    2386848 0
     3  SRR072749.3     GTTATGGAAGCTAATTATGATATTAGAGATCCCATG    IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
IIIIIIIII       1       a       36      +       gi|49175990|ref|NC_000913.2|    801124  2
C->16A9 C->26A9
     4  SRR072749.4     TATTACTTTCCGTAACCAGTTGCCTGAACTCGGCGC    IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
IIIIIIIII       1       a       36      -       gi|49175990|ref|NC_000913.2|    2793152 0
     5  SRR072749.6     GCTAAGAGGTCTTTGTGAAAAGTGCAATGACAAGCT    IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
IIIIIIIII       1       a       36      +       gi|49175990|ref|NC_000913.2|    1374842 0
[abe@lavender Annotation]$
```

（画面１６）SOAP 出力結果

ここで、左端の数字（１　SRR…）は、行番号を示す (出力結果には出力されないので注意すること)。全てタブ区切りテキストで、一行に必要な情報が全て記載されている (ここでは、画面の都合上 2 行に分かれているが、ファイル上では 1 行である)。各行に書かれている
データは、左から順番に

- １カラム目：query 配列の ID
- ２カラム目：塩基配列データ
- ３カラム目：塩基配列クオリティー値
- ４カラム目：ゲノム配列にマップされた数
- ５カラム目：実験条件のフラグ情報 (a or b)
- ６カラム目：配列長
- ７カラム目：ヒットしていた配列がヒットしていたストランド情報(+ or -)
- ８カラム目：リファレンス配列の ID
- ９カラム目：リファレンスゲノム配列においてアラインされた領域の(5’末側)開始位置
- １０カラム目：ミスマッチの有無 (0:ミスマッチなし、1～100:ミスマッチ数)

である。

５－１－３．SOAP の実行 2

SOAP などの新型シーケンサー配列データのマッピングソフトでは、短いゲノム配列断片を大量に扱うために、マッピング結果において、複数ヶ所マップされた場合でもいずれかの一箇所のみをランダムに選び、結果として出力されるのがデフォルトとなっている。そこで、ここでは、マップされた全ての箇所が出力された場合との実行結果の比較を行うために、*soap* の実行を行う。

SOAP の実行方法は以下のとおりである。ここでは、*Annotation* ディレクトリで実行する場合でコマンド例を記載しているので、注意すること。

```
./soap_1.11/soap -a ./fastq/SRR072749.fastq -d ./ NC_000913.fna -o SRR072749.soap_result2 -r 2
```

ここで、【-r　[0,1,2] how to report repeat hits, 0=none; 1=random one; 2=all, default=1】となっており、-r 2 とすることで、全ての箇所にマッピングされた結果が出力される。実行方法については、画面１５を参照されたい。

５－１－４．SOAP の実行結果の整理

出力された結果から、各配列がリファレンスゲノム上のどの位置にマップされていたかがわかる。本実習では、マップ結果を簡便化するために、SOAP 実行結果から、**クエリー配列ID、リファレンスゲノム配列においてアラインされた領域の開始位置、リファレンスゲノム配列においてアラインされた領域の終了位置**を抜き出したファイルを作成する。
ここでは、Linux に標準的にインストールされているテキスト処理のための強力なスクリプト言語である awk を利用して、上記下線部の項目の抽出を行ってみる（画面１７）。

```
[abe@lavender Annotation]$ awk '{print $1"¥t"$9"¥t"$9+$6}' SRR072749.soap_result1 | head -
5
SRR072749.1     943242  943278
SRR072749.2     2386848 2386884
SRR072749.3     801124  801160
SRR072749.4     2793152 2793188
SRR072749.6     1374842 1374878
[abe@lavender Annotation]$ awk '{print $1"¥t"$9"¥t"$9+$6}' SRR072749.soap_result1 > SRR072
749.soap_result1.cut
[abe@lavender Annotation]$
```

（画面１７）awk の実行例

ここで、awk を使い、一列目、９列目、９列目＋６列目の３項目を出力させている。初めのコマンドは、*“head -5“* を追加し、最初の５行目までをサンプルとして表示させた場合である。同様の処理は、perl でも作成可能である。興味のある方は perl でプログラムを作成してみよう。

課題 2

SOAP の解析結果を用いて、以下の課題に答えよ。
A 5-1-2 で実行した結果を用いて新型シーケンサー配列データがリファレンスゲノムにマッピングされた件数を記述せよ。
B 5-1-3 で実行した結果を用いて新型シーケンサー配列データがリファレンスゲノムにマッピングされた件数を記述せよ。

6. マッピング結果の集計方法について

前章までに、BSOAP を用いて、新型シーケンサー配列データをリファレンスゲノムにマッピング（貼り付け）を行なってきた。マッピング結果は、新型シーケンサー配列データがリファレンスゲノム（*E. coli* K-12 MG1655 株ゲノム）上にマップされた位置情報のみが得られるだけであるため、マッピング結果の解釈が非常に難しい。マッピング結果を有効活用するためには、リファレンスゲノム上での遺伝子がコードされている位置情報（以下、位置情報とする）や遺伝子の機能情報などのアノテーション情報を参照したほうが良い。

本章では、リファンレンスゲノム上の遺伝子の位置情報や機能情報などのアノテーション情報をもとに、マッピング結果の解釈を行っていく方法について紹介する。

はじめに、マッピング結果の集計方法の流れを大まかに理解してもらうために、Excel を利用した結果整理を行う。次に、Perl プログラムによるマッピング結果の集計を行う。本来であれば、大量な配列データを対象とするため、Perl などのプログラムによる解析が必須となるが、少量データの場合には、Excel も有効であるため、少量データ時の例もかねて、Excel での集計方法についても紹介する。

６－１．Perl プログラムを用いたマッピング結果の集計方法について

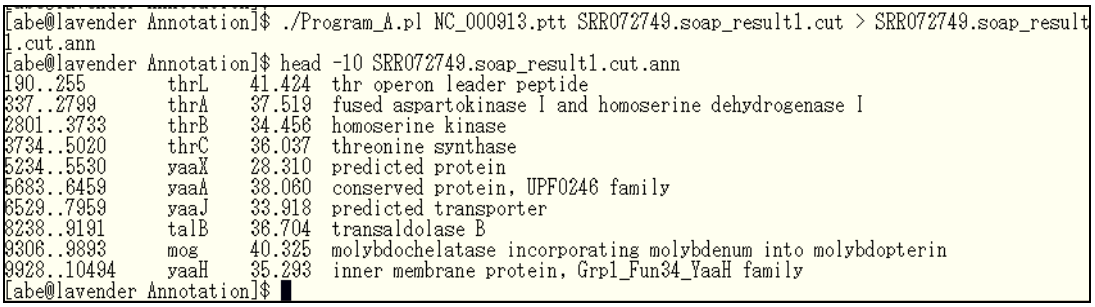
今回のような大規模な結果の場合には、全結果を対象に解析するには、Linux 上で、Perl などのプログラムによる処理が必須となる。この章では、サンプルプログラムを用いて、Perl プログラムによるマッピング結果の集計を行う。

以下の２つのサンプルプログラムを用いて、SOAP のマッピング結果【５－１－２と５－１－３】の集計を行い、以下の課題に回答せよ。
なお、*Program_A.pl* の実行方法は、以下のとおりである。

$ ./Program_A.pl NC_000913.ptt SRR072749.soap_result1.cut > SRR072749.soap_result1.cut.ann

---

実行結果を画面１８に示す。ここでは、上位１０件を表示している。出力結果は、左から、遺伝子領域（開始位置..終了位置）、遺伝子名、マップ件数（遺伝子領域にマップされた新型シーケンサー配列データの平均値）、遺伝子機能となっている。



（画面１８）Perl プログラムの実行例と実行結果例

ここまでで、新型シーケンサー配列データを用いたゲノム配列解析の生物学的な意味を引き出すために必要最低限のデータ作成が完了した。膨大な配列データを取り扱うため、UNIX（Linux）などの CUI (Command User Interface) やプログラムの必要性を理解できたかと思う。これからは、以下の課題３を通じて、作成した結果を用いて、生物学的な知識の発見を行ってみよう。

課題３

A) *Program_A.pl* による、SOAP でのマッピング全結果を対象にした実行結果を用いて、新型シーケンサー配列データがマップされていた件数の下位２０遺伝子を記述せよ。検出された遺伝子の機能の観点から特徴について考察せよ。

B) *Program_A.pl* による、SOAP での 2 種類のマッピング結果を対象にした実行結果を用いて、SOAP の 2 種類の結果でマッピングされた配列件数に差が大きい遺伝子群の上位の 20 遺伝子を記述せよ。抽出された遺伝子の機能の観点から特徴となぜ差が大きかったのかについて考察せよ。

A. Program_A.pl

```
#!/usr/bin/perl -w
use strict;

@ARGV == 2 or die "usege:    $0 (AC data) (blast data)¥n";

my $filename1 = $ARGV[0];
my $filename2 = $ARGV[1];
my @seq_hist = ();
my $length = 0;

open(IN1, $filename1) or die "$filename1 can't open.¥n";

my $line;
my $i;

$line = <IN1>;
my $protein_num = <IN1>;
my $comment = <IN1>;
if($line =~ /(¥d+)¥.¥.(¥d+)/){
        $length = $2;
}

for($i = 1; $i < $length; $i++){
        $seq_hist[$i] = 0;
}

open(IN2, $filename2) or die "$filename2 can't open.¥n";
while($line = <IN2>){
        chomp($line);
        my ($seq_name, $s_st, $s_en) = split(/¥s+/, $line);
        if($s_st > $s_en){
                for($i = $s_en - 1; $i < $s_st; $i++){
                        $seq_hist[$i]++;
                }
        }else{
                for($i = $s_st - 1; $i < $s_en; $i++){
                        $seq_hist[$i]++;
                }
        }
}
close(IN2);

while($line = <IN1>){
        chomp($line);
        my @column = split(/¥t/, $line);
        my $genename = $column[4];
        my $product = $column[8];
        my ($g_st, $g_en) = split(/¥.¥./, $column[0]);
        my $sum = 0;
                for($i = $g_st - 1; $i < $g_en; $i++){
                        $sum += $seq_hist[$i];
                }
        my $average = $sum / ($g_en - $g_st + 1);
        printf("%s¥t%s¥t%.3f¥t%s¥n", $column[0], $genename, $average,$product);
}
close(IN1);
```

(参考)

６－２．Excel を用いたマッピング結果の集計方法について

【５－１－２】で得た SOAP によるマッピング結果を用いて、Excel を用いて、マッピング結果の集計を行う。
以下で紹介する作業は、すべて、Windows 上の**【ディスクトップ】**上に作成した**【実習】**フォルダ以下での作業を例に紹介する。

６－２－１．マッピング結果の集計に使用するファイルの準備

【SOAP によるマッピング結果の形式を変換したファイル】，【E. coli K-12 MG1655 株の遺伝子アノテーション情報 (./NC_000913.ptt)】の２種類のファイルを用意する。Linux 上から、Windows 上の**【ディスクトップ】**上に作成した**【実習】**フォルダ以下に FTP などを用いて、ファイルの転送を行う。

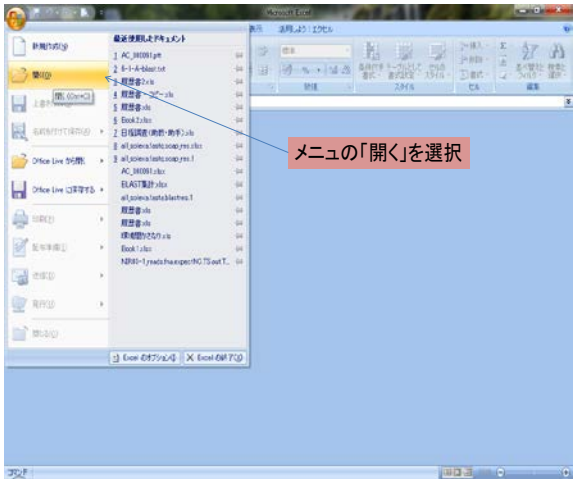
BLAST によるマッピング結果の形式を変換したファイル
【５－１－２】で作成した SOAP のタブ区切り解析結果から配列 ID とマップ位置情報のみを抜き出したファイルを作成し、上位 10000 件を抜き出し、集計用のサンプルデータとして使用する(ファイル名：SRR072749.soap_result1.10000)。形式変換方法について、画面１９を示す。

```
[abe@lavender Annotation]$ head -10000 SRR072749.soap_result1.cut > SRR072749.soap_result1.10000
[abe@lavender Annotation]$ wc  SRR072749.soap_result1.10000
 10000  30000 305427 SRR072749.soap_result1.10000
```

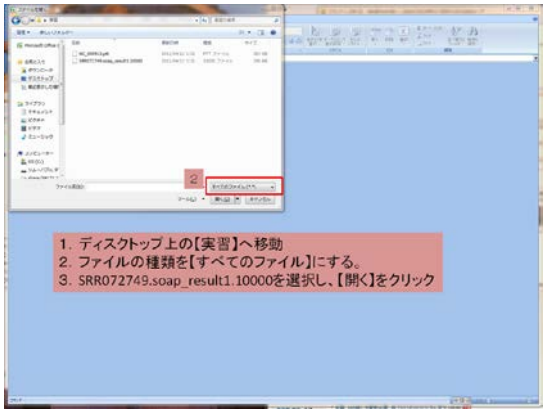
（画面１９）上位 10000 件の抽出方法 (head コマンドを利用)

６－２－２. Excel ファイルへの読み込み

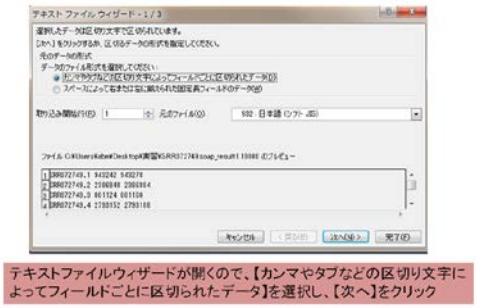
A. SOAP1 結果（サンプルファイル名：SRR072749.soap_result1.10000）の読み込み
以下の手順 2-A1～2-A4 までを参考に SOAP 結果ファイル SRR072749.soap_result1.10000 を Excel に読み込みを行う。



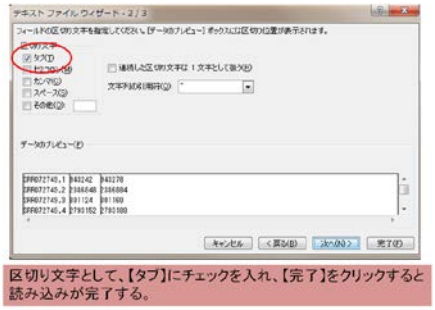
手順 2-A1



手順 2-A2



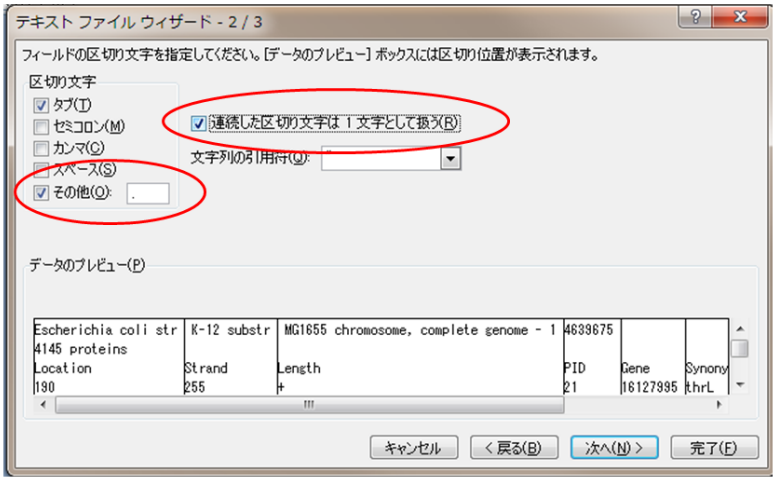
手順 2-A3



手順 2-A4

B. *E. coli* K-12 MG1655 株ゲノムのアノテーションファイル NC_000913.ptt の読み込み

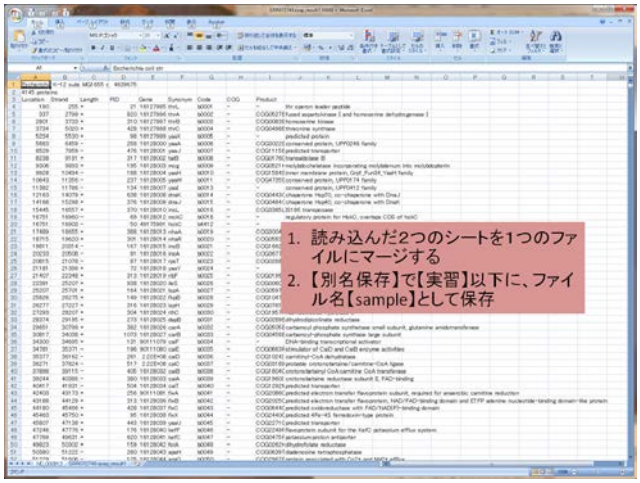
前述の 6-1-2A を参考に Excel への読み込みを行う。手順 2A-1～2A-4 までと異なる部分を手順 2-B1 に示す。



1. テキストファイルウィザードで1ステップ目を同様の設定で【次へ】をクリック
2. 区切り文字を【タブ】に加え、【その他】をチェック後、空欄に【.(ピリオド)】を入力
3. 【連続した区切り文字は1文字として扱う】にチェック後、【次へ】をクリック
4. 最終ウィザードで【完了】をクリック

手順 2-B1

C. A のファイルに B で読み込んだシートを手順 2-C1 のように、一つの Excel ファイルとして、保存する。なお、シート名は、**【NC_000913】**と**【SRR072749.soap_result1】**とする。



手順 2-C1

６－２－３．マッピング結果の集計

大腸菌ゲノムの遺伝子ごとに、各遺伝子コード領域内に何件の新型シーケンサー配列データがマップされていたかを調べる。集計結果をグラフ化する。以下の手順 3-A1 を参考に作成しよう。

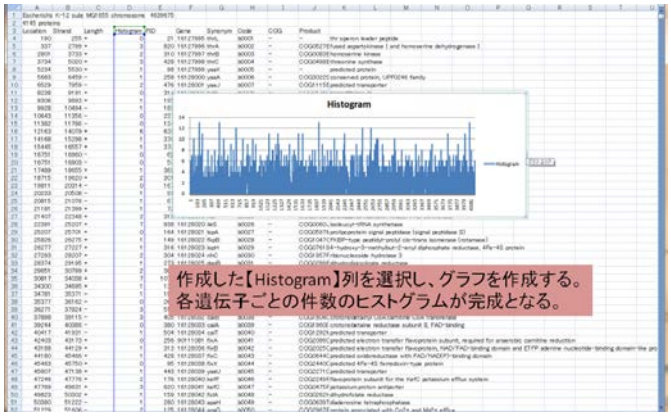
A. 各遺伝子ごとの集計

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Escherichia	K-12 subs	MG1655	chromosome	4639675										
2	4145 proteins														
3	Location	Strand	Length	Histogram	PID	Gene	Synonym	Code	COG	Product					
4	190	255	+	0	21	16127995	thrL	b0001	-	- thr operon leader peptide					
5	337	2799	+		820	16127996	thrA	b0002	-	COG0527Efused aspartokinase I and homoserine dehydrogenase I					
6	2801	3733	+		310	16127997	thrB	b0003	-	COG0083Ehomoserine kinase					
7	3734	5020	+		428	16127998	thrC	b0004	-	COG0498Ethreonine synthase					

手順３-A１

1. 【NC_000913】シート上で、D 列に列を挿入し、**【Histogram】**と 1 行目に記述する。
2. D2 セルに、
   【 =COUNTIF(SRR072749.soap_result1!$B$1:$B$10000,"<="&B4)-COUNTIF(SRR072749.soap_result1!$B$1:$B$10000,"<="&A4)】と関数を入力する。ここでは、”250 以下を満たす**【SRR072749.soap_result1】**シートの B 列の件数をカウント数- 190 以下を満たす**【SRR072749.soap_result1】**シートの B 列の件数をカウント数”を計算している。結果として、190 以上 250 以下の領域にマップされた配列件数を算出している。
3. すべての遺伝子に対し、実行する。

B. 集計結果のグラフ化



手順 3-B1

課題 4

A) (A)の BLAST の集計結果にて、新型シーケンサー配列データがマップされていた件数の上位２０遺伝子名を記述せよ。重複が多い場合には、そのうちの 20 件を記述するのみでよい。

B) (A)の BLAST の集計結果にて、新型シーケンサー配列データがマップされていた件数の下位２０遺伝子名を記述せよ。重複が多い場合には、そのうちの 20 件を記述するのみでよい。

2011 年 3 月 30 日 初版発行

発行： 長浜バイオ大学
生体分子情報学研究室（池村研究室）
阿部貴志、上原啓史、池村淑道
〒526-0829　滋賀県長浜市田村町 1266 番地
TEL　0749-64-8100（代）
FAX　0749-64-8126
URL　http://www.nagahama-i-bio.ac.jp/
MAIL　h_uehara@nagahama-i-bio.ac.jp
