

de novoアセンブリの理論の話(スライド134–147あたり)は、当日省略予定です。が、第3部初日(2016年08月01日)の、k-mer出現頻度分布の解釈や、シーケンスエラー由来k-merの除去などとも関連しますので、必要に応じて予習(または復習)しておいてください。W8(スライド159–168あたり)も状況次第で省く

第3部:NGS解析(中～上級)

～Linux環境でのデータ解析:マッピング、トリミング、アセンブリ～

東京大学・大学院農学生命科学研究科
アグリバイオインフォマティクス教育研究プログラム
門田幸二(かどた こうじ)
kadota@iu.a.u-tokyo.ac.jp
<http://www.iu.a.u-tokyo.ac.jp/~kadota/>



利用プログラムの簡単な解説

- Biostrings
 - 塩基配列の各種解析を行うためのRパッケージ。トリミング用として利用
- FaQCs (Lo and Chain, *BMC Bioinformatics*, 2014)
 - Quality Control用プログラム。クオリティフィルタリングやアダプター除去が主目的
- FastQC
 - Quality Control用プログラム。アダプターの混入などNGSデータのクオリティチェックが主目的
- FASTX-Toolkit
 - FASTAやFASTQ形式ファイルの簡単な処理を行うためのツール群。fastx_trimmerを利用
- QuasR (Gaidatzis et al., *Bioinformatics*, 2015)
 - (主に)マッピングからカウント情報取得まで行ってくれるRパッケージ
- Rockhopper2 (Tjaden, B., *Genome Biol.*, 2015)
 - バクテリア用 *de novo*トランスクリプトームアセンブラ
- Velvet (Zerbino and Birney, *Genome Res.*, 2008)
 - *de novo*ゲノムアセンブラ

おさらい

■ オリジナル(SRR616268)

- 乳酸菌paired-end RNA-seqデータで、最初の100万リードのみ抽出
- forward側(SRR616268sub_1.fastq.gz)のリード長は107 bp
- reverse側(SRR616268sub_2.fastq.gz)のリード長は93 bp

①Rockhopper2の結果に苦悩しつつ、気を取り直してRパッケージQuasRによる乳酸菌リファレンスゲノム配列へのマッピングを行うべく、Linux環境でのRの基本的な利用法を学習したのが2016年08月01日



■ FaQCs実行結果(W1-1)

- 1,000,000リード → 977,202リード (W1-3)
- forward側(QC.1.trimmed.fastq)
- reverse側(QC.2.trimmed.fastq)
- リード長はバラバラ。FastQC上で見られるIllumina adapterは消滅状態

■ *de novo*トランスクriプトームアセンブリ(Rockhopper 2)実行結果

- paired-end (QC.1.trimmed.fastqとQC.2.trimmed.fastq) : **0** transcript or contig (W5-2)
- single-end (forward側のみ; QC.1.trimmed.fastq) : **1** transcript (W6-2)
- single-end (reverse側のみ; QC.2.trimmed.fastq) : **423** transcripts (W6-4)

①

第5回原稿PDFのp195

として、FastQCによるクオリティチェックを行えばよい [W1-2]。著者らは、FastQC実行結果ファイルの項目(Overrepresented sequences)を眺めて、トリム前に見えていた既知のアダプターやプライマー配列が、トリム後に正しく見えなくなっていることを確認して安心している [W1-3]。

このデータに関して結論からいえば、forward側の107 bp のリードファイル (SRR616268sub_1.fastq.gz → QC1.trimmed.fastq) のうち、100-107 塩基付近に乳酸菌に由来しないものがトリムしきれずに多く残っている。これは、アセンブルやマッピングがうまくできない、という実害を被ることでわかる。計算時間がかかるため、できるだけ QC 段階で問題解決するという方針もある。しかし、やってみてはじめてわかることがある。以降の内容は、著者らが実際に行ったことを問題解決に至る思考回路とともに述べる。大まかに述べると、Rockhopper2¹⁸⁾によるトランスクリプトームアセンブリ、QuasR¹⁹⁾による乳酸菌ゲノムへのマッピング、そして QC 再実行である。

トランスクリプトームアセンブリ

ゲノムのアセンブリは、断片化されたゲノム配列由来リードをつなぎ合わせて、元のゲノム配列を再構築する作業である。この再構築に相当する英語がアセンブリ (assembly) であり、再構築を行うプログラムをアセンブラー (assembler) という。デノボ (*de novo*) という言葉が同時に用いられることが多いが、これは「最初から」と

か「一から」という意味である。入力として（つまり他の情報として利用せずに）ノックアラスムルする際には、*de novo assembly* という表現がなされる。トランスクリプトームアセンブリとは、アセンブリ対象がゲノムではなく解析サンプル中で発現している全転写物（トランスクリプトーム）の場合を指す。RNA-seq データのみを入力として一からアセンブリする場合は、*de novo transcriptome assembly* と呼ばれる。

今日の前半は、①RパッケージQuasRによる乳酸菌ゲノムへのマッピングの話。原因を特定しアセンブルやマッピングが改善されたことを確認するまでが第5回原稿内容

このデータに関して結論からいえば、forward側の107 bp のリードファイル (SRR616268sub_1.fastq.gz → QC1.trimmed.fastq) のうち、100-107 塩基付近に乳酸菌に由来しないものがトリムしきれずに多く残っている。これは、アセンブルやマッピングがうまくできない、という実害を被ることでわかる。①計算時間がかかるため、できるだけ QC 段階で問題解決するという方針もある。しかし、やってみてはじめてわかることがある。以降の内容は、著者らが実際に行ったことを問題解決に至る思考回路とともに述べる。大まかに述べると、Rockhopper2¹⁸⁾によるトランスクリプトームアセンブリ、QuasR¹⁹⁾による乳酸菌ゲノムへのマッピング、そして QC 再実行である。

Contents

- Illumina HiSeqデータ(トランスクリプトーム)の乳酸菌ゲノムへのマッピング
 - W14:QuasRパッケージを用いたマッピングの事前準備と本番
 - W15:結果の解説、forward側の100–107bp付近に問題があることを特定
- トリミング、*de novo*トランスクリプトームアセンブリとマッピングの再実行
 - W16:問題のある領域(forward側の100–107bp)のトリミング
 - W17:トリム後のデータでアセンブリを再実行(Rockhopper2; クラスパス設定関連Tips含む)
 - W18:トリム後のデータでマッピングを再実行(QuasR)
 - W19:トリム前のデータでクオリティチェックを再実行(FastQC)
- Illumina MiSeqデータ(乳酸菌ゲノム)の特徴と前処理
 - W4:FastQC、W5:FaQCs、W6:再度FastQC
- *de novo*ゲノムアセンブリ
 - W7:Bio-LinuxにプレインストールされているVelvet (ver. 1.2.09)を上限のk=31で実行
 - W8:k=31のアセンブリ結果をRで確認。k=141で実行し、k=31の結果と同じになるのを確認
 - W9:Velvet (ver. 1.2.10)のインストール
 - W10:Velvet (ver. 1.2.10)の実行



基本はコピペ

(Rで)塩基配列解析

～NGS、RNA-seq、ゲノム、トランскриプトーム、正規化、発現変動、統計
(last modified: 書籍 | 日本乳酸菌学会誌 |について (last modified 2016/05/12)

- 書籍 | 日本乳酸菌学会誌 | について (last modified 2016/05/12)
 - 書籍 | 日本乳酸菌学会誌 | 第1回イントロダクション (last modified 2016/05/12)
 - 書籍 | 日本乳酸菌学会誌 | 第2回GUI環境からコマンドライン環境
 - 書籍 | 日本乳酸菌学会誌 | 第3回Linux環境構築からNGSデータ処理
 - 書籍 | 日本乳酸菌学会誌 | 第4回クオリティコントロールとプロセス最適化
 - 書籍 | 日本乳酸菌学会誌 | 第5回アセンブル、マッピング、そしてSNP検出
 - 書籍 | 日本乳酸菌学会誌 | 第6回ゲノムアセンブリ (last modified 2016/05/12)

書籍 | 日本乳酸菌学会誌 | 第5回アセンブル、マッピング、そしてQC

日本乳酸菌学会誌の第5回分です。Linuxコマンドのリンク先は主に日経BP社様です。

- [第5回分PDF](#)
 - ウェブ資料PDF(2015.12.22版: 約14MB)

Linuxコマンド

- bzip2 (bzip2圧縮、および解凍)
 - cd (ディレクトリを変更)
 - echo (文字列を表示)
 - export (変数を追加)
 - file (ファイルタイプを判定)

①乳酸菌NGS連載第5回のサイト。②W14-1からスタート。例えば次のスライドは③のコピペ実行結果のスクリーンショット。私は手打ちに忙殺されて全体像の理解が追い付かないという結果になるほうが無様だと思う派ですが…、考え方はヒトそれぞれ。どうしても手打ちしたければタブ補完を有効利用して気合いでついてきましょう

マッピング (R ver. 3.2.0; QuasR ver. 1.8.4)

- ### ・入力ファイルの準備と行数確認[W14-1]

```
cd ~/Documents/srp017156  
ls
```

```
ls result2  
gzip result2/*.fastq  
ls result2
```

```
cp result2/QC.1.trimmed.fastq.gz  
cp result2/QC.2.trimmed.fastq.gz  
ls
```

```
gunzip -c SRR616268sub_1.fastq.gz | wc  
gunzip -c SRR616268sub_2.fastq.gz | wc  
gunzip -c QC.1.trimmed.fastq.gz | wc  
gunzip -c QC.2.trimmed.fastq.gz | wc
```

W14-1 : 準備

①作業ディレクトリは「~/Documents/srp017156」。②ここで見
えている2つのgzファイルは、の100万リード(400万行)からな
るpaired-end RNA-seqデータ(連載第3回W25あたりで作成)

The screenshot shows a Linux desktop environment. On the left, there is a vertical docked application bar with icons for various applications like a terminal, file manager, browser, and email. A red arrow labeled '①' points to the terminal icon. The main window is a terminal window titled 'Terminal'. It shows the following command history:

```
iu@bielinux[genomes] cd ~/Documents/srp017156 [12:06午後]
iu@bielinux[srp017156] ls [12:06午後]
result2 SRR616268sub_1.fastq.gz SRR616268sub_2.fastq.gz [12:06午後]
iu@bielinux[srp017156]
```

A red arrow labeled '②' points to the timestamp '[12:06午後]' next to the second file entry.

W14-1 : 準備

①「~/Documents/srp017156/result2」ディレクトリ上にある*.fastqファイルがFaQCs実行結果。確認。②この中の*.fastqを満たすファイル(赤線の3つ)をgzip圧縮。数分

```
File Edit View Terminal Help [12:14]
iu@bielinux[genomes] cd ~/Documents/srp017156 [12:06午後]
iu@bielinux[srp017156] ls [12:06午後]
result2 SRR616268sub_1.fastq.gz SRR616268sub_2.fastq.gz
iu@bielinux[srp017156] ls result2 [12:06午後]
fastqCount.txt QC.1.trimmed.fastq QC.unpaired.trimmed.fastq
JSLAB5_1.R QC.2.trimmed.fastq result_JSLAB1.txt
JSLAB5_2.R QC_qc_report.pdf Rockhopper_Results
nohup.out QC.stats.txt
iu@bielinux[srp017156] gzip result2/*.fastq [12:13午後]
iu@bielinux[srp017156] ls result2 [12:14午後]
fastqCount.txt QC_qc_report.pdf
JSLAB5_1.R QC.stats.txt
JSLAB5_2.R QC.unpaired.trimmed.fastq.gz
nohup.out result_JSLAB1.txt
QC.1.trimmed.fastq.gz Rockhopper_Results
QC.2.trimmed.fastq.gz
iu@bielinux[srp017156] [12:14午後]
```

W14-1 : 準備

①マッピングしたいのは赤下線の2つのファイルのみ。②これらをカレントディレクトリにコピー。③つまりココ。②のcpコマンドの最後のピリオド(.)はコピー先をカレントディレクトリにするという意味

```
File Edit View Search Terminal Help [12:22 午後]
iu@bielinux[genomes] cd ~/Documents/srp017156 [12:06 午後]
iu@bielinux[srp017156] ls [12:06 午後]
result2 SRR616268sub_1.fastq.gz SRR616268sub_2.fastq.gz
iu@bielinux[srp017156] ls result2 [12:06 午後]
fastqCount.txt QC.1.trimmed.fastq QC.unpaired.trimmed.fastq
JSLAB5_1.R QC.2.trimmed.fastq result_JSLAB1.txt
JSLAB5_2.R QC_qc_report.pdf Rockhopper_Results
nohup.out QC.stats.txt
iu@bielinux[srp017156] gzip result2/*.fastq [12:13 午後]
iu@bielinux[srp017156] ls result2 [12:14 午後]
fastqCount.txt QC_qc_report.pdf
JSLAB5_1.R QC.stats.txt
JSLAB5_2.R QC.unpaired.trimmed.fastq.gz
nohup.out result_JSLAB1.txt
QC.1.trimmed.fastq.gz Rockhopper_Results
QC.2.trimmed.fastq.gz
iu@bielinux[srp017156] cp result2/QC.1.trimmed.fastq.gz .
iu@bielinux[srp017156] cp result2/QC.2.trimmed.fastq.gz .
iu@bielinux[srp017156] ls [12:22 午後]
QC.1.trimmed.fastq.gz SRR616268sub_1.fastq.gz
QC.2.trimmed.fastq.gz SRR616268sub_2.fastq.gz
result2
iu@bielinux[srp017156] [12:22 午後]
```

W14-1：行数確認

```
File Edit View Terminal Help
iu@bielinux[srp017156] gzip result2/*.fastq
iu@bielinux[srp017156] ls result2
fastqCount.txt          QC_qc_report.pdf
JSLAB5_1.R              QC.stats.txt
JSLAB5_2.R              QC.unpaired.trimmed.fastq.gz
nohup.out                result_JSLAB1.txt
QC.1.trimmed.fastq.gz   Rockhopper_Results
QC.2.trimmed.fastq.gz
iu@bielinux[srp017156] cp result2/QC.1.trimmed.fastq.gz .
iu@bielinux[srp017156] cp result2/QC.2.trimmed.fastq.gz .
iu@bielinux[srp017156] ls
QC.1.trimmed.fastq.gz   SRR616268sub_1.fastq.gz
QC.2.trimmed.fastq.gz   SRR616268sub_2.fastq.gz
result2
iu@bielinux[srp017156] gunzip -c SRR616268sub_1.fastq.gz | wc
40000000 8000000 320760716
iu@bielinux[srp017156] gunzip -c SRR616268sub_2.fastq.gz | wc
40000000 8000000 290760716
iu@bielinux[srp017156] gunzip -c QC.1.trimmed.fastq.gz | wc
3908808 7817616 313285638
iu@bielinux[srp017156] gunzip -c QC.2.trimmed.fastq.gz | wc
3908808 7817616 281534300
iu@bielinux[srp017156]
```

-cオプションをつけて元ファイルを残したまま gzip圧縮ファイルを解凍。パイプ(|)でそのまま行数をカウントするwcコマンドに流すことで、元ファイルを変更することなくgzファイルの行数情報を得ることができる。①FaQCs実行前(pre)のファイルは4,000,000行、②実行後(post)のファイルは3,908,808行であることがわかる

[12:22午後]

←

①

[12:28午後]

②

W14-2:リストファイル

```
File Edit View Search Terminal Help  
iu@bielinux[srp017156] pwd  
/home/iu/Documents/srp017156  
iu@bielinux[srp017156] ls  
QC.1.trimmed.fastq.gz SRR616268sub_1.fastq.gz  
QC.2.trimmed.fastq.gz SRR616268sub_2.fastq.gz  
result2  
iu@bielinux[srp017156] wget -cq http://www.iu.a.u-tokyo.ac.jp/~kadota/book/JSLAB5_4.txt  
iu@bielinux[srp017156] ls  
JSLAB5_4.txt result2  
QC.1.trimmed.fastq.gz SRR616268sub_1.fastq.gz  
QC.2.trimmed.fastq.gz SRR616268sub_2.fastq.gz  
iu@bielinux[srp017156]
```

①リストファイルの作成(正確にはダウンロード)と確認。QuasRは複数サンプルのマッピングが可能。ここでは、FaQCs実行前(pre)と実行後(post)のpaired-endファイルをリストとして与えてマッピングを実行するつもり

[2:03 午後]

[2:03 午後]

[2:03 午後]

W14-2: wget失敗時は

```
File Edit View Search Terminal Help
iu@bielinux[srp017156] pwd
/home/iu/Documents/srp017156
iu@bielinux[srp017156] ls
QC.1.trimmed.fastq.gz SRR616268sub_1.fastq.gz
QC.2.trimmed.fastq.gz SRR616268sub_2.fastq.gz
result2
iu@bielinux[srp017156] wget -cq http://www.iu.a.u-tokyo.ac.jp/~kadota/book/JSLAB5_4.txt
iu@bielinux[srp017156] ls
JSLAB5_4.txt
result2
QC.1.trimmed.fastq.gz SRR616268sub_1.fastq.gz
QC.2.trimmed.fastq.gz SRR616268sub_2.fastq.gz
iu@bielinux[srp017156]
```

[2:03 午後]

[2:03 午後]

[2:03 午後]

①

• リストファイル[W14-2]
[JSLAB5_4.txt](http://www.iu.a.u-tokyo.ac.jp/~kadota/book/JSLAB5_4.txt)の作成(正確にはダウンロード)。wgetで失敗したヒトは「cp ~/Desktop/backup/JSLAB5_4.txt .」で各自対処してください。

cd ~/Documents/srp017156
pwd
ls
wget -cq http://www.iu.a.u-tokyo.ac.jp/~kadota/book/JSLAB5_4.txt
#cp ~/Desktop/backup/JSLAB5_4.txt .
ls
more JSLAB5_4.txt

②

W14-2:リストファイル

①リストファイルの中身を確認。paired-endの場合は、1行目の部分は、②「FileName1
FileName2 SampleName」と書く(固定)

```
File Edit View Search Terminal Help [ 1:48 午後 ]
iu@bielinux[srp017156] pwd
/home/iu/Documents/srp017156
iu@bielinux[srp017156] ls
QC.1.trimmed.fastq.gz SRR616268sub_1.fastq.gz
QC.2.trimmed.fastq.gz SRR616268sub_2.fastq.gz
result2
iu@bielinux[srp017156] wget -cq http://www.iu.a.u-tokyo.ac.jp/~kadota/book/JSLAB5_4.txt
iu@bielinux[srp017156] ls
JSLAB5_4.txt
result2
QC.1.trimmed.fastq.gz SRR616268sub_1.fastq.gz
QC.2.trimmed.fastq.gz SRR616268sub_2.fastq.gz
iu@bielinux[srp017156] more JSLAB5_4.txt
FileName1      FileName2      SampleName
SRR616268sub_1.fastq.gz SRR616268sub_2.fastq.gz pre
QC.1.trimmed.fastq.gz   QC.2.trimmed.fastq.gz   post
iu@bielinux[srp017156] [ 1:48 午後 ]
```

③2行目以降にマッピング
したい入力ファイル名を書く

W14-2:リストファイル

```
File Edit View Search Terminal Help [ 1:48 午後 ]
iu@bielinux[srp017156] pwd
/home/iu/Documents/srp017156
iu@bielinux[srp017156] ls [ 1:48 午後 ]
QC.1.trimmed.fastq.gz SRR616268sub_1.fastq.gz
QC.2.trimmed.fastq.gz SRR616268sub_2.fastq.gz
result2
iu@bielinux[srp017156] wget -cq http://www.iu.a.u-tokyo.ac.jp/~kadota/book/JSLAB5_4.txt ←
iu@bielinux[srp017156] ls [ 1:48 午後 ]
JSLAB5_4.txt result2
QC.1.trimmed.fastq.gz SRR616268sub_1.fastq.gz
QC.2.trimmed.fastq.gz SRR616268sub_2.fastq.gz
iu@bielinux[srp017156] more JSLAB5_4.txt [ 1:48 午後 ]
FileName1 FileName2 SampleName
SRR616268sub_1.fastq.gz SRR616268sub_2.fastq.gz pre } post
QC.1.trimmed.fastq.gz QC.2.trimmed.fastq.gz
iu@bielinux[srp017156] [ 1:48 午後 ]
```

The screenshot shows a terminal window with a scroll bar on the right. On the left, there is a vertical docked application bar with icons for various applications like a terminal, file manager, and browser.

- Annotation 1:** A red arrow points to the icon for the application containing the terminal window.
- Annotation 2:** A red arrow points to the "SampleName" column header in the table.
- Annotation 3:** A red arrow points to the brace indicating the grouping of "pre" and "post" samples.

W14-2:リストファイル

```
File Edit View Search Terminal Help  
iu@bielinux[srp017156] pwd  
/home/iu/Documents/srp017156  
iu@bielinux[srp017156] ls  
QC.1.trimmed.fastq.gz SRR616268sub_1.fastq.gz  
QC.2.trimmed.fastq.gz SRR616268sub_2.fastq.gz  
result2  
iu@bielinux[srp017156] wget -cq http://www.iu.a.u-tokyo.ac.jp/~kad←  
ota/book/JSLAB5_4.txt  
iu@bielinux[srp017156] ls  
JSLAB5_4.txt result2  
QC.1.trimmed.fastq.gz SRR616268sub_1.fastq.gz  
QC.2.trimmed.fastq.gz SRR616268sub_2.fastq.gz  
iu@bielinux[srp017156] more JSLAB5_4.txt  
FileName1      FileName2      SampleName  
SRR616268sub_1.fastq.gz SRR616268sub_2.fastq.gz pre  
QC.1.trimmed.fastq.gz QC.2.trimmed.fastq.gz post  
iu@bielinux[srp017156]
```

①1列目(赤下線部分)はforward側のファイル、
②2列目(黒下線部分)はreverse側のファイル、
③3列目(緑下線部分)は任意のサンプル名。つまりpreやpostの部分は、自由に変えてよい

[1:48 午後]

W14-3: Rスクリプト

```
File Edit View Search Terminal Help
iu@bielinux[srp017156] wget -cq http://www.dota/book/JSLAB5_5.R
iu@bielinux[srp017156] nkf JSLAB5_5.R
in_f1 <- "JSLAB5_4.txt"          #入力ファイル名を指定してin_f1に格納( RNA-seq リストファイル )
in_f2 <- "/home/iu/Documents/genomes/Lactobacillus_casei_12a.GCA_000309565.2.30.dna.toplevel.fa" #入力ファイル名を指定してin_f2に格納(リファレンス配列 )
#必要なパッケージをロード
library(QuasR)                  #パッケージの読み込み
#本番(マッピング)
out <- qAlign(in_f1, in_f2)      #マッピングを行うqAlign関数を実行した結果をoutに格納
#ファイルに保存( QC レポート用の pdf ファイル作成 )
out_f <- sub(".bam", "_QC.pdf", out@alignments[,1]) #Quality Control レポートの pdf ファイル名を作成した結果をout_fに格納
qQCReport(out, pdfFilename=out_f) #QC レポート結果をファイルに保存
iu@bielinux[srp017156]
```

① Rスクリプトファイル(JSLAB5_5.R)のダウンロードと、②中身の表示。nkfは文字化け回避用。in_f1がリストファイル(W14-2)。in_f2がリファレンス配列(W13-1)。現時点では、③のリファレンス配列ファイルはまだ gzip 圧縮状態。解凍は後で行う



カラー表示。実際のコマンドはごくわずか。
①qAlign関数部分がマッピング本番。
②qQCReport関数は、PDFレポート作成用。
これはまだコピペしない!

W14-4: カラー表示

• カラー表示[W14-4]

Rスクリプトファイル [JSLAB5_5.R](#) の中身を表示。moreでみたものと基本的に同じです。in_f1には、マップしたいFASTQファイルのリストをQuasRの入力形式に従って作成したファイルの名前([JSLAB5_4.txt](#))、in_f2にはマップされる側のリファレンス配列を指定します。ここでは作業ディレクトリ上にないRelease 30の乳酸菌ゲノムファイル(解凍したファイル)を絶対パスで指定しています。

```
in_f1 <- "JSLAB5_4.txt"          #入力ファイル名を指定してin_f1に格納(RNA-seqリソース)
in_f2 <- "/home/iu/Documents/genomes/Lactobacillus_casei_12a.GCA_000309565.2.30.dna" #リファレンス配列

#必要なパッケージをロード
library(QuasR)                  #パッケージの読み込み

#本番(マッピング)
out <- qAlign(in_f1, in_f2)      #マッピングを行うqAlign関数を実行した結果をoutに

#① ファイルに保存(QCレポート用のpdfファイル作成)
out_f <- sub(".bam", "_QC.pdf", out@alignments[,1]) #Quality Controlレポートのpdfファイル
qQCReport(out, pdfFilename=out_f) #QCレポート結果をファイルに保存
```



W14-5:解凍

```
File Edit View Terminal Help Ja 13:38
iu@bielinux[srp017156] cd ~/Documents/genomes [ 1:38午後]
iu@bielinux[genomes] ls [ 1:38午後]
JSLAB5_1.R
JSLAB5_3.R
Lactobacillus_casei_12a.GCA_000309565.1.22.dna.toplevel.fa
Lactobacillus_casei_12a.GCA_000309565.2.30.dna.toplevel.fa.gz
result_JSLAB1.txt
iu@bielinux[genomes] gunzip Lactobacillus_casei_12a.GCA_000309565
.2.30.dna.toplevel.fa.gz
iu@bielinux[genomes] ls [ 1:38午後]
JSLAB5_1.R
JSLAB5_3.R
Lactobacillus_casei_12a.GCA_000309565.1.22.dna.toplevel.fa
Lactobacillus_casei_12a.GCA_000309565.2.30.dna.toplevel.fa
result_JSLAB1.txt
iu@bielinux[genomes] ■ [ 1:38午後]
```

①QuasRでのマッピング用のRスクリプトファイルJSLAB5_5.R(W14-3)を実行。約15分

W14-5: マッピング本番

The screenshot shows a terminal window with the following command history:

```
iu@bielinux[genomes] cd ~/Documents/srp017156 [ 1:40午後]
iu@bielinux[srp017156] pwd [ 1:40午後]
/home/iu/Documents/srp017156
iu@bielinux[srp017156] ls [ 1:40午後]
JSLAB5_4.txt result2
JSLAB5_5.R SRR616268sub_1.fastq.gz
QC.1.trimmed.fastq.gz SRR616268sub_2.fastq.gz
QC.2.trimmed.fastq.gz
iu@bielinux[srp017156] R --vanilla --slave < JSLAB5_5.R
```

A red arrow points to the terminal window, and a red circle with the number 1 is placed over the terminal icon in the docked application bar.

W14-5:途中経過1

iu@bielinux[~/Documents/srp017156]

```
int,  
      rownames, sapply, setdiff, sort,  
,  
      unlist, unsplit
```

Loading required package: S4Vectors

Loading required package: stats4

Creating a generic function for 'nchar' from package 'base' in pa
ckage 'S4Vectors'

Loading required package: IRanges

Loading required package: GenomeInfoDb

Loading required package: Rbowtie

Creating .fai file for: /home/iu/Documents/genomes/Lactobacillus_
casei_12a.GCA_000309565.2.30.dna.toplevel.fa

[fai_load] build FASTA index.

alignment files missing - need to:

 create alignment index for the genome

 create 2 genomic alignment(s)

Creating an Rbowtie index for /home/iu/Documents/genomes/Lactobac
illus_casei_12a.GCA_000309565.2.30.dna.toplevel.fa

Finished creating index

Testing the compute nodes...OK

Loading QuasR on the compute nodes...■

リターンキーを押して数秒後の状態。①まず最初にやっているのは、リファレンス配列のインデックス化。インデックス化(indexing)することでマッピングを高速に行うことができます。数MB程度の乳酸菌ゲノムの場合は比較的短時間(数分のオーダー)で終わりますが、ヒトゲノムだと數十分以上はかかるのではと思います。ただし、同じリファレンス配列を使って別のデータのマッピングを行う場合には、既にインデックス化されたものを使うのでこの部分はスキップできます

①

W14-5:途中経過2

File Edit View Search Terminal Help

```
ckage 'S4Vectors'  
Loading required package: IRanges  
Loading required package: GenomeInfoDb  
Loading required package: Rbowtie  
Creating .fai file for: /home/iu/Documents/genomes/Lactobacillus_  
casei_12a.GCA_000309565.2.30.dna.toplevel.fa  
[fai_load] build FASTA index.  
alignment files missing - need to:  
  create alignment index for the genome  
  create 2 genomic alignment(s)  
Creating an Rbowtie index for /home/iu/Documents/genomes/Lactobac  
illus_casei_12a.GCA_000309565.2.30.dna.toplevel.fa  
Finished creating index  
Testing the compute nodes...OK  
Loading QuasR on the compute nodes...OK  
Available cores:  
nodeNames  
bielinux  
  1  
① Performing genomic alignments for 2 samples. See progress in the  
log file:  
/home/iu/Documents/srp017156/QuasR_log_47ee3bd7d050.txt ②
```

①マッピングがスタート。この種のプログラムは実行ログファイルを作成する場合が多いです。②QuasRも絶対パスで示したファイル名にログを書き込んでいます

W14-5:途中経過3

```
x - File Edit View Search Terminal Help
Loading required package: GenomeInfoDb
Loading required package: Rbowtie
Creating .fai file for: /home/iu/Documents/genomes/Lactobacillus_casei_12a.GCA_000309565.2.30.dna.toplevel.fa
[fai_load] build FASTA index.
alignment files missing - need to:
  create alignment index for the genome
  create 2 genomic alignment(s)
Creating an Rbowtie index for /home/iu/Documents/genomes/Lactobacillus_casei_12a.GCA_000309565.2.30.dna.toplevel.fa
Finished creating index
Testing the compute nodes...OK
Loading QuasR on the compute nodes...OK
Available cores:
nodeNames
bielinux
  1
Performing genomic alignments for 2 samples. See progress in the log file:
/home/iu/Documents/srp017156/QuasR_log_47ee3bd7d050.txt
[samopen] SAM header is present: 1 sequences.
[bam_sort_core] merging from 2 files...
```

①samやbamと書かれているが、これは多くのマッピングプログラム(QuasRのデフォルトは内部的にBowtieプログラムを利用)の結果ファイルの形式がbam形式だから。bamはsamのバイナリ版

W14-5:途中経過4

①2回目のsamやbamの記述。おそらく2つめのサンプル(リストファイルの3行目。この場合FaQCs実行後のファイルQC.*.fastq.gz)のマッピングを行っているのだろう

```
File Edit View Search Terminal Help
Creating .fai file for: /home/iu/Documents/genomes/Lactobacillus_casei_12a.GCA_000309565.2.30.dna.toplevel.fa
[fai_load] build FASTA index.
alignment files missing - need to:
    create alignment index for the genome
    create 2 genomic alignment(s)
Creating an Rbowtie index for /home/iu/Documents/genomes/Lactobacillus_casei_12a.GCA_000309565.2.30.dna.toplevel.fa
Finished creating index
Testing the compute nodes...OK
Loading QuasR on the compute nodes...OK
Available cores:
nodeNames
bielinux
    1
Performing genomic alignments for 2 samples. See progress in the log file:
/home/iu/Documents/srp017156/QuasR_log_47ee3bd7d050.txt
[samopen] SAM header is present: 1 sequences.
[bam_sort_core] merging from 2 files...
[samopen] SAM header is present: 1 sequences.
[bam_sort_core] merging from 2 files...
```

①

W14-5:途中経過5

- ①マッピングは無事に終了したようだ。
- ②QC情報を得ようとしているのだろう

```
File Edit View Search Terminal Help ↑ ↓ Ja 13:53 ⚙  
alignment files missing - need to:  
    create alignment index for the genome  
    create 2 genomic alignment(s)  
Creating an Rbowtie index for /home/iu/Documents/genomes/Lactobacillus_casei_12a.GCA_000309565.2.30.dna.toplevel.fa  
Finished creating index  
Testing the compute nodes...OK  
Loading QuasR on the compute nodes...OK  
Available cores:  
nodeNames  
bielinux  
    1  
Performing genomic alignments for 2 samples. See progress in the log file:  
/home/iu/Documents/srp017156/QuasR_log_47ee3bd7d050.txt  
[samopen] SAM header is present: 1 sequences.  
[bam_sort_core] merging from 2 files...  
[samopen] SAM header is present: 1 sequences.  
[bam_sort_core] merging from 2 files...  
Genomic alignments have been created successfully  
① collecting quality control data
```

W14-5: 無事終了

```
File Edit View Terminal Help ↑ ↓ Ja 13:54 🔍
create alignment index for the genome
create 2 genomic alignment(s)
Creating an Rbowtie index for /home/iu/Documents/genomes/Lactobacillus_casei_12a.GCA_000309565.2.30.dna.toplevel.fa
Finished creating index
Testing the compute nodes...OK
Loading QuasR on the compute nodes...OK
Available cores:
nodeNames
bielinux
    1
Performing genomic alignments for 2 samples. See progress in the log file:
/home/iu/Documents/srp017156/QuasR_log_47ee3bd7d050.txt
[samopen] SAM header is present: 1 sequences.
[bam_sort_core] merging from 2 files...
[samopen] SAM header is present: 1 sequences.
[bam_sort_core] merging from 2 files...
Genomic alignments have been created successfully

collecting quality control data
creating QC plots
iu@bielinux[srp017156] [ 1:54午後 ]
```

①

Contents

- Illumina HiSeqデータ(トランスクリプトーム)の乳酸菌ゲノムへのマッピング
 - W14: QuasRパッケージを用いたマッピングの事前準備と本番
 - W15: 結果の解説、forward側の100–107bp付近に問題があることを特定
- トリミング、*de novo*トランスクリプトームアセンブリとマッピングの再実行
 - W16: 問題のある領域(forward側の100–107bp)のトリミング
 - W17: トリム後のデータでアセンブリを再実行(Rockhopper2; クラスパス設定関連Tips含む)
 - W18: トリム後のデータでマッピングを再実行(QuasR)
 - W19: トリム前のデータでクオリティチェックを再実行(FastQC)
- Illumina MiSeqデータ(乳酸菌ゲノム)の特徴と前処理
 - W4: FastQC、W5: FaQCs、W6: 再度FastQC
- *de novo*ゲノムアセンブリ
 - W7: Bio-LinuxにプレインストールされているVelvet (ver. 1.2.09)を上限のk=31で実行
 - W8: k=31のアセンブリ結果をRで確認。k=141で実行し、k=31の結果と同じになるのを確認
 - W9: Velvet (ver. 1.2.10)のインストール
 - W10: Velvet (ver. 1.2.10)の実行



W15-1：結果の解説

①lsした結果。②多数のファイル(計8ファイル)が生成されていることがわかる。
マッピング前(W14-5)と比べてみるとよい

```
File Edit View Search Terminal Help ↑ ↓ Ja 13:56 🌐  
[samopen] SAM header is present: 1 sequences.  
[bam_sort_core] merging from 2 files...  
Genomic alignments have been created successfully  
  
collecting quality control data  
creating QC plots  
iu@bielinux[srp017156] ls [ 1:54午後 ]  
JSLAB5_4.txt  
JSLAB5_5.R  
QC.1.trimmed_47ee41c6ec8b.bam } ②  
QC.1.trimmed_47ee41c6ec8b.bam.bai } ②  
QC.1.trimmed_47ee41c6ec8b.bam.txt } ②  
QC.1.trimmed.fastq.gz  
QC.2.trimmed.fastq.gz  
QuasR_log_47ee3bd7d050.txt } ②  
result2 } ②  
SRR616268sub_1_47ee4589e65f.bam } ②  
SRR616268sub_1_47ee4589e65f.bam.bai } ②  
SRR616268sub_1_47ee4589e65f.bam.txt } ②  
SRR616268sub_1_47ee4589e65f_QC.pdf } ②  
SRR616268sub_1.fastq.gz  
SRR616268sub_2.fastq.gz  
iu@bielinux[srp017156] [ 1:56午後 ]
```

W15-1：結果の解説

```
File Edit View Search Terminal Help  
[samopen] SAM header is present: 1 sequences.  
[bam_sort_core] merging from 2 files...  
Genomic alignments have been created successful  
  
collecting quality control data  
creating QC plots  
iu@bielinux[srp017156] ls  
JSLAB5_4.txt  
JSLAB5_5.R  
QC.1.trimmed_47ee41c6ec8b.bam  
QC.1.trimmed_47ee41c6ec8b.bam.bai  
QC.1.trimmed_47ee41c6ec8b.bam.txt  
QC.1.trimmed.fastq.gz  
QC.2.trimmed.fastq.gz  
QuasR_log_47ee3bd7d050.txt  
result2  
SRR616268sub_1_47ee4589e65f.bam  
SRR616268sub_1_47ee4589e65f.bam.bai  
SRR616268sub_1_47ee4589e65f.bam.txt  
SRR616268sub_1_47ee4589e65f_QC.pdf  
SRR616268sub_1.fastq.gz  
SRR616268sub_2.fastq.gz  
iu@bielinux[srp017156]
```

マッピング結果ファイルのメインは①と②で示した.bam。この形式のファイルを入力としてその後の解析を行うプログラムが多い。③エラーが出たりすることがなければlogファイルの中身をあまり見ることはないが、この中をよく見るとマッピング時に用いたオプション情報などを読み取ることができる

[1:54午後]

[1:56午後]

赤下線部分の文字列はランダムに発生させているので、ヒトによって異なる

W15-1 : 結果の解説

```
File Edit View Search Terminal Help Ja 13:56
[samopen] SAM header is present: 1 sequences.
[bam_sort_core] merging from 2 files...
Genomic alignments have been created successfully

collecting quality control data
creating QC plots
iu@bielinux[srp017156] ls [ 1:54午後 ]
JSLAB5_4.txt
JSLAB5_5.R
QC.1.trimmed_47ee41c6ec8b.bam
QC.1.trimmed_47ee41c6ec8b.bam.bai
QC.1.trimmed_47ee41c6ec8b.bam.txt
QC.1.trimmed.fastq.gz
QC.2.trimmed.fastq.gz
QuasR_log_47ee3bd7d050.txt
result2
SRR616268sub_1_47ee4589e65f.bam
SRR616268sub_1_47ee4589e65f.bam.bai
SRR616268sub_1_47ee4589e65f.bam.txt
SRR616268sub_1_47ee4589e65f_QC.pdf
SRR616268sub_1.fastq.gz
SRR616268sub_2.fastq.gz
iu@bielinux[srp017156] [ 1:56午後 ]
```

W15-1：結果の解説

①このPDFファイル中には、入力ファイル(paired-end RNA-seqリード)のQC情報や、どれだけマップされたかなどの結果がある

```
x - File Edit View Search Terminal Help ↑ ↓ Ja 13:56
[samopen] SAM header is present: 1 sequences.
[bam_sort_core] merging from 2 files...
Genomic alignments have been created successfully

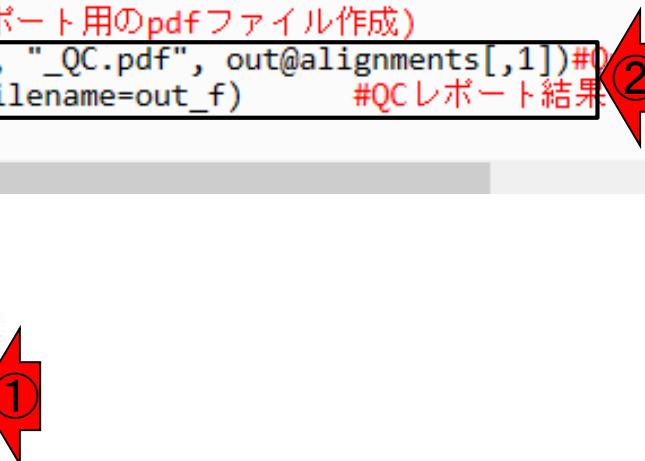
collecting quality control data
creating QC plots
iu@bielinux[srp017156] ls [ 1:54午後 ]
JSLAB5_4.txt
JSLAB5_5.R
QC.1.trimmed_47ee41c6ec8b.bam
QC.1.trimmed_47ee41c6ec8b.bam.bai
QC.1.trimmed_47ee41c6ec8b.bam.txt
QC.1.trimmed.fastq.gz
QC.2.trimmed.fastq.gz
QuasR_log_47ee3bd7d050.txt
result2
SRR616268sub_1_47ee4589e65f.bam
SRR616268sub_1_47ee4589e65f.bam.bai
SRR616268sub_1_47ee4589e65f.bam.txt
SRR616268sub_1_47ee4589e65f_QC.pdf ①
SRR616268sub_1.fastq.gz
SRR616268sub_2.fastq.gz
iu@bielinux[srp017156] [ 1:56午後 ]
```

W15-1：結果の解説

```
[samopen] SAM header is present: 1 sequences.  
[bam_sort_core] merging from 2 files...  
Genomic alignments have been created successfully  
  
collecting quality  
creating QC plots  
iu@bielinux[srp017]: JSLAB5_4.txt  
JSLAB5_5.R  
QC.1.trimmed_47ee4:  
QC.1.trimmed_47ee4:  
QC.1.trimmed_47ee4:  
QC.1.trimmed.fastq  
QC.2.trimmed.fastq  
QuasR_log_47ee3bd7  
result2  
SRR616268sub_1_47ee4589e65f.bam  
SRR616268sub_1_47ee4589e65f.bam.bai  
SRR616268sub_1_47ee4589e65f.bam.txt  
SRR616268sub_1_47ee4589e65f_QC.pdf  
SRR616268sub_1.fastq.gz  
SRR616268sub_2.fastq.gz  
iu@bielinux[srp017156]
```

①のファイルは、②の部分を実行したから生成された。わざわざ生成させたのは、Rockhopper2でアセンブルがうまくいかなかった理由が、このQCレポートファイルを眺めることでわかるから

```
in_f1 <- "JSLAB5_4.txt"          #入力ファイル名を指定してin_f1に格納(RNA-seqリソース)  
in_f2 <- "/home/iu/Documents/genomes/Lactobacillus_casei_12a.GCA_000309565.2.30.dna"  
  
#必要なパッケージをロード  
library(QuasR)                   #パッケージの読み込み  
  
#本番(マッピング)  
out <- qAlign(in_f1, in_f2)       #マッピングを行うqAlign関数を実行した結果をoutに格納  
  
#ファイルに保存(QCレポート用のpdfファイル作成)  
out_f <- sub(".bam", "_QC.pdf", out@alignments[,1])#QCレポート用のpdfファイル名を出力する  
qQCReport(out, pdfFilename=out_f)    #QCレポート結果をpdfファイルに保存
```



[1:56午後]

W15-2:リファレンスのほう

```
File Edit View Search Terminal Help  
QC.1.trimmed_47ee41c6ec8b.bam.txt  
QC.1.trimmed.fastq.gz  
QC.2.trimmed.fastq.gz  
QuasR_log_47ee3bd7d050.txt  
result2  
SRR616268sub_1_47ee4589e65f.bam  
SRR616268sub_1_47ee4589e65f.bam.bai  
SRR616268sub_1_47ee4589e65f.bam.txt  
SRR616268sub_1_47ee4589e65f_QC.pdf  
SRR616268sub_1.fastq.gz  
SRR616268sub_2.fastq.gz  
iu@bielinux[srp017156] pwd  
/home/iu/Documents/srp017156  
iu@bielinux[srp017156] ls ~/Documents/genomes  
JSLAB5_1.R  
JSLAB5_3.R  
Lactobacillus_casei_12a.GCA_000309565.1.22.dna.toplevel.fa  
Lactobacillus_casei_12a.GCA_000309565.2.30.dna.toplevel.fa  
Lactobacillus_casei_12a.GCA_000309565.2.30.dna.toplevel.fa.fai  
Lactobacillus_casei_12a.GCA_000309565.2.30.dna.toplevel.fa.md5  
Lactobacillus_casei_12a.GCA_000309565.2.30.dna.toplevel.fa.Rbowtie  
result_JSLAB1.txt  
iu@bielinux[srp017156]
```

①リファレンスゲノムファイルがあるディレクトリをls。②リファレンスとして指定したファイル。③W14-5の最初でリファレンスゲノムのインデックス化を行っていたが、そのときに作成されたのが赤枠の3ファイル。「

~/Documents/genomes」の所有者が自分なので、これらのファイルを作成することができた。が、スパコンなどで共用のリファレンスゲノムのディレクトリを利用する際には、書き込み権限がないことに起因するエラーが起こるかもしれない記憶に留めておこう

[2:06 午後]

[2:06 午後]

①

②

③

W15-3: QCレポート

```
File Edit View Search Terminal Help  
iu@bielinux[srp017156] pwd  
/home/iu/Documents/srp017156  
iu@bielinux[srp017156] ls ~/Desktop/mac_share  
JSLAB4_1.sh  
Lactobacillus_casei_12a.GCA_000309565.1.22.dna.toplevel.fa  
QC.1.trimmed_fastqc.html  
QC.1.trimmed_fastqc.zip  
QC.2.trimmed_fastqc.html  
QC.2.trimmed_fastqc.zip  
SRR616268sub_1_fastqc.html  
SRR616268sub_1_fastqc.zip  
SRR616268sub_2_fastqc.html  
SRR616268sub_2_fastqc.zip  
iu@bielinux[srp017156] cp *.pdf ~/Desktop/mac_share  
iu@bielinux[srp017156]
```

QuasRでマッピングしたのは、QC
レポートを眺めるのが主目的。こ
こでは、①pdfファイルを共有フォ
ルダ(~/Desktop/mac_share)にコ
ピーしてホストOS上で眺めるが…

[2:09 午後]

[2:09 午後]

[2:09 午後]

①

W15-3: QCレポート

①引出しアイコンをクリックしていって
ゲストOS上で眺めてもよい。反応が遅
いのでイラッとするが気長に待つべし



Ubuntu desktop environment showing a file browser window.

The file browser shows a list of files in the "Documents" folder:

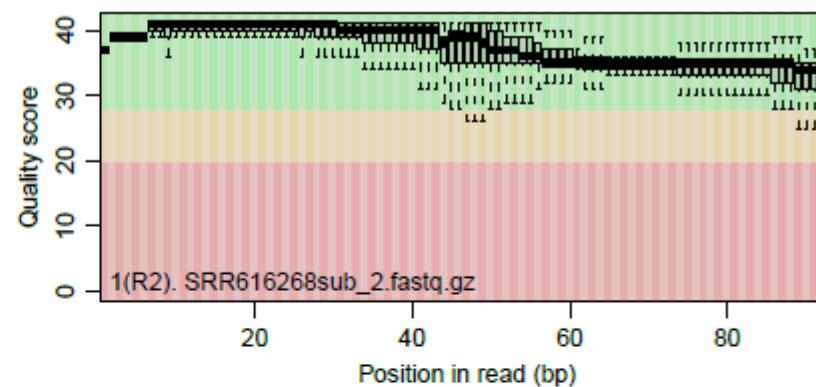
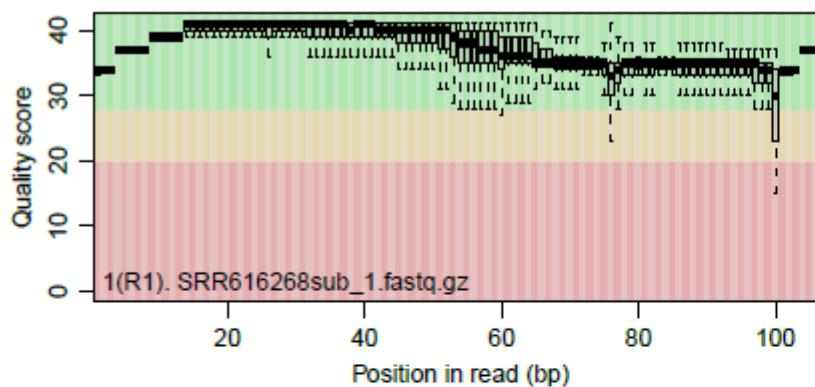
Name	Size	Type	Modified
result2	11 items	Folder	11:36
SRR616268sub_1_47ee4589e65f_QC.pdf	80.4 kB	Document	13:54
QuasR_log_47ee3bd7d050.txt	1.7 kB	Text	13:53
QC.1.trimmed_47ee41c6ec8b.bam.txt	620 bytes	Text	13:53
QC.1.trimmed_47ee41c6ec8b.bam.bai	6.2 kB	Binary	13:53
QC.1.trimmed_47ee41c6ec8b.bam	138.9 MB	Archive	13:53
SRR616268sub_1_47ee4589e65f.bam.txt	624 bytes	Text	13:47
SRR616268sub_1_47ee4589e65f.bam.bai	6.2 kB	Binary	13:47
SRR616268sub_1_47ee4589e65f.bam	142.1 MB	Archive	13:47
JSLAB5_5.R	698 bytes	Text	13:23
QC.2.trimmed.fastq.gz	65.9 MB	Archive	11:37
QC.1.trimmed.fastq.gz	73.7 MB	Archive	11:37
SRR616268sub_2.fastq.gz	68.7 MB	Archive	12月 9
SRR6162 "SRR616268sub_1_47ee4589e65f_QC.pdf" selected (80.4 kB)			

Red numbered arrows indicate the following steps:

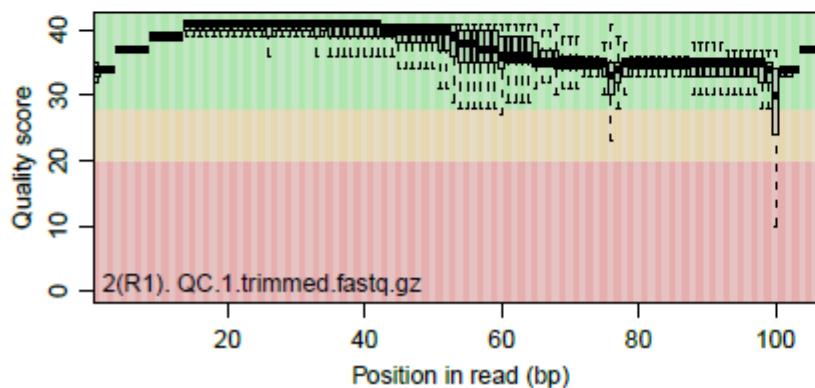
- ① Points to the Dash icon in the top left corner.
- ② Points to the folder name "srp017156" in the title bar.
- ③ Points to the PDF file "SRR616268sub_1_47ee4589e65f_QC.pdf" in the list.

W15-4 : PDF解説

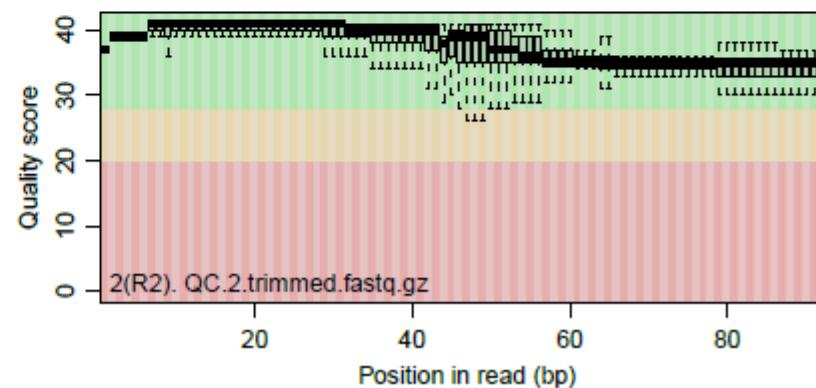
PDF1枚目。入力ファイルのQuality score分布。
FastQC Report中の項目「Per base sequence quality」と同じ。①上段がFaQCs実行前(pre)、②下段が実行後(post)。③左がforward側、④右がreverse側。ここでの目的はFaQCs実行前後の比較ではなく、マップされなかつたリードの割合や、数少ないマップされたリードの調査なので、劇的な違いはないが気にしない



① FaQCs
実行前
(pre)



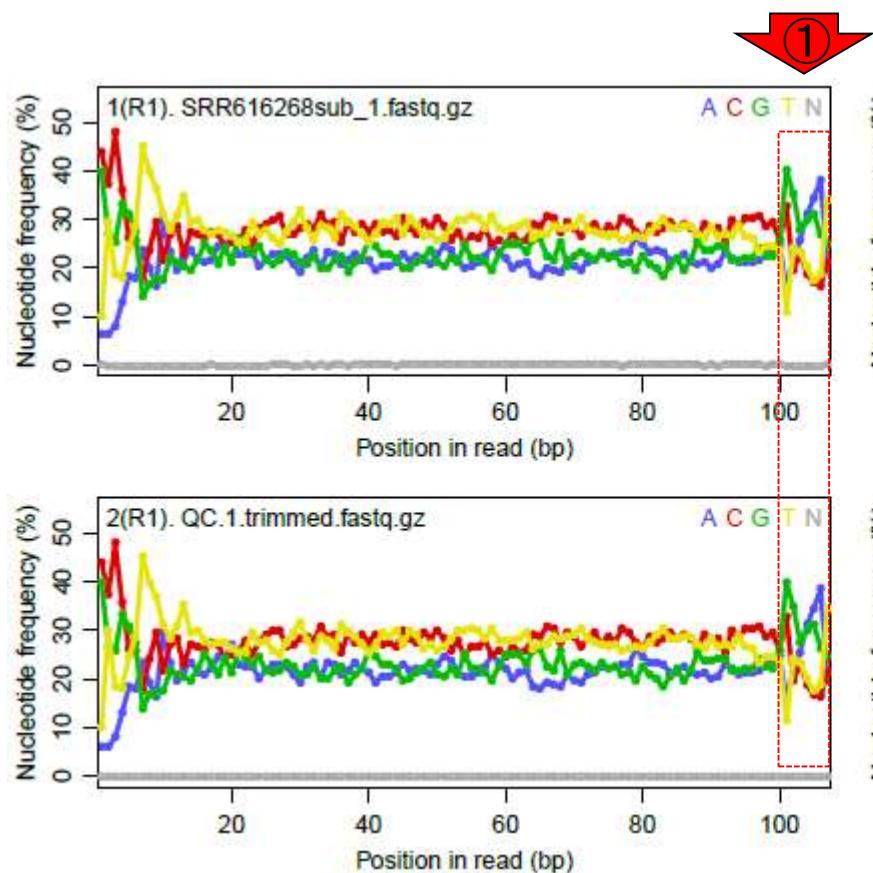
③ forward側



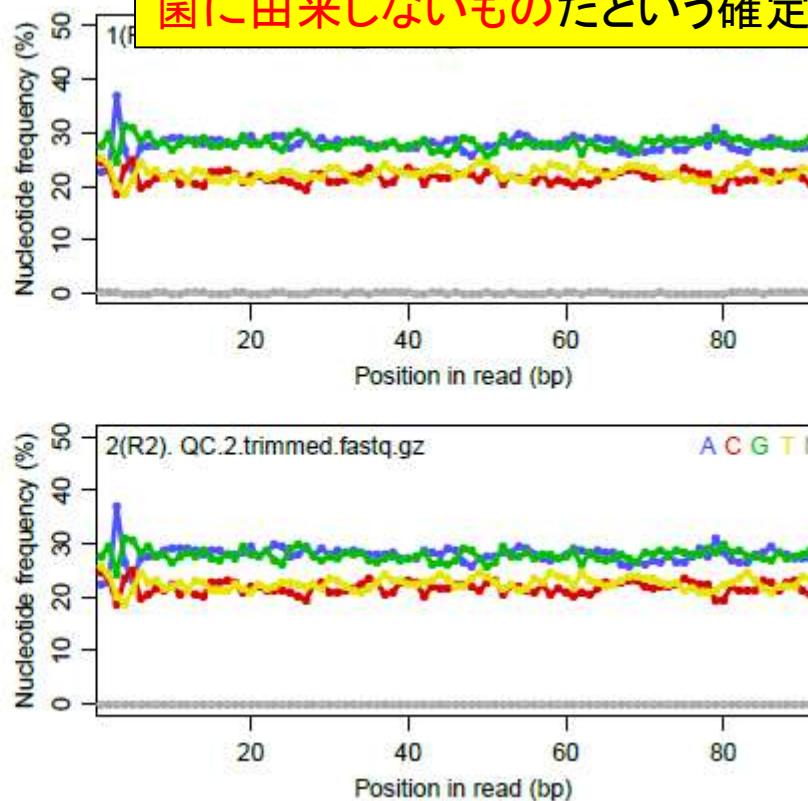
② FaQCs
実行後
(post)

④ reverse側

W15-5: PDF解説

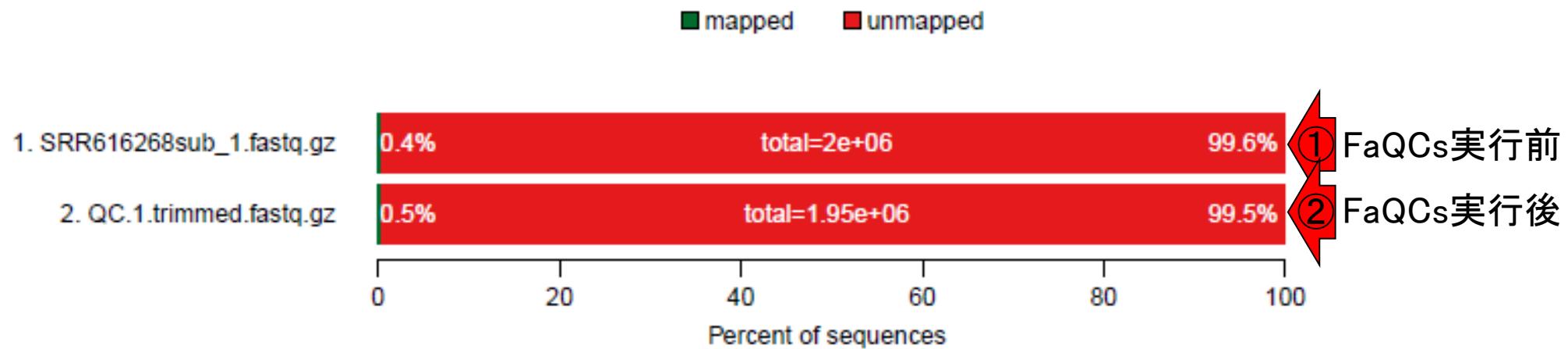


PDF2枚目。ポジションごとの塩基の出現確率。
FastQC Report中の項目「Per base sequence content」と同じ(但し色は異なる)。①forward側の
100-107bp付近(赤枠部分)の分布が不自然。このよ
うな結果は、FastQCをデフォルトオプションで実行す
ると得られない。この結果と後のほうのPDFレポート
と合わせることで、**これがトリムしきれていない乳酸
菌に由来しないものだ**という確定診断が下される



W15-6: PDF解説

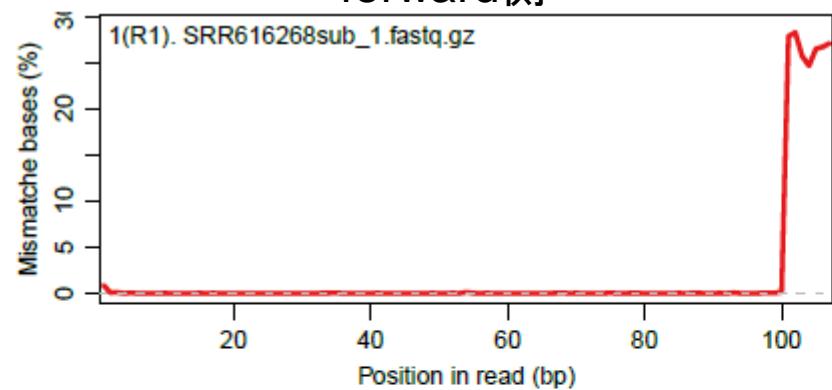
PDF4枚目。全リード(forward, reverse合わせて約200万リード)のうち、マップされたリードの割合は①FaQCs実行前(pre)が0.4%、②実行後(post)が0.5%。ほとんどマップされなかつたことを意味する



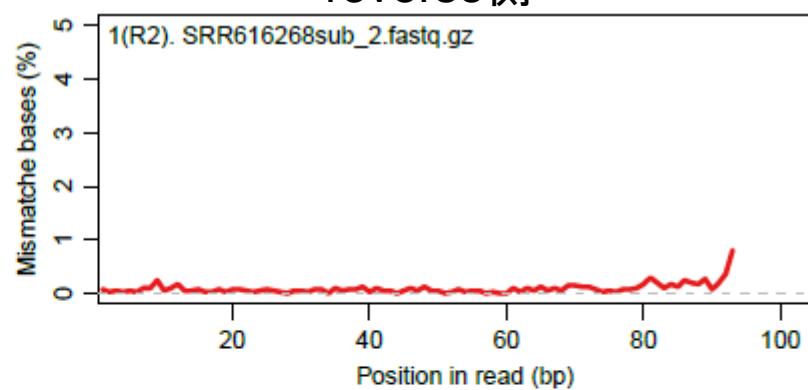
W15-7: PDF解説

PDF6枚目。マニュアルを読んでもよくわからなかつたが、おそらく縦軸がMismatch basesとなつてゐるので、ミスマッチを許容してマップされたリードの中でどこにミスマッチがあつたかを表示しているものと思われる。多少解釈が間違つてゐたとしても、①このプロット分布を見れば、「forward側の100–107 bp付近が犯人」という結論は不変

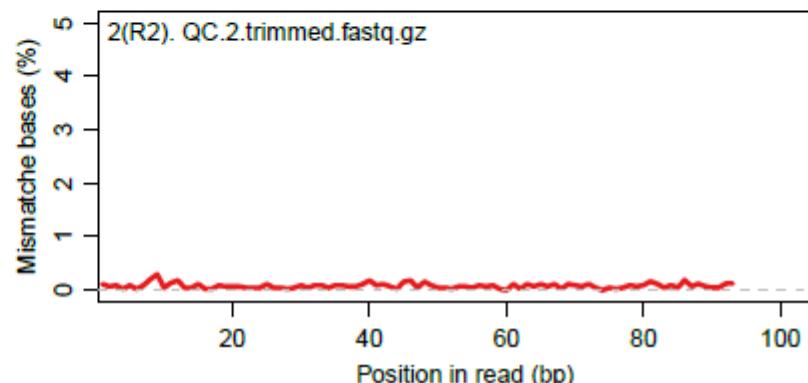
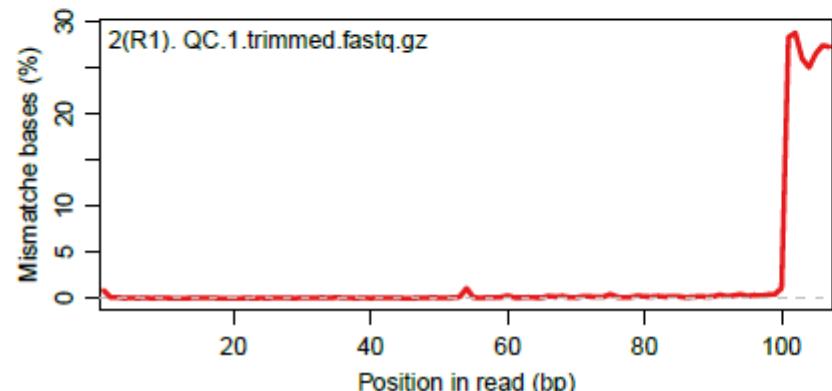
forward側



reverse側



FaQCs
実行前
(pre)

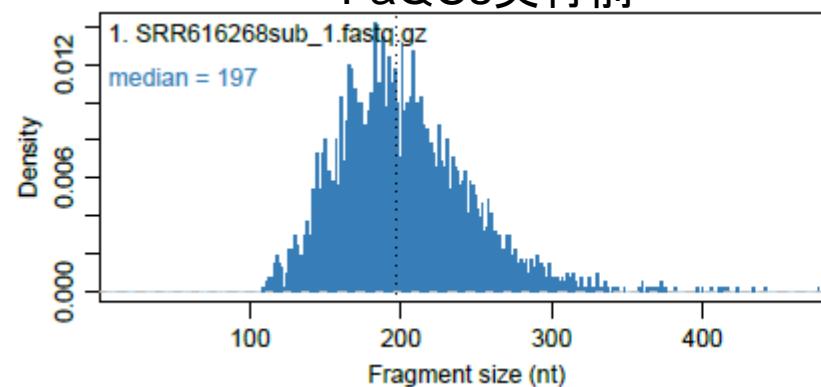


FaQCs
実行後
(post)

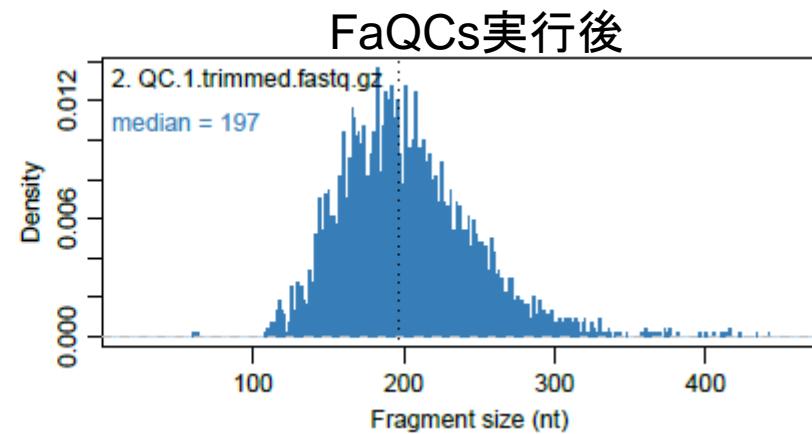
W15-8 : PDF解説

PDF8枚目。入力はpaired-endなので、おそらくforward側とreverse側両方でマップされたリードのみを取り扱っている。ゲノム配列上でのforwardとreverse間の距離分布をプロットしているものと思われる。このあたりは、よほど変な分布になっていない限り、私は気にも留めない

FaQCs実行前



FaQCs実行後



W15-9: 参考

参考

(Rで)塩基配列解析

～NGS、RNA-seq、ゲノム、トランскриプトーム、正規化、発現変動、統計、モデル、バイオイン
(last modified 2015/09/12, since 2011)

今回のRスクリプトファイル(JSLAB5_5.R)は、①のコードをテンプレートとして作成した。尚、第5回では述べないが、②カウント情報取得まで一気に行いたい場合のテンプレートなど、QuasRを用いたものは多数ある。スライドを見るだけ

What's new?

- このウェブページはインストールについての推奨手順(Windows2015.04.04版とMacintosh2015.04.03版)に従ってフリーソフトRと必要なパッケージをインストール済みであるという前提で記述しています。初心者の方は基本的な利用法(Windows2015.04.03版とMacintosh2015.04.03版)で自習してください。本ウェブページを体系的にまとめた書籍もあります。(2015/04/03)
- sample1.fastaのようなコンティグ数が1つしかない場合に、rowSums(x)の計算時にエラーが出ることがわかったので、該当箇所をapply(as.matrix(x), 1, sum)のような感じに変更しました。(2015/09/12) NEW
- NGSハンズオン講習会2015のアメリカ様分(服部先生と山口先生)の講義資料を差し替えました。(2015/09/03)
NEW
- NGSハンズオン講習会のフォローア

- マッピング | ESTレベルの長さのcontig (last modified 2014/06/24)
- マッピング | 基礎 (last modified 2013/06/19)
- マッピング | single-end | ゲノム | basic aligner(基礎) | QuasR(Gaidatzis 2015) (last modified 2014/06/21)
- マッピング | single-end | ゲノム | basic aligner(応用) | QuasR(Gaidatzis 2015) (last modified 2015/06/28)
- マッピング | single-end | ゲノム | splice-aware aligner | QuasR(Gaidatzis 2015) (last modified 2014/06/21)
- マッピング | paired-end | ゲノム | basic aligner(応用) | QuasR(Gaidatzis 2015) (last modified 2015/07/02)
- マップ後 | について (last modified 2013/06/19)
- マップ後 | 出力ファイル形式について (last modified 2013/11/05)
- マップ後 | 出力ファイルの読み込み | BAM形式 (last modified 2014/06/21)
- マップ後 | 出力ファイルの読み込み | Bowtie形式 (last modified 2013/06/18)
- マップ後 | 出力ファイルの読み込み | SOAP形式 (last modified 2013/06/19)
- マップ後 | 出力ファイルの読み込み | htSeqTools(Planet 2012) (last modified 2013/06/19)
- マップ後 | カウント情報取得 | について (last modified 2014/12/17)
 - マップ後 | カウント情報取得 | single-end | ゲノム | アノテーション有 | QuasR(Gaidatzis 2015) (last modified 2015/02/26)
 - マップ後 | カウント情報取得 | single-end | ゲノム | アノテーション無 | QuasR(Gaidatzis 2015) (last modified 2014/06/22)
 - マップ後 | カウント情報取得 | paired-end | ゲノム | アノテーション有 | QuasR(Gaidatzis 2015) (last modified 2015/07/03)
 - マップ後 | カウント情報取得 | paired-end | ゲノム | アノテーション無 | QuasR(Gaidatzis 2015) (last modified 2015/07/02)

①

②

Contents

- Illumina HiSeqデータ(トランスクリプトーム)の乳酸菌ゲノムへのマッピング
 - W14: QuasRパッケージを用いたマッピングの事前準備と本番
 - W15: 結果の解説、forward側の100–107bp付近に問題があることを特定
- トリミング、*de novo*トランスクリプトームアセンブリとマッピングの再実行
 - W16: 問題のある領域(forward側の100–107bp)のトリミング
 - W17: トリム後のデータでアセンブリを再実行(Rockhopper2; クラスパス設定関連Tips含む)
 - W18: トリム後のデータでマッピングを再実行(QuasR)
 - W19: トリム前のデータでクオリティチェックを再実行(FastQC)
- Illumina MiSeqデータ(乳酸菌ゲノム)の特徴と前処理
 - W4: FastQC、W5: FaQCs、W6: 再度FastQC
- *de novo*ゲノムアセンブリ
 - W7: Bio-LinuxにプレインストールされているVelvet (ver. 1.2.09)を上限のk=31で実行
 - W8: k=31のアセンブリ結果をRで確認。k=141で実行し、k=31の結果と同じになるのを確認
 - W9: Velvet (ver. 1.2.10)のインストール
 - W10: Velvet (ver. 1.2.10)の実行



第5回原稿PDFのp195

として、FastQCによるクオリティチェックを行えばよい[W1-2]。著者らは、FastQC実行結果ファイルの項目(Overrepresented sequences)を眺めて、トリム前に見えていた既知のアダプターやプライマー配列が、トリム後に正しく見えなくなっていることを確認して安心している[W1-3]。

このデータに関して結論からいえば、forward側の107 bp のリードファイル (SRR616268sub_1.fastq.gz → QC1.trimmed.fastq) のうち、100–107 塩基付近に乳酸菌に由来しないものがトリムしきれずに多く残っている。これは、アセンブルやマッピングがうまくできない、という実害を被ることでわかる。計算時間がかかるため、できるだけ QC 段階で問題解決するという方針もある。しかし、やってみてはじめてわかることもある。以降の内容は、著者らが実際に行ったことを問題解決に至る思考回路とともに述べる。大まかに述べると、Rockhopper²¹⁸⁾によるトランスクリプトームアセンブリ、QuasR¹⁹⁾による乳酸菌ゲノムへのマッピング、そして QC 再実行である。

トランスタリフトームアセンブリ

ゲノムのアセンブリは、断片化されたゲノム配列由来リードをつなぎ合わせて、元のゲノム配列を再構築する作業である。この再構築に相当する英語がアセンブリ(assembly)であり、再構築を行うプログラムをアセンブラー(assembler)という。デノボ(*de novo*)という言葉が同時に用いられることが多いが、これは「最初から」と

か「一から」という意味である
入力として(つまり他の情報
ルする際には、*de novo* asse
トランスクリプトームアセンブ
ゲノムではなく解析サンプル
(トランスクリプトーム)の場

のみを入力して一からアセンブルする場合は、*de novo* trans-assemblyなどと呼ばれる。

このデータ
の 107 bp の リ
gz → QC.1.trim
酸菌に由来し
る。これは、ア
という実害を被
できるだけ QC
しかし、やって
容は、著者らが
路とともに述べ
によるトランス

現在、原因(forward側の100–107 bp)が特定された状態。この部分をトリミング(除去)しましょう、というのが次の話。新しいディレクトリ「~/Documents/srp017156_trim1」上にトリム後のファイルなどを置いて作業する

このデータについて結論からいえば、forward側の107 bp のリードファイル (SRR616268sub_1.fastq.gz → QC.1.trimmed.fastq) のうち、100-107 塩基付近に乳酸菌に由来しないものがトリムしきれずに多く残っている。これは、アセンブルやマッピングがうまくできない、という実害を被ることでわかる。 算時間がかかるため、できるだけ QC 段階で問題解決するという方針もある。しかし、やってみてはじめてわかることがある。以降の内容は、著者らが実際に行ったことを問題解決に至る思考回路とともに述べる。大まかに述べると、Rockhopper¹⁸⁾によるトランск립トームアセンブリ、QuasR¹⁹⁾による乳酸菌ゲノムへのマッピング、そして QC 再実行である。

W16-1:トリミング

(Rで)塩基配列解析

～NGS、RNA-seq、ゲノム、トランскриプトーム、正規化、発現変動、統計、モデル、ハイオインフォマティクス～
(last modified 2015/09/12, since 2011)

目的は、forward側リードの100–107塩基付近の乳酸菌に由来しないものを除去。末端8塩基分を除去するためのRスクリプト(次スライドのJSLAB5_6.R)のテンプレートは、①の例題4をベースに作成。スライドを見るだけ

What's new?

- このウェブページは[インストール](#)についての推奨手順(Windows2015.04.04版とMacintosh2015.04.03版)に従ってフリーソフトRと必要なパッケージをインストール済みです。前提で記述しています。初心者の上に基本的な手順があります。(2015/04/03)
- [sample1.fasta](#)のようなコンティグ数が1つしかない該当箇所をapply(as.matrix(x), 1, sum)のような戻り値を返す。
- [NGSハンズオン講習会](#)2015のアメリカ様分(版 NEW)
- NGSハンズオン講習会のフォローアップ勉強会
- [前処理 | クオリティコントロール](#)について (last modified 2015/08/21) NEW
- [前処理 | クオリティチェック | QuasR\(Gaidatzis 2015\)](#) (last modified 2015/06/15)
- [前処理 | クオリティチェック | qrqc](#) (last modified 2014/07/17)
- [前処理 | クオリティチェック | PHREDスコアに変換](#) (last modified 2013/06/18)
- [前処理 | クオリティチェック | 配列長分布を調べる](#) (last modified 2015/06/22)
- [前処理 | クオリティチェック | Overrepresented sequences | ShortRead\(Morgan 2009\)](#) (last modified 2015/07/29)
- [前処理 | トリミング | ポリA配列除去 | ShortRead\(Morgan 2009\)](#) (last modified 2014/06/11)
- [前処理 | トリミング | アダプター配列除去\(基礎\) | QuasR\(Gaidatzis 2015\)](#) (last modified 2015/06/26) 推奨
- [前処理 | トリミング | アダプター配列除去\(基礎\) | girafe\(Toedling 2010\)](#) (last modified 2014/06/11)
- [前処理 | トリミング | アダプター配列除去\(基礎\) | ShortRead\(Morgan 2009\)](#) (last modified 2014/06/21)
- [前処理 | トリミング | アダプター配列除去\(応用\) | QuasR\(Gaidatzis 2015\)](#) (last modified 2015/06/26) 推奨
- [前処理 | トリミング | アダプター配列除去\(応用\) | ShortRead\(Morgan 2009\)](#) (last modified 2015/09/12) NEW
- [前処理 | トリミング | 指定した末端塩基数だけ除去](#) (last modified 2015/06/29)
- [前処理 | フィルタリング | PHREDスコアが低い塩基をNに置換](#) (last modified 2014/03/03)
- [前処理 | フィルタリング | PHREDスコアが低い配列\(リード\)を除去](#) (last modified 2014/08/27)
- [前処理 | フィルタリング | ACGTのみからなる配列を抽出](#) (last modified 2015/09/12) NEW
- [前処理 | フィルタリング | ACGT以外のcharacter "-" をNに変換](#) (last modified 2013/06/18)
- [前処理 | フィルタリング | ACGT以外の文字数が閾値以下の配列を抽出](#) (last modified 2015/09/12) NEW
- [前処理 | フィルタリング | 重複のない配列セットを作成](#) (last modified 2013/06/18)
- [前処理 | フィルタリング | 指定した長さ以上の配列を抽出](#) (last modified 2014/02/07)
- [前処理 | フィルタリング | 任意のリード\(サブセット\)を抽出](#) (last modified 2014/08/21)
- [前処理 | フィルタリング | 指定した長さの範囲の配列を抽出](#) (last modified 2015/02/26)



W16-1:トリミング

```
File Edit View Terminal Help  
iu@bielinux[srp017156] cd ~/Documents/  
iu@bielinux[Documents] mkdir srp017156_trim1  
iu@bielinux[Documents] cd srp017156_trim1  
iu@bielinux[srp017156_trim1] pwd  
/home/iu/Documents/srp017156_trim1  
iu@bielinux[srp017156_trim1] wget -cq http://www.iu.a.u-tokyo.ac.jp/~kadota/book/JSLAB5_6.R  
iu@bielinux[srp017156_trim1] nkf JSLAB5_6.R | head -n 3  
in_f <- "~/Documents/srp017156/SRR616268sub_1.fastq.gz">#入力ファイル名を指定してin_fに格納  
out_f <- "hoge_1.fastq.gz" #出力ファイル名を指定してout_fに格納  
param_trim <- 8 #3'末端のトリムしたい塩基数を指定  
iu@bielinux[srp017156_trim1] R --vanilla --slave < JSLAB5_6.R
```

①作業ディレクトリはここ。②ダウンロードしたJSLAB5_6.Rの最初の3行分を表示。③入力ファイルは相対パスで示したSRR616268sub_1.fastq.gz。④3'末端の8塩基を除去した結果を⑤hoge_1.fastq.gzというファイル名で保存。⑥スクリプトファイルの実行。約10秒



W16-1:トリミング

する。107 bpから8 bpト
実行したので、99 bpに
width seq
[1] 99 bbbeeeeegggggiiiiiiiiiiiiiiig...
ccddbccaaacccb
[2] 99 bbbeeeeegggggiiiiiiiiiiifgghhi...dc]bcccccabbdc
bccbbccbbcdcc
[3] 99 bbbceeeegggggiiiiiiihiiiiiiiii...bbcbbcacc
cbc _ bbbccc
[4] 99 bbbeeeeegggggiiihiiiiiiiiiehi...d]bceecdcdcc
bbcbbbbcccbcac
[5] 99 abbeeccgggggiiigfhihhidfghihi...cccbc^bbc
ccbcdccccccbc
...
[999996] 99 abbeeccgggggiiihiiifhiihihi...ccccc`bcc
cccccbcbccbc`
[999997] 99 abbeecegggggiiihghihiiiiifhghi...dccb`b^acc
cbPT`aa^bcaca
[999998] 99 bbbeeeefgggghiihiighiiiiiihii...ccccccccccca
ac]^acdcca^a
[999999] 99 ab_eeeebeeggghiiiiiiiiiiighi...cccccbccccccca
accc^`acaac_ac
[1000000] 99 bbbeeeeegggggiiiefghiiigiihii...eeddddddcccccc
cccccccccccc
iu@bielinux[srp017156 trim1] [2:36午後]

スクリプトファイル実行後の状態。①width列の数値が99になっているのがわかる。これは、トリミング後のリード長が99 bpであることを意味する。107 bpから8 bpトリムするプログラムを実行したので、99 bpになっているのは妥当

W16-1:トリミング

File Edit View Search Terminal Help

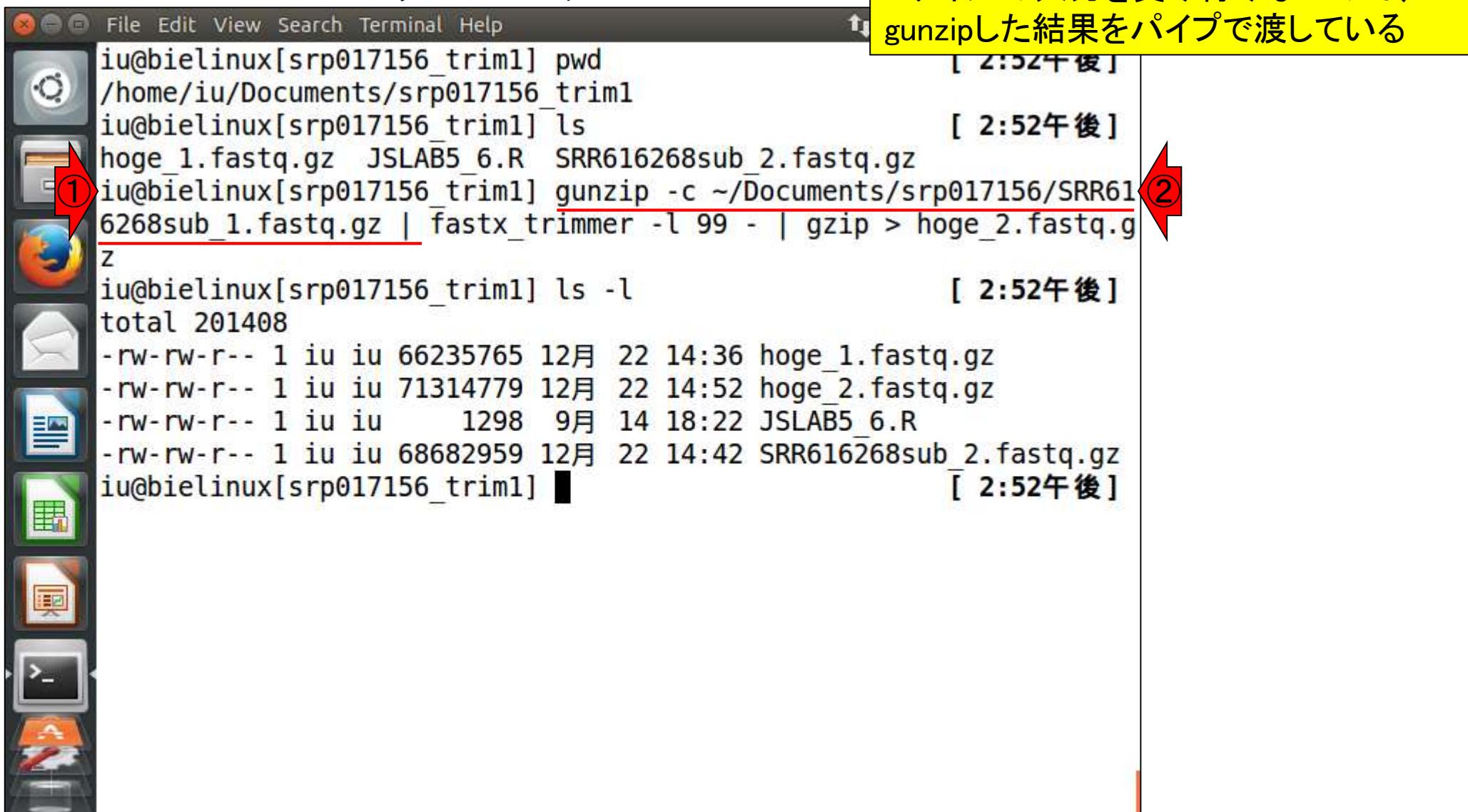
```
...     ...     ...
[999996] 99 abeeeccgggggiiiihi...
cccccbcbbc]
[999997] 99 abeeecegggggiiihghihiiiiifhghi...dccc`b^accaacc
cbPT`aa^bcaca
[999998] 99 bbbeeeeefggggghiihiighiiiiiihii...ccccccccccccca_
ac]^acdcca^a
[999999] 99 ab_eeeebeeggghiiiiiiiiiiighi...cccccbcbbbbccca
accc^`acaac_ac
[1000000] 99 bbbeeeeeggggiiiefghiiigiihii...eeddddddccccccccc
cccccccccccccc
iu@bielinux[srp017156_trim1] cp ~/Documents/srp017156/SRR616268sub
_2.fastq.gz .
iu@bielinux[srp017156_trim1] ls -l [ 2:42 午後 ]
total 131764
-rw-rw-r-- 1 iu iu 66235765 12月 22 14:36 hoge_1.fastq.gz ②
-rw-rw-r-- 1 iu iu 1298 9月 14 18:22 JSLAB5_6.R
-rw-rw-r-- 1 iu iu 68682959 12月 22 14:42 SRR616268sub_2.fastq.gz
iu@bielinux[srp017156_trim1] ls -l ~/Documents/srp017156/SRR616268
sub_1.fastq.gz
-rw-rw-r-- 1 iu iu 76659501 12月 9 15:24 /home/iu/Documents/srp01
7156/SRR616268sub_1.fastq.gz
iu@bielinux[srp017156_trim1] [ 2:42 午後 ]
```

①reverse側のファイルとしてSRR616268sub_2.fastq.gzを作業ディレクトリにコピー。②hoge_1.fastq.gz (ファイルサイズ66,235,765 bytes)は、JSLAB5_6.Rの実行結果ファイル。③ JSLAB5_6.Rの入力ファイル (SRR616268sub_1.fastq.gz) は76,659,501 bytes。107 bpが99 bpになったファイルサイズの減少度合い的に妥当

W16-2:トリミング

```
iu@bielinux[srp017156_trim1] pwd [ 2:52 午後]
/home/iu/Documents/srp017156_trim1
iu@bielinux[srp017156_trim1] ls [ 2:52 午後]
hoge_1.fastq.gz JSLAB5_6.R SRR616268sub_2.fastq.gz
iu@bielinux[srp017156_trim1] gunzip -c ~/Documents/srp017156/SRR616268sub_1.fastq.gz | fastx_trimmer -l 99 - | gzip > hoge_2.fastq.gz [ 2:52 午後]
iu@bielinux[srp017156_trim1] ls -l [ 2:52 午後]
total 201408
-rw-rw-r-- 1 iu iu 66235765 12月 22 14:36 hoge_1.fastq.gz
-rw-rw-r-- 1 iu iu 71314779 12月 22 14:52 hoge_2.fastq.gz
-rw-rw-r-- 1 iu iu 1298 9月 14 18:22 JSLAB5_6.R
-rw-rw-r-- 1 iu iu 68682959 12月 22 14:42 SRR616268sub_2.fastq.gz
iu@bielinux[srp017156_trim1] [ 2:52 午後]
```

①FASTX-Toolkitのfastx_trimmerを利用するやり方。②fastx_trimmerはgzip圧縮ファイルの入力を受け付けないので、gunzipした結果をパイプで渡している



W16-2:トリミング

③この「- (ハイフン)」は、パイプで渡したものに入力として受け付けるという明示的な意思表示。省略することができるコマンド(or プログラム)もあるが、fastx_trimmerは明示しないと怒られるのでつけています

```
iu@bielinux[srp017156_trim1] pwd [ 2:52 午後]
/home/iu/Documents/srp017156_trim1
iu@bielinux[srp017156_trim1] ls [ 2:52 午後]
hoge_1.fastq.gz JSLAB5_6.R SRR616268sub_2.fastq.gz
iu@bielinux[srp017156_trim1] gunzip -c ~/Documents/srp017156/SRR61
6268sub_1.fastq.gz | fastx_trimmer -l 99 - | gzip > hoge_2.fastq.g
z
iu@bielinux[srp017156_trim1] ls -l [ 2:52 午後]
total 201408
-rw-rw-r-- 1 iu iu 66235765 12月 22 14:36 hoge_1.fastq.gz
-rw-rw-r-- 1 iu iu 71314779 12月 22 14:52 hoge_2.fastq.gz
-rw-rw-r-- 1 iu iu 1298 9月 14 18:22 JSLAB5_6.R
-rw-rw-r-- 1 iu iu 68682959 12月 22 14:42 SRR616268sub_2.fastq.gz
iu@bielinux[srp017156_trim1] [ 2:52 午後]
```

③

W16-2:トリミング

④fastx_trimmer実行の本体部分。`-l`でリードの何塩基目までを残すかを指定。ここでは、(100塩基目以降をトリムしたいので)99塩基目まで残すという指定を行っている

```
iu@bielinux[srp017156_trim1] pwd [ 2:52 午後]
/home/iu/Documents/srp017156_trim1
iu@bielinux[srp017156_trim1] ls [ 2:52 午後]
hoge_1.fastq.gz JSLAB5_6.R SRR616268sub_2.fastq.gz
iu@bielinux[srp017156_trim1] gunzip -c ~/Documents/srp017156/SRR61
6268sub_1.fastq.gz | fastx_trimmer -l 99 - | gzip > hoge_2.fastq.g
z
iu@bielinux[srp017156_trim1] ls -l [ 2:52 午後] ④
total 201408
-rw-rw-r-- 1 iu iu 66235765 12月 22 14:36 hoge_1.fastq.gz
-rw-rw-r-- 1 iu iu 71314779 12月 22 14:52 hoge_2.fastq.gz
-rw-rw-r-- 1 iu iu 1298 9月 14 18:22 JSLAB5_6.R
-rw-rw-r-- 1 iu iu 68682959 12月 22 14:42 SRR616268sub_2.fastq.gz
iu@bielinux[srp017156_trim1] [ 2:52 午後]
```

W16-2:トリミング

```
File Edit View Search Terminal Help  
iu@bielinux[srp017156_trim1] pwd  
/home/iu/Documents/srp017156_trim1  
iu@bielinux[srp017156_trim1] ls [ 2:52 午後 ]  
hoge_1.fastq.gz JSLAB5_6.R SRR616268sub_2.fastq.gz  
iu@bielinux[srp017156_trim1] gunzip -c ~/Documents/srp017156/SRR61  
6268sub_1.fastq.gz | fastx_trimmer -l 99 - | gzip > hoge_2.fastq.g  
z  
iu@bielinux[srp017156_trim1] ls -l [ 2:52 午後 ]  
total 201408  
-rw-rw-r-- 1 iu iu 66235765 12月 22 14:36 hoge_1.fastq.gz  
-rw-rw-r-- 1 iu iu 71314779 12月 22 14:52 hoge_2.fastq.gz  
-rw-rw-r-- 1 iu iu 1298 9月 14 18:22 JSLAB5_6.R  
-rw-rw-r-- 1 iu iu 68682959 12月 22 14:42 SRR616268sub_2.fastq.gz  
iu@bielinux[srp017156_trim1] [ 2:52 午後 ]
```

⑤ここでは、トリムした結果をパイプで流してgzip圧縮してhoge_2.fastq.gzというファイル名で保存するという指令。「| gzip -> …」とハイフン(-)を明示してもよい。fastx_trimmerの-zや-oオプションを使う書き方もある。表現方法はいろいろ

⑤

W16-3：確認

```
File Edit View Search Terminal Help
iu@bielinux[srp017156_trim1] pwd
/home/iu/Documents/srp017156_trim1
iu@bielinux[srp017156_trim1] ls [ 2:52午後 ]
hoge_1.fastq.gz  JSLAB5_6.R  SRR616268sub_2.fastq.gz
iu@bielinux[srp017156_trim1] gunzip -c ~/Documents/srp017156/SRR61
6268sub_1.fastq.gz | fastx_trimmer -l 99 - | gzip > hoge_2.fastq.g
z
iu@bielinux[srp017156_trim1] ls -l [ 2:52午後 ]
total 201408
-rw-rw-r-- 1 iu iu 66235765 12月 22 14:36 hoge_1.fastq.gz
-rw-rw-r-- 1 iu iu 71314779 12月 22 14:52 hoge_2.fastq.gz
-rw-rw-r-- 1 iu iu 1298 9月 14 18:22 JSLAB5_6.R
-rw-rw-r-- 1 iu iu 68682959 12月 22 14:42 SRR616268sub_2.fastq.gz
iu@bielinux[srp017156_trim1] [ 2:52午後 ]
```

①ls -lでファイルサイズを確認。RのBiostringsでの実行結果(hoge_1.fastq.gz)とfastx_trimmerでの実行結果(hoge_2.fastq.gz)のファイルサイズが異なっている。この場合、どちらかのプログラムにバグがある可能性を考えるのが自然

W16-3：確認

①RのBiostringsでの実行結果(hoge_1.fastq.gz)と②fastx_trimmerでの実行結果(hoge_2.fastq.gz)の③最初の4行分を表示。両者の違いは赤枠のdescription情報の有無だけのようだ。バグではなさそう

```
iu@bielinux[srp017156_trim1] ls -l [ 2:57 午後]
total 201408
-rw-rw-r-- 1 iu iu 66235765 12月 22 14:36 hoge_1.fastq.gz
-rw-rw-r-- 1 iu iu 71314779 12月 22 14:52 hoge_2.fastq.gz
-rw-rw-r-- 1 iu iu 1298 9月 14 18:22 JSLAB5_6.R
-rw-rw-r-- 1 iu iu 68682959 12月 22 14:42 SRR616268sub_2.fastq.gz
iu@bielinux[srp017156_trim1] gunzip -c hoge_1.fastq.gz | head -n 4
①
②
③
@SRR616268.7 2291:6:1101:1412:2249 length=107
AGCCCGACTTTCGTCCTGCTCGACTTGTCAAGCTCCCTATACCTTACACTC
TGCGAATGATTCCAACCATTCTGAGGGAACCT
+
bbbeeeeeggggiiiiiiiiiiiiiiighiihiihiiihfihiiighhiggggggeeeee
dddcaccccccddcccccdedbccaaacccb
iu@bielinux[srp017156_trim1] gunzip -c hoge_2.fastq.gz | head -n 4
@SRR616268.7 2291:6:1101:1412:2249 length=107
AGCCCGACTTTCGTCCTGCTCGACTTGTCAAGCTCCCTATACCTTACACTC
TGCGAATGATTCCAACCATTCTGAGGGAACCT
+
SRR616268.7 2291:6:1101:1412:2249 length=107
bbbeeeeeggggiiiiiiiiiiiiighiihiihiiihfihiiighhiggggggeeeee
dddcaccccccddcccccdedbccaaacccb
iu@bielinux[srp017156_trim1] [ 2:57 午後]
```

W16-3：確認

- ①RのBiostringsでの実行結果(hoge_1.fastq.gz)と
②fastx_trimmerでの実行結果(hoge_2.fastq.gz)の
③最後の4行分を表示。大丈夫そうだ

```
iu@bielinux[srp017156_trim1] ls -l [ 2:57 午後]
total 201408
-rw-rw-r-- 1 iu iu 66235765 12月 22 14:36 hoge_1.fastq.gz
-rw-rw-r-- 1 iu iu 71314779 12月 22 14:52 hoge_2.fastq.gz
-rw-rw-r-- 1 iu iu 1298 9月 14 18:22 JSLAB5_6.R
-rw-rw-r-- 1 iu iu 68682959 12月 22 14:42 SRR616268sub_2.fastq.gz
iu@bielinux[srp017156_trim1] gunzip -c hoge_1.fastq.gz | tail -n 4
①
@SRR616268.1000860 2291:6:1101:11638:95311 length=107
GCCTTGTCAATCAAGGTGAGCATGTCGCCATGCCAGAATTGGTTGCCATGCGATCAGGATAG
AAGACATCCAGCGCATCCATCTTTCACCTTGA
+
bbbeeeeegggggiiefghiiigiiihiiiiiiiiiihifhiiiiiiiiihgggggg
eeeeedddddcccccccccccccccccccccccc
②
iu@bielinux[srp017156_trim1] gunzip -c hoge_2.fastq.gz | tail -n 4
③
@SRR616268.1000860 2291:6:1101:11638:95311 length=107
GCCTTGTCAATCAAGGTGAGCATGTCGCCATGCCAGAATTGGTTGCCATGCGATCAGGATAG
AAGACATCCAGCGCATCCATCTTTCACCTTGA
+
SRR616268.1000860 2291:6:1101:11638:95311 length=107
bbbeeeeegggggiiefghiiigiiihiiiiiiiiiihifhiiiiiiiiihgggggg
eeeeedddddcccccccccccccccccccc
iu@bielinux[srp017156_trim1] [ 2:58 午後]
```

W16-4 : Tips

素朴な疑問として、よく赤下線部分の「description情報の記述が変わってないけど...」という質問が出ます。これはdescription行部分の①スペース以降の記述は任意のため、トリム用プログラムは、この赤下線部分は「ただの文字列」として取り扱います。そんなもんです

```
iu@bielinux[srp017156_trim1] ls -l  
total 201408
```

```
-rw-rw-r-- 1 iu iu 66235765 12月 22 14:36 hoge_1.fastq.gz  
-rw-rw-r-- 1 iu iu 71314779 12月 22 14:52 hoge_2.fastq.gz  
-rw-rw-r-- 1 iu iu 1298 9月 14 18:22 JSLAB5_6.R  
-rw-rw-r-- 1 iu iu 68682959 12月 22 14:42 SRR616268sub_2.fastq.gz  
iu@bielinux[srp017156_trim1] gunzip -c hoge_1.fastq.gz | tail -n 4
```

①

```
@SRR616268.1000860 2291:6:1101:11638:95311 length=107  
GCCTTGTCAATCAAGGTGAGCATGTCGCCATGCCAGAATTGGTTGGCCATGCGATCAGGATAG  
AAGACATCCAGCGCATCCATCTTTCACCTTGA
```

```
+  
bbbeeeeeggggiiiefghiiigiiihiiiiiiiiiihifhiiiiiiiiihgggggg  
eeeeedddddccccccccc  
iu@bielinux[srp017156_trim1] gunzip -c hoge_2.fastq.gz | tail -n 4
```

①

```
@SRR616268.1000860 2291:6:1101:11638:95311 length=107  
GCCTTGTCAATCAAGGTGAGCATGTCGCCATGCCAGAATTGGTTGGCCATGCGATCAGGATAG  
AAGACATCCAGCGCATCCATCTTTCACCTTGA  
+SRR616268.1000860 2291:6:1101:11638:95311 length=107  
bbbeeeeeggggiiiefghiiigiiihiiiiiiiiiihifhiiiiiiiiihgggggg  
eeeeedddddccccccccc  
iu@bielinux[srp017156_trim1]
```

[2:58 午後]

Contents

- Illumina HiSeqデータ(トランスクリプトーム)の乳酸菌ゲノムへのマッピング
 - W14:QuasRパッケージを用いたマッピングの事前準備と本番
 - W15:結果の解説、forward側の100–107bp付近に問題があることを特定
- トリミング、*de novo*トランスクリプトームアセンブリとマッピングの再実行
 - W16:問題のある領域(forward側の100–107bp)のトリミング
 - W17:トリム後のデータでアセンブリを再実行(Rockhopper2; クラスパス設定関連Tips含む)
 - W18:トリム後のデータでマッピングを再実行(QuasR)
 - W19:トリム前のデータでクオリティチェックを再実行(FastQC)
- Illumina MiSeqデータ(乳酸菌ゲノム)の特徴と前処理
 - W4:FastQC、W5:FaQCs、W6:再度FastQC
- *de novo*ゲノムアセンブリ
 - W7:Bio-LinuxにプレインストールされているVelvet (ver. 1.2.09)を上限のk=31で実行
 - W8:k=31のアセンブリ結果をRで確認。k=141で実行し、k=31の結果と同じになるのを確認
 - W9:Velvet (ver. 1.2.10)のインストール
 - W10:Velvet (ver. 1.2.10)の実行



W17-1 : Rockhopper

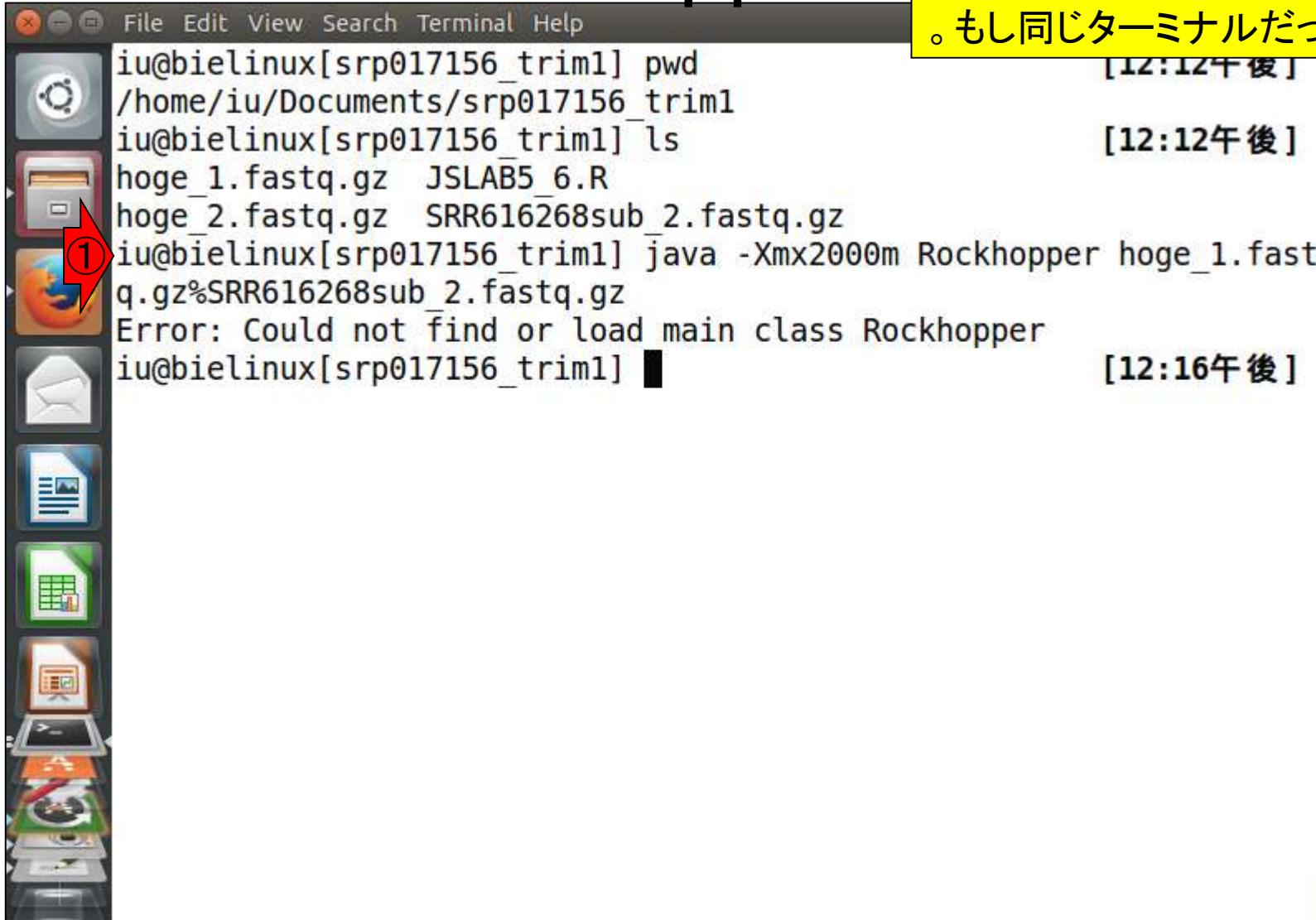
```
File Edit View Terminal Help  
iu@bielinux[srp017156_trim1] pwd  
/home/iu/Documents/srp017156_trim1  
iu@bielinux[srp017156_trim1] ls  
hoge_1.fastq.gz JSLAB5_6.R  
hoge_2.fastq.gz SRR616268sub_2.fastq.gz  
iu@bielinux[srp017156_trim1] java -Xmx2000m Rockhopper hoge_1.fastq.gz%SRR616268sub_2.fastq.gz
```

[12:12午後]

The screenshot shows a terminal window with a command-line interface. The terminal window has a dark header bar with the title 'File Edit View Terminal Help'. Below the header, there is a list of files in the current directory: 'hoge_1.fastq.gz', 'JSLAB5_6.R', 'hoge_2.fastq.gz', and 'SRR616268sub_2.fastq.gz'. The terminal is running on a system named 'bielinux' with the command 'java -Xmx2000m Rockhopper hoge_1.fastq.gz%SRR616268sub_2.fastq.gz'. To the left of the terminal, there is a vertical docked application bar containing icons for various applications, including a browser (Firefox), file manager, and other system tools. A red arrow with a circled number '1' points to the Firefox icon in the dock.

②Rockhopper2による*de novo* transcriptome assemblyをトリム後のデータで再実行。forward側はRのBiostringsを用いて得られたファイル(hoge_1.fastq.gz)、reverse側は特に何もしていないSRR616268sub_2.fastq.gzを入力として与えている

W17-1 : Rockhopper



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title bar says "File Edit View Search Terminal Help". The terminal content is as follows:

```
iu@bielinux[srp017156_trim1] pwd  
/home/iu/Documents/srp017156_trim1  
iu@bielinux[srp017156_trim1] ls  
hoge_1.fastq.gz JSLAB5_6.R  
hoge_2.fastq.gz SRR616268sub_2.fastq.gz  
iu@bielinux[srp017156_trim1] java -Xmx2000m Rockhopper hoge_1.fastq.gz%SRR616268sub_2.fastq.gz  
Error: Could not find or load main class Rockhopper  
iu@bielinux[srp017156_trim1] █
```

A red arrow points to the Firefox icon in the dock, which is circled with a red number "1".

[12:12午後] [12:12午後] [12:16午後]

実行結果。エラーが出ていることがわかる。これは、今実行したターミナルは、クラスパスを設定したターミナル(W4)とは異なるものだから。もし同じターミナルだったら、エラーは出ない

W17-2:echoで書き込み

The screenshot shows a terminal window with a command history. A red arrow labeled ① points to the terminal window title bar. A red arrow labeled ② points to the first few lines of the command history, specifically the command `tail -n 5 ~/.zshrc`. A red dashed box highlights the command `export PATH=$PATH:/home/iu/Downloads/FastQC`.

```
File Edit View Search Terminal Help
iu@bielinux[srp017156_trim1] echo $CLASSPATH
iu@bielinux[srp017156_trim1] tail -n 5 ~/.zshrc
# screen -xRR
# fi

export PATH=$PATH:/home/iu/Downloads/FastQC

iu@bielinux[srp017156_trim1]
```

①W4で設定したクラスパスが、このターミナルでは無効になっていることを確認。環境設定の永続化は、第4回W10-3で行った、`~/.zshrc`ファイルへの書き込み。
②`~/.zshrc`ファイルの最後の5行分を表示。これがクラスパス書き込み前の状態

[3:16午後] ←

W17-2:echoで書き込み

```
File Edit View Search Terminal Help  
iu@bielinux[srp017156_trim1] echo $CLASSPATH  
iu@bielinux[srp017156_trim1] tail -n 5 ~/.zshrc  
# screen -xRR  
# fi  
  
export PATH=$PATH:/home/iu/Downloads/FastQC  
  
① iu@bielinux[srp017156_trim1] echo 'hoge' [ 3:16 午後 ] ←  
hoge  
② iu@bielinux[srp017156_trim1] echo 'export CLASSPATH=/home/iu/Downloads/Rockhopper.jar'  
export CLASSPATH=/home/iu/Downloads/Rockhopper.jar  
iu@bielinux[srp017156_trim1] [ 3:18 午後 ]
```

「gedit ~/.zshrc」で.zshrcファイルを編集してもよいが、せっかくなので「echoで表示させた文字列をファイルに追加書き込みする」やり方を伝授。①や②で示すように、シングルクオーテーション('')で囲まれた文字列を画面上に出力するのがecho

W17-2:>>で追加書き込み

```
File Edit View Search Terminal Help
iu@bielinux[srp017156_trim1] echo $CLASSPATH
iu@bielinux[srp017156_trim1] tail -n 5 ~/.zshrc
# screen -xRR
# fi

export PATH=$PATH:/home/iu/Downloads/FastQC

iu@bielinux[srp017156_trim1] echo 'hoge'
hoge
iu@bielinux[srp017156_trim1] echo 'export CLASSPATH=/home/iu/Downloads/Rockhopper.jar'
export CLASSPATH=/home/iu/Downloads/Rockhopper.jar
iu@bielinux[srp017156_trim1] echo 'export CLASSPATH=/home/iu/Downloads/Rockhopper.jar' >> ~/.zshrc
iu@bielinux[srp017156_trim1] tail -n 5 ~/.zshrc
# fi

export PATH=$PATH:/home/iu/Downloads/FastQC

export CLASSPATH=/home/iu/Downloads/Rockhopper.jar
iu@bielinux[srp017156_trim1] █
```

①echoで表示させた、`~/.zshrc`ファイルの最後に書き込みたい内容を「`>>`」で追加書き込み。「`>`」では追加ではなく上書きになってしまうので注意!「`cp ~/.zshrc ~/.zshrc_org`」などとしてバックアップファイルを作成しておくほうがいいかもしれない。②追加書き込み後にtailコマンドで最後の5行分を再表示。追加書き込みが正常終了

[3:16 午後]



[3:19 午後]

[3:19 午後]

W17-3:sourceして確認

The screenshot shows a terminal window with the following command history:

```
iu@bielinux[srp017156_trim1] echo $CLASSPATH  
iu@bielinux[srp017156_trim1] source ~/.zshrc  
iu@bielinux[srp017156_trim1] echo $CLASSPATH  
/home/iu/Downloads/Rockhopper.jar  
iu@bielinux[srp017156_trim1]
```

Red numbered arrows point to specific parts of the terminal output:

- ① Points to the first command: `iu@bielinux[srp017156_trim1] echo $CLASSPATH`.
- ② Points to the second command: `iu@bielinux[srp017156_trim1] source ~/.zshrc`.
- ③ Points to the third command: `iu@bielinux[srp017156_trim1] echo $CLASSPATH`, which shows the result after sourcing.

ただの復習(第4回のW10-4)。~/.zshrcにきちんと書き込みできたとしても、それを反映させるには、②sourceを実行して環境設定ファイル(~/.zshrc)のリロードを行わねばならない。①リロード前と③リロード後で「echo \$CLASSPATH」実行結果が異なっていることがわかる

[3:26午後]

W17-4 : Rockhopper

Rockhopper2を再々トライ。RのBiostringsを利用したファイルhoge_1.fastq.gzをforward側として入力する場合(paired-end)。約2分

```
File Edit View Terminal Help  
iu@bielinux[srp017156_trim1] echo $CLASSPATH [ 3:26午後]  
iu@bielinux[srp017156_trim1] source ~/.zshrc [ 3:26午後]  
iu@bielinux[srp017156_trim1] echo $CLASSPATH [ 3:26午後]  
/home/iu/Downloads/Rockhopper.jar [ 3:26午後]  
iu@bielinux[srp017156_trim1] java -Xmx2000m Rockhopper hoge_1.fast  
q.gz%SRR616268sub_2.fastq.gz [ 3:26午後]
```

①

W17-4:途中経過

```
File Edit View Search Terminal Help  
iu@bielinux[srp017156_trim1] echo $CLASSPATH [ 3:26午後]  
iu@bielinux[srp017156_trim1] source ~/.zshrc [ 3:26午後]  
iu@bielinux[srp017156_trim1] echo $CLASSPATH [ 3:26午後]  
/home/iu/Downloads/Rockhopper.jar  
iu@bielinux[srp017156_trim1] java -Xmx2000m Rockhopper hoge_1.fastq.gz%SRR616268sub_2.fastq.gz  
  
Assembling transcripts from reads in files:  
    hoge_1.fastq.gz  
    SRR616268sub_2.fastq.gz
```

①

W17-4: 実行結果

iu@bielinux[~/Documents/srp017156_trim1]

hoge_1.fastq.gz
SRR616268sub_2.fastq.gz

トリム前の無残な結果(W5-2)やreverse側のsingle-endのみの結果(W6-4)と比べても、①転写物数(794 transcripts)や②総塩基数(449,115 bases)の点で劇的にアセンブリ結果が改善されたことがわかる!

Aligning reads to assembled transcripts using files:

hoge_1.fastq.gz
SRR616268sub_2.fastq.gz

Total reads in files: 987886

Perfectly aligned reads: 579770 59%

Total number of assembled transcripts: 794

Average transcript length: 565

Median transcript length: 306

Total number of assembled bases: 449115

Summary of results written to file:

_Results/summary.txt

Details of assembled transcripts written to file: Rockhopper

_Results/transcripts.txt

FINISHED.

iu@bielinux[srp017156_trim1] [3:36午後]

①

②

Rockhopper

Rockhopper

W17-5: Rockhopper

```
File Edit View Terminal Help  
  
Total reads in files: 987886  
Perfectly aligned reads: 579770 59%  
  
Total number of assembled transcripts: 794  
Average transcript length: 565  
Median transcript length: 306  
Total number of assembled bases: 449115  
  
Summary of results written to file:  
_Results/summary.txt  
Details of assembled transcripts written to file:  
_Results/transcripts.txt  
  
FINISHED.  
  
iu@bielinux[srp017156_trim1] ls [ 3:36 午後 ]  
hoge_1.fastq.gz JSLAB5_6.R SRR616268sub_2.fastq.gz  
hoge_2.fastq.gz Rockhopper_Results  
iu@bielinux[srp017156_trim1] ls Rockhopper_Results [ 3:42 午後 ]  
genomeBrowserFiles intermediary summary.txt transcripts.txt  
iu@bielinux[srp017156_trim1] java -Xmx2000m Rockhopper hoge_2.fast  
q.gz%SRR616268sub_2.fastq.gz
```

①fastx_trimmerでの実行結果ファイル(hoge_2.fastq.gz)を入力として、念のため実行。②Rockhopper_Resultsディレクトリ中の以前の実行結果ファイルは上書きされてなくなるので注意！様々なオプションや入力ファイルの結果を保存したい場合は、summary.txtやtranscripts.txtのファイル名をその都度変更しておく。ここは同じ結果になることを確認するだけなので気にしない

Rockhopper



W17-5: Rockhopper

確かに同じ結果になった！2つのトリミングプログラムとともに正しく動作していることも、ポジティブなアセンブル結果から証明されたといえる

File Edit View Search Terminal Help

↑ ↓ Ja 15:47

hoge_2.fastq.gz
SRR616268sub_2.fastq.gz

Aligning reads to assembled transcripts using files:

hoge_2.fastq.gz
SRR616268sub_2.fastq.gz

Total reads in files: 987886

Perfectly aligned reads: 579770 59%

Total number of assembled transcripts: 794

Average transcript length: 565

Median transcript length: 306

Total number of assembled bases: 449115

Summary of results written to file: Rockhopper

_Results/summary.txt

Details of assembled transcripts written to file: Rockhopper

_Results/transcripts.txt

FINISHED.

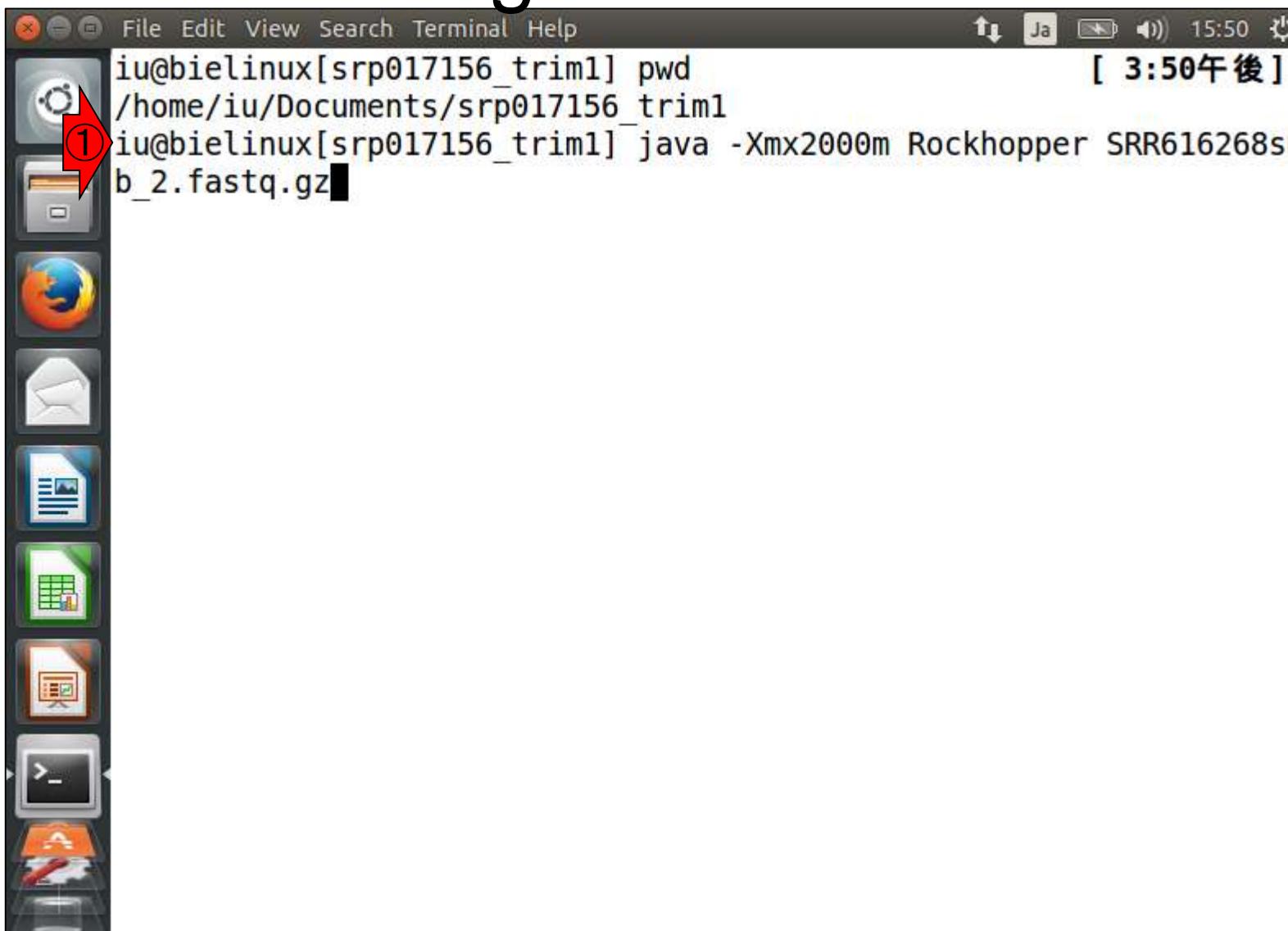
iu@bielinux[srp017156_trim1] └

[3:44午後]

W17-6:single-end

参考

比較用に①何もしていないreverse側のsingle-endのみ(SRR616268sub_2.fastq.gz)で実行



The screenshot shows a terminal window on a Linux desktop environment. The terminal window has a dark header bar with the title "iu@bielinux[srp017156_trim1]". Below the header is a scrollable text area. On the left side of the screen, there is a vertical dock containing icons for various applications, with the terminal icon being highlighted by a red circle labeled "①". The terminal text area contains the following command:

```
iu@bielinux[srp017156_trim1] pwd  
/home/iu/Documents/srp017156_trim1  
iu@bielinux[srp017156_trim1] java -Xmx2000m Rockhopper SRR616268su  
b_2.fastq.gz
```

The terminal window also displays the system tray at the top right, showing the date and time as "3:50 午後".

W17-6:single-end

参考

File Edit View Search Terminal Help

Assembling transcripts from reads in file:
ub_2.fastq.gz

Aligning reads to assembled transcripts using file: SRR616268s
ub_2.fastq.gz

Total reads in file: 983854
Perfectly aligned reads: 710393 72%

③

①

②

Total number of assembled transcripts: 424
Average transcript length: 436
Median transcript length: 228
Total number of assembled bases: 185233

Summary of results written to file:

_Results/summary.txt

Details of assembled transcripts written to file:

_Results/transcripts.txt

Rockhopper

Rockhopper

FINISHED.

iu@bielinux[srp017156_trim1] █

[3:51午後]

①アセンブルされた転写物数は424個、②総塩基数は185,233。③入力リード数983,854個のうち、72% (710,393個)がマップされていることがわかる。FaQCs実行後のファイルを入力とした結果(W6-4)よりもわずかによい結果といえるかもしれないが、事実上誤差範囲

ここまでまとめ

①が追加実行分。アセンブリ結果へのインパクトが大きいのは、forward側の100–107 bp部分のトリム(W6–2 → W17–4)

■ オリジナル(SRR616268)

- 乳酸菌paired-end RNA-seqデータで、最初の100万リードのみ抽出
- forward側(SRR616268sub_1.fastq.gz)のリード長は107 bp
- reverse側(SRR616268sub_2.fastq.gz)のリード長は93 bp



■ FaQCs実行結果(W1–1)

- 1,000,000リード → 977,202リード (W1–3)
- forward側(QC.1.trimmed.fastq)
- reverse側(QC.2.trimmed.fastq)
- リード長はバラバラ。FastQC上で見られるIllumina adapterは消滅状態

■ *de novo*トランスクリプトームアセンブリ(Rockhopper2)実行結果

- paired-end (QC.1.trimmed.fastqとQC.2.trimmed.fastq) : **0** transcript or contig (W5–2)
- single-end (forward側のみ; QC.1.trimmed.fastq) : **1** transcript (W6–2)
- single-end (reverse側のみ; QC.2.trimmed.fastq) : **423** transcripts (W6–4)
- paired-end (hoge_1.fastq.gzとSRR616268sub_2.fastq.gz) : **794** transcripts (W17–4)
- single-end (reverse側のみ; SRR616268sub_2.fastq.gz) : **424** transcripts (W17–6)

①

処理の順番について

今回、トリミングは①を入力として②を作成した。なぜ③のFaQCs実行後のファイルを入力としなかったのか?について思考回路を解説

■ オリジナル(SRR616268)

- 乳酸菌paired-end RNA-seqデータで、最初の100万リードのみ抽出
- forward側(SRR616268sub_1.fastq.gz) ①リード長は107 bp 107 bp
- reverse側(SRR616268sub_2.fastq.gz)のリード長は93 bp 93 bp



■ FaQCs実行結果(W1-1)

- 1,000,000リード → 977,202リード (W1-3)
- forward側(QC.1.trimmed.fastq) ③
- reverse側(QC.2.trimmed.fastq)
- リード長はバラバラ。FastQC上で見られるIllumina adapterは消滅状態

■ *de novo*トランスクリプトームアセンブリ(Rockhopper2)実行結果

- paired-end (QC.1.trimmed.fastqとQC.2.trimmed.fastq) : 0 transcript or contig (W5-2)
- single-end (forward側のみ; QC.1.trimmed.fastq) : 1 transcript (W6-2)
- single-end (reverse側のみ; QC.2.trimmed.fastq) : 423 transcripts (W6-4)
- paired-end (hoge_1.fastq.gz ② SRR616268sub_2.fastq.gz) : 794 transcripts (W17-4)
- single-end (reverse側のみ; SRR616268sub_2.fastq.gz) : 424 transcripts (W17-6)

処理の順番について

■ オリジナル(SRR616268)

- 乳酸菌paired-end RNA-seqデータで、最初のforward側(SRR616268sub_1.fastq.gz)①リード長は93 bp
- forward側(SRR616268sub_1.fastq.gz)①リード長は93 bp
- reverse側(SRR616268sub_2.fastq.gz)のリード長は93 bp

まず、結論としては、③を入力として3'側の8 bpをトリムしてもほとんど同じ結果となるだろうと予想した。理由は、④FaQCs実行で、リード数自体が3%程度しか減少していないから。また、FastQC実行結果(第4回W17-2)で見えていたアダプター配列の出現回数自体も1,296回と全体に与える影響度合い的には誤差範囲



■ FaQCs実行結果(W1-1)

- 1,000,000リード → 977,202リード(W1-3)④
- forward側(QC.1.trimmed.fastq)③
- reverse側(QC.2.trimmed.fastq)
- リード長はバラバラ。FastQC上で見られるIllumina adapterは消滅状態

■ *de novo*トランスクリプトームアセンブリ(Rockhopper 2)実行結果

- paired-end (QC.1.trimmed.fastqとQC.2.trimmed.fastq) : 0 transcript or contig (W5-2)
- single-end (forward側のみ; QC.1.trimmed.fastq) : 1 transcript (W6-2)
- single-end (reverse側のみ; QC.2.trimmed.fastq) : 423 transcripts (W6-4)
- paired-end (hoge_1.fastq.gz②RR616268sub_2.fastq.gz) : 794 transcripts (W17-4)
- single-end (reverse側のみ; SRR616268sub_2.fastq.gz) : 424 transcripts (W17-6)

処理の順番について

```
File Edit View Search Terminal Help  
iu@bielinux[result2] pwd  
/home/iu/Documents/srp017156/result2  
iu@bielinux[result2] ls  
fastqCount.txt          QC.2.trimmed.fastq.gz  
nohup.out                QC_qc_report.pdf  
QC.1.trimmed_fastqc.html QC.stats.txt  
QC.1.trimmed_fastqc.zip  QC.unpaired.trimmed.fastq.gz  
QC.1.trimmed.fastq.gz    Rockhopper_Results  
iu@bielinux[result2] tail -n 15 QC.stats.txt [12:13午後]  
  
Discarded reads #: 27365 (1.37 %) ①  
Trimmed bases: 4053286 (2.03 %) ②  
Reads Filtered by length cutoff (50 bp): 26089 (1.30 %)  
Bases Filtered by length cutoff: 602583 (0.30 %)  
Reads Filtered by continuous base "N" (2): 1270 (0.06 %)  
Bases Filtered by continuous base "N": 125675 (0.06 %)  
Reads Filtered by low complexity ratio (0.8): 6 (0.00 %)  
Bases Filtered by low complexity ratio: 600 (0.00 %)  
Reads Trimmed by quality (5.0): 148776 (7.44 %)  
Bases Trimmed by quality: 3044709 (1.52 %)  
Reads Trimmed with Adapters/Primers: 7418 (0.37 %)  
Bases Trimmed with Adapters/Primers: 279719 (0.14 %)  
  Nextera-primer-adapter-1 7270 reads (0.36 %) 272839 bases (0.14 %)  
  Nextera-primer-adapter-2 148 reads (0.01 %) 6880 bases (0.00 %)  
iu@bielinux[result2] [12:13午後]
```

また、実は裏で①FaQCsのQC結果のサマリーファイル(QC.stats.txt)もずっと眺めている。細かいことはよくわからなくても、②トリムされた塩基が全体の2%程度というのをみた段階で、「FaQCs実行結果ファイルを入力としても同じだろう」と断定。このファイルについては、第6回W5-3でも解説

W1-3: FastQCで確認

さらに、①第5回W1-3(2016.08.01のスライド25)でFastQC実行結果を眺めているが、この時に②で配列長分布も見ている。つまり…

FaQCs(ver. 1.34)によるQ

①

- 連載第5回[W1-1]結果との比較用に、FaQCs実行前のデータでFastQC (ver. 0.11.3)を実行。
連載第4回の[W7, W8, W9-7]あたり。

```
fastqc2 -v
fastqc2 -q SRR616268sub_1.fa
fastqc2 -q SRR616268sub_2.fa
```

連載第4回[W9-5]の手順通りに行なった
です。実行結果ファイルSRR616268su
結果ですが、連載第4回W8-6に示すよ
と3」がリード中に含まれていることが
「Illumina Single End PCR Primer 1」が

- [Skewer: Jiang et al., BMC Bioinformatics](#)
- [FaQCs: Lo and Chain, BMC Bioinformatics](#)
- 図1。第4回の[W17-3]と基本的に同じ。

```
cd ~/Documents/srp017156
rm -f hoge*
rm -f JS*
rm -rf result*
rm -f *.bz2

pwd
ls -lh
fastqc2 -v
FaQCs.pl -v
time FaQCs.pl -adapter -p SRR616268sub_1
ls result2
```

```
pwd
ls -lh
fastqc2 -v
FaQCs.pl -v
time FaQCs.pl -adapter -p SRR616268sub_1.fastq.gz SRR616268sub_2.fastq.gz -d result2
ls result2
```

FaQCs ver. 1.34を実行し、結果をresult2ディレクトリに保存(約25分)。result2ディレクトリ上には、計6ファイルが生成されている。そのうちのQC結果のサマリーレポートは次の2つ:[QC.stats.txt](#)と[QC qc_report.pdf](#)

- トリム後(FaQCs実行後)のデータを入力としてFastQC (ver. 0.11.3)をデフォルトで実行。[W1-2]

```
pwd
ls result2/*.fastq
fastqc2 -q result2/QC.1.trimmed.fastq --outdir=/home/iu/Desktop/mac_share
fastqc2 -q result2/QC.2.trimmed.fastq --outdir=/home/iu/Desktop/mac_share
date
ls -lh /home/iu/Desktop/mac_share/QC.*
```

出力結果を共有フォルダ(/home/iu/Desktop/mac_share)に直接保存。実行結果ファイル
[QC.1.trimmed_fastqc.html](#)(forward側)と[QC.2.trimmed_fastqc.html](#)(reverse側)とともに、Overrepresented
sequences項目に見えていたアダプターやプライマー配列情報がなくなっているのがわかります。[W1-3]

- [Rockhopper 2: Tjaden B, Genome Biol., 2015](#)
- [QuasR: Gaidatzis et al., Bioinformatics, 2015](#)

②

トランскryptームアセンブリ

処理の順番について

FastQC Report

Sun 20 Dec 2015
QC.1.trimmed.fastq

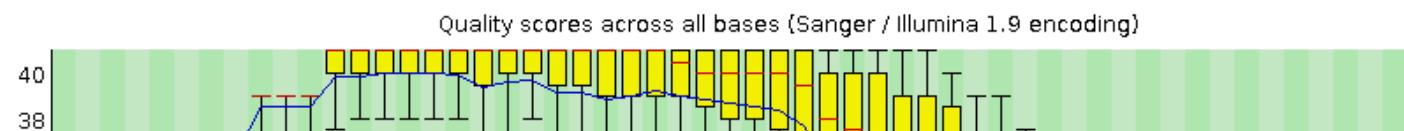
Summary

- Basic Statistics ①
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content
- Kmer Content

Basic Statistics

Measure	Value
Filename	QC.1.trimmed.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	977202
Sequences flagged as poor quality	0
Sequence length	50-107 ②
%GC	50

Per base sequence quality



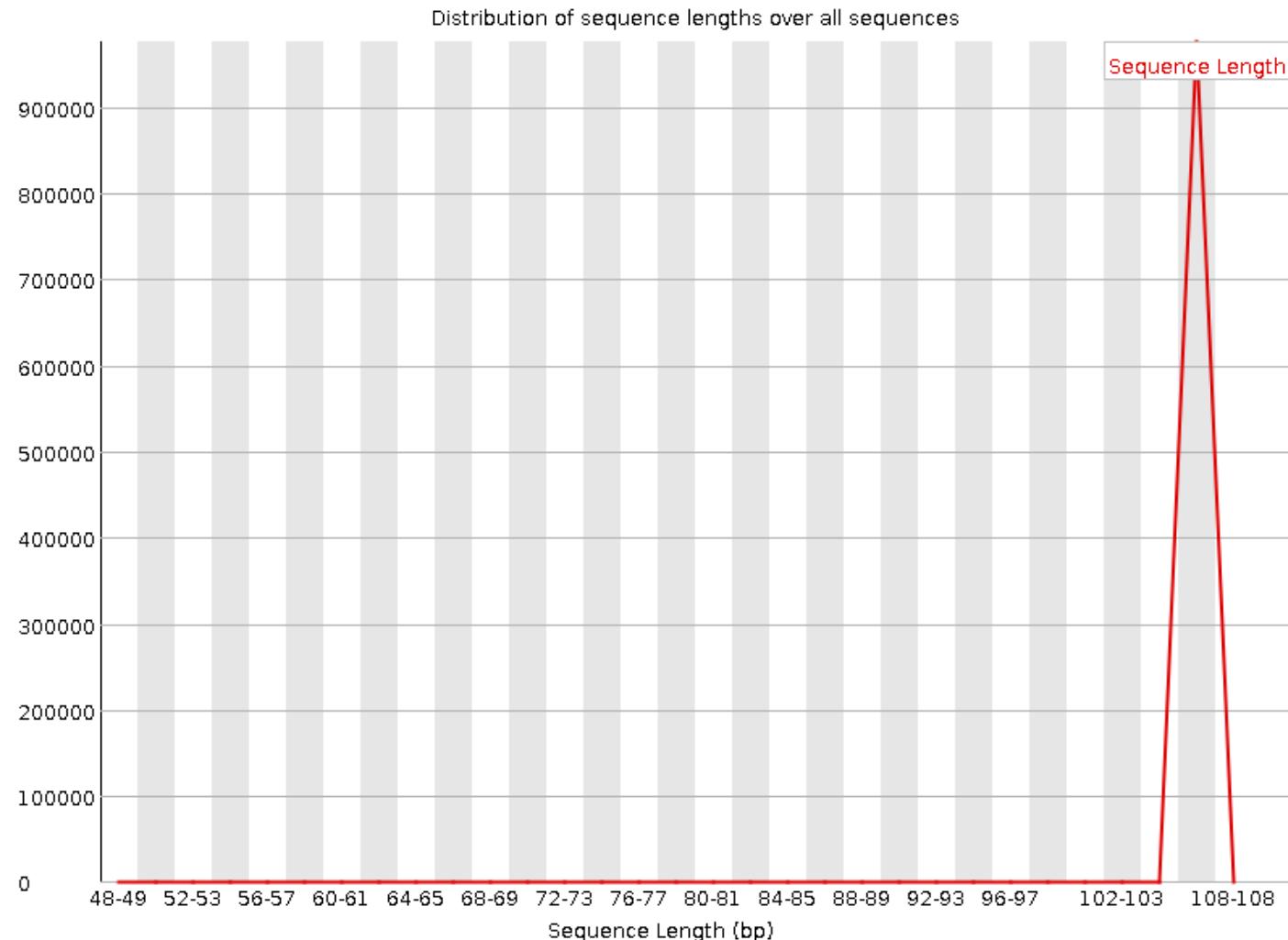
処理の順番について

FastQC Report

Summary

- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution ①
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content
- Kmer Content

Sequence Length Distribution



①Sequence Length Distributionのところを眺めて、リード数としては少ないが、最短で50 bpになっているリードもあることを認識する。そしてこれらに対して無条件で8 bpトリムするのは、心情的に忍びない、という思考回路。8 bpトリム後にFaQCsはアリ。むしろ推奨

Contents

- Illumina HiSeqデータ(トランスクリプトーム)の乳酸菌ゲノムへのマッピング
 - W14:QuasRパッケージを用いたマッピングの事前準備と本番
 - W15:結果の解説、forward側の100–107bp付近に問題があることを特定
- トリミング、*de novo*トランスクリプトームアセンブリとマッピングの再実行
 - W16:問題のある領域(forward側の100–107bp)のトリミング
 - W17:トリム後のデータでアセンブリを再実行(Rockhopper2; クラスパス設定関連Tips含む)
 - W18:トリム後のデータでマッピングを再実行(QuasR)
 - W19:トリム前のデータでクオリティチェックを再実行(FastQC)
- Illumina MiSeqデータ(乳酸菌ゲノム)の特徴と前処理
 - W4:FastQC、W5:FaQCs、W6:再度FastQC
- *de novo*ゲノムアセンブリ
 - W7:Bio-LinuxにプレインストールされているVelvet (ver. 1.2.09)を上限のk=31で実行
 - W8:k=31のアセンブリ結果をRで確認。k=141で実行し、k=31の結果と同じになるのを確認
 - W9:Velvet (ver. 1.2.10)のインストール
 - W10:Velvet (ver. 1.2.10)の実行



W18-1 : QuasR

```
File Edit View Search Terminal Help  
iu@bielinux[srp017156_trim1] pwd  
/home/iu/Documents/srp017156_trim1  
iu@bielinux[srp017156_trim1] ls  
hoge_1.fastq.gz JSLAB5_6.R  
hoge_2.fastq.gz Rockhopper_Results  
iu@bielinux[srp017156_trim1] wget -cq http://www.iu.a.u-tokyo.ac.jp/~kadota/book/JSLAB5_7.txt  
iu@bielinux[srp017156_trim1] more JSLAB5_7.txt [ 4:00午後]  
FileName1 FileName2 SampleName  
SRR616268sub_1.fastq.gz SRR616268sub_2.fastq.gz pre_7bp_trim  
hoge_1.fastq.gz SRR616268sub_2.fastq.gz post_7bp_trim  
iu@bielinux[srp017156_trim1]
```

①トリム後のデータでマッピングを再実行すべく、リストファイル(JSLAB5_7.txt)をダウンロード。forward側はhoge_1.fastq.gz、reverse側は特に何もしていないSRR616268sub_2.fastq.gz。②moreで中身を表示。③比較のため、特に何もしていないforward側のファイルでのマッピングも行っている(pre_7bp_trimの行に相当)

SRR616268sub_2.fastq.gz

[4:00午後]

[4:00午後]

W18-2:QuasR

①Rスクリプトファイル(JSLAB5_8.R)をダウンロードし、②最初の2行分を表示させ、③入出力ファイルを確認

```
iu@bielinux[srp017156_trim1] pwd [ 4:00午後]
/home/iu/Documents/srp017156_trim1
iu@bielinux[srp017156_trim1] ls [ 4:00午後]
hoge_1.fastq.gz JSLAB5_6.R SRR616268sub_2.fastq.gz
hoge_2.fastq.gz Rockhopper_Results
iu@bielinux[srp017156_trim1] wget -cq http://www.iu.a.u-tokyo.ac.jp/~kadota/book/JSLAB5_7.txt [ 4:00午後]
iu@bielinux[srp017156_trim1] more JSLAB5_7.txt
FileName1 FileName2 SampleName
SRR616268sub_1.fastq.gz SRR616268sub_2.fastq.gz pre_7bp_trim
hoge_1.fastq.gz SRR616268sub_2.fastq.gz post_7bp_trim
iu@bielinux[srp017156_trim1] wget -cq http://www.iu.a.u-tokyo.ac.jp/~kadota/book/JSLAB5_8.R
iu@bielinux[srp017156_trim1] nkf JSLAB5_8.R | head -n 2 -
in_f1 <- "JSLAB5_7.txt" #入力ファイル名を指定してin_f1に格納( RNA-seq リストファイル )
in_f2 <- "/home/iu/Documents/genomes/Lactobacillus_casei_12a.GCA_00309565.2.30.dna.toplevel.fa" #入力ファイル名を指定してin_f2に格納(リファレンス配列 )
iu@bielinux[srp017156_trim1] [ 4:09午後]
```

W18-3:QuasR

①比較用入力ファイル
(SRR616268sub_1.fastq.gz)のコピーと②確認

```
iu@bielinux[~/Documents/srp017156_trim1] pwd [ 4:14午後]
/home/iu/Documents/srp017156_trim1
iu@bielinux[srp017156_trim1] ls [ 4:14午後]
hoge_1.fastq.gz JSLAB5_7.txt SRR616268sub_2.fastq.gz
hoge_2.fastq.gz JSLAB5_8.R
JSLAB5_6.R Rockhopper_Results
iu@bielinux[srp017156_trim1] cp ~/Documents/srp017156/SRR616268sub
_1.fastq.gz .
iu@bielinux[srp017156_trim1] ls [ 4:14午後]
hoge_1.fastq.gz JSLAB5_7.txt SRR616268sub_1.fastq.gz
hoge_2.fastq.gz JSLAB5_8.R SRR616268sub_2.fastq.gz
JSLAB5_6.R Rockhopper_Results
iu@bielinux[srp017156_trim1] [ 4:14午後]
```

W18-4 : QuasR

```
File Edit View Search Terminal Help [ 4:14 午後 ]
iu@bielinux[srp017156_trim1] pwd
/home/iu/Documents/srp017156_trim1
iu@bielinux[srp017156_trim1] ls [ 4:14 午後 ]
hoge_1.fastq.gz JSLAB5_7.txt SRR616268sub_2.fastq.gz
hoge_2.fastq.gz JSLAB5_8.R
JSLAB5_6.R Rockhopper_Results
iu@bielinux[srp017156_trim1] cp ~/Documents/srp017156/SRR616268sub
1.fastq.gz .
iu@bielinux[srp017156_trim1] ls [ 4:14 午後 ]
hoge_1.fastq.gz JSLAB5_7.txt SRR616268sub_1.fastq.gz
hoge_2.fastq.gz JSLAB5_8.R SRR616268sub_2.fastq.gz
JSLAB5_6.R Rockhopper_Results
iu@bielinux[srp017156_trim1] R --vanilla --slave < JSLAB5_8.R
```

①

W18-4 : QuasR

無事終了。①lsで確認。bamファイルや②QCLポートファイルが作成されていることがわかる。
③pdfファイルを共有フォルダにコピーして眺める

```
File Edit View Search Terminal Help  
bielinux  
1  
Performing genomic alignments for 2 samples. See progress in the log file:  
/home/iu/Documents/srp017156_trim1/QuasR_log_663c661fd224.txt  
[samopen] SAM header is present: 1 sequences.  
[bam_sort_core] merging from 2 files...  
[samopen] SAM header is present: 1 sequences.  
[bam_sort_core] merging from 2 files...  
Genomic alignments have been created successfully  
  
collecting quality control data  
creating QC plots  
iu@bielinux[srp017156_trim1] ls [ 4:30 午後 ]  
hoge_1_663c66aaf313.bam QuasR_log_663c661fd224.txt  
hoge_1_663c66aaf313.bam.bai Rockhopper_Results  
hoge_1_663c66aaf313.bam.txt SRR616268sub_1_663c2ebdd882.bam  
hoge_1.fastq.gz SRR616268sub_1_663c2ebdd882.bam.bai  
hoge_2.fastq.gz SRR616268sub_1_663c2ebdd882.bam.txt  
JSLAB5_6.R SRR616268sub_1_663c2ebdd882_QC.pdf  
JSLAB5_7.txt SRR616268sub_1.fastq.gz  
JSLAB5_8.R SRR616268sub_2.fastq.gz  
③ iu@bielinux[srp017156_trim1] cp *.pdf ~/Desktop/mac_share
```

W18-5: PDFはここにもあります

- QuasR[W18-4]

Rスクリプトファイル(JSLAB5_8.R)の実行。

```
R --vanilla --slave < JSLAB5_8.R  
ls  
cp *.pdf ~/Desktop/mac_share
```

- 得られたQCレポートPDFファイル([SRR616268sub_1_663c2ebdd882_QC.pdf](#))の解説[W18-5]



おわりに

- FastQC[W19-1]

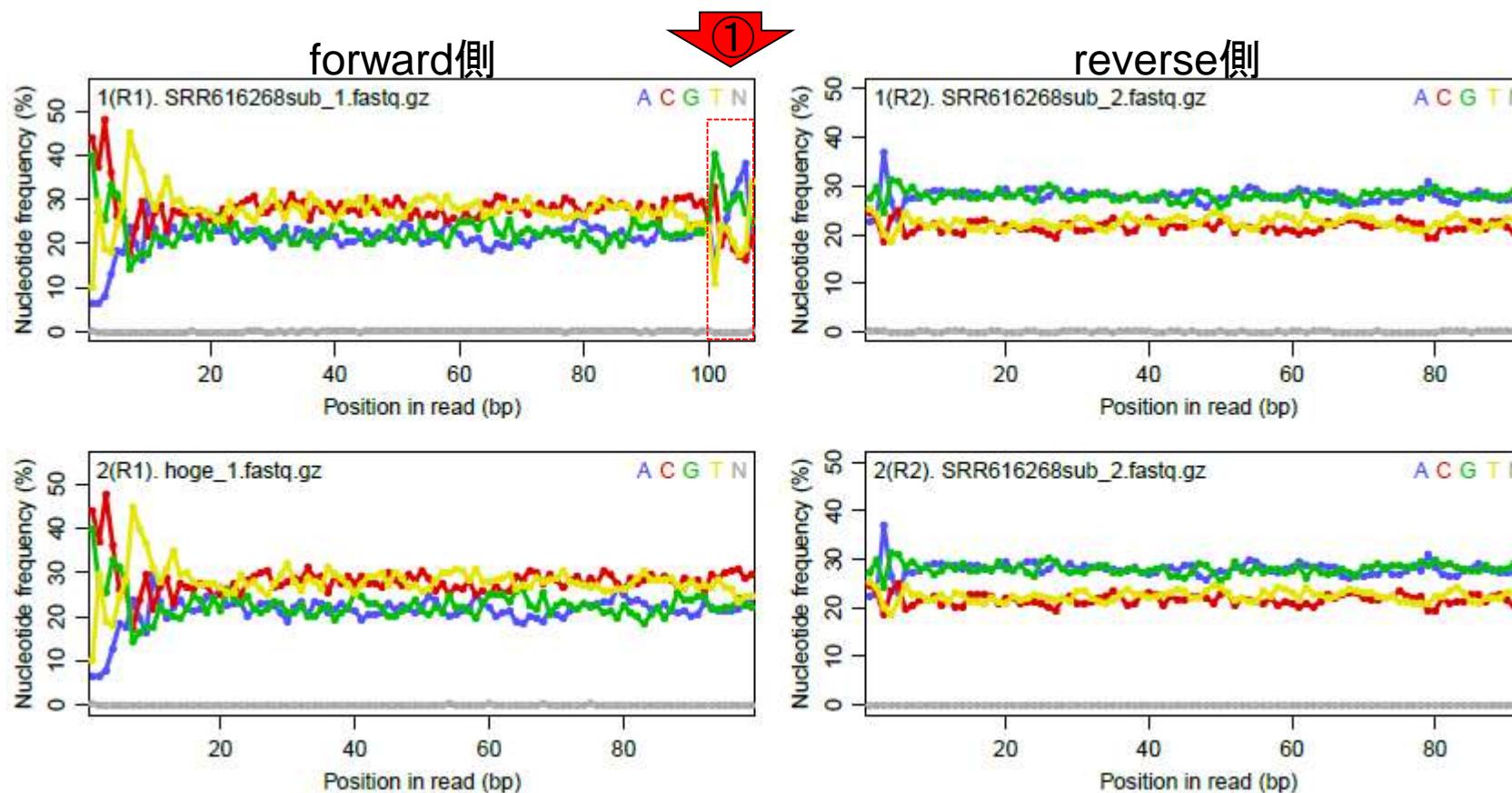
トリム前(FaQCs実行前)のforward側のリードを入力として **-nogroup**オプションをつけて FastQC (ver. 0.11.4)をデフォルトで実行。-qオプションは、途中経過を表示させないだけのオプションなので、結果自体には無関係。

```
pwd  
ls SRR616268sub_1*  
fastqc2 -v  
fastqc2 -q --nogroup SRR616268sub_1.fastq.gz  
ls SRR616268sub_1*  
mv SRR616268sub_1_fastqc.html SRR616268sub_1_nogroup.html  
cp SRR616268sub_1_nogroup.html ~/Desktop/mac_share
```

何らかの原因でファイルが作成されなかった場合は、①こちらをご覧になってください

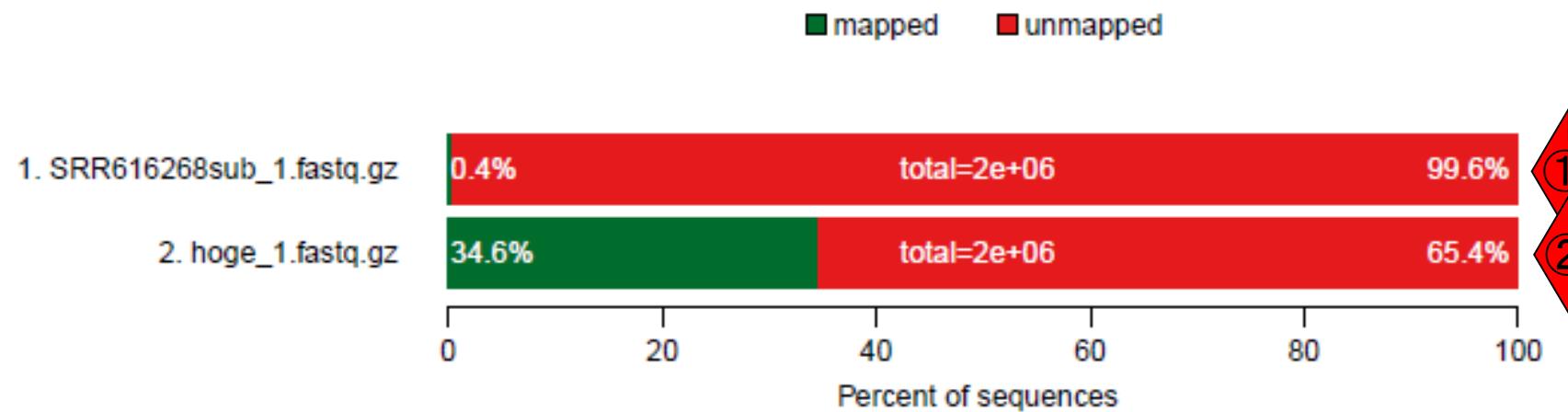
W18-5: PDF解説

PDF2枚目。ポジションごとの塩基の出現確率。FastQC Report中の項目「Per base sequence content」と同じ(但し色は異なる)。赤枠部分をトリムしたおかげで、アセンブルやマッピングが劇的に改善したことになる



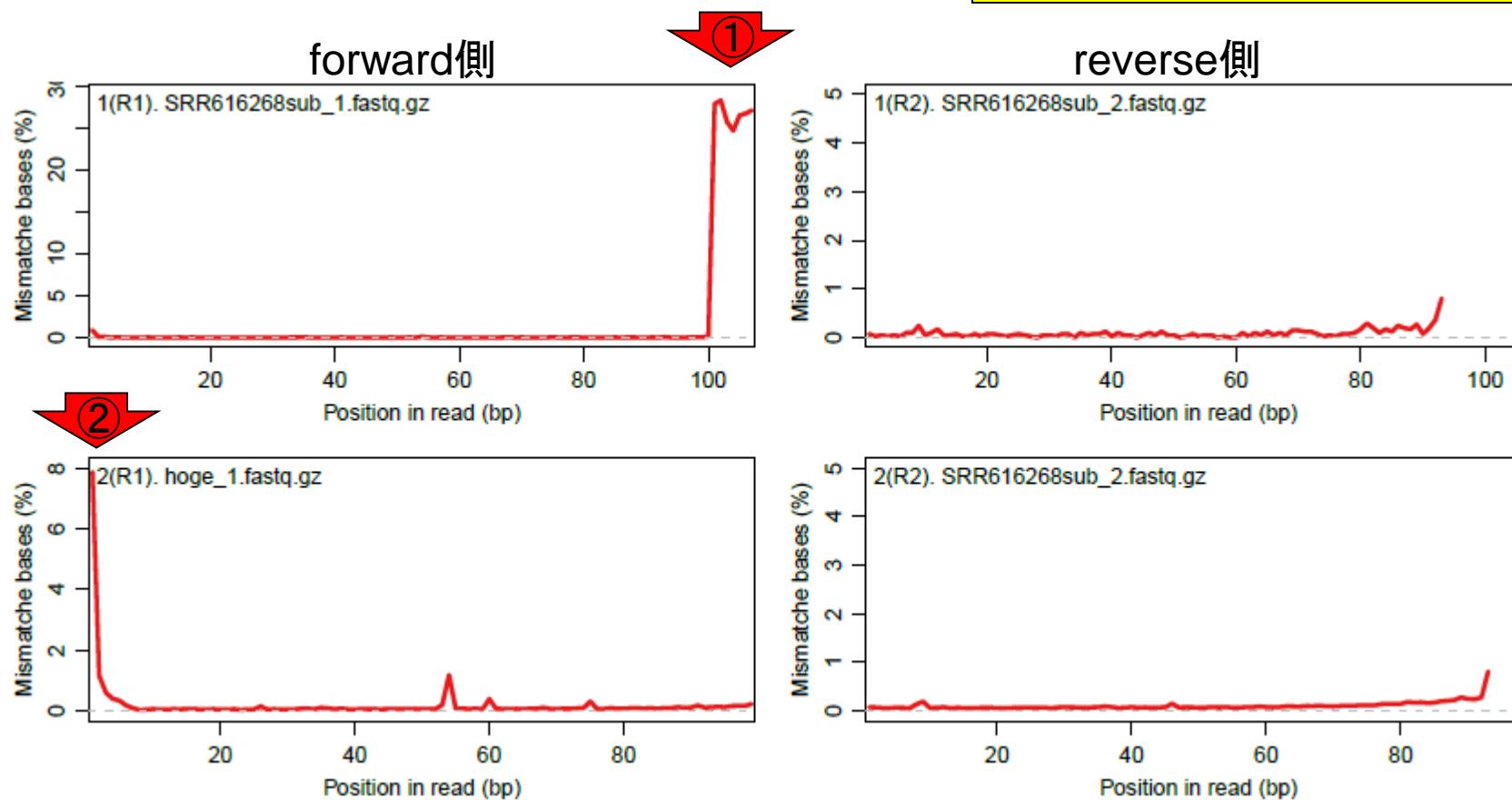
W18-6: PDF解説

PDF4枚目。全リード(forward, reverse合わせて200万リード)のうち、マップされたリードの割合は①トリム実行前が0.4%、②実行後が34.6%。トリム後のマップ率が劇的に向上!



W18-7: PDF解説

PDF6枚目。forward側の100–107 bpをトリムしたおかげで①のミスマッチ塩基の割合が劇的に低下していることがわかる。そのおかげで、相対的なインパクトが弱かった②forward側の1塩基あたりもミスマッチ率が高かったことがわかる。ここもトリムしておけばいいかも…と妄想する



Contents

- Illumina HiSeqデータ(トランスクリプトーム)の乳酸菌ゲノムへのマッピング
 - W14: QuasRパッケージを用いたマッピングの事前準備と本番
 - W15: 結果の解説、forward側の100–107bp付近に問題があることを特定
- トリミング、*de novo*トランスクリプトームアセンブリとマッピングの再実行
 - W16: 問題のある領域(forward側の100–107bp)のトリミング
 - W17: トリム後のデータでアセンブリを再実行(Rockhopper2; クラスパス設定関連Tips含む)
 - W18: トリム後のデータでマッピングを再実行(QuasR)
 - W19: トリム前のデータでクオリティチェックを再実行(FastQC)
- Illumina MiSeqデータ(乳酸菌ゲノム)の特徴と前処理
 - W4: FastQC、W5: FaQCs、W6: 再度FastQC
- *de novo*ゲノムアセンブリ
 - W7: Bio-LinuxにプレインストールされているVelvet (ver. 1.2.09)を上限のk=31で実行
 - W8: k=31のアセンブリ結果をRで確認。k=141で実行し、k=31の結果と同じになるのを確認
 - W9: Velvet (ver. 1.2.10)のインストール
 - W10: Velvet (ver. 1.2.10)の実行



W19-1 : FastQC

①--nogroupオプションをつけてFastQCを実行
②赤枠が出力ファイル。③htmlレポートのファイル名をSRR616268sub_1_nogroup.htmlに変更

```

File Edit View Search Terminal Help [ 4:49 午後 ]
iu@bielinux[srp017156_trim1] pwd
/home/iu/Documents/srp017156_trim1
iu@bielinux[srp017156_trim1] ls SRR616268sub_1*
SRR616268sub_1_663c2ebdd882.bam [ 4:50 午後 ]
SRR616268sub_1_663c2ebdd882.bam.bai
SRR616268sub_1_663c2ebdd882.bam.txt
SRR616268sub_1_663c2ebdd882_QC.pdf
SRR616268sub_1.fastq.gz
iu@bielinux[srp017156_trim1] fastqc2 -v [ 4:50 午後 ]
FastQC v0.11.4
iu@bielinux[srp017156_trim1] fastqc2 -q --nogroup SRR616268sub_1.f
astq.gz
iu@bielinux[srp017156_trim1] ls SRR616268sub_1* [ 4:50 午後 ]
SRR616268sub_1_663c2ebdd882.bam SRR616268sub_1_fastqc.html
SRR616268sub_1_663c2ebdd882.bam.bai SRR616268sub_1_fastqc.zip
SRR616268sub_1_663c2ebdd882.bam.txt SRR616268sub_1.fastq.gz
SRR616268sub_1_663c2ebdd882_QC.pdf
iu@bielinux[srp017156_trim1] mv SRR616268sub_1_fastqc.html SRR6162
68sub_1_nogroup.html
iu@bielinux[srp017156_trim1] cp SRR616268sub_1_nogroup.html ~/Des
ktop/mac_share
iu@bielinux[srp017156_trim1] [ 4:55 午後 ]

```

W19-2:FastQC

①SRR616268sub_1_nogroup.htmlのKmer Content項目を表示。1-59塩基目には極端に多いk-merの上位6個は存在しないことがわかる

Wed 16 Sep 2015
SRR616268sub_1.fastq.gz

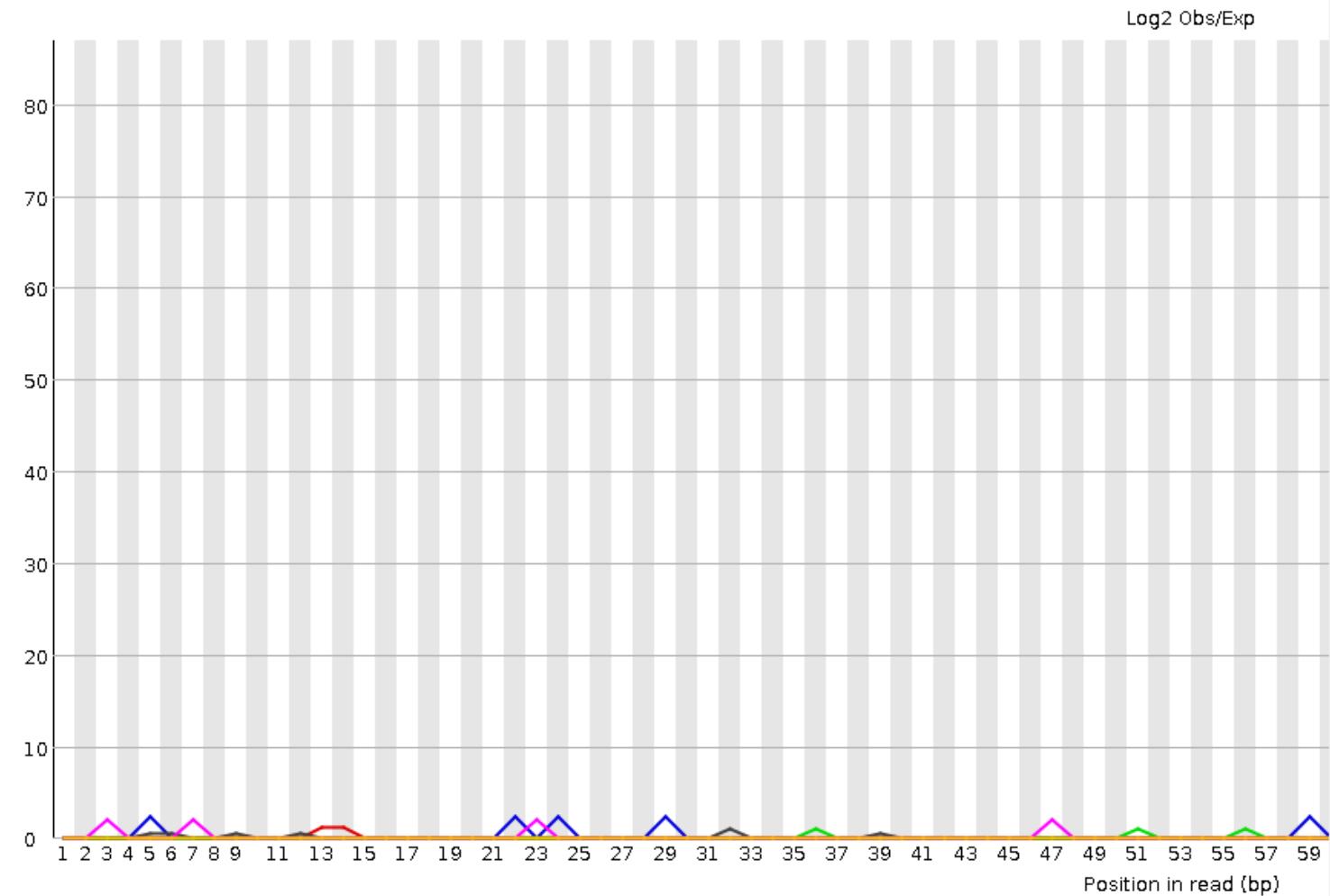
FastQC Report

Summary

- [Basic Statistics](#)
- [Per base sequence quality](#)
- [Per tile sequence quality](#)
- [Per sequence quality scores](#)
- [Per base sequence content](#)
- [Per sequence GC content](#)
- [Per base N content](#)
- [Sequence Length Distribution](#)
- [Sequence Duplication Levels](#)
- [Overrepresented sequences](#)
- [Adapter Content](#)
- [Kmer Content](#)

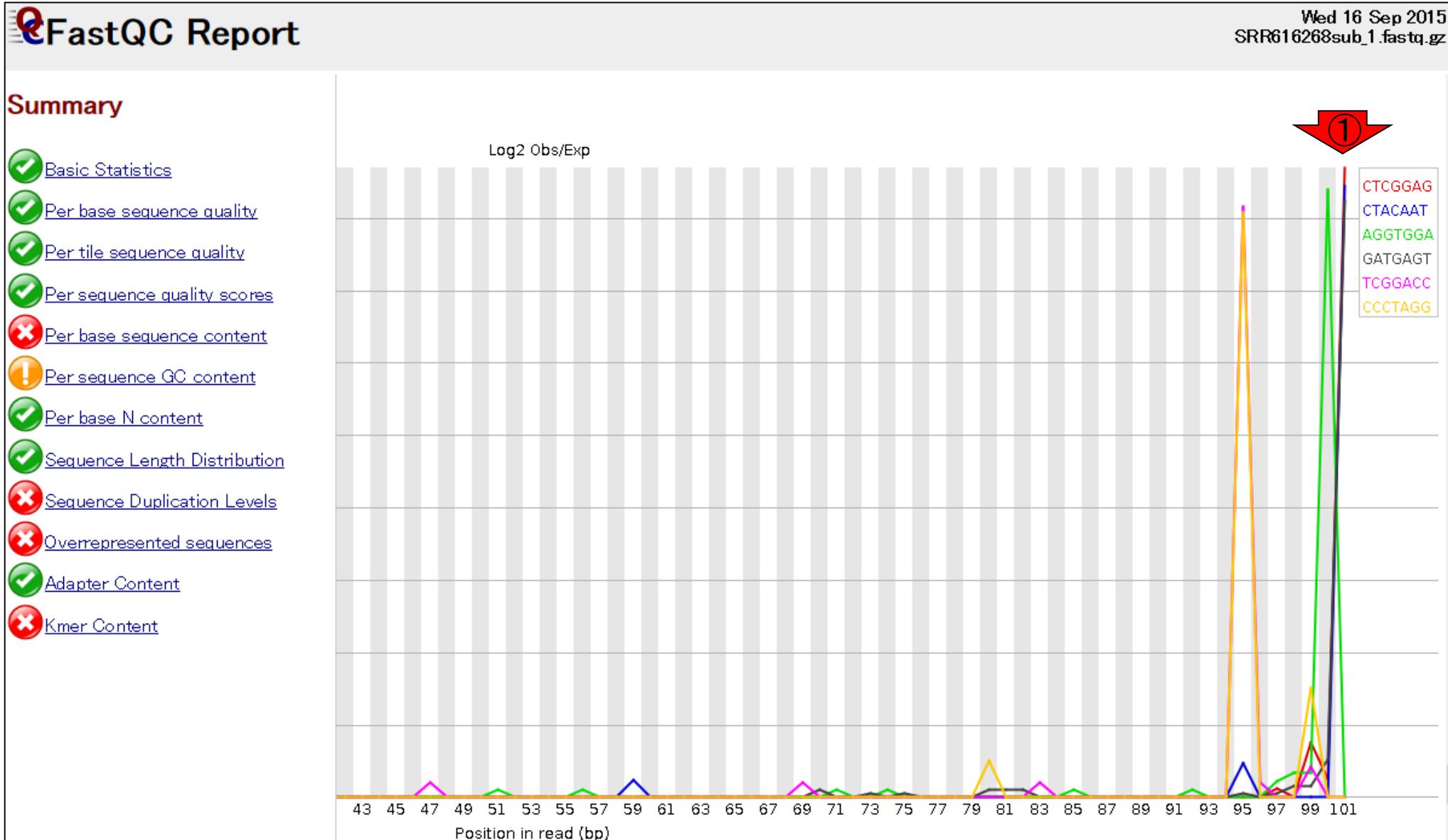
①

Kmer Content



W19-2:FastQC

①リードの右側(3'側)を表示。極端に多いk-merの上位6個が右側に偏って存在することがわかる



W19-2:FastQC

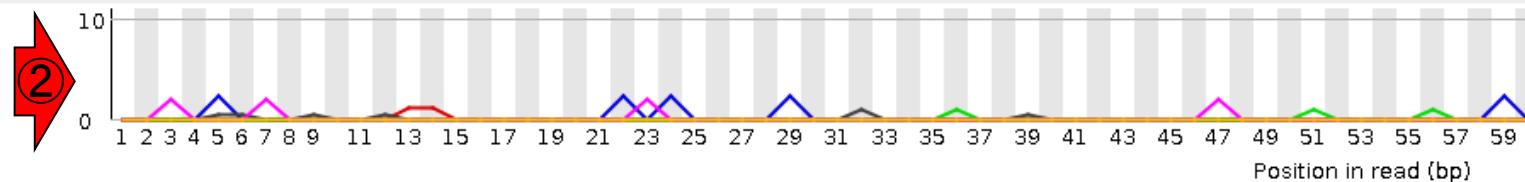
① Kmer Content項目のちょっと下のほうを表示。②上の折れ線グラフは、③赤枠で示す観測値/期待値が大きい上位6個をプロットしたもの

Wed 16 Sep 2015
SRR616268sub_1.fastq.gz

FastQC Report

Summary

- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content
- Kmer Content



Sequence	Count	PValue	Obs/Exp Max	Max Obs/Exp Position
CTCGGAG	395	0.0	86.890045	101
CTACAAT	215	0.0	84.51276	101
AGGTGGA	445	0.0	83.98286	100
GATGAGT	960	0.0	82.54421	101
TCGGACC	235	0.0	81.66475	95
CCCTAGG	100	0.0	80.80512	95
OCTAGGT	45	1.4897523E-9	78.56054	96
GTTGTGG	965	0.0	75.840096	101
GGCCCTG	180	0.0	75.75481	100
TACTACA	20	0.0016262057	75.75481	96
TAGGTGG	225	0.0	74.07136	99
GTTCTCT	75	0.0	74.02692	101
CGGGCCT	235	0.0	71.00822	1
TCCCTCG	50	3.430614E-9	70.70448	98
CCCACAC	440	0.0	70.0158	95
TACTTAC	405	0.0	69.5948	95

①デフォルトでFastQCを実行。②htmlレポートの
ファイル名をSRR616268sub_1_default.htmlに変更

W19-3:FastQC

```
iu@bielinux[srp017156_trim1] pwd [ 5:09 午後]
/home/iu/Documents/srp017156_trim1
iu@bielinux[srp017156_trim1] ls SRR616268sub_1* [ 5:09 午後]
SRR616268sub_1_663c2ebdd882.bam      SRR616268sub_1_fastqc.zip
SRR616268sub_1_663c2ebdd882.bam.bai   SRR616268sub_1.fastq.gz
SRR616268sub_1_663c2ebdd882.bam.txt   SRR616268sub_1_nogroup.html
SRR616268sub_1_663c2ebdd882_QC.pdf
iu@bielinux[srp017156_trim1] fastqc2 -q SRR616268sub_1.fastq.gz
iu@bielinux[srp017156_trim1] mv SRR616268sub_1_fastqc.html SRR616268sub_1_default.html
iu@bielinux[srp017156_trim1] ls SRR616268sub_1* [ 5:09 午後]
SRR616268sub_1_663c2ebdd882.bam      SRR616268sub_1_default.html
SRR616268sub_1_663c2ebdd882.bam.bai   SRR616268sub_1_fastqc.zip
SRR616268sub_1_663c2ebdd882.bam.txt   SRR616268sub_1.fastq.gz
SRR616268sub_1_663c2ebdd882_QC.pdf   SRR616268sub_1_nogroup.html
iu@bielinux[srp017156_trim1] cp SRR616268sub_1_default.html ~/Desktop/mac_share
iu@bielinux[srp017156_trim1] [ 5:09 午後]
```

W19-4 : FastQC

①SRR616268sub_1_default.htmlのKmer Content項目を表示。極端に多いk-merの上位6個が左側(5'側)に偏って存在していることがわかる

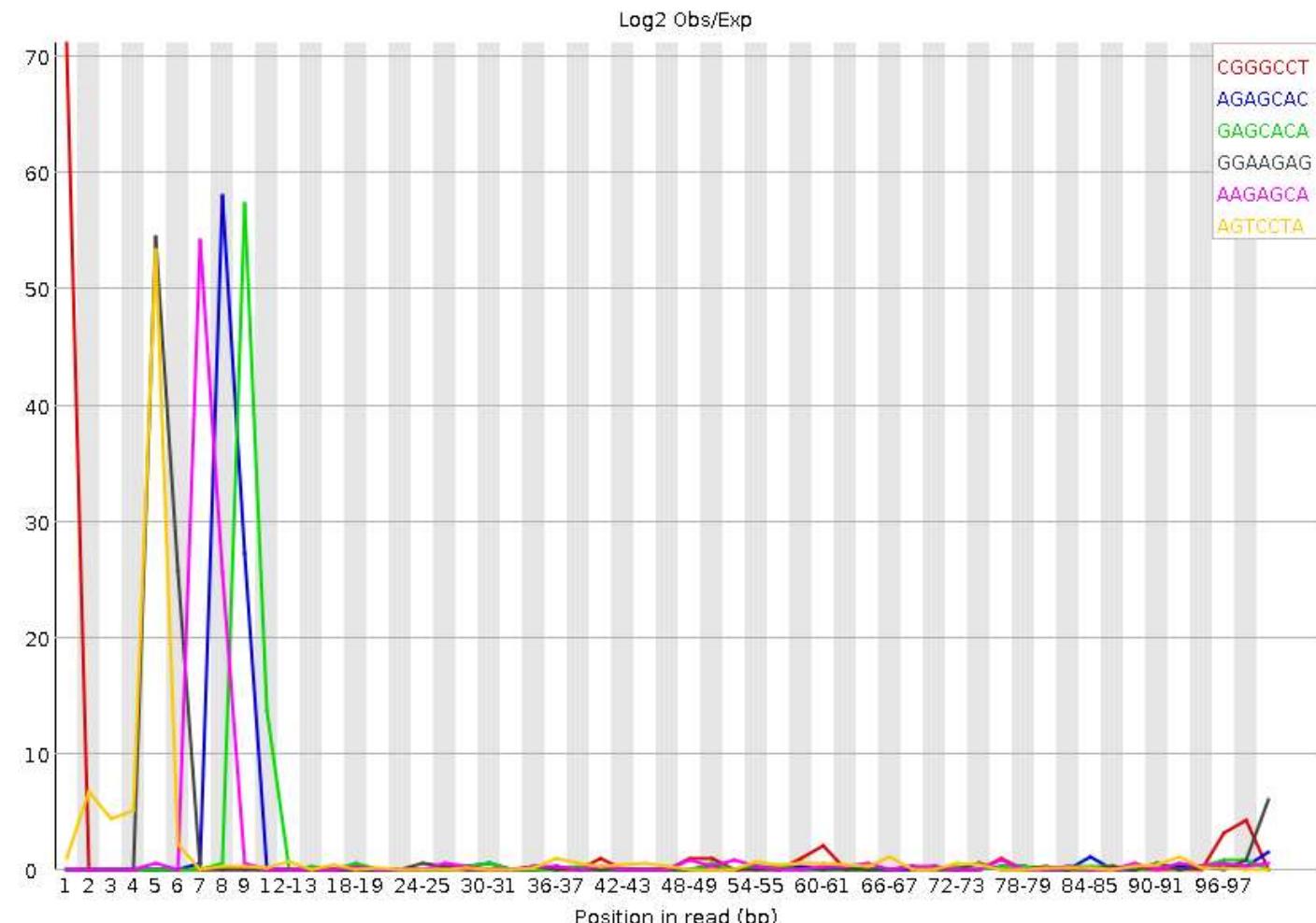
Wed 16 Sep 2015
SRR616268sub_1.fastq.gz

FastQC Report

Summary

- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content
- Kmer Content

Kmer Content



W19-4 : FastQC

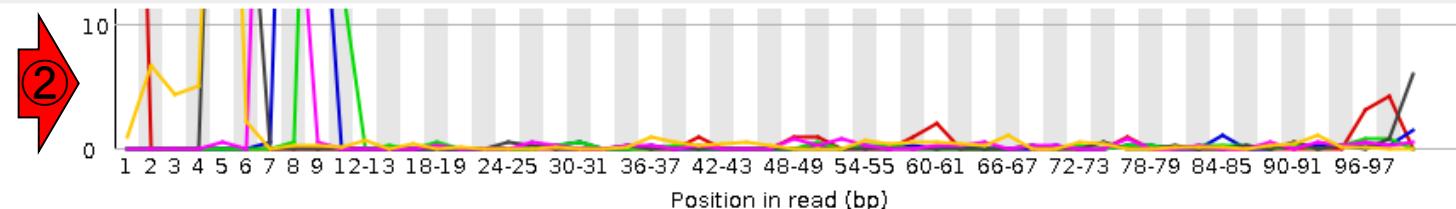
① Kmer Content 項目のちょっと下のほうを表示。
②上の折れ線グラフは、③赤枠で示す観測値/期待値が大きい上位6個をプロットしたもの

Wed 16 Sep 2015
SRR616268sub_1.fastq.gz

FastQC Report

Summary

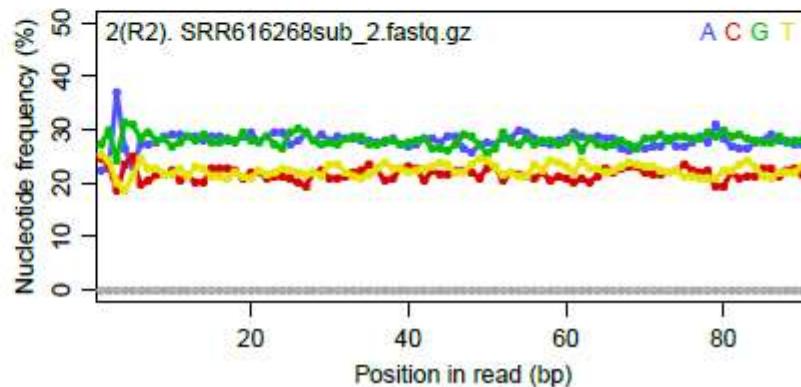
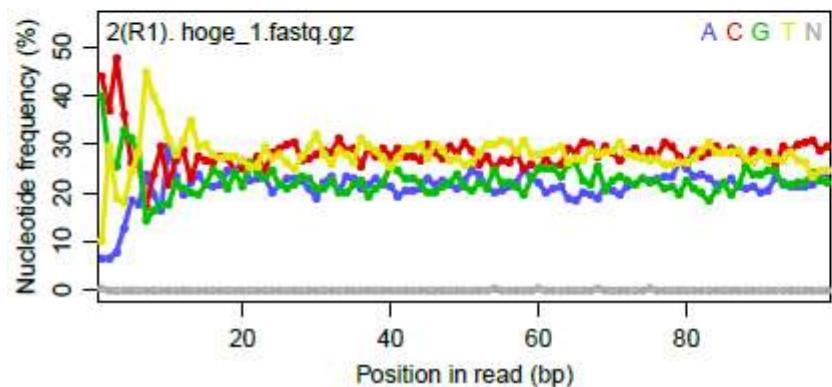
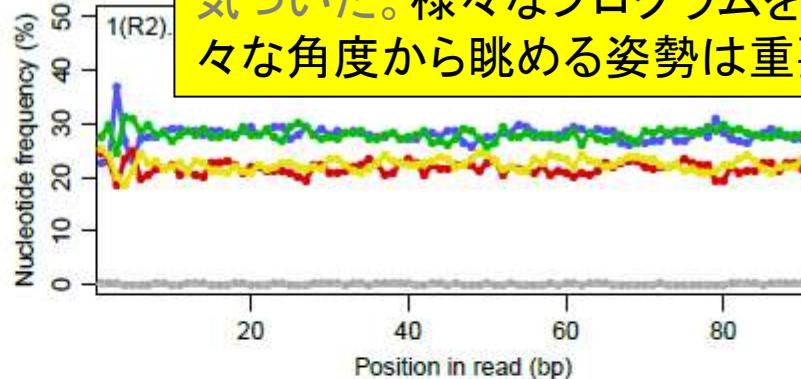
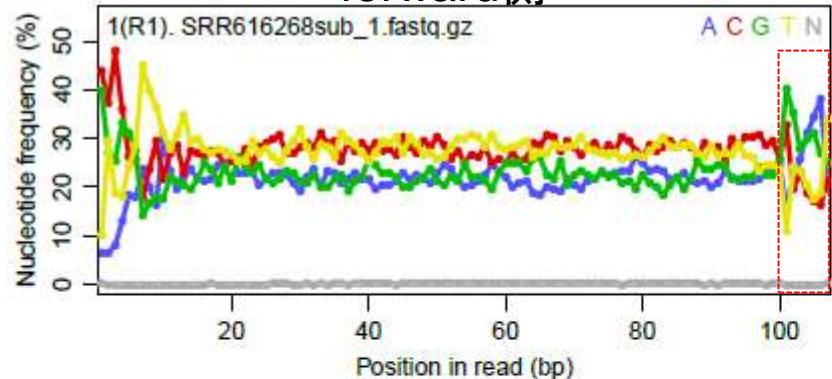
- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content
- Kmer Content



Sequence	Count	PValue	Obs/Exp Max	Max Obs/Exp Position
CGGGCCT	235	0.0	71.00822	1
AGAGCAC	835	0.0	58.01711	8
GAGCACA	845	0.0	57.330517	9
GGAAGAG	880	0.0	54.476883	5
AAGAGCA	885	0.0	54.169106	7
AGTCCTA	1570	0.0	53.35557	5
GTCCAGT	1555	0.0	53.00536	1
CCAGTCC	1655	0.0	51.225075	3
CAGTCCT	1650	0.0	51.07447	4
CCTCTAT	30	0.008129199	50.4628	2
GAAGAGC	975	0.0	49.68645	6
GTCCTAC	1720	0.0	49.289246	6
TCCTACA	1760	0.0	48.169033	7
TCGGAAG	1005	0.0	47.701153	3
CCCCCCT	160	0.0	47.40806	1

遺言(独り言)

forward側



FastQC実行時に「--nogroup」オプションをつけた場合(W19-2)と、つけるないデフォルトの場合(W19-4)でKmer_Content項目の結果が変わるのはいかがなものか…(個人の感想です)。門田は①QuasR実行結果(PDFファイル)を眺めることで、アダプタ一除去プログラム(FaQCs)でもとりきれないものがあることを学び、FastQCの--nogroupオプションに気づいた。様々なプログラムを活用し、結果を様々な角度から眺める姿勢は重要ではないだろうか

第5回原稿PDFのp200

もしれない。

話の展開上本文中では省略したが、結論としては $f_{100-107}$ 問題に QC 段階で気づくことはできる [W15-5]。具体的には、`--nogroup` オプションをつけて FastQC を実行した結果を眺めればよい。特に Kmer Contents の項目は、ゲノムアセンブリのところでも述べた k -mer (ver. 0.11.3 のデフォルトは $k=7$) の出現頻度をリードのポジションごとに調べ、出現頻度の期待値に比べて実測値が極端に多い上位の k -mer とその位置をリストアップしたものである。また、`--nogroup` は「長いリードの場合に 10 番目以降のポジションを一定幅でグループ化する（デフォルト）」機能をオフにするオプションである [W19-1]。著者らは、`--nogroup` オプションの有無によって Kmer Contents 項目の結果までが異なることを最近まで知らなかった。つまり、`--nogroup` オプションをつけずにデフォルトで実行し

第5回は、乳酸菌RNA-seqデータ解析の話。
①第6回は、乳酸菌ゲノムデータの *de novo* ゲノムアセンブリの話。第5回(*L. casei* 12A)と第6回(②*L. hokkaidonensis* LOOC260^T)では、乳酸菌株が異なる点に注意!

た *Fa* と第6回(② *L. hokkaidonensis* LOOC260^T)で
conte は、乳酸菌株が異なる点に注意!
けなかったのである [W19-4]。第6回は、アセンブルプ
ログラム Velvet をオプションつきでインストールする
ことで指定可能な数値範囲を変更できること、複数の異なる
k-mer で実行した乳酸菌ゲノムアセンブル結果の違いな
どを紹介する予定である。



謝 詞

本連載の一部は、国立研究開発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。乳酸菌 *Lactobacillus hokkaidonensis* LOOC260^T ゲノム配列決定部分については、原著論文著者（遠野雅徳氏、谷澤靖洋氏、神沼英里氏、中村保一氏、有田正規氏）より詳細情報をいただきました。

2

Contents

- Illumina HiSeqデータ(トランスクリプトーム)の乳酸菌ゲノムへのマッピング
 - W14: QuasRパッケージを用いたマッピングの事前準備と本番
 - W15: 結果の解説、forward側の100–107bp付近に問題があることを特定
- トリミング、*de novo*トランスクリプトームアセンブリとマッピングの再実行
 - W16: 問題のある領域(forward側の100–107bp)のトリミング
 - W17: トリム後のデータでアセンブリを再実行(Rockhopper2; クラスパス設定関連Tips含む)
 - W18: トリム後のデータでマッピングを再実行(QuasR)
 - W19: トリム前のデータでクオリティチェックを再実行(FastQC)
- Illumina MiSeqデータ(乳酸菌ゲノム)の特徴と前処理
 - W4: FastQC、W5: FaQCs、W6: 再度FastQC
- *de novo*ゲノムアセンブリ
 - W7: Bio-LinuxにプレインストールされているVelvet (ver. 1.2.09)を上限のk=31で実行
 - W8: k=31のアセンブリ結果をRで確認。k=141で実行し、k=31の結果と同じになるのを確認
 - W9: Velvet (ver. 1.2.10)のインストール
 - W10: Velvet (ver. 1.2.10)の実行



ここからは...

(Rで)塩基配列解析

~NGS、RNA-seq、ゲノム、トランскриプトーム、正規化、発現変動、統計、モラ
(last modified 2016/06/03, since 2011)

What's
このウ
リソソ
法(W
ホリキ

- 書籍 | 日本乳酸菌学会誌 | について (last modified 2016/05/12) NEW
- 書籍 | 日本乳酸菌学会誌 | 第1回イントロダクション (last modified 2015/09/11)
- 書籍 | 日本乳酸菌学会誌 | 第2回GUI環境からコマンドライン環境へ (last modified 2016/05/12)
- 書籍 | 日本乳酸菌学会誌 | 第3回Linux環境構築からNGSデータ取得まで (last modified 2016/05/12)
- 書籍 | 日本乳酸菌学会誌 | 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 (last modified 2016/05/12)
- 書籍 | 日本乳酸菌学会誌 | 第5回アセンブル、マッピング、そしてQC (last modified 2016/05/12)
- 書籍 | 日本乳酸菌学会誌 | 第6回ゲノムアセンブリ (last modified 2016/04/23) 
- 書籍 | 日本乳酸菌学会誌 | 第7回ロングリードアセンブリ (last modified 2016/05/12)
- イントロ | 一般 | ランダムに行き来を自由に (last modified 2014/07/17)

日本乳酸菌学会誌の①連載第6回ゲノムアセンブリの話。公共DBやデータ取得関連の話は省略するので、必要に応じて②オリジナルのウェブ資料で自習してください

書籍 | 日本乳酸菌学会誌 | 第6回ゲノムアセンブリ

日本乳酸菌学会誌の第6回分です。Linuxコマンドのリンク先は主に日経BP社様です。原稿PDFの「はじめに」項目のところにtypoがあります。誤:するであれば、正:するの"であれば、ですねm(_)_m。また、「配列長によるフィルタリング」項目の最後の文章「これらについては、第7回で詳述する予定である。」についてですが、これは「これらについては、"第8回以降"で詳述する予定である。」と読み替えてください。第7回ドラフト原稿作成時点で、これらの内容は含まれていないためです(2016年4月23日追加)。

- 原稿PDF
- ウェブ資料PDF 
 - Windows用(2016.03.29版; 約25MB)
 - Macintosh用
- (共有フォルダ設定情報を含む)連載第3回終了時点以降のovaファイルからのインストール手順:
 - Windows用(2015.12.28版; 約2MB)
 - Macintosh用(2015.12.28版; 約2MB)

W2-1：乳酸菌データ

http://www.ncbi.nlm.nih.gov/pubmed/25879859 Complete genome sequ... x

NCBI Resources How To

PubMed Advanced Search

Abstract

BMC Genomics. 2015 Mar 25;16:240. doi: 10.1186/s12864-015-1435-2.

Complete genome sequence and analysis of *Lactobacillus hokkaidonensis* LOOC260(T), a psychrotrophic lactic acid bacterium isolated from silage.

Tanizawa Y^{1,2}, Tohno M³, Kaminuma E⁴, Nakamura Y⁵, Arita M^{6,7}.

Author information

Abstract

BACKGROUND: *Lactobacillus hokkaidonensis* is an obligate heterofermentative lactic acid bacterium, which is isolated from Timothy grass silage in Hokkaido, a subarctic region of Japan. This bacterium is expected to be useful as a silage starter culture in cold regions because of its remarkable psychrotolerance; it can grow at temperatures as low as 4°C. To elucidate its genetic background, particularly in relation to the source of psychrotolerance, we constructed the complete genome sequence of *L. hokkaidonensis* LOOC260(T) using PacBio single-molecule real-time sequencing technology.

RESULTS: The genome of LOOC260(T) comprises one circular chromosome (2.28 Mbp) and two circular plasmids: pLOOC260-1 (81.6 kbp) and pLOOC260-2 (41.0 kbp). We identified diverse genetic elements, such as prophages, integrated and conjugative elements, and conjugative plasmids, which may reflect adaptation to plant-associated niches. Comparative genome analysis also revealed unique genomic features, such as genes involved in pentose assimilation and NADPH gene.

CONCLUSIONS: This is the first complete genome in the *L. vaccinostercus* group, which is characterized by a unique combination of traits. The genomic information obtained in this study provides insight into the genetic evolution of this group. We also found several factors that may contribute to the ability of *L. hokkaidonensis* to grow at cold temperatures. The results of this study will facilitate further research on the cold-tolerance mechanism of *L. hokkaidonensis*.

PMID: 25879859 [PubMed - in process] PMCID: PMC4377027 Free PMC Article



原著論文(PMID: 25879859)の①Full textリンク先で全文を見られる。②Availability of supporting dataという項目をよく眺めると、NGS生データがDDBJ Sequence Read Archive (DDBJ SRA; 略してDRA)に DRR024500とDRR024501というIDで登録されていることがわかる。スライドを見るだけ

Send to: ▾

Full text links

Read free full text at BioMed Central

PMC Full text **FREE**

Save items

Add to Favorites

Similar articles

Availability of supporting data

The complete genome sequence of *L. hokkaidonensis* LOOC260^T and its annotations were deposited at DDBJ/ENA/GenBank under accession numbers AP014680 (chromosome), AP014681 (plasmid pLOOC260-1), and AP014682 (plasmid pLOOC260-2). All of the sequencing data were deposited in the DDBJ Sequence Read Archive under accession numbers DRR024500 and DRR024501. The phylogenetic tree and associated data matrix for in Additional file 1: Figure S2 are available in TreeBASE database (Accession URL: <http://purl.org/phylo/treebase/phylows/study/TB2:S17206>).

Related information

...
...
...

解析データ

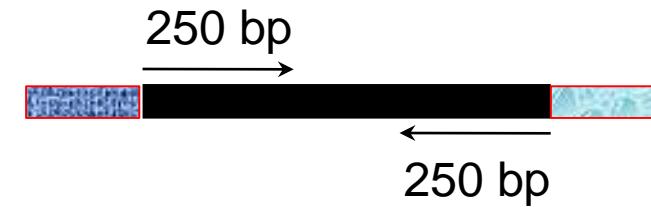
PacBio RS IIデータ(DRR024500)

- DRR024500は登録内容に問題があったことが判明し消滅
- 4セル分のデータ。DRR054113–054116に差し替えられている
- セルあたり約15万リード。4セル分なので約60万リード

Illumina MiSeqデータ(DRR024501)

- paired-endゲノムデータ
- リード長は、forward側とreverse側共に250 bp
- オリジナルは2,971,310リード。最初の300,000リードを解析
- forward側(DRR024501sub_1.fastq.gz)
- reverse側(DRR024501sub_2.fastq.gz)

①乳酸菌ゲノム配列決定論文は、2種類のNGS機器から得られたデータを併用している。第6回はIllumina MiSeqデータ(DRR024501)を、そして第7回はPacBioデータを取り扱っている



W4-1 : FastQC

DRR024501の最初の30万リードに限定したファイルに対して、①W4-1のFastQCを実行するところから行います。②をコピペ

書籍 | 日本乳酸菌学会誌 | 第6回ゲノムアセンブリ

日本乳酸菌学会誌の第6回分です。Linuxコマンドのリンク先は主に日経BP社様です。原稿PDFの「はじめに」項目のところにtypoがあります。誤:するであれば、正:する"の"であれば、ですねm()m。また、「配列長によるフィルタリング」項目の最後の文章「これらについては、第7回で詳述する予定である。」についてですが、これは「これらについては、"第8回以降"で詳述する予定である。」と読み替えてください。

第7回ドラフト原稿作成時点での内容は含まれ

- ・ [原稿PDF](#)
- ・ ウェブ資料PDF
 - [Windows用](#)(2016.03.29版; 約25MB)
 - Macintosh用
- ・ [\(共有フォルダ設定情報を含む\)連載第3回終了時](#)
 - [Windows用](#)(2015.12.28版; 約2MB)
 - [Macintosh用](#)(2015.12.28版; 約2MB)

```
wget -cq ftp://ftp.ddbj.nig.ac.jp/ddbj_database/dra/fastq/DRA002/DRA002643/DRX6  
ls -lh
```

① [FastQC \(ver. 0.11.4\)\[W4-1\]](#)

FastQC実行結果を共有フォルダ(/home/iu/Desktop/mac_share)に保存している。

```
cd ~/Documents/DRR024501
```

②
pwd

```
ls -l
```

```
time fastqc2 -q DRR024501sub_1.fastq.gz --outdir=/home/iu/Desktop/mac_share
```

```
time fastqc2 -q DRR024501sub_2.fastq.gz --outdir=/home/iu/Desktop/mac_share
```

```
ls -l ~/Desktop/mac_share
```

```
date
```

- ・ FastQC実行結果を眺める(forward側)[W4-2]

[DRR024501sub_1_fastqc.html](#)

- ・ FastQC実行結果を眺める(reverse側)[W4-3]

[DRR024501sub_2_fastqc.html](#)

アダプタートrimming

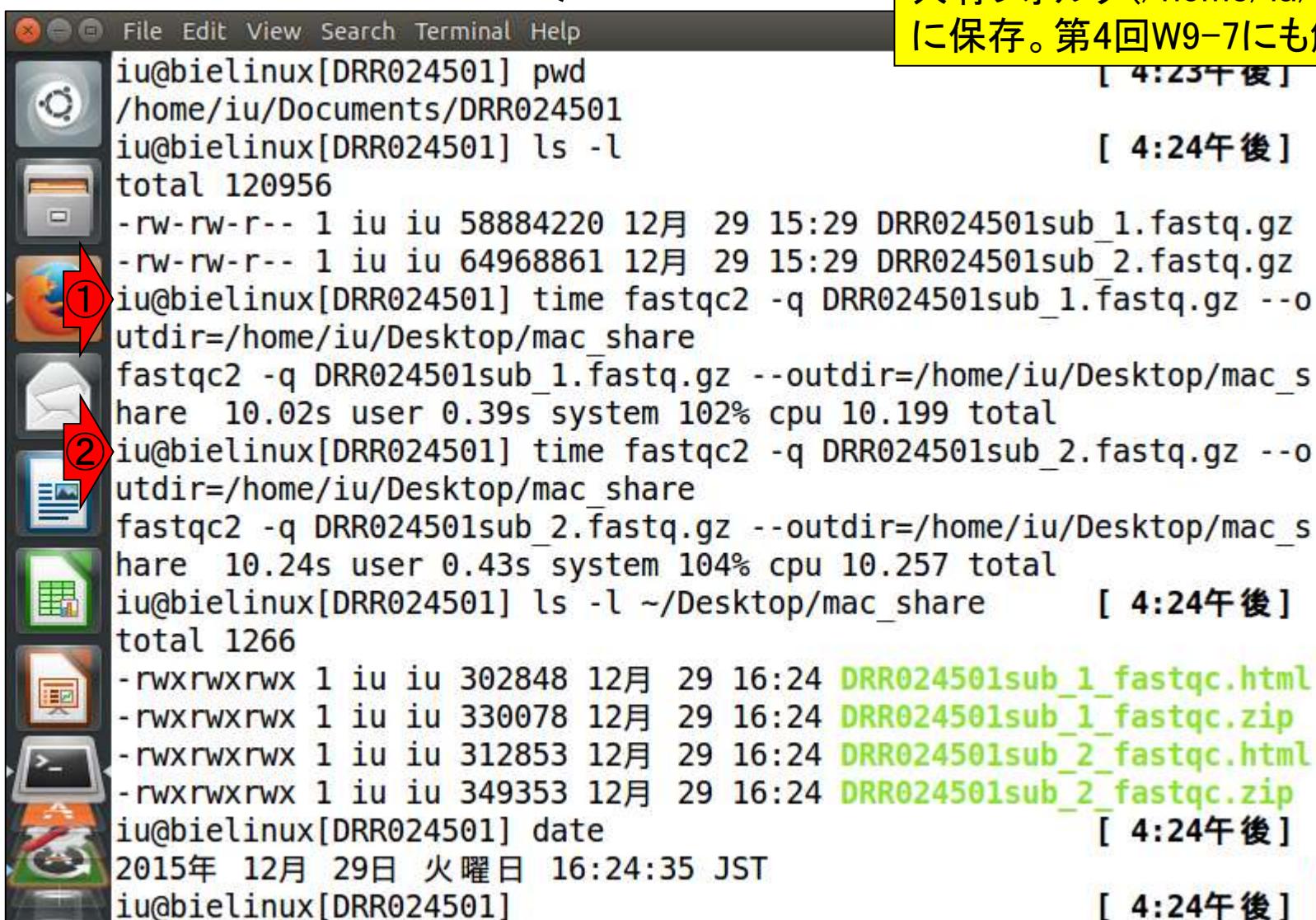
- ・ [FaQCs \(Lo and Chain, BMC Bioinformatics, 2014\)を実行\[W5-1\]](#)

アダプター配列や低クオリティリードの除去を行うプログラムFaQCsを実行し、結果をresultディレクトリに

W4-1 : FastQC

①②FastQC (ver. 0.11.4)をtimeコマンドをつけて実行。各ファイルにつき約10秒。実行結果は共有フォルダ(/home/iu/Desktop/mac_share)に保存。第4回W9-7にも解説あり

```
File Edit View Search Terminal Help
iu@bielinux[DRR024501] pwd [ 4:23 午後 ]
/home/iu/Documents/DRR024501
iu@bielinux[DRR024501] ls -l [ 4:24 午後 ]
total 120956
-rw-rw-r-- 1 iu iu 58884220 12月 29 15:29 DRR024501sub_1.fastq.gz
-rw-rw-r-- 1 iu iu 64968861 12月 29 15:29 DRR024501sub_2.fastq.gz
iu@bielinux[DRR024501] time fastqc2 -q DRR024501sub_1.fastq.gz --o utdir=/home/iu/Desktop/mac_share
fastqc2 -q DRR024501sub_1.fastq.gz --outdir=/home/iu/Desktop/mac_share 10.02s user 0.39s system 102% cpu 10.199 total
iu@bielinux[DRR024501] time fastqc2 -q DRR024501sub_2.fastq.gz --o utdir=/home/iu/Desktop/mac_share
fastqc2 -q DRR024501sub_2.fastq.gz --outdir=/home/iu/Desktop/mac_share 10.24s user 0.43s system 104% cpu 10.257 total
iu@bielinux[DRR024501] ls -l ~/Desktop/mac_share [ 4:24 午後 ]
total 1266
-rwxrwxrwx 1 iu iu 302848 12月 29 16:24 DRR024501sub_1_fastqc.html
-rwxrwxrwx 1 iu iu 330078 12月 29 16:24 DRR024501sub_1_fastqc.zip
-rwxrwxrwx 1 iu iu 312853 12月 29 16:24 DRR024501sub_2_fastqc.html
-rwxrwxrwx 1 iu iu 349353 12月 29 16:24 DRR024501sub_2_fastqc.zip
iu@bielinux[DRR024501] date [ 4:24 午後 ]
2015年 12月 29日 火曜日 16:24:35 JST
iu@bielinux[DRR024501] [ 4:24 午後 ]
```



W4-2: 結果を眺める

①共有フォルダに保存することで、使いなれたホストOS(この場合Windows)上でFastQC実行結果ファイルを眺めることができる。②forward側の結果

The screenshot shows a Windows desktop environment. On the left, a file browser window is open, showing a folder structure under 'share'. Two red arrows with the number '1' point to the 'share' folder icon and the 'share' folder entry in the list. Another red arrow with the number '2' points to the 'DRR024501sub_1_fastqc.html' file entry. The title bar of the browser window indicates the path is 'C:\Users\kadota\Desktop\share\DRR024501sub_1_fastqc.html'.
On the right, a 'FastQC Report' window is displayed. The title bar shows the file path 'C:\Users\kadota\Desktop\share\DRR024501sub_1_fastqc.html' and the date 'Tue 29 Dec 2015'. The window contains a 'Summary' section with various quality check links, a 'Basic Statistics' table, and a 'Per base sequence quality' chart.

Measure	Value
Filename	DRR024501sub_1.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	300000
Sequences flagged as poor quality	0
Sequence length	251
%GC	38

Produced by FastQC (version 0.11.4)

W4-2: 結果を眺める

FastQC実行結果の解説は第4回W8-2とW17-2にもあり。①入力ファイル。これはforward側の結果。②リード数。30万リードであることがわかる。③配列長。251 bpであることがわかる

FastQC Report

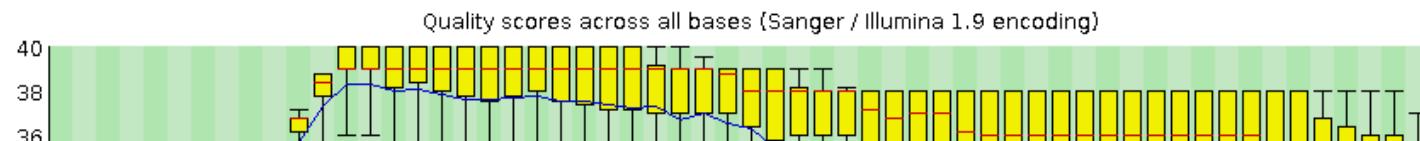
Summary

- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content
- Kmer Content

Basic Statistics

Measure	Value
Filename	DRR024501sub_1.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	300000
Sequences flagged as poor quality	0
Sequence length	251
%GC	38

Per base sequence quality



W4-2: 結果を眺める

FastQC Report

Summary

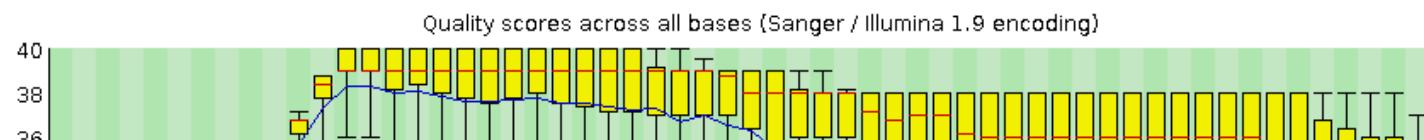
- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content
- Kmer Content



Basic Statistics

Measure	Value
Filename	DRR024501sub_1.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	300000
Sequences flagged as poor quality	0
Sequence length	251
%GC	38

Per base sequence quality



全体的なクオリティは、①赤枠内の色でわかる。概ね、信号通りの理解でよい。第4回W17-2のRNA-seqデータのFastQC結果と比較するとよい。ゲノムデータの場合はRNA-seqデータよりもcoverageが一定なので、一般によりよい結果になる

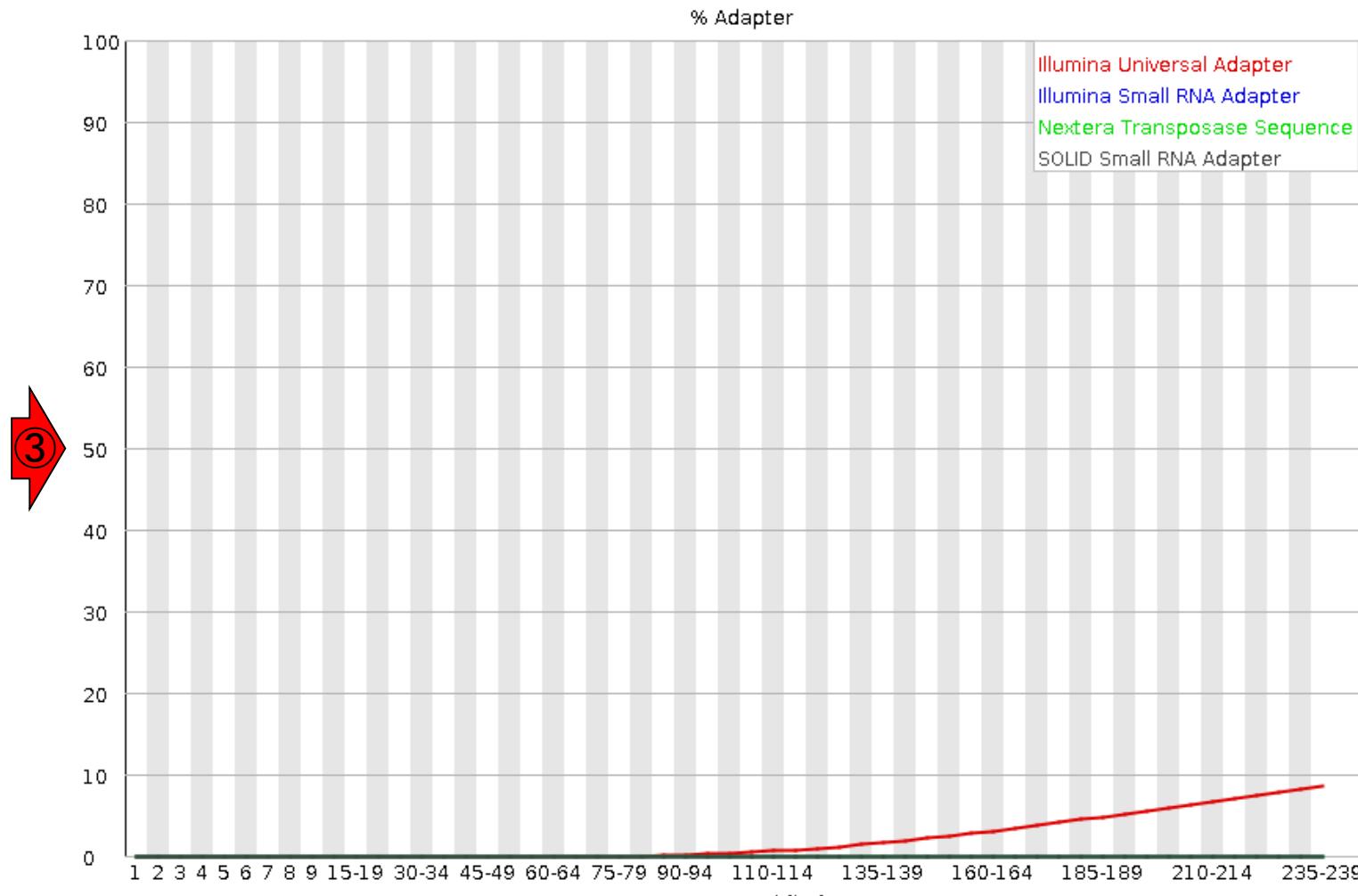
W4-2: 結果を眺める

例えば黄色の項目①Adapter Contentを眺める。②横軸は全251 bpからなるリードのポジション、③縦軸はポジションごとの既知のアダプター配列を含む割合を示す

Summary

- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content
- Kmer Content

Adapter Content



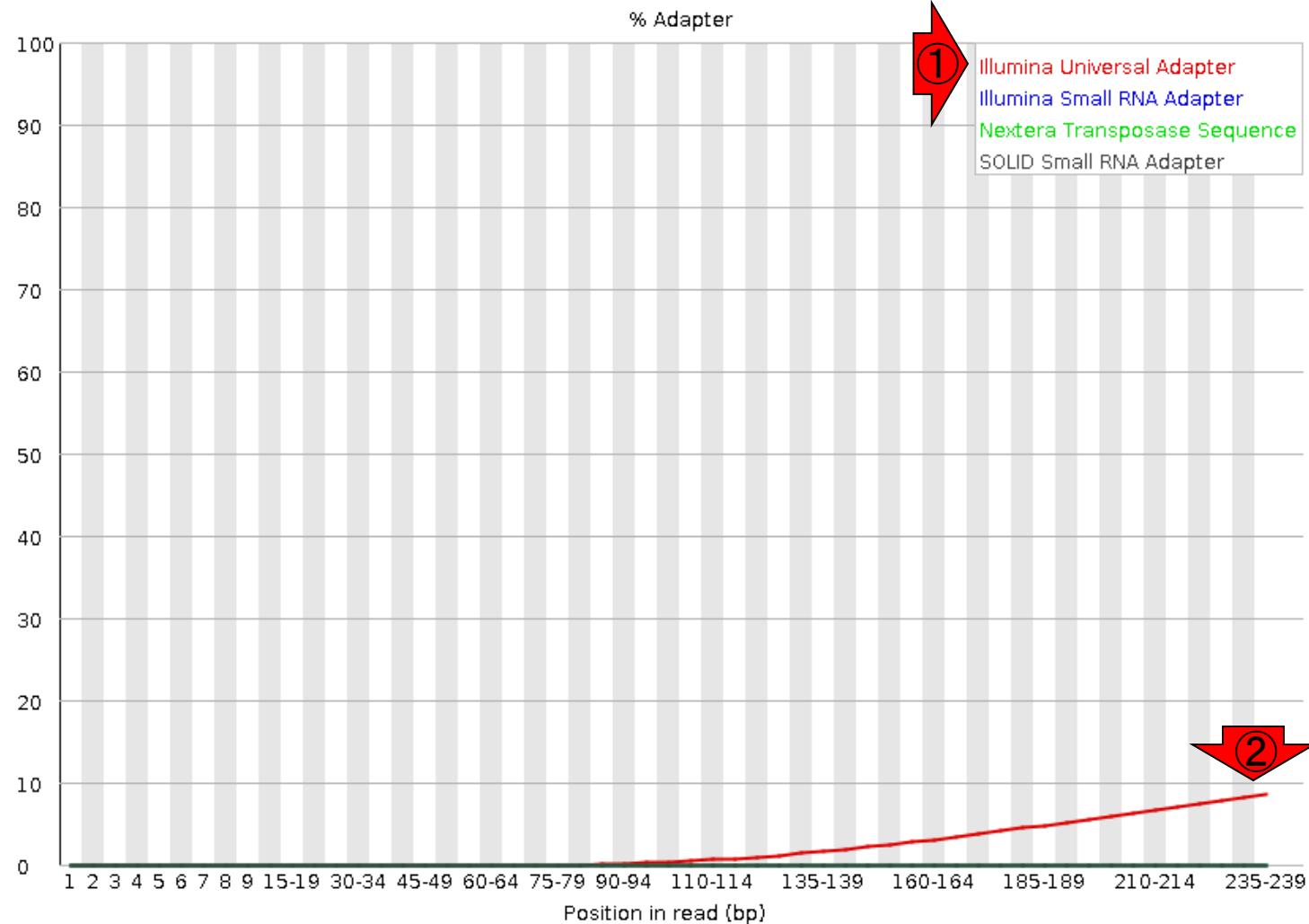
W4-2: 結果を眺める

① Illumina Universal Adapterが3'側に多く含まれており、②終端付近では全リードの10%弱に含まれるほどであることもわかる

Summary

- [Basic Statistics](#)
- [Per base sequence quality](#)
- [Per tile sequence quality](#)
- [Per sequence quality scores](#)
- [Per base sequence content](#)
- [Per sequence GC content](#)
- [Per base N content](#)
- [Sequence Length Distribution](#)
- [Sequence Duplication Levels](#)
- [Overrepresented sequences](#)
- [Adapter Content](#)
- [Kmer Content](#)

Adapter Content



W4-3: 結果を眺める

Tue 29 Dec 2015
DRR024501sub_2.fastq.gz

FastQC Report

Summary

- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content
- Kmer Content

Basic Statistics

Measure	Value
Filename	DRR024501sub_2.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	300000
Sequences flagged as poor quality	0
Sequence length	251
%GC	38

①

②

③

Per base sequence quality

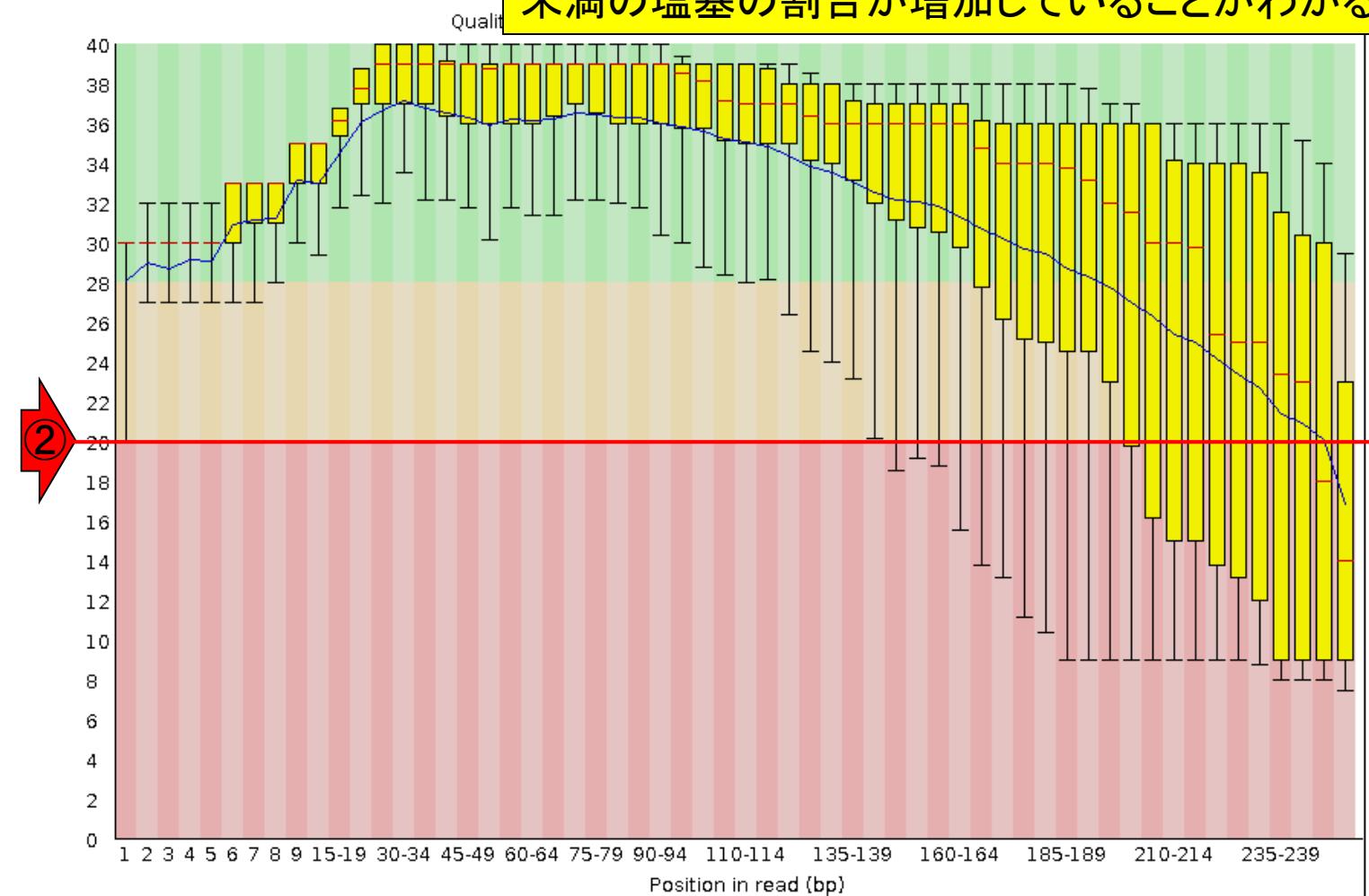


W4-3: 結果を眺める

Summary

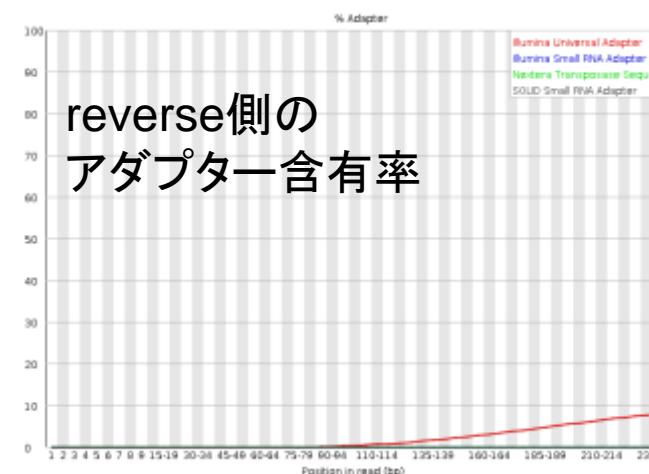
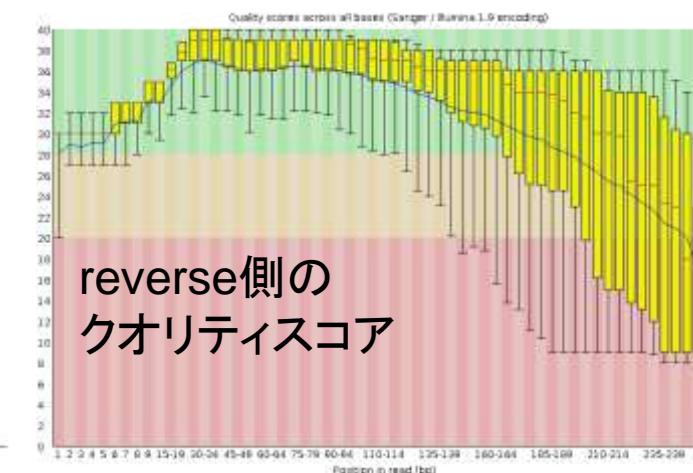
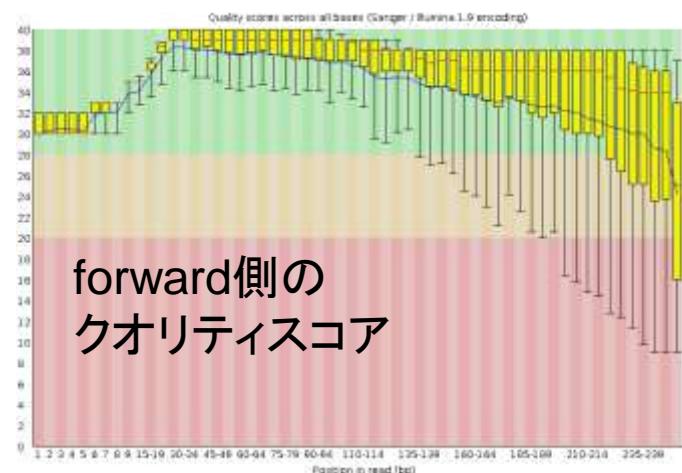
- Basic Statistics
- Per base sequence quality ①
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content
- Kmer Content

✖ Per base sequence quality



①Per base sequence qualityの項目が赤になっているので眺める。この図の縦軸はクオリティスコア。赤線のスコア20を超えているかどうかが1つの目安。この場合、3'末端にいくほどスコア20未満の塩基の割合が増加していることがわかる

W4-4: FastQCまとめ



Contents

- Illumina HiSeqデータ(トランスクリプトーム)の乳酸菌ゲノムへのマッピング
 - W14:QuasRパッケージを用いたマッピングの事前準備と本番
 - W15:結果の解説、forward側の100–107bp付近に問題があることを特定
- トリミング、*de novo*トランスクリプトームアセンブリとマッピングの再実行
 - W16:問題のある領域(forward側の100–107bp)のトリミング
 - W17:トリム後のデータでアセンブリを再実行(Rockhopper2; クラスパス設定関連Tips含む)
 - W18:トリム後のデータでマッピングを再実行(QuasR)
 - W19:トリム前のデータでクオリティチェックを再実行(FastQC)
- Illumina MiSeqデータ(乳酸菌ゲノム)の特徴と前処理
 - W4:FastQC、W5:FaQCs、W6:再度FastQC
- *de novo*ゲノムアセンブリ
 - W7:Bio-LinuxにプレインストールされているVelvet (ver. 1.2.09)を上限のk=31で実行
 - W8:k=31のアセンブリ結果をRで確認。k=141で実行し、k=31の結果と同じになるのを確認
 - W9:Velvet (ver. 1.2.10)のインストール
 - W10:Velvet (ver. 1.2.10)の実行



W5-1 : FaQCs実行

```
File Edit View Search Terminal Help
iu@bielinux[DRR024501] pwd
/home/iu/Documents/DRR024501
iu@bielinux[DRR024501] ls -l
total 120956
-rw-rw-r-- 1 iu iu 58884220 12月 29 15:29 DRR024501sub_1.fastq.gz
-rw-rw-r-- 1 iu iu 64968861 12月 29 15:29 DRR024501sub_2.fastq.gz
iu@bielinux[DRR024501] time FaQCs.pl -adapter -p DRR024501sub_1.fa
stq.gz DRR024501sub_2.fastq.gz -d result
Bwa extension trimming algorithm is used.
Processing DRR024501sub_1.fastq.gz DRR024501sub_2.fastq.gz file
```

①アダプター配列や低クオリティリードの除去を行う
プログラムFaQCsを実行し、結果をresultディレクトリ
に保存。約12分。画面は開始1分後の状況。FaQCs
は第4回W17-1、および第5回W1-1でも実行している

[6:13 午後]

[6:13 午後]

1

W5-2: 実行結果

FaQCs実行後の状態。①timeコマンドをつけて実行したので、②赤枠分が余分に表示されている

```
iu@bielinux[~/Documents/DRR024501]          [ 6:13 午後 ]
iu@bielinux[DRR024501] pwd
/home/iu/Documents/DRR024501
iu@bielinux[DRR024501] ls -l                [ 6:13 午後 ]
total 120956
-rw-rw-r-- 1 iu iu 58884220 12月 29 15:29 DRR024501sub_1.fastq.gz
-rw-rw-r-- 1 iu iu 6496558 12月 29 15:29 DRR024501sub_2.fastq.gz
iu@bielinux[DRR024501] time FaQCs.pl -adapter -p DRR024501sub_1.fa
stq.gz DRR024501sub_2.fastq.gz -d result
Bwa extension trimming algorithm is used.
Processing DRR024501sub_1.fastq.gz DRR024501sub_2.fastq.gz file
Processed 600000/600000
Post Trimming Length(Mean, Std, Median, Max, Min) of 596558 reads ←
with Overall quality 33.33
(244.29, 24.89, 251.0, 251, 50)
FaQCs.pl -adapter -p DRR024501sub_1.fastq.gz DRR024501sub_2.fastq.    ②
gz -d result 639.02s user 5.27s system 93% cpu 11:27.49 total
iu@bielinux[DRR024501]          [ 6:24 午後 ]
```

W5-2: 実行結果

結果が保存されているresultディレクトリを①lsしている。②FaQCsの主な実行結果ファイルは、QC.1.trimmed.fastqとQC.2.trimmed.fastq

```
File Edit View Search Terminal Help ↑ ↓ Ja 18:27
total 120956
-rw-rw-r-- 1 iu iu 58884220 12月 29 15:29 DRR024501sub_1.fastq.gz
-rw-rw-r-- 1 iu iu 64968861 12月 29 15:29 DRR024501sub_2.fastq.gz
iu@bielinux[DRR024501] time FaQCs.pl -adapter -p DRR024501sub_1.fa
stq.gz DRR024501sub_2.fastq.gz -d result
Bwa extension trimming algorithm is used.
Processing DRR024501sub_1.fastq.gz DRR024501sub_2.fastq.gz file
Processed 600000/600000
Post Trimming Length(Mean, Std, Median, Max, Min) of 596558 reads ←
with Overall quality 33.33
(244.29, 24.89, 251.0, 251, 50)
FaQCs.pl -adapter -p DRR024501sub_1.fastq.gz DRR024501sub_2.fastq.
gz -d resul 639.02s user 5.27s system 93% cpu 11:27.49 total
iu@bielinux[DRR024501] ls -l result [ 6:24午後 ]
total 371472
-rw-rw-r-- 1 iu iu 94 12月 29 18:13 fastqCount.txt
-rw-rw-r-- 1 iu iu 189492182 12月 29 18:24 QC.1.trimmed.fastq [②]
-rw-rw-r-- 1 iu iu 189628416 12月 29 18:24 QC.2.trimmed.fastq
-rw-rw-r-- 1 iu iu 424499 12月 29 18:24 QC_qc_report.pdf
-rw-rw-r-- 1 iu iu 1073 12月 29 18:24 QC.stats.txt
-rw-rw-r-- 1 iu iu 821270 12月 29 18:24 QC.unpaired.trimmed.fas
tq
iu@bielinux[DRR024501] [ 6:27午後 ]
```

W5-2: 実行結果

```
File Edit View Search Terminal Help
iu@bielinux[DRR024501] ls
DRR024501sub_1.fastq.gz  DRR024501sub_2.fastq.gz
iu@bielinux[DRR024501] cd result
iu@bielinux[result] pwd
/home/iu/Documents/DRR024501/result
iu@bielinux[result] ls -l
total 371472
-rw-rw-r-- 1 iu iu      94 12月 29 18:13 fastqCount.txt
-rw-rw-r-- 1 iu iu 189492182 12月 29 18:24 QC.1.trimmed.fastq
-rw-rw-r-- 1 iu iu 189628416 12月 29 18:24 QC.2.trimmed.fastq
-rw-rw-r-- 1 iu iu    424499 12月 29 18:24 QC_qc_report.pdf
-rw-rw-r-- 1 iu iu     1073 12月 29 18:24 QC.stats.txt
-rw-rw-r-- 1 iu iu   821270 12月 29 18:24 QC.unpaired.trimmed.fastq
iu@bielinux[result] wc QC.1.trimmed.fastq
1190532 2381064 189492182 QC.1.trimmed.fastq
iu@bielinux[result] wc QC.2.trimmed.fastq
1190532 2381064 189628416 QC.2.trimmed.fastq
iu@bielinux[result]
```

①resultディレクトリに移動し、FaQCs実行結果ファイルの行数確認。②forward側、③reverse側ともに $1,190,532$ 行 / 4 = 297,633リードが出力ファイルに含まれることがわかる。
2つのファイル合わせて $297,633 \times 2 = 595,266$ リード。このリード数を記憶に留めておく

[7:01 午後]

W5-3: 結果の概要

①FaQCs実行結果の概要は、この2つのファイル中にある。どちらも同じ内容なので、ここでは②moreでQC.stats.txtを眺める

```
File Edit View Search Terminal Help [ 7:01午後 ]
iu@bielinux[DRR024501] ls
DRR024501sub_1.fastq.gz DRR024501sub_2.fastq.gz result [ 7:01午後 ]
iu@bielinux[DRR024501] cd result [ 7:01午後 ]
iu@bielinux[result] pwd [ 7:01午後 ]
/home/iu/Documents/DRR024501/result
iu@bielinux[result] ls -l [ 7:01午後 ]
total 371472
-rw-rw-r-- 1 iu iu 94 12月 29 18:13 fastqCount.txt
-rw-rw-r-- 1 iu iu 189492182 12月 29 18:24 QC.1.trimmed.fastq
-rw-rw-r-- 1 iu iu 189628416 12月 29 18:24 QC.2.trimmed.fastq
-rw-rw-r-- 1 iu iu 424499 12月 29 18:24 QC_qc_report.pdf
-rw-rw-r-- 1 iu iu 1073 12月 29 18:24 QC.stats.txt
-rw-rw-r-- 1 iu iu 821270 12月 29 18:24 QC.unpaired.trimmed.fastq
iu@bielinux[result] wc QC.1.trimmed.fastq [ 7:01午後 ]
1190532 2381064 189492182 QC.1.trimmed.fastq
iu@bielinux[result] wc QC.2.trimmed.fastq [ 7:01午後 ]
1190532 2381064 189628416 QC.2.trimmed.fastq
iu@bielinux[result] more QC.stats.txt [ 7:01午後 ]
```

W5-3: 結果の概要

①FaQCs実行前(Before Trimming)の入力データの概要。片側30万リードなので、Readsのところが600000になっているのは妥当

```
File Edit View Search Terminal Help  
Before Trimming  
Reads #: 600000  
Total bases: 150600000  
Reads Length: 251.00  
  
After Trimming  
Reads #: 596558 (99.43 %)  
Total bases: 145731275 (96.77 %)  
Mean Reads Length: 244.29  
Paired Reads #: 595266 (99.78 %)  
Paired total bases: 145416561 (99.78 %)  
Unpaired Reads #: 1292 (0.22 %)  
Unpaired total bases: 314714 (0.22 %)  
  
Discarded reads #: 3442 (0.57 %)  
Trimmed bases: 4868725 (3.23 %)  
Reads Filtered by length cutoff (50 bp): 42 (0.01 %)  
Bases Filtered by length cutoff: 1314 (0.00 %)  
Reads Filtered by continuous base "N" (2): 3396 (0.57 %)  
Bases Filtered by continuous base "N": 839009 (0.56 %)  
Reads Filtered by low complexity ratio (0.8): 4 (0.00 %)  
Bases Filtered by low complexity ratio: 1004 (0.00 %)  
--More-- (67%)
```

W5-3: 結果の概要

```
File Edit View Search Terminal Help 19:11  
Before Trimming  
Reads #: 600000  
Total bases: 150600000  
Reads Length: 251.00  
  
After Trimming  
Reads #: 596558 (99.43 %)  
Total bases: 145731275 (96.77 %)  
Mean Reads Length: 244.29  
Paired Reads #: 595266 (99.78 %)  
Paired total bases: 145416561 (99.78 %)  
Unpaired Reads #: 1292 (0.22 %)  
Unpaired total bases: 314714 (0.22 %)  
  
Discarded reads #: 3442 (0.57 %)  
Trimmed bases: 4868725 (3.23 %)  
Reads Filtered by length cutoff (50 bp): 42 (0.01 %)  
Bases Filtered by length cutoff: 1314 (0.00 %)  
Reads Filtered by continuous base "N" (2): 3396 (0.57 %)  
Bases Filtered by continuous base "N": 839009 (0.56 %)  
Reads Filtered by low complexity ratio (0.8): 4 (0.00 %)  
Bases Filtered by low complexity ratio: 1004 (0.00 %)  
--More-- (67%)
```

1

W5-3: 結果の概要

```
X - File Edit View Search Terminal Help  
Before Trimming  
Reads #: 600000  
Total bases: 150600000  
Reads Length: 251.00  
  
After Trimming  
① Reads #: 596558 (99.43 %)  
Total bases: 145731275 (96.77 %)  
Mean Reads Length: 244.29  
Paired Reads #: 595266 (99.78 %) ②  
Paired total bases: 145416561 (99.78 %)  
Unpaired Reads #: 1292 (0.22 %)  
Unpaired total bases: 314714 (0.22 %)  
  
Discarded reads #: 3442 (0.57 %)  
Trimmed bases: 4868725 (3.23 %)  
Reads Filtered by length cutoff (50 bp): 42 (0.01 %)  
Bases Filtered by length cutoff: 1314 (0.00 %)  
Reads Filtered by continuous base "N" (2): 3396 (0.57 %)  
Bases Filtered by continuous base "N": 839009 (0.56 %)  
Reads Filtered by low complexity ratio (0.8): 4 (0.00 %)  
Bases Filtered by low complexity ratio: 1004 (0.00 %)  
--More-- (67%)
```

①トリム後に596,558リード残ったことがわかる。
FaQCsはforward側とreverse側を独立に実行したのち、両方で生き残ったものを②Paired Reads (595,266個)、片側のみで生き残ったものを③Unpaired Reads (1,292個)としてレポートしている

W5-3: 結果の概要

```
File Edit View Search Terminal Help ↑ ↓ Ja 19:11 🎧
Before Trimming
Reads #: 600000
Total bases: 150600000
Reads Length: 251.00

After Trimming
Reads #: 596558 (99.43 %)
Total bases: 145731275 (96.77 %)
Mean Reads Length: 244.29
    Paired Reads #: 595266 (99.78 %)
    Paired total bases: 145416561 (99.78 %)
    Unpaired Reads #: 1292 (0.22 %)
    Unpaired total bases: 314714 (0.22 %)

Discarded reads #: 3442 (0.57 %)
Trimmed bases: 4868725 (3.23 %)
    Reads Filtered by length cutoff (50 bp): 42 (0.01 %)
    Bases Filtered by length cutoff: 1314 (0.00 %)
    Reads Filtered by continuous base "N" (2): 3396 (0.57 %)
    Bases Filtered by continuous base "N": 839009 (0.56 %)
    Reads Filtered by low complexity ratio (0.8): 4 (0.00 %)
    Bases Filtered by low complexity ratio: 1004 (0.00 %)
--More-- (67%)
```

①

W5-3: 結果の概要

①特に言及しなかったQC.unpaired.trimmed.fastq
ファイルの中身は想像がつく。②なぜ行数が5,168
になるのかまでは、使わないので深追いしない

```
File Edit View Search Terminal Help [ 19:28 ]  
iu@bielinux[result] pwd [ 7:27 午後 ]  
/home/iu/Documents/DRR024501/result  
iu@bielinux[result] ls -l [ 7:27 午後 ]  
total 371472  
-rw-rw-r-- 1 iu iu 94 12月 29 18:13 fastqCount.txt  
-rw-rw-r-- 1 iu iu 189492182 12月 29 18:24 QC.1.trimmed.fastq  
-rw-rw-r-- 1 iu iu 189628416 12月 29 18:24 QC.2.trimmed.fastq  
-rw-rw-r-- 1 iu iu 424499 12月 29 18:24 QC_qc_report.pdf  
-rw-rw-r-- 1 iu iu 1073 12月 29 18:24 QC.stats.txt  
-rw-rw-r-- 1 iu iu 821270 12月 29 18:24 QC.unpaired.trimmed.fastq ①  
iu@bielinux[result] wc QC.unpaired.trimmed.fastq [ 7:27 午後 ]  
5168 10336 821270 QC.unpaired.trimmed.fastq  
iu@bielinux[result] [ 7:28 午後 ]
```

W5-4 : gzip圧縮

```
File Edit View Search Terminal Help [ 7:38 午後 ]
iu@bielinux[result] pwd
/home/iu/Documents/DRR024501/result
iu@bielinux[result] ls -l [ 7:39 午後 ]
total 371472
-rw-rw-r-- 1 iu iu      94 12月 29 18:13 fastqCount.txt
-rw-rw-r-- 1 iu iu 189492182 12月 29 18:24 QC.1.trimmed.fastq
-rw-rw-r-- 1 iu iu 189628416 12月 29 18:24 QC.2.trimmed.fastq
-rw-rw-r-- 1 iu iu    424499 12月 29 18:24 QC_qc_report.pdf
-rw-rw-r-- 1 iu iu      1073 12月 29 18:24 QC.stats.txt
-rw-rw-r-- 1 iu iu   821270 12月 29 18:24 QC.unpaired.trimmed.fastq
iu@bielinux[result] gzip QC.1.trimmed.fastq [ 7:39 午後 ]
iu@bielinux[result] gzip QC.2.trimmed.fastq [ 7:39 午後 ]
iu@bielinux[result] ls -l *.gz [ 7:39 午後 ]
-rw-rw-r-- 1 iu iu 57061392 12月 29 18:24 QC.1.trimmed.fastq.gz
-rw-rw-r-- 1 iu iu 62989289 12月 29 18:24 QC.2.trimmed.fastq.gz
iu@bielinux[result] [ 7:39 午後 ]
```



Contents

- Illumina HiSeqデータ(トランスクリプトーム)の乳酸菌ゲノムへのマッピング
 - W14: QuasRパッケージを用いたマッピングの事前準備と本番
 - W15: 結果の解説、forward側の100–107bp付近に問題があることを特定
- トリミング、*de novo*トランスクリプトームアセンブリとマッピングの再実行
 - W16: 問題のある領域(forward側の100–107bp)のトリミング
 - W17: トリム後のデータでアセンブリを再実行(Rockhopper2; クラスパス設定関連Tips含む)
 - W18: トリム後のデータでマッピングを再実行(QuasR)
 - W19: トリム前のデータでクオリティチェックを再実行(FastQC)
- Illumina MiSeqデータ(乳酸菌ゲノム)の特徴と前処理
 - W4: FastQC、W5: FaQCs、W6: 再度FastQC
- *de novo*ゲノムアセンブリ
 - W7: Bio-LinuxにプレインストールされているVelvet (ver. 1.2.09)を上限のk=31で実行
 - W8: k=31のアセンブリ結果をRで確認。k=141で実行し、k=31の結果と同じになるのを確認
 - W9: Velvet (ver. 1.2.10)のインストール
 - W10: Velvet (ver. 1.2.10)の実行

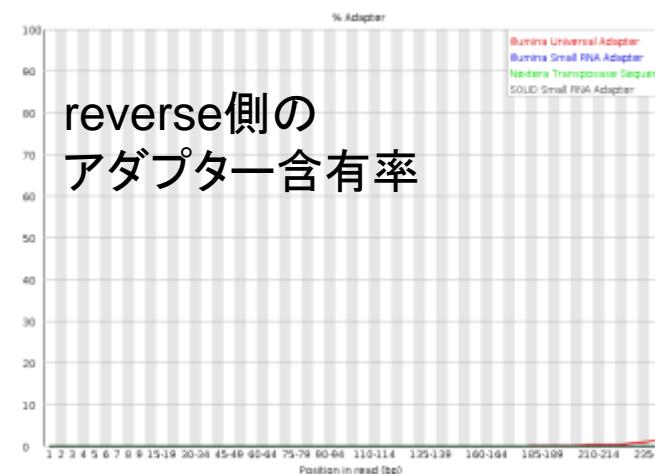
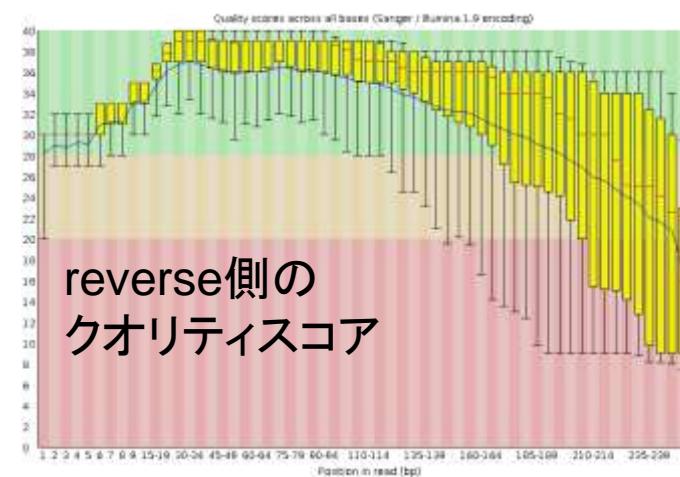
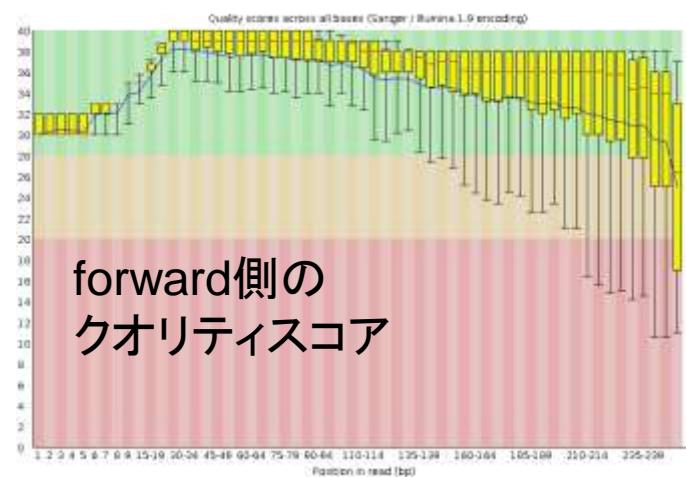


W6-1：再度FastQC

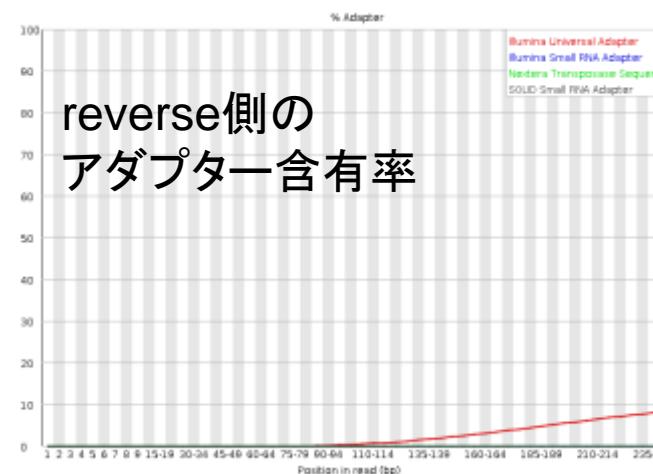
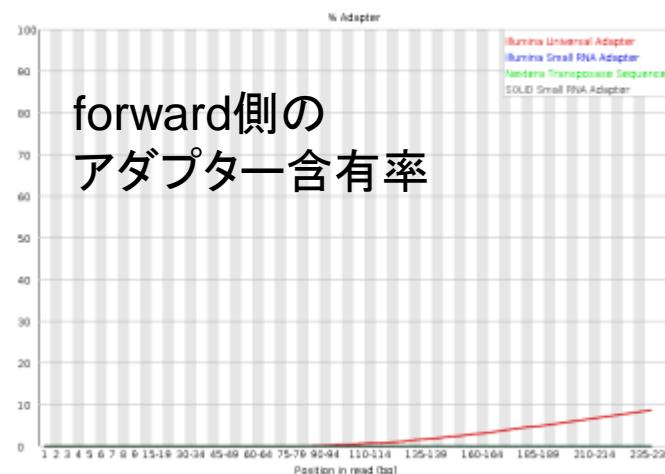
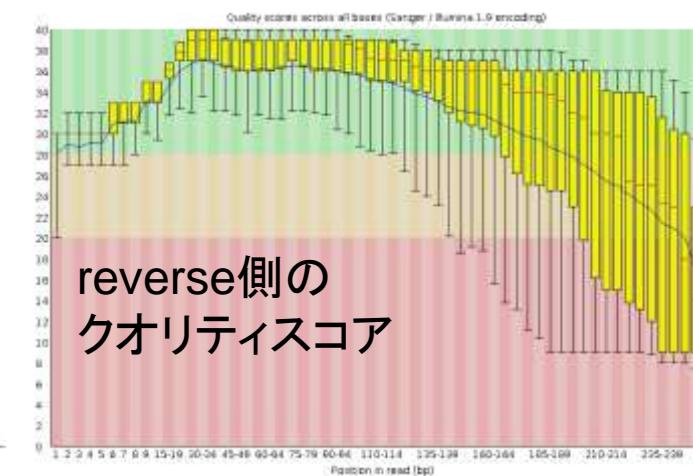
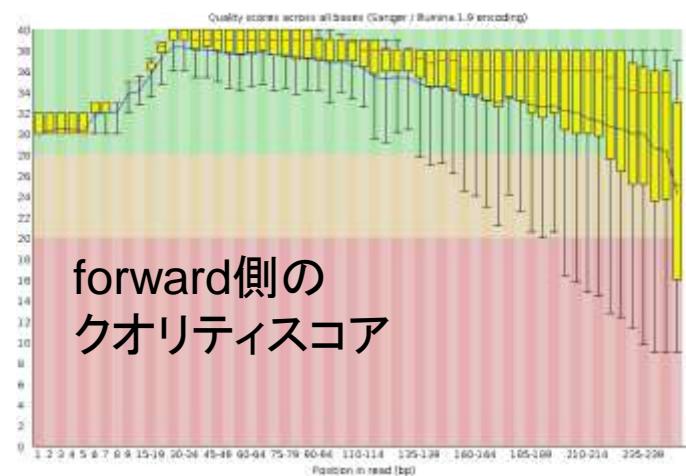
```
File Edit View Search Terminal Help  
iu@bielinux[result] pwd [ 7:49 午後 ]  
/home/iu/Documents/DRR024501/result  
iu@bielinux[result] ls -l *.gz [ 7:49 午後 ]  
-rw-rw-r-- 1 iu iu 57061392 12月 29 18:24 QC.1.trimmed.fastq.gz  
-rw-rw-r-- 1 iu iu 62989289 12月 29 18:24 QC.2.trimmed.fastq.gz  
iu@bielinux[result] fastqc2 -q QC.1.trimmed.fastq.gz --outdir=/home/iu/Desktop/mac_share [ 7:49 午後 ]  
iu@bielinux[result] fastqc2 -q QC.2.trimmed.fastq.gz --outdir=/home/iu/Desktop/mac_share [ 7:49 午後 ]  
iu@bielinux[result] ls -l ~/Desktop/mac_share [ 7:50 午後 ]  
total 2523  
-rwxrwxrwx 1 iu iu 302848 12月 29 18:12 DRR024501sub_1_fastqc.html  
-rwxrwxrwx 1 iu iu 330078 12月 29 18:12 DRR024501sub_1_fastqc.zip  
-rwxrwxrwx 1 iu iu 312853 12月 29 18:12 DRR024501sub_2_fastqc.html  
-rwxrwxrwx 1 iu iu 349353 12月 29 18:12 DRR024501sub_2_fastqc.zip  
-rwxrwxrwx 1 iu iu 311960 12月 29 19:50 QC.1.trimmed_fastqc.html  
-rwxrwxrwx 1 iu iu 338669 12月 29 19:50 QC.1.trimmed_fastqc.zip  
-rwxrwxrwx 1 iu iu 305072 12月 29 19:50 QC.2.trimmed_fastqc.html  
-rwxrwxrwx 1 iu iu 330187 12月 29 19:50 QC.2.trimmed_fastqc.zip  
iu@bielinux[result] date [ 7:50 午後 ]  
2015年 12月 29日 火曜日 19:50:14 JST  
iu@bielinux[result] [ 7:50 午後 ]
```

①FaQCs実行結果のgzip圧縮ファイルを入力として、FastQC (ver. 0.11.4)を実行し、結果を共有フォルダに保存。約1分。目的は、アダプターがちゃんと消えているかどうかの確認

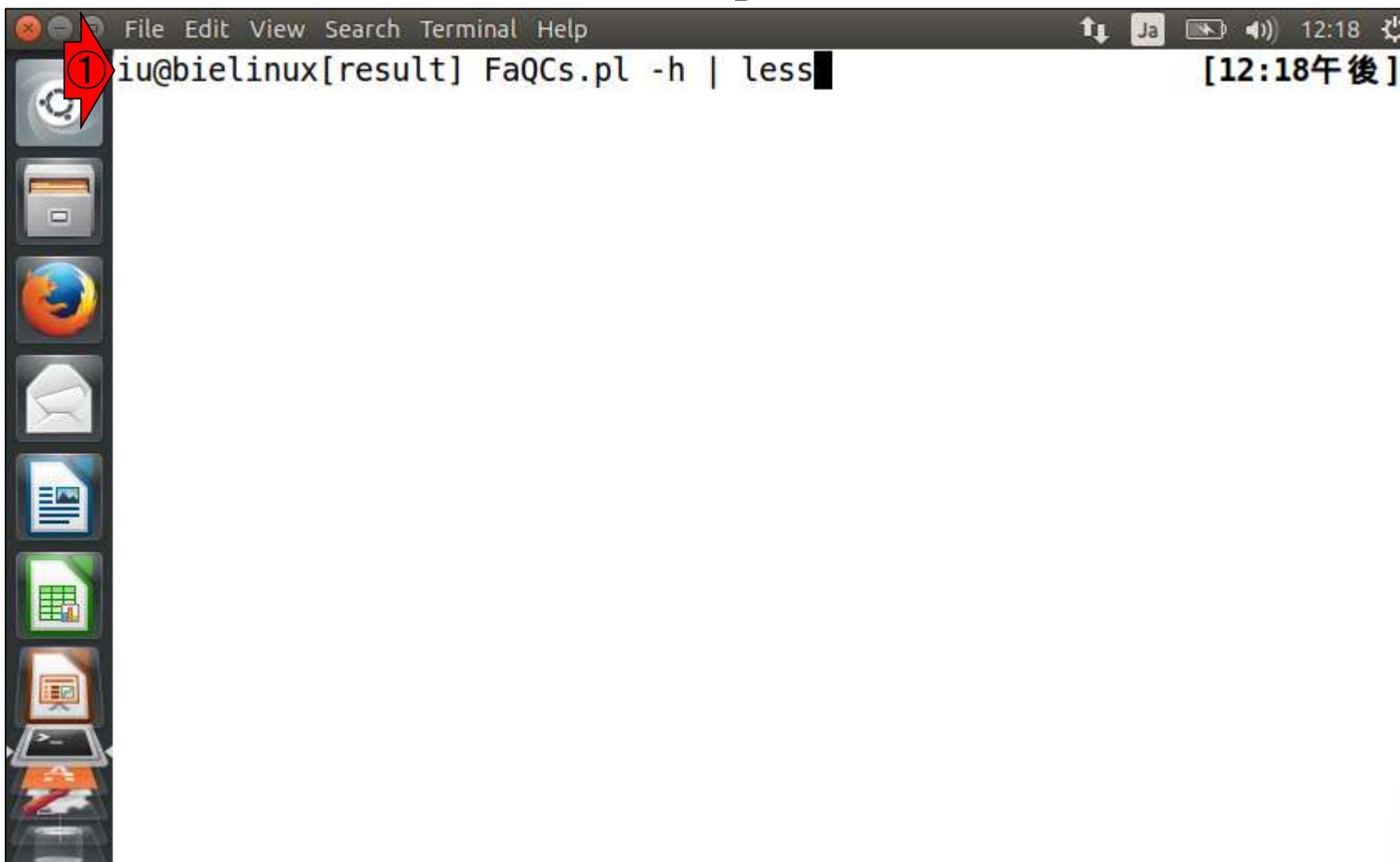
W6-2: FastQCまとめ(FaQCs実行後)



W4-4: FastQCまとめ(FaQCs実行前)



W6-3: FaQCsのオプション



File Edit View Terminal Help
iu@bielinux[result] FaQCs.pl -h | less [12:18午後]

例えば①-qオプションのデフォルトが5なので、10にするとトリムされる塩基数が増えるのだろうとか…

W6-3: FaQCsのオプション

```
File Edit View Search Terminal Help Ja 12:21
Usage: perl /usr/local/bin/FaQCs.pl [options] [-u unpaired.fastq]
-p reads1.fastq reads2.fastq -d out_directory
Version 1.34
Input File: (can use more than once)
  -u          <Files> Unpaired reads

  -p          <Files> Paired reads in two files and separate
e by space
Trim:
  -mode       "HARD" or "BWA" or "BWA_plus" (default BWA_pl
us)
               BWA trim is NOT A HARD cutoff! (see bwa's bwa
_trim_read() function in bwaseqio.c)

  -q          <INT> Targets # as quality level (default 5)
for trimming
               1
  -5end      <INT> Cut # bp from 5 end before quality trim
ming/filtering
  :
```

①

W6-3: FaQCsのオプション

①-avg_qオプションのデフォルトが0なので、20にすると平均クオリティスコアが20未満の(以下かも)リードが除去されるのだろうとか…いろいろと妄想する

```
File Edit View Search Terminal Help
-x -y -z -3end      <INT> Cut # bp from 3 end before
ming/filtering

-adapter      <bool> Trim reads with illumina adapter/primer
rs (default: no)
              -rate   <FLOAT> Mismatch ratio of adapters' l
ength (default: 0.2, allow 20% mismatches)

-artifactFile <File>    additional artifact (adapters/primer
ers/contaminations) reference file in fasta format
Filters:
-min_L        <INT> Trimmed read should have to be at least
this minimum length (default:50)

-filtering)   -avg_q       <NUM> Average quality cutoff (default:0, no f
iltration)
              -n          <INT> Trimmed read has more than this number
of continuous base "N" will be discarded.
              (default: 2, "NN")
:
```

1

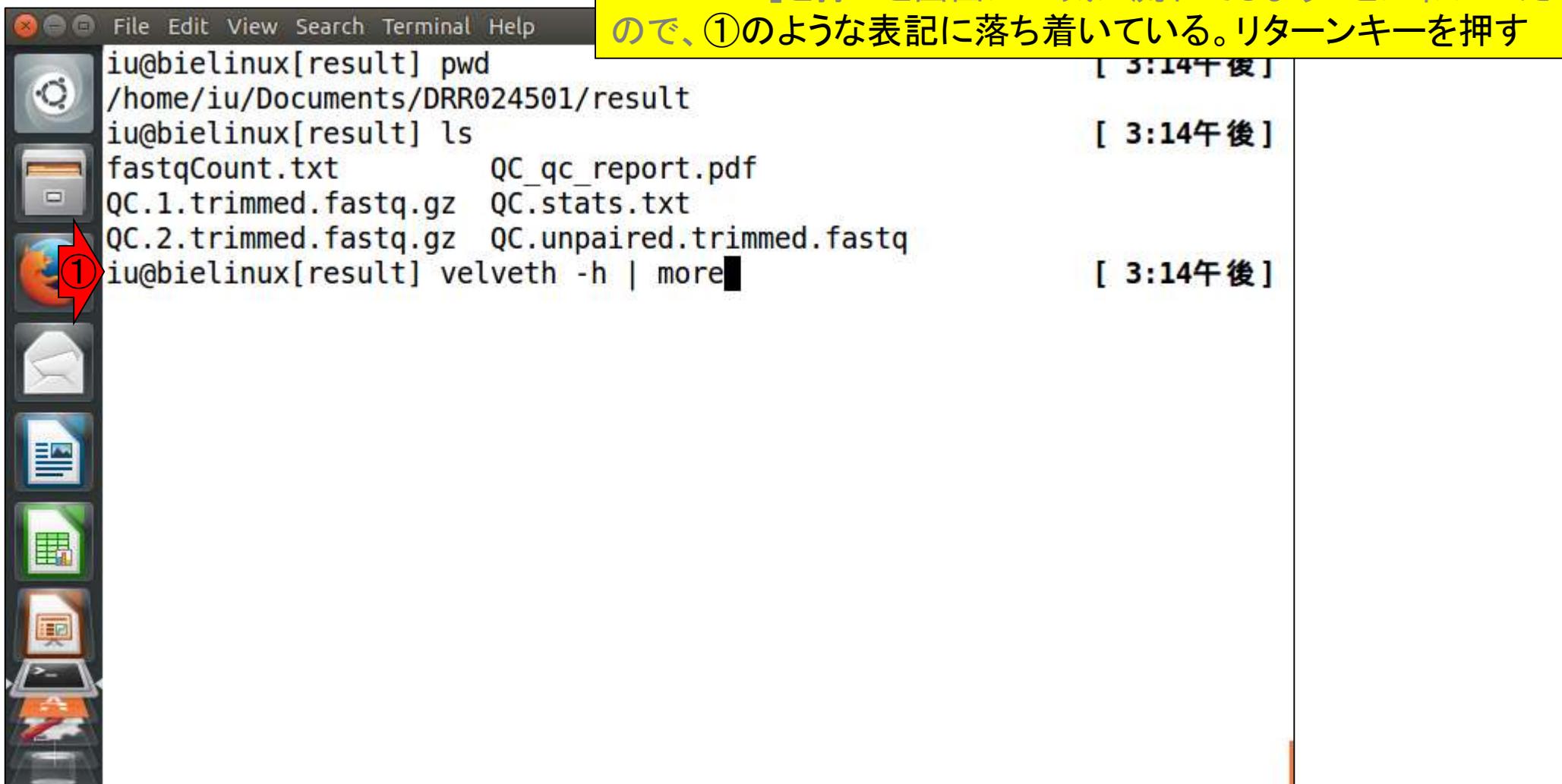
Contents

- Illumina HiSeqデータ(トランスクリプトーム)の乳酸菌ゲノムへのマッピング
 - W14:QuasRパッケージを用いたマッピングの事前準備と本番
 - W15:結果の解説、forward側の100–107bp付近に問題があることを特定
- トリミング、*de novo*トランスクリプトームアセンブリとマッピングの再実行
 - W16:問題のある領域(forward側の100–107bp)のトリミング
 - W17:トリム後のデータでアセンブリを再実行(Rockhopper2; クラスパス設定関連Tips含む)
 - W18:トリム後のデータでマッピングを再実行(QuasR)
 - W19:トリム前のデータでクオリティチェックを再実行(FastQC)
- Illumina MiSeqデータ(乳酸菌ゲノム)の特徴と前処理
 - W4:FastQC、W5:FaQCs、W6:再度FastQC
- *de novo*ゲノムアセンブリ
 - W7:Bio-LinuxにプレインストールされているVelvet (ver. 1.2.09)を上限のk=31で実行
 - W8:k=31のアセンブリ結果をRで確認。k=141で実行し、k=31の結果と同じになるのを確認
 - W9:Velvet (ver. 1.2.10)のインストール
 - W10:Velvet (ver. 1.2.10)の実行



W7-1 : Velvet

```
iu@bielinux[result] pwd [ 3:14 午後]
/home/iu/Documents/DRR024501/result
iu@bielinux[result] ls [ 3:14 午後]
fastqCount.txt      QC_qc_report.pdf
QC.1.trimmed.fastq.gz  QC.stats.txt
QC.2.trimmed.fastq.gz  QC.unpaired.trimmed.fastq
iu@bielinux[result] velveth -h | more [ 3:14 午後]
```



①いきなり「velveth -h | more」と書いてヘルプを表示させているが、「velvet -h」と打つとvelvethに修正を試みること、「velveth -h」と打つと画面が一気に流れてしまうことがわかったので、①のような表記に落ち着いている。リターンキーを押す

W7-2: Usage

```
File Edit View Search Terminal Help  
velveth - simple hashing program  
Version 1.2.09  
Copyright 2007, 2008 Daniel Zerbino (zerbino@ebi.ac.uk)  
This is free software; see the source for copying conditions. There is  
NO  
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  
Compilation settings:  
CATEGORIES = 2  
MAXKMERLENGTH = 31  
Usage:  
. ./velveth directory hash_length {[-file_format][-read_type][-separate|-  
interleaved] filename1 [filename2 ...]} {...} [options]  
directory : directory name for output files  
hash_length : EITHER an odd integer (if even, it will be de-  
cremented) <= 31 (if above, will be reduced)  
- -More --
```

W7-3:k値指定

The terminal window shows the following output:

```
File Edit View Search Terminal Help
velveth - simple hashing program
Version 1.2.09

Copyright 2007, 2008 Daniel Zerbino (zerbino@ebi.ac.uk)
This is free software; see the source for copying conditions. There is
NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Compilation settings:
CATEGORIES = 2
MAXKMERLENGTH = 31

Usage:
./velveth directory hash_length {[-file_format][-read_type][-separate|-
interleaved] filename1 [filename2 ...]} {...} [options]

    directory      : directory name for output files
    hash_length    : EITHER an odd integer (if even, it will be de-
cremented) <= 31 (if above, will be reduced)
--More--
```

Annotations with red numbers:

- ① A red arrow points to the `hash_length` parameter in the usage section.
- ② A red box highlights the `hash_length` parameter in the detailed description, and a red arrow points to the explanatory text: "EITHER an odd integer (if even, it will be decremented) <= 31 (if above, will be reduced)".

①赤下線の`hash_length`のところで指定する数値がアセンブリ時の主要なオプションであるk-merのk値。第5回でも述べたように、②通常は奇数(an odd integer)が指定される。そしてこのプログラムの場合は、31以上の値が指定されても31にされてしまうようだと読み解く

W7-3:k値指定

②

スペースキーを1回押して、次のページに移動。①
まだhash_lengthの説明部分で、「either A or B」の
②ORに相当する部分。W10-2で利用しているが、あ
まり有難みを感じないので講習会ではやりません

: OR: m,M,s where m and M are odd integers (11
not, they will be decremented) with m < M <= 31 (if above, will be reduced)

①

and s is a step (even number). Velvet will then hash from k=m to k=M with a step of s
filename : path to sequence file or - for standard input

File format options:

-fasta -fastq -raw -fasta.gz -fastq.gz -raw.gz
-sam -bam -fmtAuto

(Note: -fmtAuto will detect fasta or fastq, and will try the following programs for decompression : gunzip, pbunzip2, bunzip2

File layout options for paired reads (only for fasta and fastq formats)

:

-interleaved : File contains paired reads interleaved in the one file (default)
-separate : Read 2 separate files for paired reads

Read type options:

--More--

de novoアセンブリ

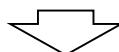
省略予定

*de novo*アセンブリとは、リードの塩基配列情報のみを頼りに、元のリード長よりも長い配列(コンティグ)を出力する作業。この例の場合、赤下線が一致部分。出力は、元のリード長よりも2 bp長いコンティグとなる

入力: NGSリードファイル(FASTA/FASTQ)

リード1: CACCAGGACATGAAGACGCG

リード2: CCAGGACATGAAGACGCGTT



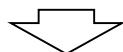
出力: コンティグ(FASTA/FASTQ)

CACCAGGACATGAAGACGCGTT

de novoアセンブリ

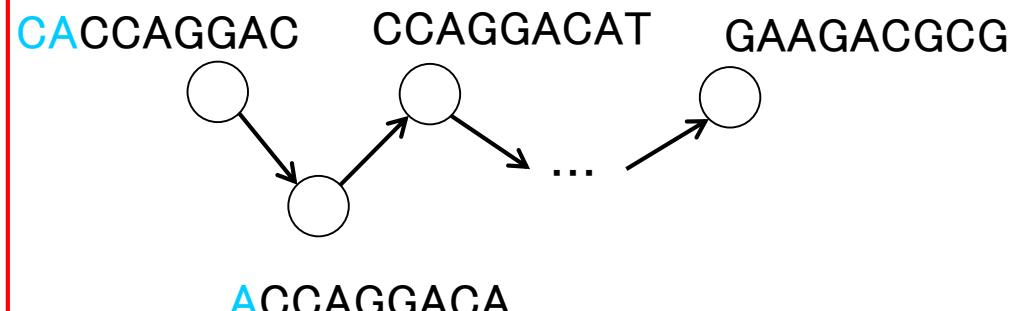
省略予定

リード1: CA
CCAGGACATGAAGACGCG



CA
CCAGGAC
ACCAGGACA
CCAGGACAT
...

ATGAAGACG
TGAAGACGC
GAAGACGCG



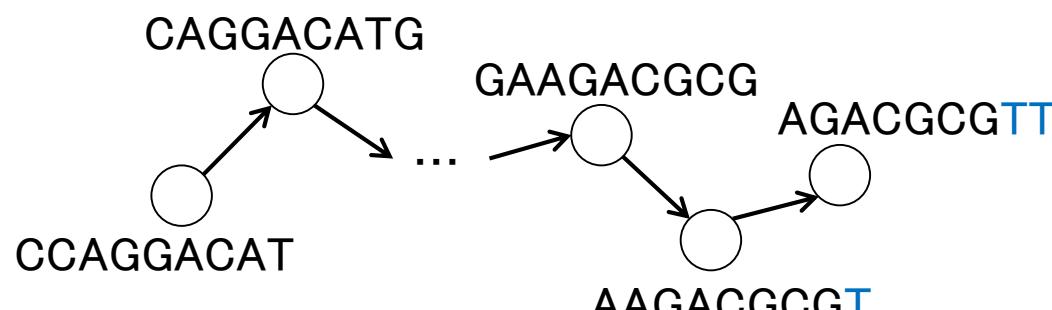
k-merアプローチ(de Bruijnグラフ)の説明。リードを全ての可能なk-merに分割し、①有向グラフを作成(これはk=9の例)。矢印で向きが決まっているので有向(directed)といいます。私はこっち方面のヒトではないので多少ミスがあるかもしれませんのが大筋では大丈夫

リード2: CCAGGACATGAAGACGCGTT



CCAGGACAT
CAGGACATG
AGGACATGA
...

GAAGACGCG
AAGACGCGT
AGACGCGTT

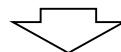


①

de novoアセンブリ

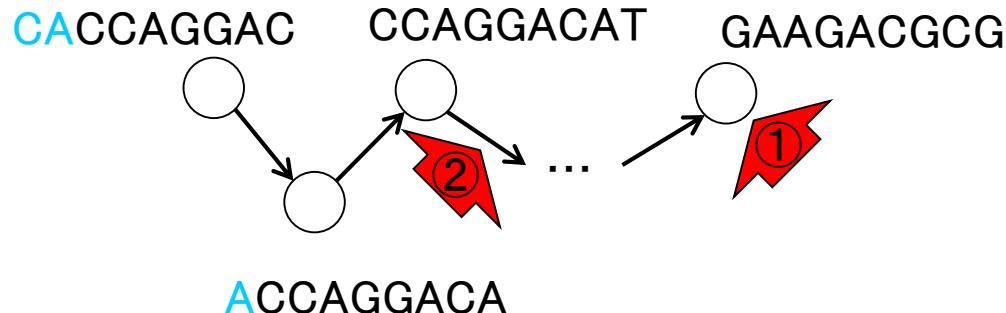
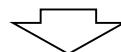
省略予定

リード1: CA₁CCAGGACATGAAGACGCG

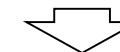


CA₁CCAGGAC
A₂CCAGGACA
CCAGGACAT
...

ATGAAGACG
TGAAGACGC
GAAGACGCG

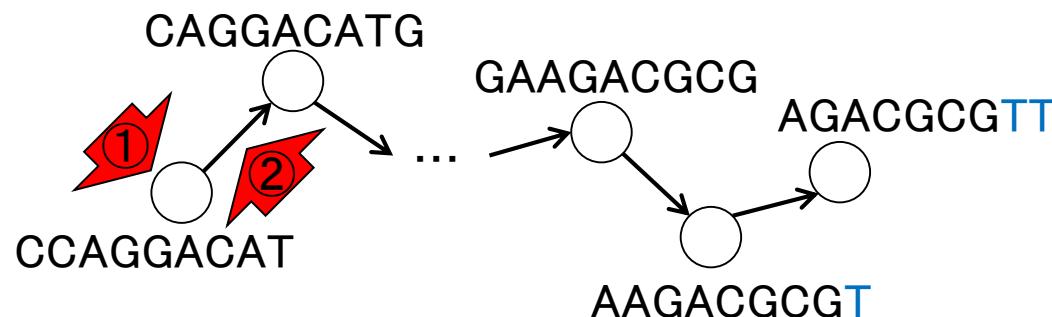
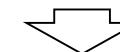


リード2: CCAGGACATGAAGACGCGT₁T



CCAGGACAT
CAGGACATG
AGGACATGA
...

GAAGACGCG
AAGACGCGT₂
AGACGCGT₁T



de novoアセンブリ

省略予定

①赤枠のものが(異なるリード間で共有されている)同一ノードの例

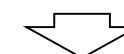
リード1: CA CCAGGACAT GAAGACGCG



CCAGGAC
ACCAGGACA
CCAGGACAT
...

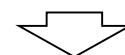
ATGAAGACG
TGAAGACGC
GAAGACGCG

リード2: CCAGGACAT GAAGACGCG TT



CAGGACAT
AGGACATGA
...

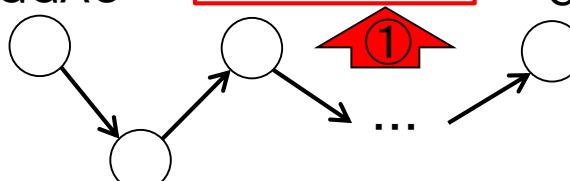
GAAGACGCG
AAGACGCGT
AGACGCGTT



CCAGGACAT



GAAGACGCG



ACCAGGACA

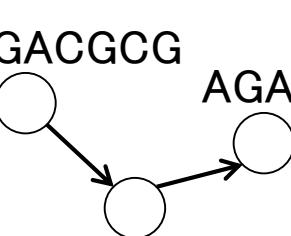
CAGGACATG

CCAGGACAT



GAAGACGCG

AGACGCGTT



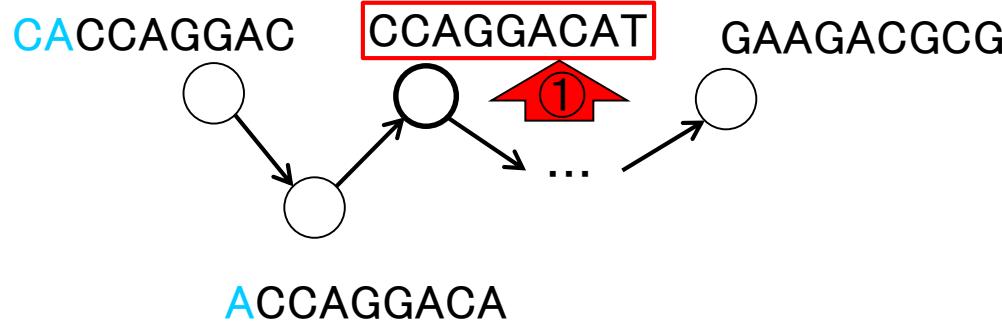
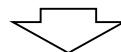
AAGACGCGT

de novoアセンブリ

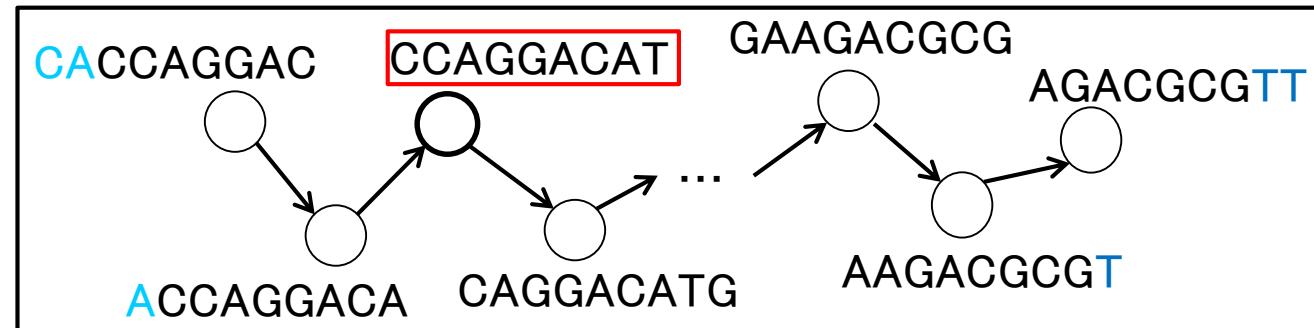
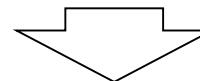
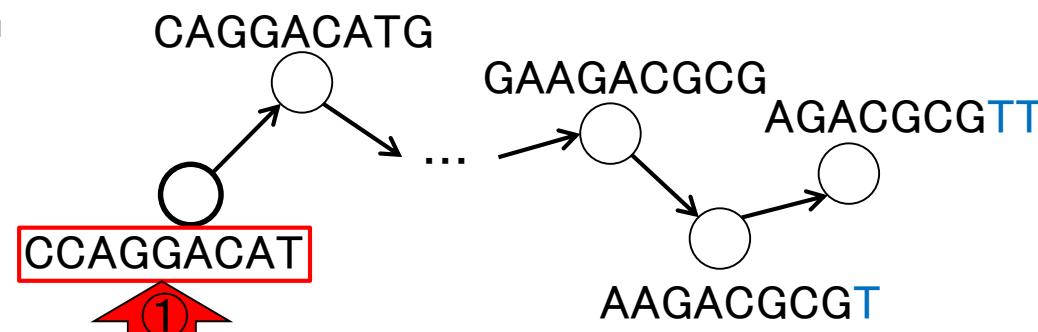
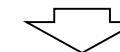
省略予定

②同一ノードをマージしたグラフを作成。これが「k-merグラフ」とか「de Bruijnグラフ」と言われるものです。ここまでがグラフ構築(graph construction)作業

リード1: CA**CCAGGACAT**GAAGACGCG



リード2: CCAGGACAT**GAAGACGCGTT**



de novoアセンブリ

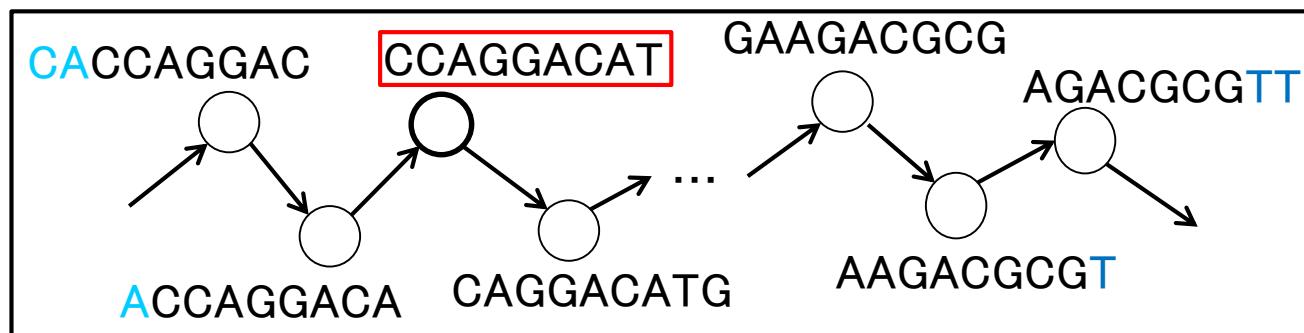
省略予定

③グラフ簡易化(graph simplification)作業のイメージ。実際には、大量のリードから複雑なde Bruijnグラフが作成されるのでできるだけシンプルにする必要がある。実際に行うのは、ここで示されているような「連続したノード(頂点)」や、次スライドで示す「バブル構造」のマージ

リード1: CA[CCAGGACAT]GAAGACGCG



リード2: CCAGGACATGAAGACGCGTT



→ CA[CCAGGACAT]GAAGACGCGTT →

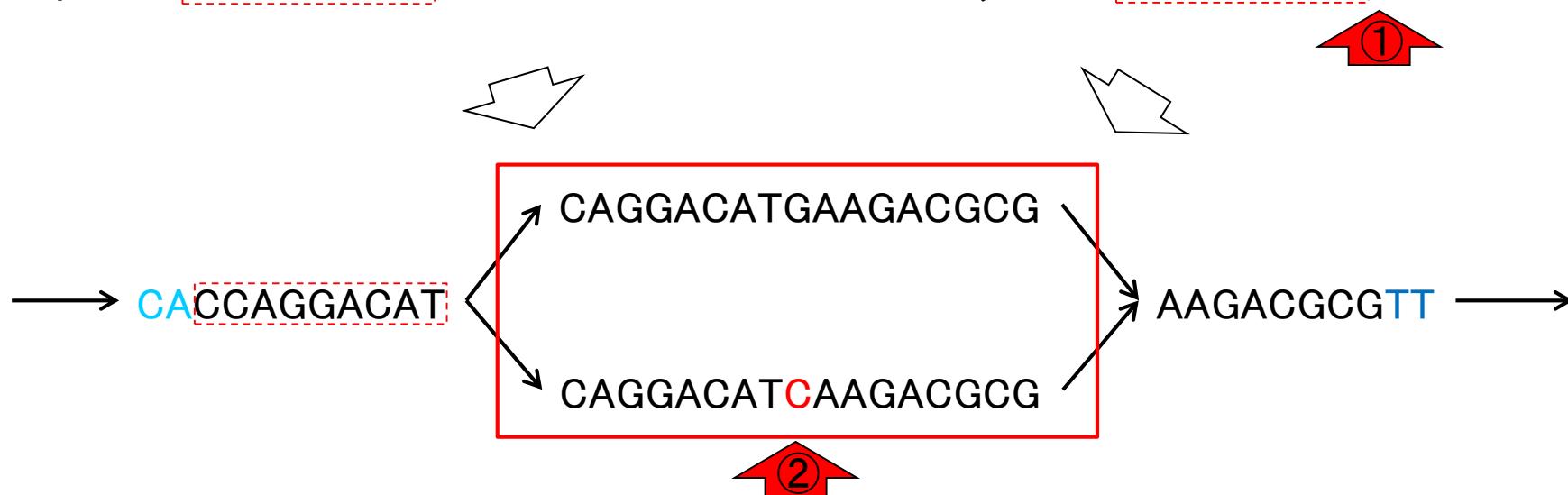
de novoアセンブリ

省略予定

バブル(泡)構造の例。①シークエンスエラーやSNPなど、塩基に違いがあれば②バブル構造になります。エラーが沢山あると、バブルだらけになってグラフ構築や簡易化作業が困難になるであろうことは容易に想像がつきます。クオリティフィルタリングの重要性も分かるでしょう。このあたりは、2014.06.25の講義資料をベースに作成

リード1: CA CCAGGACAT GAAGACGCG

リード2: CCAGGACAT CAAGACGCG TT



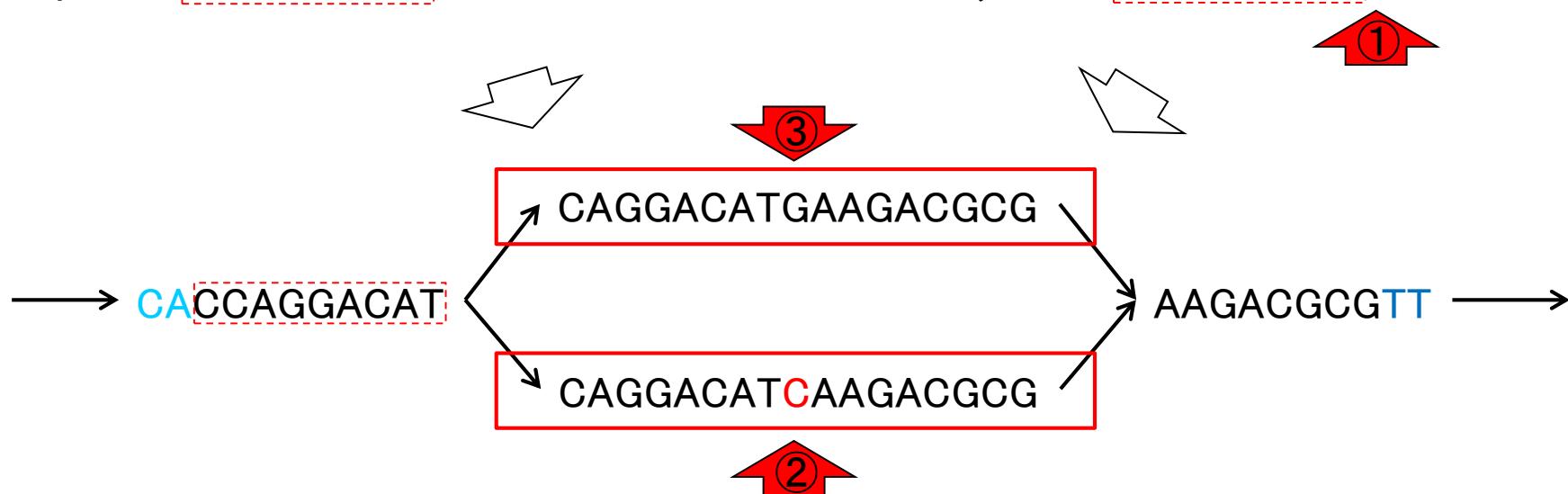
de novoアセンブリ

省略予定

通常、②のようなシークエンスエラー由来の部分配列の出現回数は、③のような本物のサンプル由来のものに比べて低い、といった事柄(多数決ルール)をおそらく内部的に利用しているはず

リード1: CA**CCAGGACAT**GAAGACGCG

リード2: CCAGGACAT**CAAGACGCGTT**

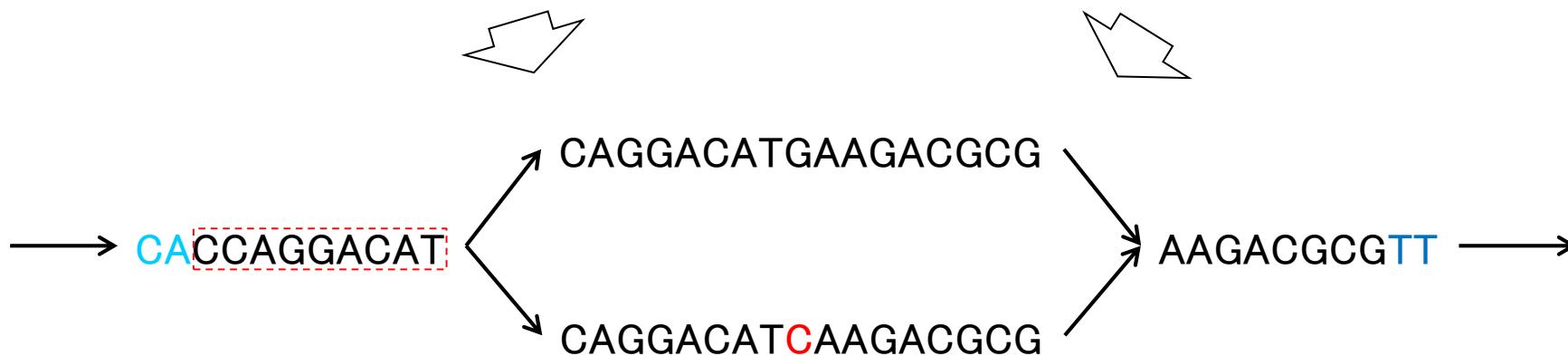


de novoアセンブリ

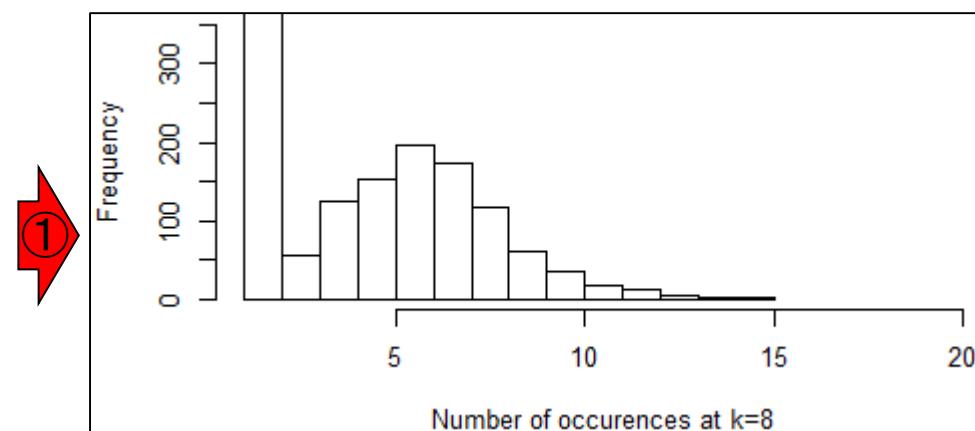
省略予定

①は、k-mer出現頻度分布(2016.07.20の統計解析の最後のスライド)です

リード1: CA[CCAGGACAT]GAAGACGCG



リード2: CCAGGACAT[CAAGACGCGTT]



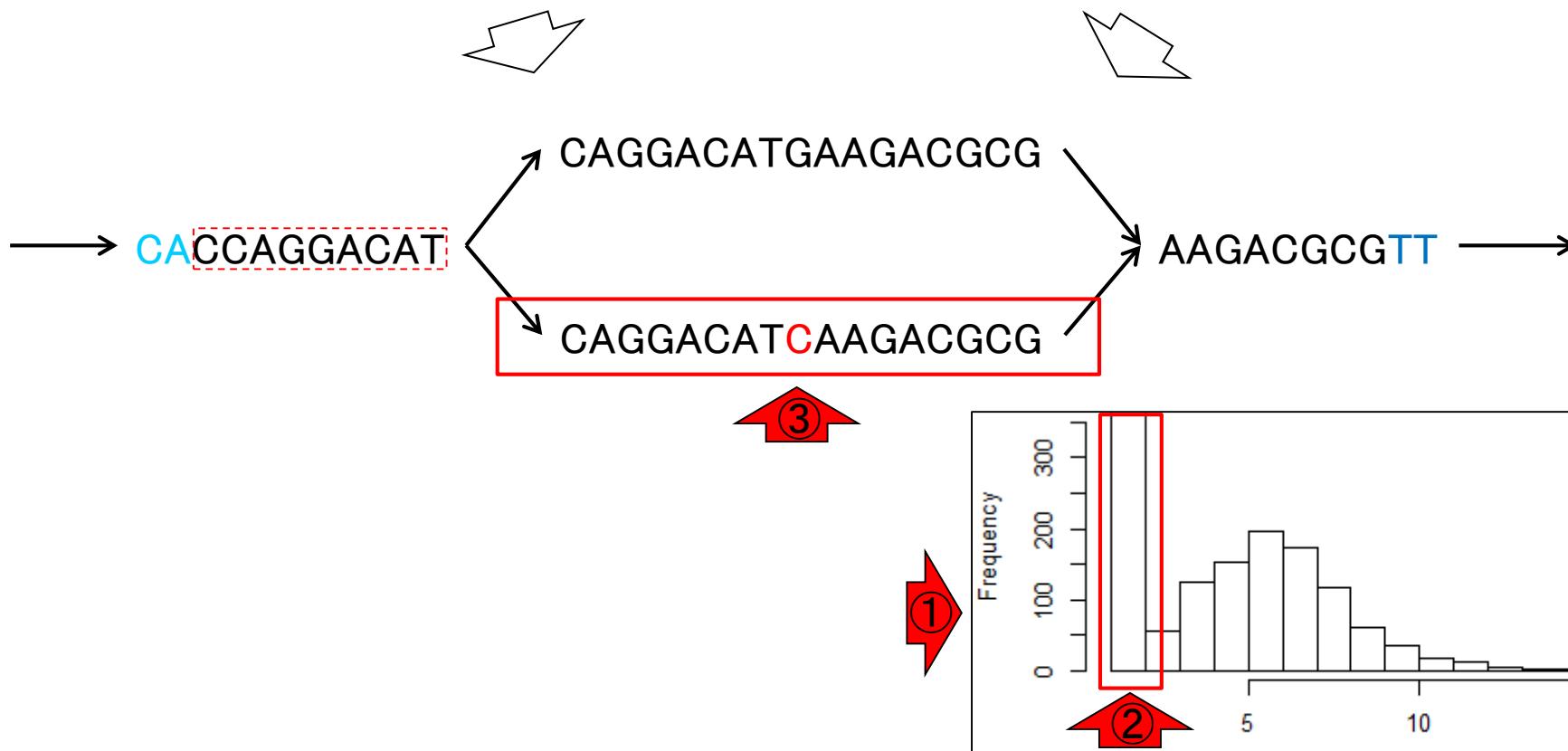
de novoアセンブリ

省略予定

②シークエンスエラー由来k-merをフィルタリングしておくことで、バブル構造中の③消されるべきパスがより明瞭になることでしょう



リード1: CA[CCAGGACAT]GAAGACGCG



de novoアセンブリ

省略予定

父親由来ゲノム

リード1: CA**CCAGGACAT**GAAGACGCG



ヘテロ接合度の高い2倍体ゲノム(heterozygous diploid genomes)の場合は、出現割合も同程度なのでバブル構造が解けにくい(どちらのパスを残すべきか決めづらい)のですが…

母親由来ゲノム

リード2: CCAGGACAT**CAAGACGCGTT**

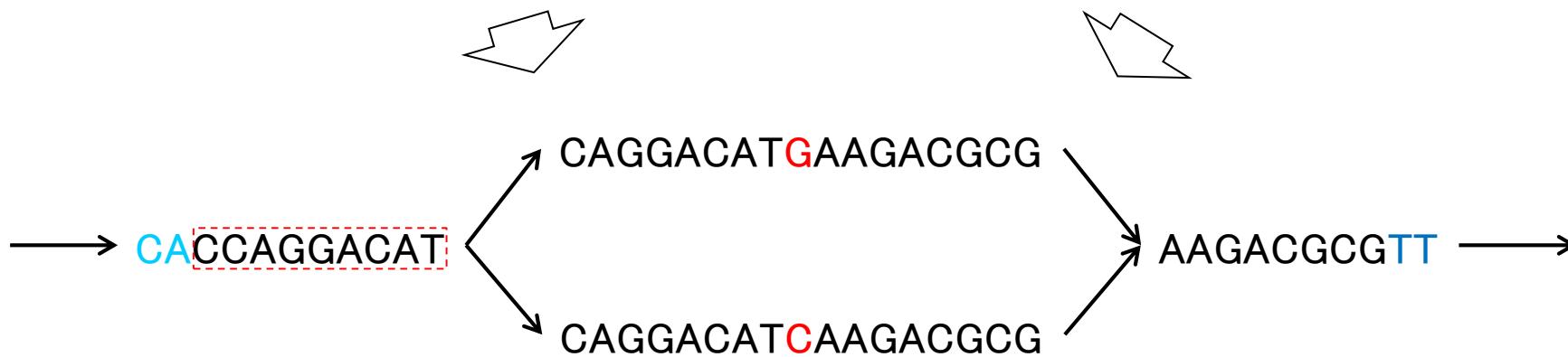
父親由来ゲノム
...CA**CCAGGACAT**GAAGACGCGTTCA...
...CA**CCAGGACAT**CAAGACGCGTTCA...
母親由来ゲノム

de novoアセンブリ

省略予定

父親由来ゲノム

リード1: CA**CCAGGACAT**GAAGACGCG

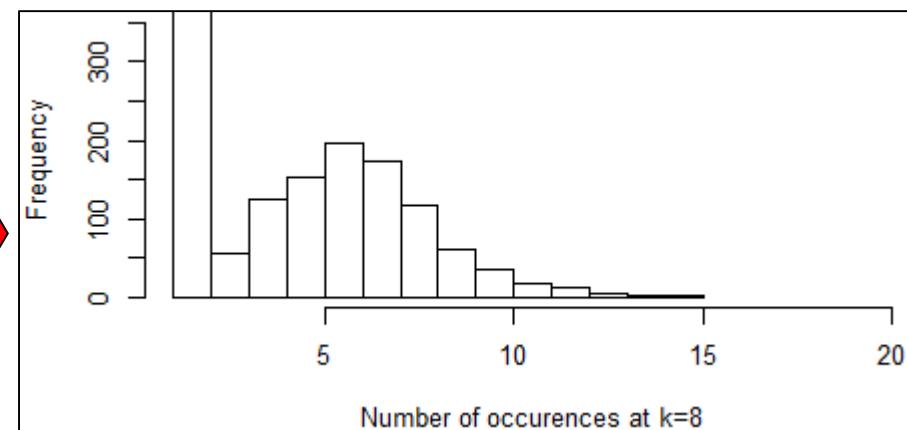


ヘテロ接合性(heterozygosity)に関する知見も①のk-mer出現頻度分布を眺めることでわかります



母親由来ゲノム

リード2: CCAGGACAT**CAAGACGCGTT**



de novoアセンブリ

省略予定

父親由来ゲノム

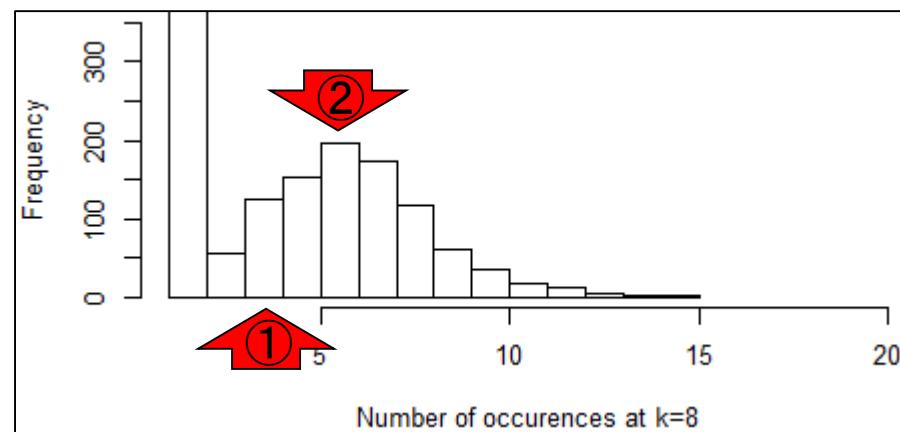
リード1: CA**CCAGGACAT**GAAGACGCG



両親間で①配列の異なる領域を含むk-merの出現頻度は、②配列が同じ領域由来k-merの出現頻度(c)の $1/2$ となる。この例の場合は、②の分布のみだが、もしheterozygosityがあれば、①の付近にピークのある分布も見られるのです

母親由来ゲノム

リード2: CCAGGACAT**CAAGACGCG**TT



de novoアセンブリ

ヘテロ接合度の高いゲノムもうまく取り扱えるde Bruijnグラフに基づくアセンブラーがPlatanus。DDBJ Pipelineにも実装されており、2016.08.03に利用します

Genome Res. 2014 Aug;24(8):1384-95. doi: 10.1101/gr.170720.113. Epub 2014 Apr 22.

省略予定

Efficient de novo assembly of highly heterozygous genomes from whole-genome shotgun short reads.

Kajitani R¹, Toshimoto K², Nozuchi H³, Toyoda A⁴, Ogura Y⁵, Okuno M¹, Yabana M¹, Harada M¹, Nagayasu E⁶, Maruyama H⁶, Kohara Y⁷, Fujiyama A⁴, Hayashi T⁵, Itoh T¹.

⊕ Author information

Abstract

Although many de novo genome assembly projects have recently been conducted using high-throughput sequencers, assembling highly heterozygous diploid genomes is a substantial challenge due to the increased complexity of the de Bruijn graph structure predominantly used. To address the increasing demand for sequencing of nonmodel and/or wild-type samples, in most cases inbred lines or fosmid-based hierarchical sequencing methods are used to overcome such problems. However, these methods are costly and time consuming,  forfeiting the advantages of massive parallel sequencing. Here, we describe a novel de novo assembler, Platanus, that can effectively manage high-throughput data from heterozygous samples. Platanus assembles DNA fragments (reads) into contigs by constructing de Bruijn graphs with automatically optimized k-mer sizes followed by the scaffolding of contigs based on paired-end information. The complicated graph structures that result from the heterozygosity are simplified during not only the contig assembly step but also the scaffolding step. We evaluated the assembly results on eukaryotic samples with various levels of heterozygosity. Compared with other assemblers, Platanus yields assembly results that have a larger scaffold NG50 length without any accompanying loss of accuracy in both simulated and real data. In addition, Platanus recorded the largest scaffold NG50 values for two of the three low-heterozygosity species used in the de novo assembly contest, Assemblathon 2. Platanus therefore provides a novel and efficient approach for the assembly of gigabase-sized highly heterozygous genomes and is an attractive alternative to the existing assemblers designed for genomes of lower heterozygosity.

W7-4: 入力ファイル形式

①入力ファイル形式の説明部分。
②*.fastq.gzは大丈夫なようだ。他にも、-fmtAutoオプションが便利
そうであり、-separateオプションも
使わないといけなさうだと学習

: OR: m,M,s where m and M are o
not, they will be decremented) with m < M <= 31 (if above, will be read
ced)
and s is a step (even number). Velvet w
ill then hash from k=m to k=M with a step of s
filename : path to sequence file or - for standard input

File format options:

-fasta -fastq -raw -fasta.gz -fastq.gz - raw.gz
-sam -bam -fmtAuto

(Note: -fmtAuto will detect fasta or fastq, and will try the fo
llowing programs for decompression : gunzip, pbunzip2, bunzip2

File layout options for paired reads (only for fasta and fastq formats)

:
-interleaved : File contains paired reads interleaved in the
one file (default)
-separate : Read 2 separate files for paired reads

Read type options:

--More--

②

①

W7-5: Synopsis

File Edit View Search Terminal Help

Ja 16:26

- Short single end reads:

```
velveth Assem 29 -short -fastq s_1_sequence.txt
```

- Paired-end short reads (remember to interleave paired reads):

```
velveth Assem 31 -shortPaired -fasta interleaved.fna
```

- Paired-end short reads (using separate files for the paired reads)

```
velveth Assem 31 -shortPaired -fasta -separate left.fa right.fa
```

- Two channels and some long reads:

```
velveth Assem 43 -short -fastq unmapped.fna -longPaired -fasta  
SangerReads.fasta
```

- Three channels:

```
velveth Assem 35 -shortPaired -fasta pe_lib1.fasta -shortPaired  
2 pe_lib2.fasta -short3 se_lib1.fa
```

Output:

directory/Roadmaps

--More--



作業ディレクトリはどこでもよい。①velvethのマニュアルをそのまま任意のファイル名(hoge.txt)で共有フォルダ(~/Desktop/mac_share)に保存して、使い慣れたホストOS上のテキストエディタで見ることもできる。②headコマンドで最初の10行分(デフォルト)を表示

W7-6: Tips

```
File Edit View Search Terminal Help
iu@bielinux[result] pwd
/home/iu/Documents/DRR024501/result
iu@bielinux[result] date
2016年 1月 1日 金曜日 11:43:35 JST
iu@bielinux[result] velveth -h > ~/Desktop/mac_share/hoge.txt [11:43午前]
iu@bielinux[result] ls -l ~/Desktop/mac_share/hoge*
-rwxrwxrwx 1 iu iu 2632 1月 1 2016 /home/iu/Desktop/mac_share/hoge.txt [11:43午前]
iu@bielinux[result] head ~/Desktop/mac share/hoge.txt [11:43午前]
velveth - simple hashing program
Version 1.2.09

Copyright 2007, 2008 Daniel Zerbino (zerbino@ebi.ac.uk)
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Compilation settings:
CATEGORIES = 2
MAXKMERLENGTH = 31
iu@bielinux[result] [11:43午前]
```

W7-7:velveth

```
File Edit View Search Terminal Help
iu@bielinux[result] pwd
/home/iu/Documents/DRR024501/result
iu@bielinux[result] ls
fastqCount.txt      QC.2.trimmed.fastq.gz  QC.stats.txt
QC.1.trimmed.fastq.gz  QC_qc_report.pdf    QC.unpaired.trimmed.fastq
iu@bielinux[result] velveth uge 31 -shortPaired -fmtAuto -separate QC.1.trim
med.fastq.gz QC.2.trimmed.fastq.gz
```

[11:58午前]

Velvetは、velvethとvelvetgの2つのコマンドから構成される。まずは①velvethコマンドの実行。②lsで見られる pairedの2つのfastq.gzを入力とする。③-fmtAutoは入力ファイル形式がFASTAやFASTQのどれでも通用するので、-fastaや-fastqの代わりにお約束的に用いてよい

②k=31で実行し、結果をugeディレクトリに保存。画面はリターンキーを押して数秒後の状態。約1分

W7-7:velveth

```
File Edit View Search Terminal Help [11:58午前]
iu@bielinux[result] pwd
/home/iu/Documents/DRR024501/result [11:58午前]
iu@bielinux[result] ls
fastqCount.txt QC.2.trimmed.fastq.gz QC.stats.txt
QC.1.trimmed.fastq.gz QC_qc_report.pdf QC.unpaired.trimmed.fastq
iu@bielinux[result] velveth uge 31 -shortPaired -fmtAuto -separate QC.1.trim
med.fastq.gz QC.2.trimmed.fastq.gz
[0.000000] Reading file 'QC.1.trimmed.fastq.gz' using 'gunzip' as FastQ
[0.005418] Reading file 'QC.2.trimmed.fastq.gz' using 'gunzip' as FastQ
[5.034392] 595266 sequences found in total in the paired sequence files
[5.034420] Done
[5.034451] Reading read set file uge/Sequences;
[5.215596] 595266 sequences found
[5.839366] Done
[5.839403] 595266 sequences in total.
[5.839442] Writing into roadmap file uge/Roadmaps...
[7.303562] Inputting sequences...
[7.303605] Inputting sequence 0 / 595266
```

①velveth実行後の状態。②lsすると、③確かにugeディレクトリが作成されていることがわかる

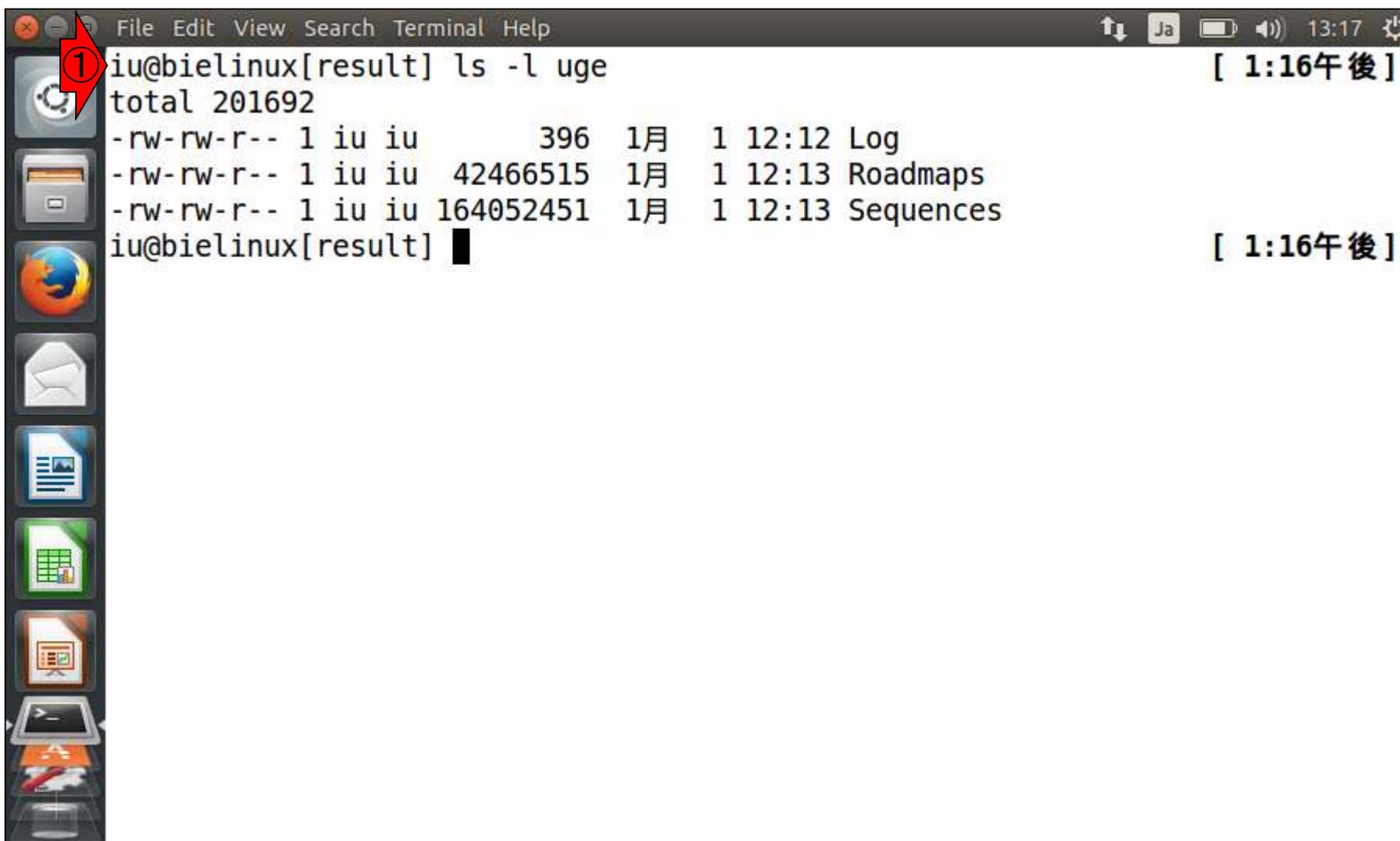
W7-7:velveth

```
File Edit View Terminal Help  
iu@bielinux[result] velveth uge 31 -shortPaired -fmtAuto -separate QC.1.trimmed.fastq.gz QC.2.trimmed.fastq.gz  
[0.000000] Reading file 'QC.1.trimmed.fastq.gz' using 'gunzip' as FastQ  
[0.005418] Reading file 'QC.2.trimmed.fastq.gz' using 'gunzip' as FastQ  
[5.034392] 595266 sequences found in total in the paired sequence files  
[5.034420] Done  
[5.034451] Reading read set file uge/Sequences;  
[5.215596] 595266 sequences found  
[5.839366] Done  
[5.839403] 595266 sequences in total.  
[5.839442] Writing into roadmap file uge/Roadmaps...  
[7.303562] Inputting sequences...  
[7.303605] Inputting sequence 0 / 595266  
[28.061760] === Sequences loaded in 20.758200 s  
[28.061854] Done inputting sequences  
[28.061861] Destroying splay table  
[28.083910] Splay table destroyed  
iu@bielinux[result] ls  
fastqCount.txt QC_qc_report.pdf  
QC.1.trimmed.fastq.gz QC.stats.txt  
QC.2.trimmed.fastq.gz QC.unpaired.trimmed.fastq  
iu@bielinux[result]
```

uge [12:13午後] (3)

[12:16午後]

W7-7:velveth



File Edit View Search Terminal Help [1:16午後]
iu@bielinux[result] ls -l uge
total 201692
-rw-rw-r-- 1 iu iu 396 1月 1 12:12 Log
-rw-rw-r-- 1 iu iu 42466515 1月 1 12:13 Roadmaps
-rw-rw-r-- 1 iu iu 164052451 1月 1 12:13 Sequences
iu@bielinux[result] [1:16午後]

The terminal window shows the command 'ls -l uge' being run, listing three files: Log, Roadmaps, and Sequences. The 'Log' file is small (396 bytes), while 'Roadmaps' and 'Sequences' are much larger (42466515 and 164052451 bytes respectively). The timestamp indicates the operation occurred at 12:12 PM on January 1st. The terminal window has a dark theme and includes a dock on the left side with various application icons.

W7-8: velvetg

Velvetは、velvethとvelvetgの2つのコマンドから構成される。①velvetgは、velvethで作成したugeディレクトリを入力として指定する。画面は実行数秒後の状態。約2分

```
File Edit View Search Terminal Help [ 1:16 午後 ]
iu@bielinux[result] ls -l uge
total 201692
-rw-rw-r-- 1 iu iu      396  1月   1 12:12 Log
-rw-rw-r-- 1 iu iu 42466515  1月   1 12:13 Roadmaps
-rw-rw-r-- 1 iu iu 164052451  1月   1 12:13 Sequences
iu@bielinux[result] velvetg uge
[0.000001] Reading roadmap file uge/Roadmaps
[1.195365] 595266 roadmaps read
[1.195756] Creating insertion markers
[1.306829] Ordering insertion markers
[1.979029] Counting preNodes
[2.083571] 1402149 preNodes counted, creating them now
```

W7-8:velvetg

```
File Edit View Search Terminal Help  
[92.292377] Removing reference contigs with co  
[92.298273] Concatenation...  
[92.311109] Renumbering nodes  
[92.311147] Initial node count 55222  
[92.311207] Removed 0 null nodes  
[92.311215] Concatenation over!  
[92.315264] Writing contigs into uge/contigs.fa...  
[92.574157] Writing into stats file uge/stats.txt...  
[92.697270] Writing into graph file uge/LastGraph...  
Final graph has 55222 nodes and n50 of 115, max 2106, total 3522957, using 0  
/595266 reads  
iu@bielinux[result] ls -l uge  
total 265888  
-rw-rw-r-- 1 iu iu 5214281 1月 1 13:23 contigs.fa  
-rw-rw-r-- 1 iu iu 9939193 1月 1 13:23 Graph  
-rw-rw-r-- 1 iu iu 9939193 1月 1 13:23 LastGraph  
-rw-rw-r-- 1 iu iu 803 1月 1 13:23 Log  
-rw-rw-r-- 1 iu iu 37033601 1月 1 13:21 PreGraph  
-rw-rw-r-- 1 iu iu 42466515 1月 1 12:13 Roadmaps  
-rw-rw-r-- 1 iu iu 164052451 1月 1 12:13 Sequences  
-rw-rw-r-- 1 iu iu 3592487 1月 1 13:23 stats.txt  
iu@bielinux[result]
```

velvetg終了後の状態。①ugeディレクトリの中身を表示。velvetg実行前は3つのファイルしかなかったが、計8ファイルに増えていることがわかる。このうち、②contigs.faが主なアセンブリ結果のmulti-FASTAファイル

[1:23午後]

[1:24午後]

W7-8:velvetg

```
File Edit View Search Terminal Help
iu@bielinux[result] cd uge
iu@bielinux[uge] pwd
/home/iu/Documents/DRR024501/result/uge
iu@bielinux[uge] ls
contigs.fa Graph LastGraph Log PreGraph Roadmaps Sequences stats.txt
iu@bielinux[uge] grep -c ">" contigs.fa
29502
①
②
iu@bielinux[uge] head -n 11 contigs.fa
>NODE_1_length_47_cov_49.276596
GCATACTGAAATTGAAGCTGATCAACACGAAACCATTCCAGCTGGCAAGGGTAATCTAAA
CCACCCATTAGCTGTAA
>NODE_2_length_85_cov_53.811764
AGGGTAATCTAACCAACCCATTAGCTGTTATTGAAGCTTGAGCAACGAGTTGATGATA
AAATGACCCTTCGGTTGATGTGGGGAGCCATTATTTGGATGGCCCGGCACCTT
>NODE_3_length_31_cov_53.387096
CGCCATTATTGTTAGTAATGGATGCAGACGCTGGAGTGGCGTACCTTGGTCAATT
T
>NODE_4_length_120_cov_50.200001
TGGGCCAAATTAAACCGCGGCATCTTACCGTATTCATAATTCTTGGAAACGAACCAT
iu@bielinux[uge] ■
```

[10:35午前]

[10:35午前]

[10:35午前]

[10:35午前]

①配列数をカウント。grepのcオプションは、入力ファイル(contigs.fa)中の”>”を含む行数を出力するという。29,502個の配列があることがわかる。数万というオーダーは、アセンブリ結果としてはよくないといえる。②headコマンドでcontigs.faの最初の11行を表示。grepとwcは第3回W14にもあり

Contents

- Illumina HiSeqデータ(トランスクリプトーム)の乳酸菌ゲノムへのマッピング
 - W14:QuasRパッケージを用いたマッピングの事前準備と本番
 - W15:結果の解説、forward側の100–107bp付近に問題があることを特定
- トリミング、*de novo*トランスクリプトームアセンブリとマッピングの再実行
 - W16:問題のある領域(forward側の100–107bp)のトリミング
 - W17:トリム後のデータでアセンブリを再実行(Rockhopper2; クラスパス設定関連Tips含む)
 - W18:トリム後のデータでマッピングを再実行(QuasR)
 - W19:トリム前のデータでクオリティチェックを再実行(FastQC)
- Illumina MiSeqデータ(乳酸菌ゲノム)の特徴と前処理
 - W4:FastQC、W5:FaQCs、W6:再度FastQC
- *de novo*ゲノムアセンブリ
 - W7:Bio-LinuxにプレインストールされているVelvet (ver. 1.2.09)を上限のk=31で実行
 - W8:k=31のアセンブリ結果をRで確認。k=141で実行し、k=31の結果と同じになるのを確認
 - W9:Velvet (ver. 1.2.10)のインストール
 - W10:Velvet (ver. 1.2.10)の実行



状況次第で省略

W8-1 : Rで解析

File Edit View Search Terminal Help

```

iu@bielinux[uge] pwd
/home/iu/Documents/DRR024501/result/uge
iu@bielinux[uge] ls
contigs.fa Graph LastGraph Log PreGraph Roadmaps Sequences stats.txt
iu@bielinux[uge] wget -cq http://www.iu.a.u-tokyo.ac.jp/~kadota/book/JSLAB6_1.R
iu@bielinux[uge] ls
contigs.fa JSLAB6_1.R Log Roadmaps stats.txt
Graph LastGraph PreGraph Sequences
iu@bielinux[uge] nkf JSLAB6_1.R | head -n 2
in_f <- "contigs.fa"          #入力ファイル名を指定してin_fに格納
out_f <- "result_hoge.txt"    #出力ファイル名を指定してout_fに格納
iu@bielinux[uge] R --vanilla --slave < JSLAB6_1.R

```

[3:11午後] [3:11午後] [3:13午後]

① ② ③

第5回W12-3 (2016.08.01のスライド165)と同じ解析をcontigs.faを入力として実行。
①wgetでJSLAB6_1.Rをダウンロード。②最初の2行分を表示。JSLAB5_2.Rとの違いは、赤下線の入出力ファイル名部分のみ。③JSLAB6_1.Rをバッチモードで実行

W8-1 : Rで解析

計算自体は数秒。①lsで確認。②JSLAB6_1.R上で出力ファイル名として指定したresult_hoge.txtが確かに作成されていることがわかる

```

File Edit View Search Terminal Help

Loading required package: S4Vectors
Loading required package: stats4
Creating a generic function for 'nchar' from package 'base' in package 'S4Vector
s'
Loading required package: IRanges
Loading required package: XVector
iu@bielinux[uge] ls -l
total 265896
-rw-rw-r-- 1 iu iu 5214281 1月 1 13:23 contigs.fa
-rw-rw-r-- 1 iu iu 9939193 1月 1 13:23 Graph
-rw-rw-r-- 1 iu iu 2013 1月 3 14:57 JSLAB6_1.R
-rw-rw-r-- 1 iu iu 9939193 1月 1 13:23 LastGraph
-rw-rw-r-- 1 iu iu 803 1月 1 13:23 Log
-rw-rw-r-- 1 iu iu 37033601 1月 1 13:21 PreGraph
-rw-rw-r-- 1 iu iu 167 1月 3 15:22 result_hoge.txt
-rw-rw-r-- 1 iu iu 42466515 1月 1 12:13 Roadmaps
-rw-rw-r-- 1 iu iu 164052451 1月 1 12:13 Sequences
-rw-rw-r-- 1 iu iu 3592487 1月 1 13:23 stats.txt
iu@bielinux[uge] date
2016年 1月 3日 日曜日 15:22:18 JST
iu@bielinux[uge] █

```

[3:22午後]

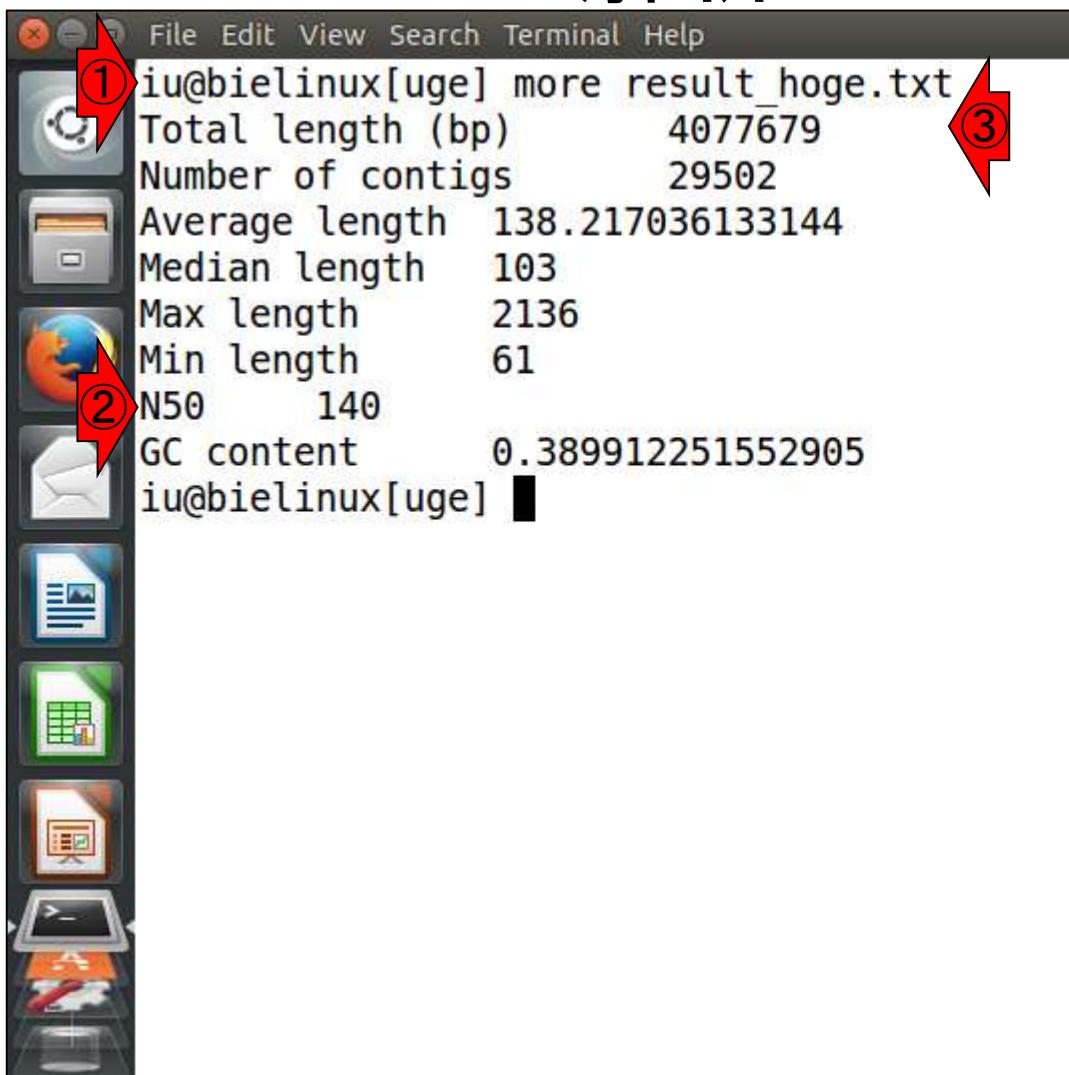
①

②

[3:22午後]

[3:22午後]

W8-1 : Rで解析



```
File Edit View Search Terminal Help
① iu@bielinux[uge] more result_hoge.txt
Total length (bp)      4077679
Number of contigs      29502
Average length         138.217036133144
Median length          103
Max length             2136
Min length             61
N50                   140
GC content             0.389912251552905
iu@bielinux[uge] ■
```

①result_hoge.txtの中身をmoreで表示。W7-8の「grep -c ">" contigs.fa」の結果と同じく、29,502 contigsになっていることがわかる。②N50は140、③総塩基数は4,077,679 bpとなっている。これはゲノムサイズが約4.08MBということを意味する(実際は約2.4MB)。明らかに配列(コンティグ)数が多くあてにはならない

[3:31午後]

W8-1: コードの中身は...

JSLAB6_1.Rの中身は、①の例題1と基本的に同じ。違いは③赤枠の入出力ファイル名部分のみ。スライドを見るだけ

(Rで)塩基配列解析

～NGS、RNA-seq、ゲノム、トランскриプトーム、正規化、発現変動、統計、モデル、バイオインフォマティクス～
(last modified 2015/12/22, since 2011)

状況次第で省略

What's new?

- このウェブペ Rと必要なパ (Windows20 (2015/04/03)
- 多群間比較

- ・ イントロ | NGS | アノテーション情報取得 | TranscriptDb | [GenomicFeatures\(Lawrence 2013\)](#) (last modified 2015/02/19)
- ・ イントロ | NGS | アノテーション情報取得 | TranscriptDb | [GFF/GTF形式ファイルから](#) (last modified 2015/03/04)
- ・ イントロ | NGS | 読み込み | BSgenome | [基本情報を取得](#) (last modified 2015/09/12)
- ・ イントロ | NGS | 読み込み | FASTA形式 | [基本情報を取得](#) (last modified 2015/09/12)
- ・ イントロ | NGS | 読み込み | FASTA形式 | [description行の記述を整形](#) (last modified 2014/04/05)
- ・ イントロ | NGS | 読み込み | FASTQ形式 | [基礎](#) (last modified 2015/07/26)
- ・ イントロ | NGS | 読み込み | FASTQ形式 | [応用](#) (last modified 2015/06/18)
- ・ イントロ | NGS | 読み込み | FASTQ形式 | [description行の記述を整形](#) (last modified 2014/08/21)
- ・ イントロ | NGS | 読み込み | [Illuminaの*.seq.txt](#) (last modified 2013/06/13)
- ・ イントロ | NGS | 読み込み | [Illumina](#)
- ・ [イントロ | ファイル形式の変換](#) | につい
- ・ イントロ | ファイル形式の変換 | [BAN](#)

①

イントロ | NGS | 読み込み | FASTA形式 | 基本情報を取得

multi-FASTAファイルを読み込んで、Total lengthやaverage lengthなどの各種情報取得を行うためのやり方を示します。例題6以降は、ヒトやマウスレベルの巨大ファイルを取り扱うためのコードです。具体的には、塩基数を整数(integer)ではなく実数(real number)として取り扱うためのas.numeric関数を追加しています。
「ファイル」→「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピペ。

②

1. イントロ | 一般 | ランダムな塩基配列を作成の4.を実行して得られたmulti-FASTAファイル([hoge4.fa](#))の場合:

```
in_f <- "hoge4.fa"                                # 入力ファイル名を指定してin_fに格納
out_f <- "hoge1.txt"                               # 出力ファイル名を指定してout_fに格納

# 必要なパッケージをロード
library(Biostrings)                                # パッケージの読み込み

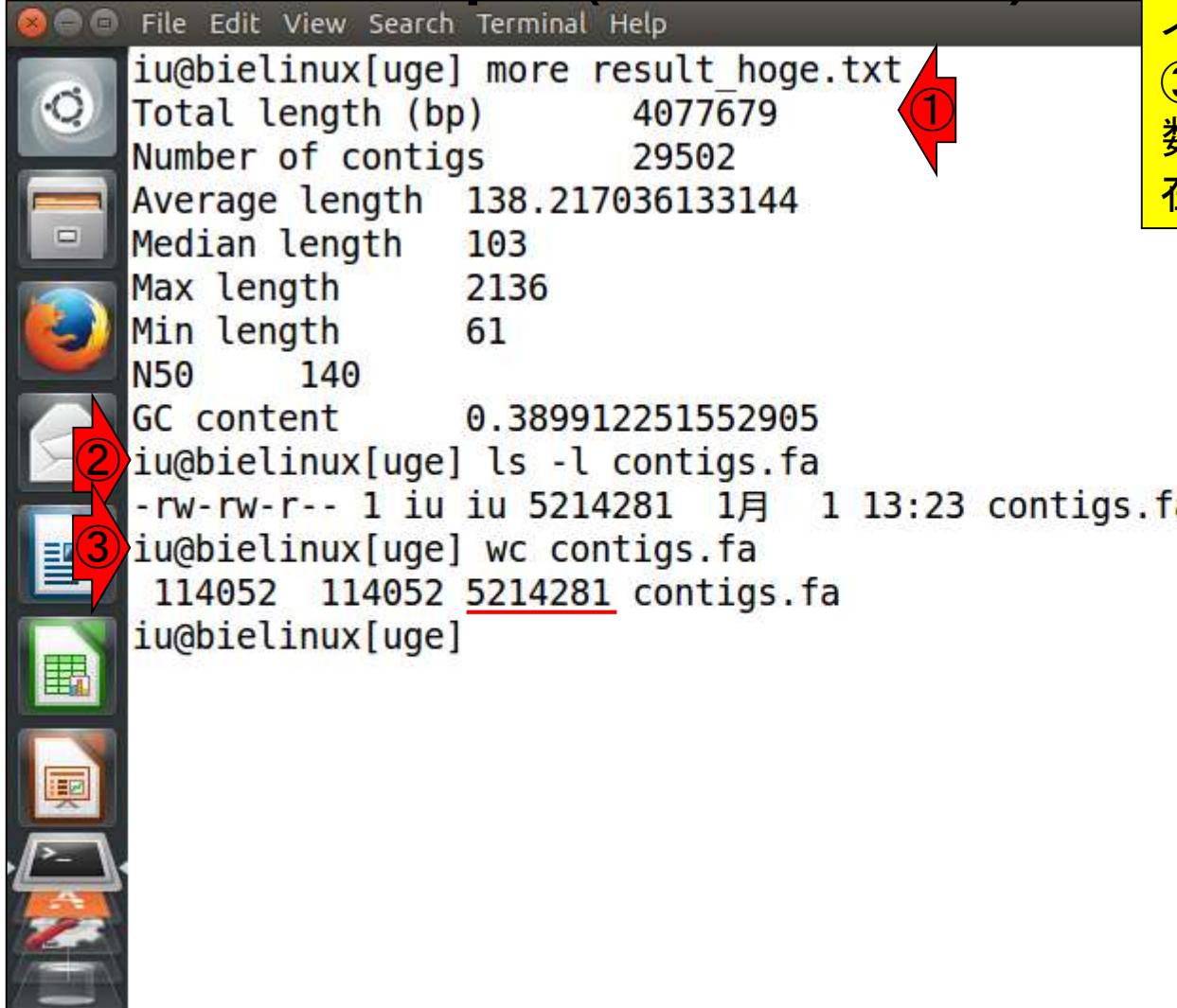
# 入力ファイルの読み込み
fasta <- readDNAStringSet(in_f, format="fasta") # in_fで指定したファイルの読み込み

# 本番(基本情報取得)
Total_len <- sum(width(fasta))                  # 配列の「トータルの長さ」を取得
Number_of_contigs <- length(fasta)               # 「配列数」を取得
```

③

W8-2: Tips(総塩基数)

```
File Edit View Search Terminal Help  
iu@bielinux[uge] more result_hoge.txt  
Total length (bp) 4077679  
Number of contigs 29502  
Average length 138.217036133144  
Median length 103  
Max length 2136  
Min length 61  
N50 140  
GC content 0.389912251552905  
iu@bielinux[uge] ls -l contigs.fa  
-rw-rw-r-- 1 iu iu 5214281 1月 1 13:23 contigs.fa  
iu@bielinux[uge] wc contigs.fa  
114052 114052 5214281 contigs.fa  
iu@bielinux[uge]
```



①総塩基数(4,077,679 bp)は、contigs.fa中のdescription行を除いたものから把握可能であるというTips。②contigs.faのファイルサイズは、5,214,281 bytes。これは③wc実行結果の一番右側の数値(バイト数)と同じ。この数値は、1行につき1つ存在する改行コード分を含む文字数と同じ

状況次第で省略

[4:27午後]

[4:27午後]

[4:27午後]

W8-2: Tips(総塩基数)

iu@bielinux[~/Documents/DRR024501/result/uge]

```
iu@bielinux[uge] more result_hoge.txt
Total length (bp)      4077679
Number of contigs       29502
Average length          138.217036133144
Median length           103
Max length              2136
Min length              61
N50                     140
GC content              0.389912251552905
```

```
iu@bielinux[uge] ls -l contigs.fa
-rw-rw-r-- 1 iu iu 5214281 1月 1 13:23 contigs.fa
```

```
iu@bielinux[uge] wc contigs.fa
114052 114052 5214281 contigs.fa
```

```
iu@bielinux[uge] grep -v ">" contigs.fa | wc
84550 84550 4162229
```

①

②

③

④

そのため、①contigs.faからdescription行を除いた(grep -v ">"に相当)分を、パイプでつなげてwcすることによって得られた②4,162,229 bytesを見ることで、「だいたい総塩基数はこれくらいね」とわかる。③実際の総塩基数(4,077,679 bp)よりも若干大きな値になる理由は、④計84,550行分だけ改行コードが含まれているから。それゆえ、wc実行結果の②4,162,229 - ④84,550)からも、実際の総塩基数情報(4,077,679 bp)を正確に得られる

状況次第で省略

[4:14午後]

[4:14午後]

[4:14午後]

[4:14午後]

W8-2: Tips(総塩基数)

状況次第で省略

①と②の比較で、「grep -v ">"」によってdescription行をうまく除去できていることがわかる

```

File Edit View Search Terminal Help [ 4:08 午後 ]
iu@bielinux[uge] ls -l contigs.fa
-rw-rw-r-- 1 iu iu 5214281 1月 1 13:23 contigs.fa [ 4:08 午後 ]
iu@bielinux[uge] wc contigs.fa
114052 114052 5214281 contigs.fa [ 4:08 午後 ]
iu@bielinux[uge] grep -v ">" contigs.fa | wc
84550 84550 4162229 [ 4:08 午後 ]
iu@bielinux[uge] head -n 8 contigs.fa [ 4:09 午後 ]
>NODE_1_length_47_cov_49.276596
GCATACTGAAATTGAAGCTGATCAACACGAAACCATTCCAGCTGGCAAGGGTAATCTAAA
CCACCCATTAGCTGTTA
>NODE_2_length_85_cov_53.811764
AGGGTAATCTAACCAACCCATTAGCTGTTATTGAAGCTTGCAGCAACGAGTTGATGATA
AAATGACCGTTCGGTTGATGTGGGGAGCCATTATATTGGATGGCCCGGCACTT } [ 4:11 午後 ]
>NODE_3_length_31_cov_53.387096
CGCCATTATTGTTTAGTAATGGGATGCAGACGCTTGGAGTGGCGCTACCTTGGTCAATT
iu@bielinux[uge] grep -v ">" contigs.fa | head -n 5
GCATACTGAAATTGAAGCTGATCAACACGAAACCATTCCAGCTGGCAAGGGTAATCTAAA
CCACCCATTAGCTGTTA } [ 4:11 午後 ]
AGGGTAATCTAACCAACCCATTAGCTGTTATTGAAGCTTGCAGCAACGAGTTGATGATA
AAATGACCGTTCGGTTGATGTGGGGAGCCATTATATTGGATGGCCCGGCACTT
CGCCATTATTGTTTAGTAATGGGATGCAGACGCTTGGAGTGGCGCTACCTTGGTCAATT
iu@bielinux[uge] ■

```

W8-3:k=141で実行

File Edit View Search Terminal Help

```

iu@bielinux[uge] cd ~/Documents/DRR024501/result
iu@bielinux[result] pwd
/home/iu/Documents/DRR024501/result
iu@bielinux[result] ls
fastqCount.txt      QC.2.trimmed.fastq.gz  QC.stats.txt
QC.1.trimmed.fastq.gz  QC_qc_report.pdf    QC.unpaired.trimmed.fastq
iu@bielinux[result] velveth mongee 141 -shortPaired -fmtAuto -separate QC.1.tr
immed.fastq.gz QC.2.trimmed.fastq.gz
[0.000000] Velvet can't handle k-mers as long as 141! We'll stick to 31 if you
don't mind.
[0.093173] Reading file 'QC.1.trimmed.fastq.gz' using 'gunzip' as FastQ
[0.103678] Reading file 'QC.2.trimmed.fastq.gz' using 'gunzip' as FastQ

```

[11:12午後] uge

W7-3でも示したが、k=31以上の数値を指定すると31になるとことを一応確認すべくk=141でアセンブルを試みる。②のあたりで既に「そんな長いk-merはハンドリングできません!とりあえず31でやります」と書かれていることがわかる

参考 ①のあたりでも上限は31だと書かれている。②「そんなファイルはない!」と文句を言われているようだが、③とりあえず無視してvelvetgを実行

W8-3:k=141で実行

```
File Edit View Terminal Help
iu@bielinux[result] velvet mongee 141 -shortPaired -fmtAuto -separate QC.1.trimmed.fastq.gz QC.2.trimmed.fastq.gz
[0.000000] Velvet can't handle k-mers as long as 141! We'll stick to 31 if you don't mind.
[0.093173] Reading file 'QC.1.trimmed.fastq.gz' using 'gunzip' as FastQ
[0.103678] Reading file 'QC.2.trimmed.fastq.gz' using 'gunzip' as FastQ
[18.491968] 595266 sequences found in total in the paired sequence files
[18.492057] Done
[18.492121] Reading read set file mongee/Sequences;
[18.629979] 595266 sequences found
[19.405358] Done
[19.405463] 595266 sequences in total.
[19.405537] Writing into roadmap file mongee/Roadmaps...
[20.927204] Inputting sequences...
[20.927256] Inputting sequence 0 / 595266
[42.378089] === Sequences loaded in 21.450889 s
[42.378174] Done inputting sequences
[42.378183] Destroying splay table
[42.416255] Splay table destroyed
velveth: Word length 33 greater than max allowed value (31).
Recompile Velvet to deal with this word length.: No such file or directory
iu@bielinux[result] velvetg mongee [11:13午後]
```

① ② ③

W8-3:k=141で実行

無事終了したようだ。①配列数を数えた結果は、予定通りk=31のときと同じ29,502個。②ファイルサイズも、③明示的にk=31で実行したときのものと同じことを確認

状況次第で省略

```

File Edit View Terminal Help
[87.089220] Removed 0 null nodes
[87.089226] Concatenation over!
[87.089230] Removing reference contigs with coverage < -1.000000...
[87.094801] Concatenation...
[87.107378] Renumbering nodes
[87.107411] Initial node count 55222
[87.107463] Removed 0 null nodes
[87.107468] Concatenation over!
[87.110919] Writing contigs into mongee/contigs.fa...
[87.371751] Writing into stats file mongee/stats.txt...
[87.503387] Writing into graph file mongee/LastGraph...
Final graph has 55222 nodes and n50 of 115, max 2106, total 3522957, using 0/5
95266 reads
①iu@bielinux[result] grep -c ">" mongee/contigs.fa [11:22午後]
29502
②iu@bielinux[result] ls -l mongee/contigs.fa [11:22午後]
-rw-rw-r-- 1 iu iu 5214281 1月 3 23:22 mongee/contigs.fa
iu@bielinux[result] date [11:22午後]
2016年 1月 3日 日曜日 23:22:57 JST
③iu@bielinux[result] ls -l uge/contigs.fa [11:22午後]
-rw-rw-r-- 1 iu iu 5214281 1月 1 13:23 uge/contigs.fa
iu@bielinux[result]

```

Contents

- Illumina HiSeqデータ(トランスクリプトーム)の乳酸菌ゲノムへのマッピング
 - W14:QuasRパッケージを用いたマッピングの事前準備と本番
 - W15:結果の解説、forward側の100–107bp付近に問題があることを特定
- トリミング、*de novo*トランスクリプトームアセンブリとマッピングの再実行
 - W16:問題のある領域(forward側の100–107bp)のトリミング
 - W17:トリム後のデータでアセンブリを再実行(Rockhopper2; クラスパス設定関連Tips含む)
 - W18:トリム後のデータでマッピングを再実行(QuasR)
 - W19:トリム前のデータでクオリティチェックを再実行(FastQC)
- Illumina MiSeqデータ(乳酸菌ゲノム)の特徴と前処理
 - W4:FastQC、W5:FaQCs、W6:再度FastQC
- *de novo*ゲノムアセンブリ
 - W7:Bio-LinuxにプレインストールされているVelvet (ver. 1.2.09)を上限のk=31で実行
 - W8:k=31のアセンブリ結果をRで確認。k=141で実行し、k=31の結果と同じになるのを確認
 - W9:Velvet (ver. 1.2.10)のインストール
 - W10:Velvet (ver. 1.2.10)の実行



W9-1 : Velvetマニュアル

EMBL-EBI



Velvet

Sequence assembler for very short reads

- [Current version: 1.2.10](#)
- [Manual and extension for Columbus in pdf format](#)
-  [① Git URL: git clone git://github.com/dzerbino/velvet.git](#)
- [For up-to-date info, you can consult and/or subscribe to the mailing list.](#)
- [For transcriptomic assembly Velvet is extended by Sanger](#)

<http://www.ebi.ac.uk/~zerbino/velvet/>

Velvet Manual - version 1.1		
Daniel Zerbino		
August 29, 2008		
Contents		
1	For impatient people	2
2	Installation	2
2.1	Requirements	2
2.2	Compiling instructions	3
2.3	Compilation settings	3
2.3.1	Colorspace Velvet	3
2.3.2	CATEGORIES	3
2.3.3	MAXKMERLENGTH	3
2.3.4	BIGASSEMBLY	4
2.3.5	LONGSEQUENCES	4
2.3.6	OPENMP	4
2.3.7	RUNFILEZIP	4

W9-1 : Velvetマニュアル

2.3.3 MAXKMERLENGTH.

Another useful compilation parameter is the MAXKMERLENGTH. As explained in 5.2, the hash length can be crucial to getting optimal assemblies. Depending on the dataset, you might wish to use long hash lengths.

By default, hash-lengths are limited to 31bp, but you can push up this limit by adjusting the MAXKMERLENGTH parameter at compilation time:

```
make 'MAXKMERLENGTH=57'
```

①

(Note the single quotes and absence of spacing.)

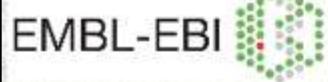
By storing longer words, Velvet will be requiring more memory, so adjust this variable according to your needs and memory resources.

makeというコマンドでインストールを行うのが基本。①MAXKMERLENGTHオプションをつけてコンパイル(velvethやvelvetgコマンドを作成する作業に相当)を行うことで、デフォルトの31よりも大きい任意のk値を指定可能になる。ただし、②大きな値を指定するほどメモリを消費するので、実際にどこまでアセンブルできるかはやってみないとわからない

②

W9-2: ダウンロード

①Current versionのところで右クリックし、②ショートカットのコピーで、wgetでダウンロードするときに必要なURL情報を取得できる。スライドを見るだけ



Velvet

Sequence assembler for very short reads

- ・ [Current version: 1.2.10](#)
- ・ [Manual and extension 1](#)
- ・ [Public Git URL: git clone https://github.com/phac-nml/velvet.git](#)
- ・ [For up-to-date info, you can subscribe to the mailing list.](#)
- ・ [For transcriptomic assembly, see Oases.](#)

開く(O)
新しいタブで開く(W)
新しいウィンドウで開く(N)
対象をファイルに保存(A)...
対象を印刷(P)

切り取り
コピー(C)
ショートカットのコピー(T)
貼り付け(P)

[at](#)
[no/velvet.git](#)

[subscribe to the mailing list.](#)

[by Oases.](#)

W9-2: ダウンロード

The terminal window shows the user's session on a bielinux system. The user navigates to the Downloads directory, lists its contents, performs a wget download, lists the contents again, and finally uses ls -l to show detailed file information. Red numbered arrows point to specific actions: ① points to the terminal window icon in the dock; ② points to the wget command; ③ points to the completed download file in the ls -l output.

```
File Edit View Search Terminal Help
① iu@bielinux[result] cd ~/Downloads
iu@bielinux[Downloads] ls
FaQCs  fastqc_v0.11.4.zip  IGV_2.3.67.zip  Rockhopper.jar
FastQC  IGV_2.3.67          nohup.out        Rockhopper_Results
iu@bielinux[Downloads] wget -cq http://www.ebi.ac.uk/~zerbino/velvet/velvet_1.2.10.tgz
iu@bielinux[Downloads] ls
FaQCs                  IGV_2.3.67          Rockhopper.jar
FastQC                 IGV_2.3.67.zip       Rockhopper_Results
fastqc_v0.11.4.zip    nohup.out          velvet_1.2.10.tgz
iu@bielinux[Downloads] ls -l
total 69364
drwxrwxr-x 6 iu iu      4096 12月 11 16:09 FaQCs
drwxrwxr-x 8 iu iu      4096 12月 10 10:09 FastQC
-rw-rw-r-- 1 iu iu 10026266 10月  9 20:55 fastqc_v0.11.4.zip
drwxr-xr-x 2 iu iu      4096 11月 30 22:10 IGV_2.3.67
-rw-rw-r-- 1 iu iu 28120604 12月  1 12:10 IGV_2.3.67.zip
-rw----- 1 iu iu        0 12月 20 15:57 nohup.out
-rw-rw-r-- 1 iu iu 14039789  3月 17 2015 Rockhopper.jar
drwxrwxr-x 3 iu iu      4096 12月 20 14:28 Rockhopper_Results
-rw-rw-r-- 1 iu iu 18818559 11月 17 2013 velvet_1.2.10.tgz
iu@bielinux[Downloads]
```

①本連載では、プログラムの大元は ~/Downloadsに置くようしているが、 /usr/local/srcというディレクトリに置くのが 正統派らしい。②wgetでダウンロードした つもり。③20MB弱。ダウンロード済みです

W9-3: tgzファイルの解凍

```
iu@bielinux[Downloads] pwd  
/home/iu/Downloads  
iu@bielinux[Downloads] ls  
boost_1_61_0.tar.bz2      master.zip  
Bridger_r2014-12-01.tar.gz  nohup.out  
FaQCs                      Rockhopper.jar  
FastQC                     Rockhopper_Results  
fastqc_v0.11.4.zip          sratoolkit.2.6.3-ubuntu64.tar.gz  
IGV_2.3.67                  v2.2.0.tar.gz  
IGV_2.3.67.zip              velvet_1.2.10.tgz ①  
kmergenie-1.6982.tar.gz  
iu@bielinux[Downloads] tar -zxvf velvet_1.2.10.tgz [ 2:50午後]
```

①.tgzという拡張子は、.tar.gzの略。このファイル形式も比較的よく目に見る。解凍は、②tarコマンドを-zxvfというオプションをつけて実行すればよい。**ここからは手を動かす**

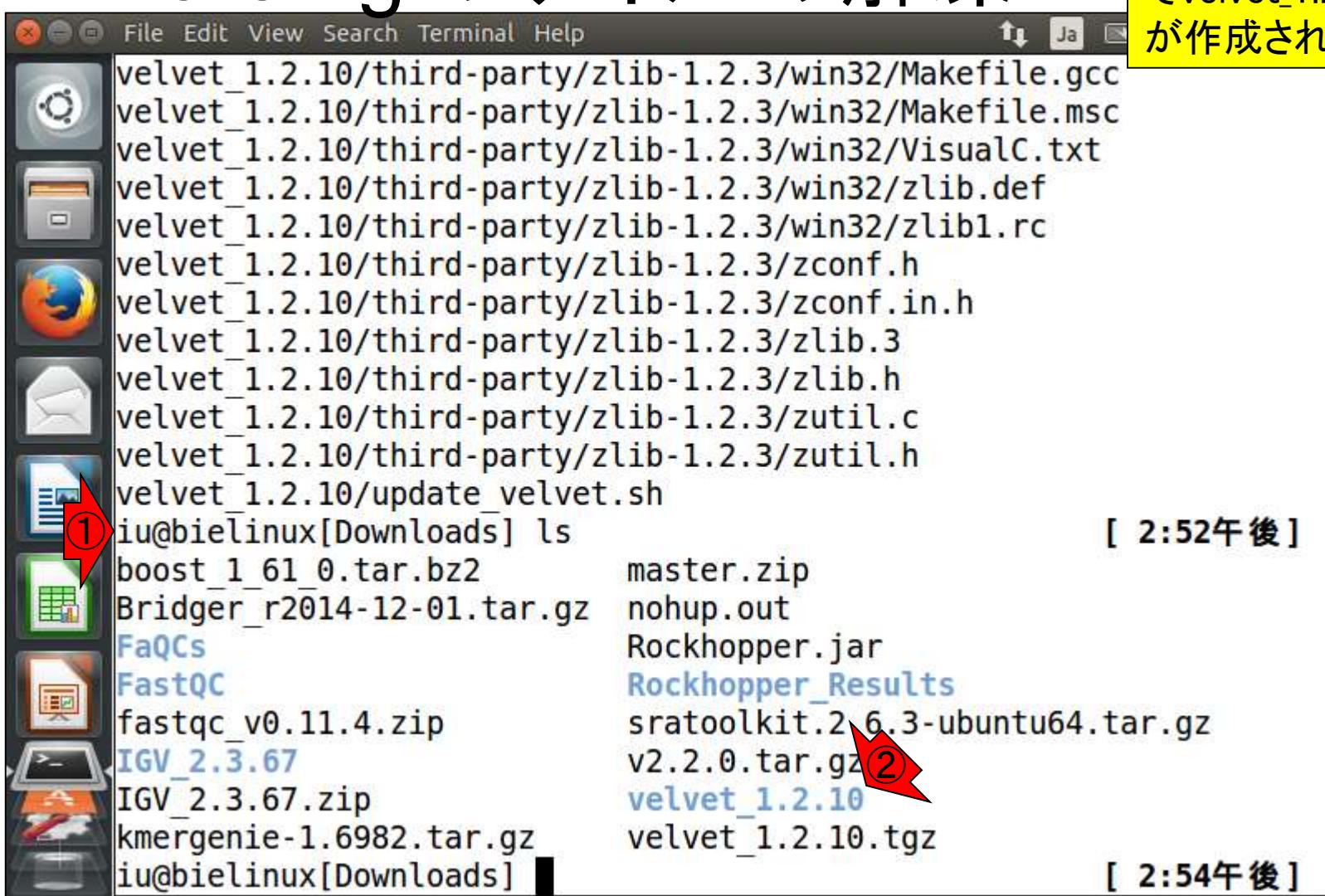
[2:50午後]

[2:50午後]

W9-3: tgzファイルの解凍

解凍は、ほぼ一瞬で終わる。①lsで確認。②確かに無事解凍されてvelvet_1.2.10というディレクトリが作成されていることがわかる

```
File Edit View Search Terminal Help Ja
velvet_1.2.10/third-party/zlib-1.2.3/win32/Makefile.gcc
velvet_1.2.10/third-party/zlib-1.2.3/win32/Makefile.msc
velvet_1.2.10/third-party/zlib-1.2.3/win32/VisualC.txt
velvet_1.2.10/third-party/zlib-1.2.3/win32/zlib.def
velvet_1.2.10/third-party/zlib-1.2.3/win32/zlib1.rc
velvet_1.2.10/third-party/zlib-1.2.3/zconf.h
velvet_1.2.10/third-party/zlib-1.2.3/zconf.in.h
velvet_1.2.10/third-party/zlib-1.2.3/zlib.3
velvet_1.2.10/third-party/zlib-1.2.3/zlib.h
velvet_1.2.10/third-party/zlib-1.2.3/zutil.c
velvet_1.2.10/third-party/zlib-1.2.3/zutil.h
velvet_1.2.10/update_velvet.sh
iu@bielinux[Downloads] ls [ 2:52 午後 ]
boost_1_61_0.tar.bz2      master.zip
Bridger_r2014-12-01.tar.gz nohup.out
FaQCs                     Rockhopper.jar
FastQC                    Rockhopper_Results
fastqc_v0.11.4.zip        sratoolkit.2.6.3-ubuntu64.tar.gz
IGV_2.3.67                v2.2.0.tar.gz
IGV_2.3.67.zip            velvet_1.2.10
kmergenie-1.6982.tar.gz   velvet_1.2.10.tgz
iu@bielinux[Downloads] [ 2:54 午後 ]
```



W9-3: tgzファイルの解凍

```
File Edit View Search Terminal Help
velvet_1.2.10/third-party/zlib-1.2.3/zutil.h
velvet_1.2.10/update_velvet.sh
iu@bielinux[Downloads] ls
boost_1_61_0.tar.bz2      master.zip
Bridger_r2014-12-01.tar.gz nohup.out
FaQCs                      Rockhopper.jar
FastQC                     Rockhopper_Results
fastqc_v0.11.4.zip          sratoolkit.2.6.3-ubuntu64.tar.gz
IGV_2.3.67                  v2.2.0.tar.gz
IGV_2.3.67.zip              velvet_1.2.10
kmergenie-1.6982.tar.gz    velvet_1.2.10.tgz
iu@bielinux[Downloads] cd velvet_1.2.10
iu@bielinux[velvet_1.2.10] pwd
/home/iu/Downloads/velvet_1.2.10
iu@bielinux[velvet_1.2.10] ls
ChangeLog                   doc
Columbus_manual.pdf        For_MAC_or_SPARC_users.txt
contrib                      LICENSE.txt
CREDITS.txt                 Makefile
data                         Manual.pdf
debian                       README.txt
iu@bielinux[velvet_1.2.10] [ 2:54 午後]
src                          [ 2:55 午後]
tests                        [ 2:55 午後]
third-party
update_velvet.sh            [ 2:55 午後]
```

①velvet_1.2.10ディレクトリに移動し、②lsで中身を確認。マニュアルの2.2節を見ると「make(と打つのが基本)」と書いている。アセンブリなど計算時間のかかるプログラムはmakeというコマンドを打ち込んで、実行ファイル(この場合velvethとvelvetg)を作成する場合が多い。経験上、③Makefileというものが存在する場合は、makeと打てばよい

W9-4 : make

```
File Edit View Search Terminal Help
iu@bielinux[velvet_1.2.10] pwd
/home/iu/Downloads/velvet_1.2.10
iu@bielinux[velvet_1.2.10] ls
ChangeLog          doc
Columbus_manual.pdf For_MAC_or_SPARC_users.txt
contrib            LICENSE.txt
CREDITS.txt        Makefile
data               Manual.pdf
debian             README.txt
iu@bielinux[velvet_1.2.10] make 'MAXKMERLENGTH=201' [ 3:11午後]
```

① ②

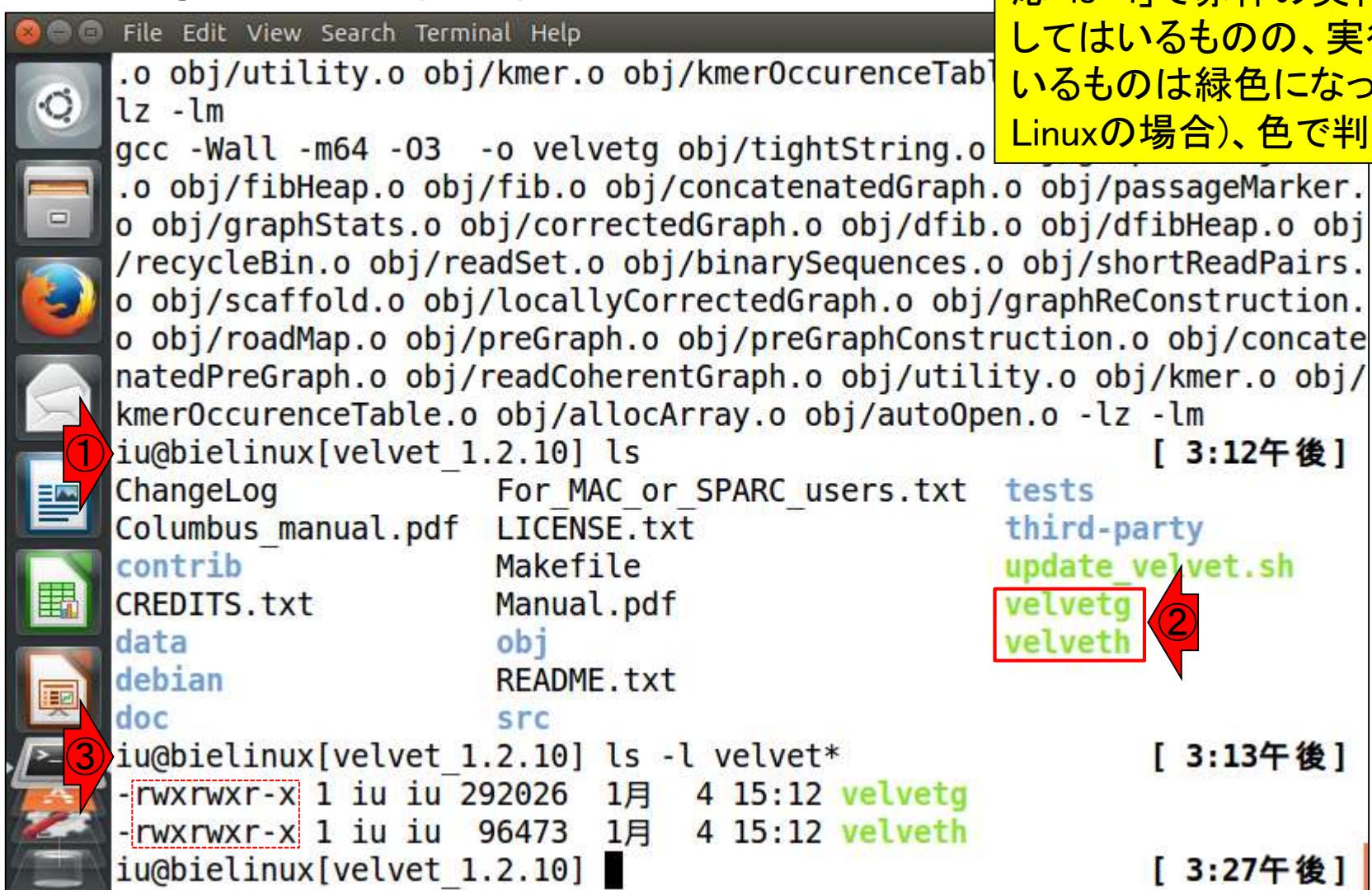
[3:11午後]

src
tests
third-party
update_velvet.sh

①k値の指定可能な最大値を201にしたい場合。マニュアルには明記されていないが、②Makefileが存在するディレクトリ上で実行するのが(たぶん)常識。このmakeを実行する作業を「コンパイル(compile)する」という

W9-4 : make

```
File Edit View Search Terminal Help
.o obj/utility.o obj/kmer.o obj/kmerOccurrenceTable.o obj/allocArray.o obj/autoOpen.o -lz -lm
gcc -Wall -m64 -O3 -o velvetg obj/tightString.o
.o obj/fibHeap.o obj/fib.o obj/concatenatedGraph.o obj/passageMarker.o obj/graphStats.o obj/correctedGraph.o obj/dfib.o obj/dfibHeap.o obj/recycleBin.o obj/readSet.o obj/binarySequences.o obj/shortReadPairs.o obj/scaffold.o obj/locallyCorrectedGraph.o obj/graphReConstruction.o obj/roadMap.o obj/preGraph.o obj/preGraphConstruction.o obj/concatenatedPreGraph.o obj/readCoherentGraph.o obj/utility.o obj/kmer.o obj/kmerOccurrenceTable.o obj/allocArray.o obj/autoOpen.o -lz -lm
iu@bielinux[velvet_1.2.10] ls [ 3:12 午後 ]
ChangeLog           For_MAC_or_SPARC_users.txt   tests
Columbus_manual.pdf LICENSE.txt                 third-party
contrib              Makefile                   update_velvet.sh
CREDITS.txt         Manual.pdf                velvet
data                obj
debian               README.txt
doc                 src
iu@bielinux[velvet_1.2.10] ls -l velvet* [ 3:13 午後 ]
-rwxrwxr-x 1 iu iu 292026 1月 4 15:12 velvetg
-rwxrwxr-x 1 iu iu 96473 1月 4 15:12 velveth
iu@bielinux[velvet_1.2.10] [ 3:27 午後 ]
```



make実行時間は数秒。①lsで確認。確かに②velvetgとvelvethができている。③一応「ls -l」で赤枠の実行権限部分を確認してはいるものの、実行権限(x)がついているものは緑色になっているので(Bio-Linuxの場合)、色で判断してもよい

W9-5: 復習と確認

```
File Edit View Search Terminal Help
iu@bielinux[velvet_1.2.10] pwd
/home/iu/Downloads/velvet_1.2.10
iu@bielinux[velvet_1.2.10] ls -l velvet*
-rwxrwxr-x 1 iu iu 292026 1月 4 15:12 velvetg
-rwxrwxr-x 1 iu iu 96473 1月 4 15:12 velveth
iu@bielinux[velvet_1.2.10] velveth -h | head
velveth - simple hashing program
Version 1.2.09

Copyright 2007, 2008 Daniel Zerbino (zerbino@ebi.ac.uk)
This is free software; see the source for copying conditions. There
is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PU
RPOSE.

Compilation settings:
CATEGORIES = 2
MAXKMERLENGTH = 31
iu@bielinux[velvet_1.2.10] where velveth
/usr/bin/velveth
iu@bielinux[velvet_1.2.10]
```

第4回のW9-3でパスの概念を説明した通り、「~/Downloads/velvet_1.2.10」上で①velvethと打っても、Bio-Linuxにプレインストールされているver. 1.2.09が実行されるので、②指定可能なk値の上限は31のまま

[3:55 午後]

[3:55 午後]

[3:55 午後]

[3:55 午後]

W9-5: 復習と確認

```
File Edit View Search Terminal Help
iu@bielinux[velvet_1.2.10] pwd
/home/iu/Downloads/velvet_1.2.10
iu@bielinux[velvet_1.2.10] ls -l velvet*
-rwxrwxr-x 1 iu iu 292026 1月 4 15:12 velvetg
-rwxrwxr-x 1 iu iu 96473 1月 4 15:12 velveth
iu@bielinux[velvet_1.2.10] ./velveth -h | head [ 4:08 午後 ]
velveth - simple hashing program
Version 1.2.10

Copyright 2007, 2008 Daniel Zerbino (zerbino@ebi.ac.uk)
This is free software; see the source for copying conditions. There
is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PU
RPOSE.

Compilation settings:
CATEGORIES = 2
MAXKMERLENGTH = 201
iu@bielinux[velvet_1.2.10] [ 4:08 午後 ]
```

カレントディレクトリの「~/Downloads/velvet_1.2.10」上に存在するvelvethを実行したい場合は、①./velvethと打てばよい(第4回のW3-1, W9-3, W15-7)。確かにさきほどインストールしたver. 1.2.10が実行され、②指定可能なk値の上限は201になっている。ちなみにこれは相対パス指定のやり方

W9-5: 復習と確認

iu@bielinux[~/Downloads/velvet_1.2.10]

```
iu@bielinux[velvet_1.2.10] pwd  
/home/iu/Downloads/velvet_1.2.10
```

[4:19 午後]

```
iu@bielinux[velvet_1.2.10] ls -l velvet*  
-rwxrwxr-x 1 iu iu 292026 1月 4 15:12 velvetg  
-rwxrwxr-x 1 iu iu 96473 1月 4 15:12 velveth
```

[4:19 午後]

```
iu@bielinux[velvet_1.2.10] /home/iu/Downloads/velvet_1.2.10/velveth -  
h | head
```

```
velveth - simple hashing program  
Version 1.2.10
```

```
Copyright 2007, 2008 Daniel Zerbino (zerbino@ebi.ac.uk)  
This is free software; see the source for copying conditions. There  
is NO  
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PU  
RPOSE.
```

Compilation settings:

CATEGORIES = 2

MAXKMERLENGTH = 201

```
iu@bielinux[velvet_1.2.10]
```

[4:19 午後]

絶対パス指定の場合は、①こんな感じ。当然ユーザー名がiuでないヒトはこれをそのまま打ち込んでもダメ! また、wget実行時に、~/Downloadsにvelvet_1.2.10.tgzを保存しなかったヒトも、基本的にダメ!



Contents

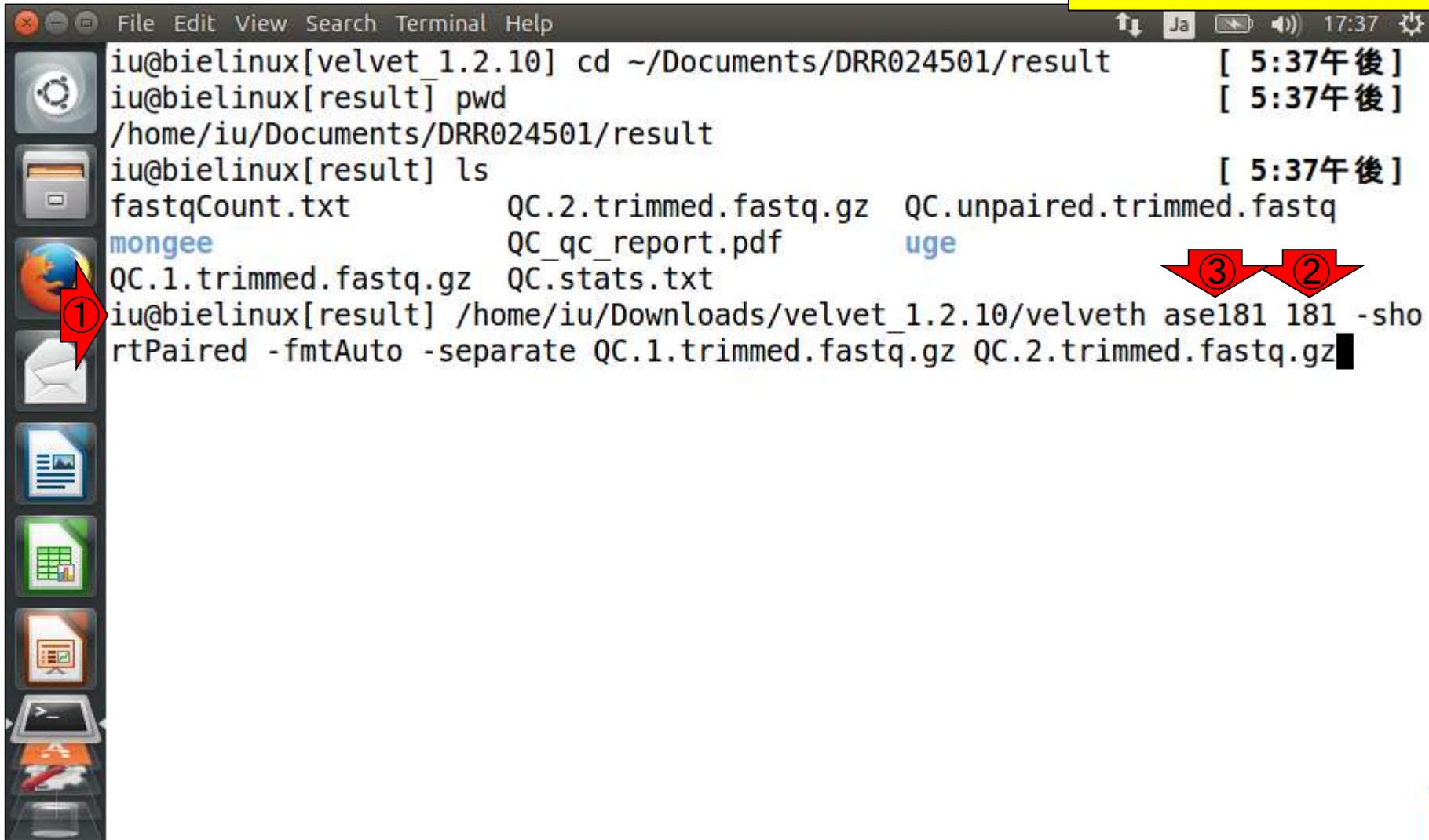
- Illumina HiSeqデータ(トランスクリプトーム)の乳酸菌ゲノムへのマッピング
 - W14: QuasRパッケージを用いたマッピングの事前準備と本番
 - W15: 結果の解説、forward側の100–107bp付近に問題があることを特定
- トリミング、*de novo*トランスクリプトームアセンブリとマッピングの再実行
 - W16: 問題のある領域(forward側の100–107bp)のトリミング
 - W17: トリム後のデータでアセンブリを再実行(Rockhopper2; クラスパス設定関連Tips含む)
 - W18: トリム後のデータでマッピングを再実行(QuasR)
 - W19: トリム前のデータでクオリティチェックを再実行(FastQC)
- Illumina MiSeqデータ(乳酸菌ゲノム)の特徴と前処理
 - W4: FastQC、W5: FaQCs、W6: 再度FastQC
- *de novo*ゲノムアセンブリ
 - W7: Bio-LinuxにプレインストールされているVelvet (ver. 1.2.09)を上限のk=31で実行
 - W8: k=31のアセンブリ結果をRで確認。k=141で実行し、k=31の結果と同じになるのを確認
 - W9: Velvet (ver. 1.2.10)のインストール
 - W10: Velvet (ver. 1.2.10)の実行



W10-1 : Velvet再実行

①絶対パス指定でvelvethを実行
。② k=181で実行し、結果は③
ase181ディレクトリに保存。約2分

```
File Edit View Search Terminal Help
iu@bielinux[velvet_1.2.10] cd ~/Documents/DRR024501/result [ 5:37 午後]
iu@bielinux[result] pwd [ 5:37 午後]
/home/iu/Documents/DRR024501/result
iu@bielinux[result] ls [ 5:37 午後]
fastqCount.txt QC.2.trimmed.fastq.gz QC.unpaired.trimmed.fastq
mongee QC_qc_report.pdf uge
QC.1.trimmed.fastq.gz QC.stats.txt
iu@bielinux[result] /home/iu/Downloads/velvet_1.2.10/velveth ase181 181 -shortPaired -fmtAuto -separate QC.1.trimmed.fastq.gz QC.2.trimmed.fastq.gz
```



W10-1 : Velvet再実行

```
iu@bielinux[result] ls [ 5:37 午後 ]
fastqCount.txt      QC.2.trimmed.fastq.gz  QC.unpaired.trimmed.fastq
mongee              QC_qc_report.pdf       uge
QC.1.trimmed.fastq.gz  QC.stats.txt

iu@bielinux[result] /home/iu/Downloads/velvet_1.2.10/velveth ase181 181 -shortPaired -fmtAuto -separate QC.1.trimmed.fastq.gz QC.2.trimmed.fastq.gz
[0.000001] Reading file 'QC.1.trimmed.fastq.gz' using 'gunzip' as FastQ
[0.014876] Reading file 'QC.2.trimmed.fastq.gz' using 'gunzip' as FastQ
[13.120874] 595266 sequences found in total in the paired sequence files
[13.120919] Done
[13.120974] Reading read set file ase181/Sequences;
[13.265461] 595266 sequences found
[13.905028] Done
[13.905120] 595266 sequences in total.
[13.905188] Writing into roadmap file ase181/Roadmaps...
[15.426667] Inputting sequences...
[15.426712] Inputting sequence 0 / 595266
[34.538723] === Sequences loaded in 19.112059 s
[39.323715] Done inputting sequences
[39.323748] Destroying splay table
[39.431186] Splay table destroyed
iu@bielinux[result] /home/iu/Downloads/velvet_1.2.10/velvetg ase181
```

W10-1 : Velvet再実行

```
File Edit View Terminal Help
[34.742196] Removed 0 null nodes
[34.742200] Concatenation over!
[34.742205] Removing reference contigs with cov...
[34.742221] Concatenation...
[34.742257] Renumbering nodes
[34.742263] Initial node count 305
[34.742272] Removed 0 null nodes
[34.742278] Concatenation over!
[34.745229] Writing contigs into ase181/contigs.fa...
[34.894429] Writing into stats file ase181/stats.txt...
[34.895101] Writing into graph file ase181/LastGraph...
Final graph has 305 nodes and n50 of 32897, max 135797, total 2356728, using
0/595266 reads
iu@bielinux[result] grep -c ">" ase181/contigs.fa [ 5:45午後]
198
iu@bielinux[result] ls -l ase181/contigs.fa [ 5:46午後]
-rw-rw-r-- 1 iu iu 2432842 1月 4 17:45 ase181/contigs.fa
iu@bielinux[result] grep -v ">" ase181/contigs.fa | wc [ 5:46午後]
 39873 39873 2425921
iu@bielinux[result] date [ 5:46午後]
2016年 1月 4日 月曜日 17:47:01 JST
iu@bielinux[result]
```

①得られた配列数は198個。k=31のとき(29,502個; W7-8)と比べて激減していることがわかる。②description行を除いたファイルサイズは2,425,921 bytes。ここから行数分だけの改行コード(39,873 bytes)を差し引いた残りの $2,425,921 - 39,873 = 2,386,048$ bytesが総塩基数、つまりゲノムサイズに相当する(W8-2)。原著論文の実際の数値も約2.4MBとなっており妥当

W10-5: 再挑戦

・再挑戦[W10-5]

一旦アセンブル結果を全て削除し、今度はk=111,121, 131, 151,

ls

(多少話が飛躍するが)計6個のk値で一気にVelvetを実行。赤枠のような一連のコマンドからなるファイルを用意しておき、シェルスクリプト or コピペで実行すると(デフォルトの2GBメモリで)約1時間。しかし、VirtualBox上で4GBメモリに変更すると約10分で終了することが判明したので、設定変更して実行します

```
① /home/iu/Downloads/velvet_1.2.10/velveth hoge_111 111 -shortPaired -fmtf  
/home/iu/Downloads/velvet_1.2.10/velveth hoge_121 121 -shortPaired -fmtf  
/home/iu/Downloads/velvet_1.2.10/velveth hoge_131 131 -shortPaired -fmtf  
/home/iu/Downloads/velvet_1.2.10/velveth hoge_151 151 -shortPaired -fmtf  
/home/iu/Downloads/velvet_1.2.10/velveth hoge_171 171 -shortPaired -fmtf  
/home/iu/Downloads/velvet_1.2.10/velveth hoge_191 191 -shortPaired -fmtf  
② /home/iu/Downloads/velvet_1.2.10/velvetg hoge_111  
/home/iu/Downloads/velvet_1.2.10/velvetg hoge_121  
/home/iu/Downloads/velvet_1.2.10/velvetg hoge_131  
/home/iu/Downloads/velvet_1.2.10/velvetg hoge_151  
/home/iu/Downloads/velvet_1.2.10/velvetg hoge_171  
/home/iu/Downloads/velvet_1.2.10/velvetg hoge_191  
grep -c ">" hoge_*/*contigs.fa  
grep -v ">" hoge_111/*contigs.fa | wc  
grep -v ">" hoge_121/*contigs.fa | wc  
grep -v ">" hoge_131/*contigs.fa | wc  
grep -v ">" hoge_151/*contigs.fa | wc  
grep -v ">" hoge_171/*contigs.fa | wc  
grep -v ">" hoge_191/*contigs.fa | wc
```



使用メモリ増加

(Rで)塩基配列解析

～NGS、RNA-seq、ゲノム、トランскриプトーム、正規化、発現変動、統計、モラ
(last modified 2016/06/03, since 2011)

What's new
このウェブリソース
法(Win
本山圭子)

- 基本的な利用法 (last modified 2015/04/03)
- サンプルデータ (last modified 2016/05/13)
- バイオインフォマティクス人材育成カリキュラム(次世代シーケンサ) | NGSハンズオン講習会2016 (last modified 2016/07/19)
- バイオインフォマティクス人材育成カリキュラム(次世代シーケンサ) | NGSハンズオン講習会2015 (last modified 2015/06/23)
- バイオインフォマティクス人材育成カリキュラム(次世代シーケンサ) | 速習コース2014 (last modified 2015/02/28)



バイオインフォマティクス人材育成カリキュラム(次世代シーケンサ) | NGSハンズオン講習会2016

平成28年度NGSハンズオン講習会の実習で用いるリンク先、データファイル、コピペ用コード集などはここで示します。

- はじめに(講習会参加者必読) (last modified 2016/07/04)
- 事前準備 | Bio-Linux 8とRのインストール状況確認(2016.07.19) (last modified 2016/07/04)
- 第1部 | 統計解析 | について (last modified 2016/06/24)
 - 第1部 | 統計解析 | ゲノム 解析、塩基配列解析(2016.07.20) (last modified 2016/06/23)
 - 第1部 | 統計解析 | トランスクリプトーム 解析1(2016.07.21) (last modified 2016/06/24)
 - 第1部 | 統計解析 | トランスクリプトーム 解析2(2016.07.22) (last modified 2016/06/23)
- 第2部 | NGS解析(初～中級) | について (last modified 2016/06/16)
 - 第2部 | NGS解析(初～中級) | NGS解析基礎(2016.07.25) (last modified 2016/04/26)
 - 第2部 | NGS解析(初～中級) | ゲノム Reseq、変異解析(2016.07.26) (last modified 2016/04/26)
 - 第2部 | NGS解析(初～中級) | RNA-seq(2016.07.27) (last modified 2016/04/26)
 - 第2部 | NGS解析(初～中級) | ChIP-seq(2016.07.28) (last modified 2016/04/26)
- 第3部 | NGS解析(中～上級) | について (last modified 2016/07/04)
 - 第3部 | NGS解析(中～上級) | Linux環境でのデータ解析: JavaやRの利用法(2016.08.01) (last modified 2016/06/13)
 - 第3部 | NGS解析(中～上級) | Linux環境でのデータ解析: マッピング、トリミング、アセンブリ(2016.08.02) (last modified 2016/06/21)
 - 第3部 | NGS解析(中～上級) | クラウド環境との連携、ロングリードデータの解析(2016.08.03) (last modified 2016/06/20)
 - 第3部 | NGS解析(中～上級) | トランスクリプトームアセンブリ、発現量推定(2016.08.04) (last modified 2016/07/04)



使用メモリ増加

事前準備 | Bio-Linux 8とRのインストール状況確認(2016.07.19)

作成途中です。基本的に昨年度と同じく自習です。スタッフが複数人常駐予定ですので、インストールで躊躇した箇所の相談など個別対応してもらってください。

1. Bio-Linux8(第2部および3部で利用するovaファイル)の導入確認

第2部および3部で利用するovaファイルは6GBから10GB程度になります。そしてそれを使って第2部および3部を受講してもらいます。ovaファイルは独立に提供(URLをお知らせする)予定ですので、例えば全日程参加者は、第2部用ovaファイルと第3部用ovaファイルを独立にダウンロードしておいてください。なるべく早い段階でovaファイルをダウンロード可能な状態にする予定ですが....ネットワーク環境的にダウンロードできないヒトも一定数いると思われます。当日は、それらの人々用に第2部および3部のovaファイルが入ったUSBメモリを講義室で用意する予定ですので、下記を参考にして各自のPCにコピーしてBio-Linuxを導入してください(もちろん乳酸菌NGS連載第2回最後の「1. VirtualBox、および2. Extension Pack」のインストールが完了しているという前提です)。

- ovaファイルの導入手順: [Windows用](#)(2015.12.28版; 約2MB)
- ovaファイルの導入手順: [Macintosh用](#)(2015.12.28版; 約2MB)

2. 共有フォルダ設定完了確認

第2部および第3部参加者のみ。

3. 基本的なLinuxコマンドの習得状況確認

第2部および第3部参加者のみ。

4. R本体およびパッケージのインストール確認

基本的に第1部参加者のみでよい。

5. 講師指定の事前予習内容の再確認

- [第1部 | 統計解析 |について](#) (last modified 2016/06/20)
- [第2部 | NGS解析\(初～中級\) |について](#) (last modified 2016/06/16)
- [第3部 | NGS解析\(中～上級\) |について](#) (last modified 2016/07/04)

6. 講習会期間中に貸与されるノートPCを用いた各種動作確認

基本的に貸与希望者のみではありますが、持込PCの不具合時にはアグリバイオPCを貸し出します。この際に普段利用するPC環境でないので戸惑うかもしれません。この日を利用して貸与PCに慣れておくというのもアリでしょう。

7. 無線LANの設定(持ち込みPCのみ)

会場で利用可能なアクセスポイントの把握とパスワードの設定を行ってください(PWは当日スタッフから教えてもらってください)。



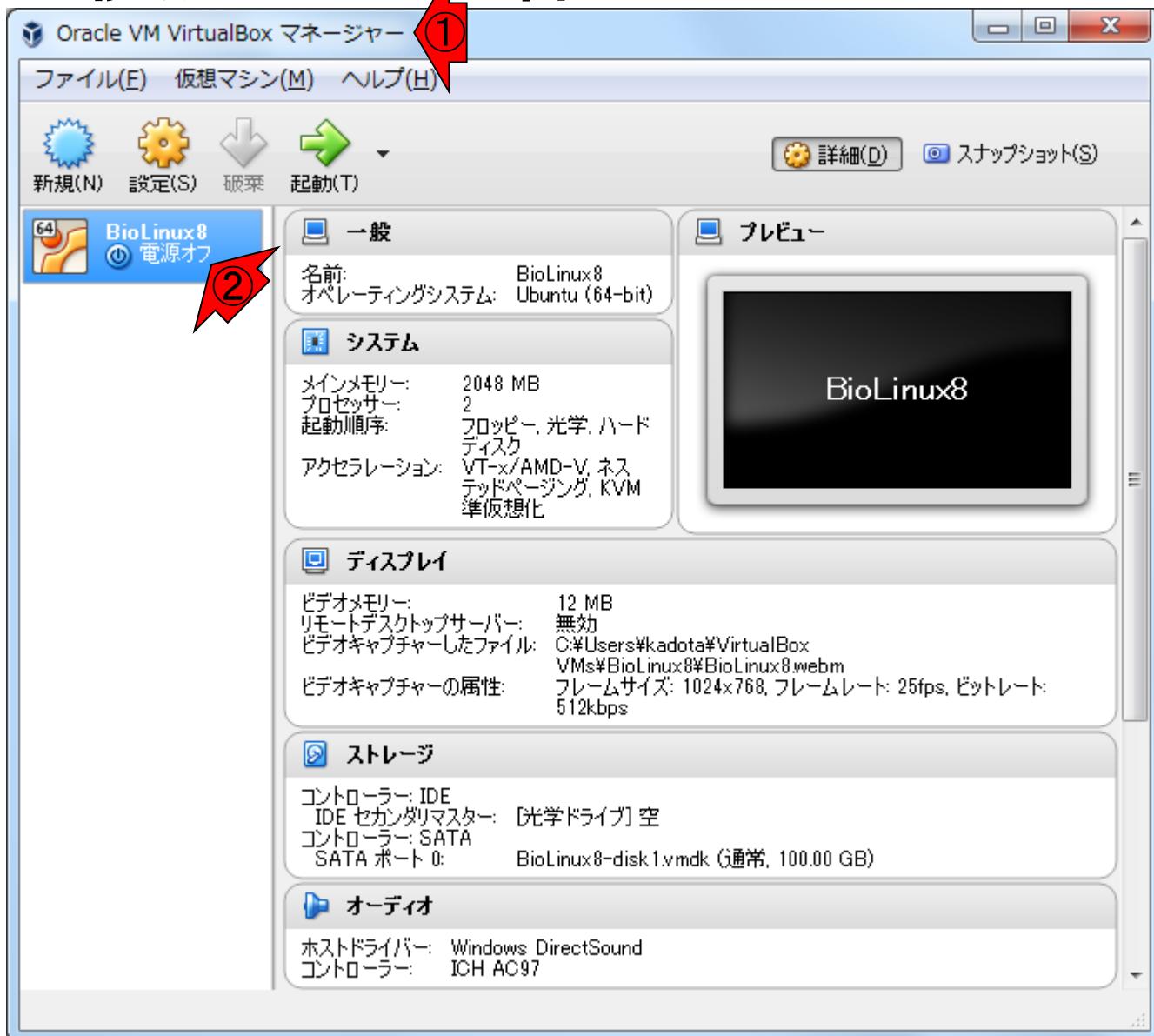
8. [VirtualBox関連Tips](#)(2016.07.04版)

下記内容が含まれます。

- BioLinux8の名前を変えたい
- 使用メモリを変更したい
- ...

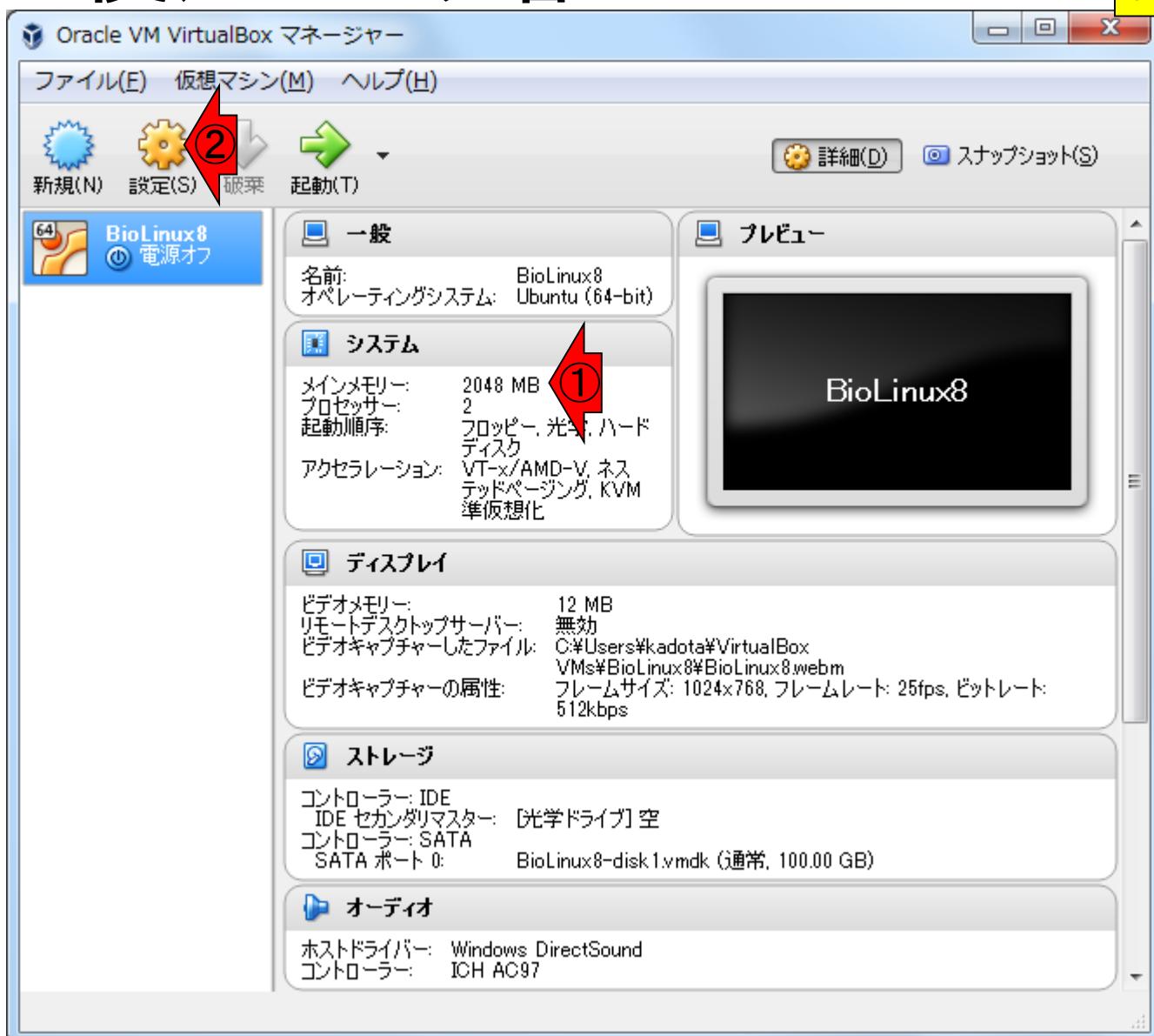
使用メモリ増加

一旦BioLinux8を保存せずに終了。すると、①VirtualBox上では、②電源オフになっているはず



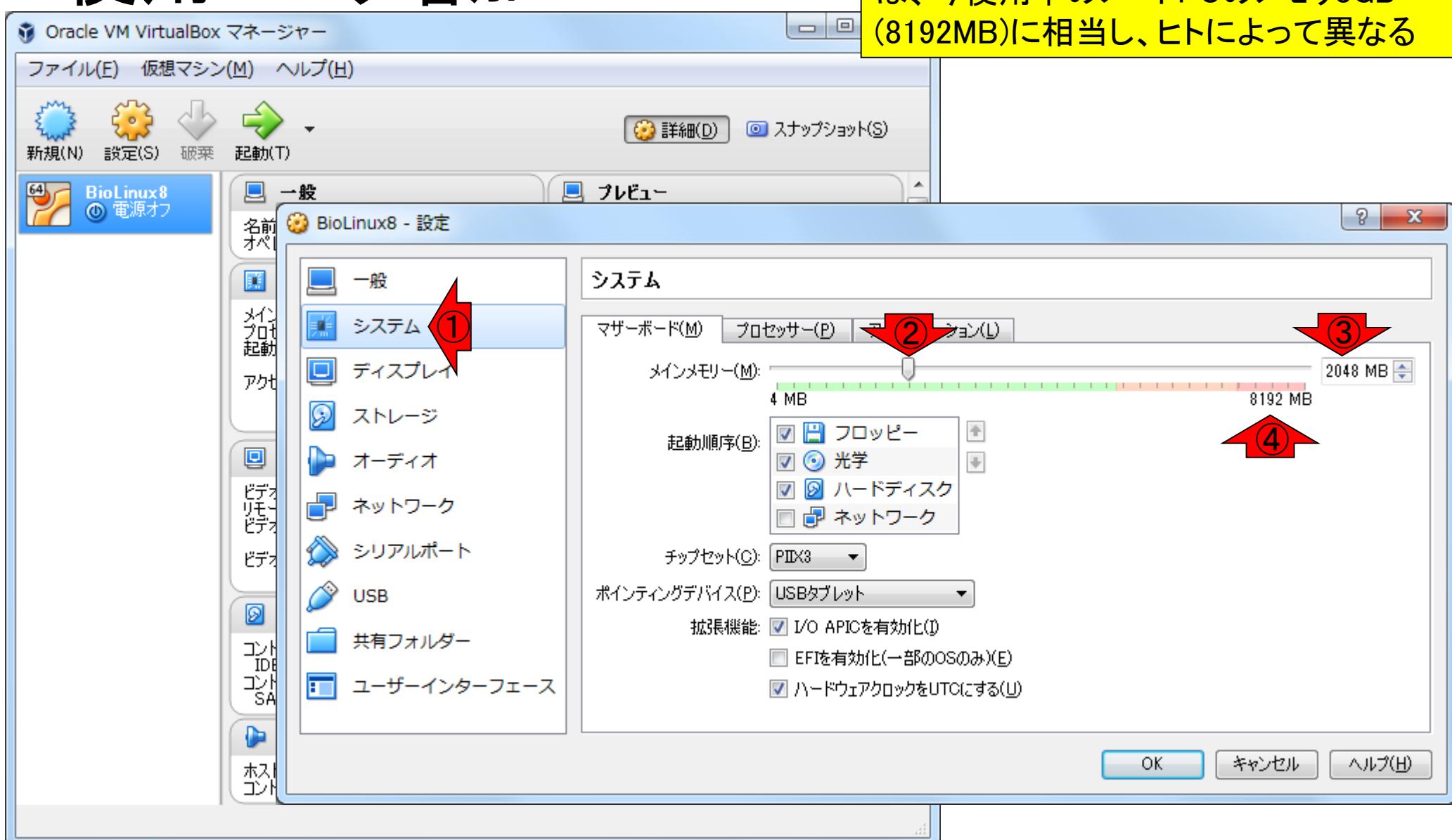
使用メモリ増加

①現在は2GB (2048MB) 割り当てている状況です。これを4GB (4096MB) にするやり方を示します。②設定



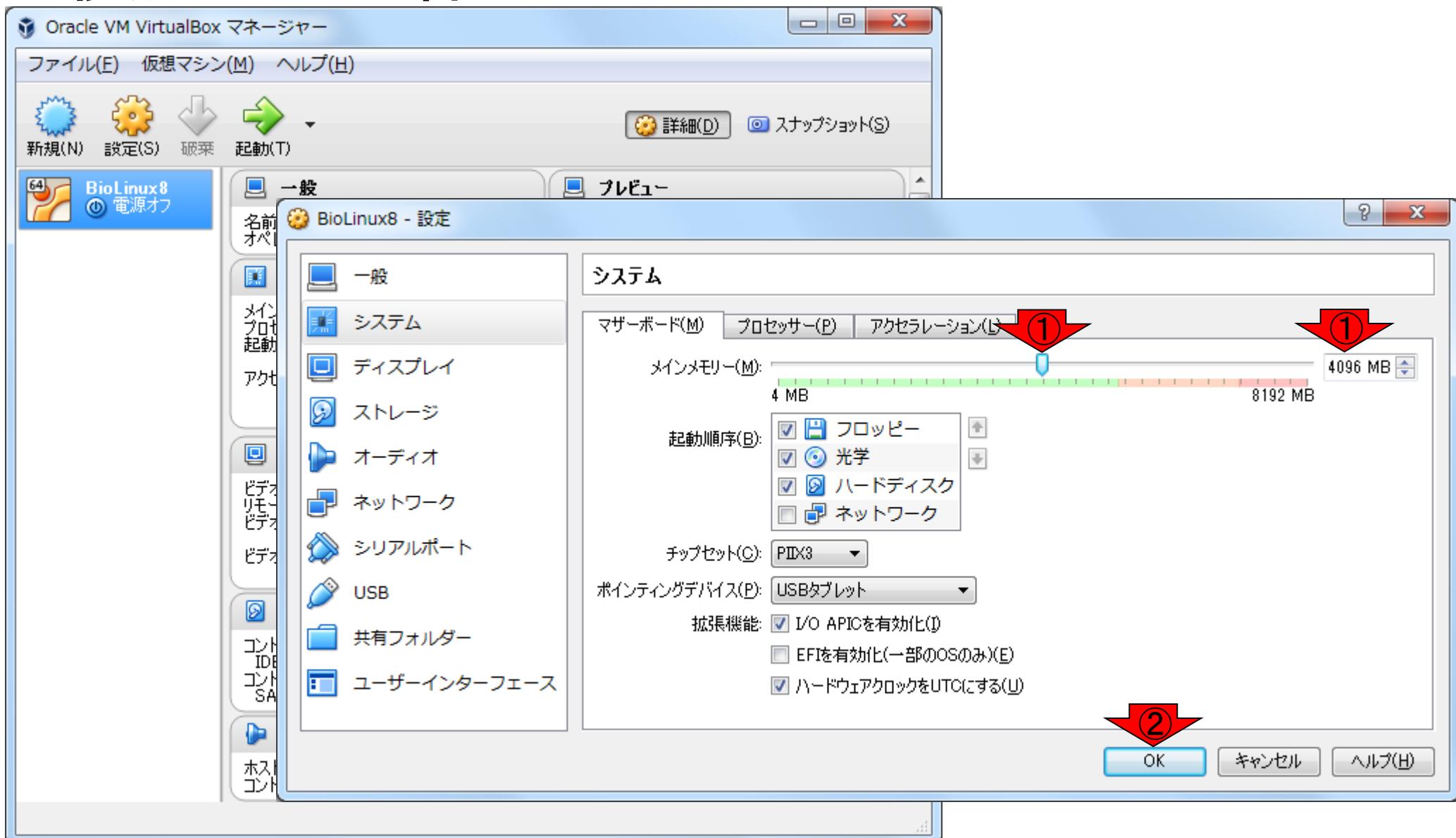
使用メモリ増加

①システム。②現在はここ。具体的な数値は③に表示されている。④この数値は、今使用中のノートPCのメモリ8GB(8192MB)に相当し、ヒトによって異なる



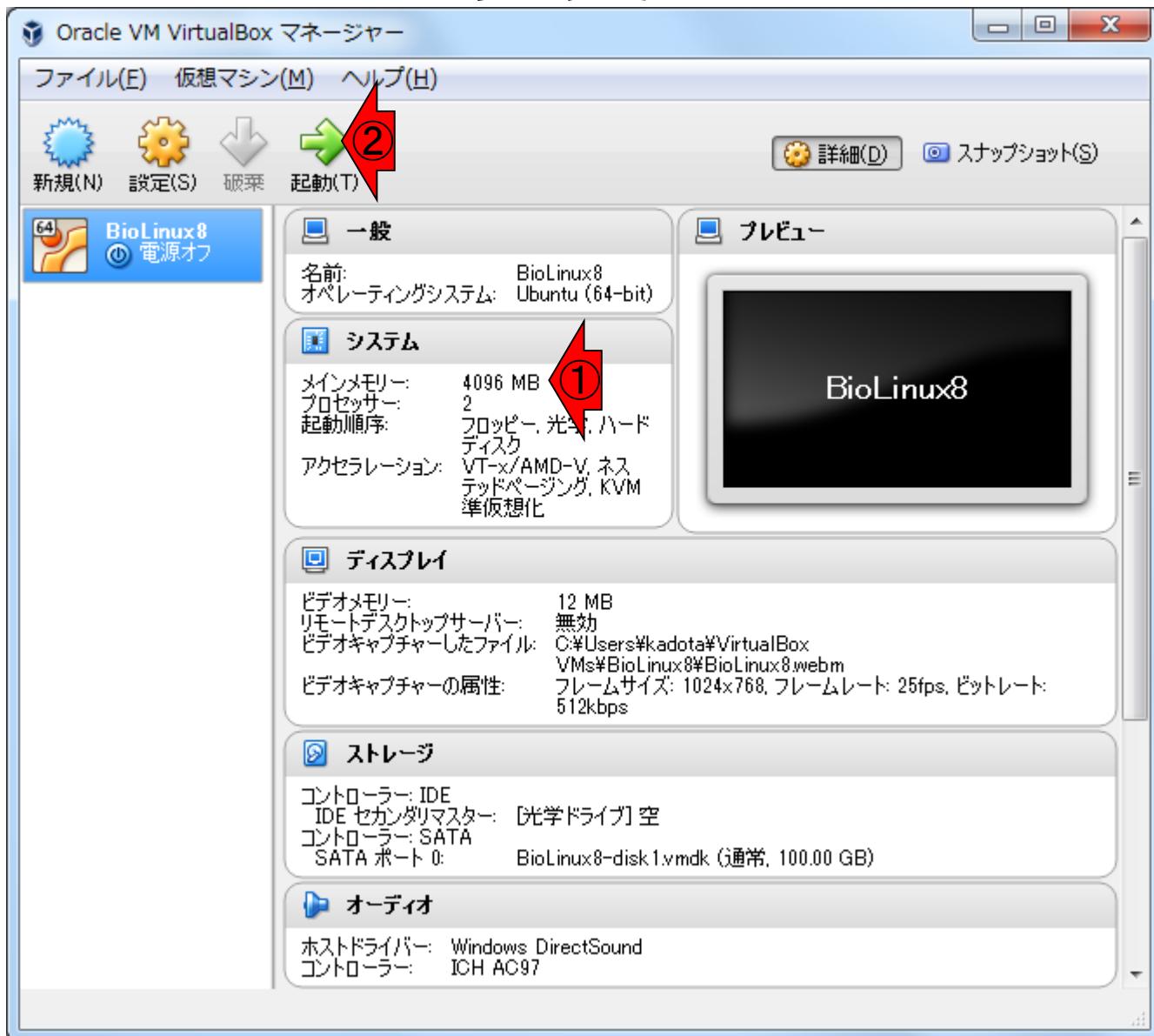
①メモリ4GBに変更して、②OK

使用メモリ増加



メモリ4GB変更後の状態

①メモリ4GBに確かに変更されたことが分かる。②起動して



W10-5: 再挑戦

- ### • 再挑戦[W10-5]

一旦アセンブル結果を全て削除し、今度は $k=111, 121, 131, 151$

```
cd ~/Documents/DRR024501/result
```

pwd

1s

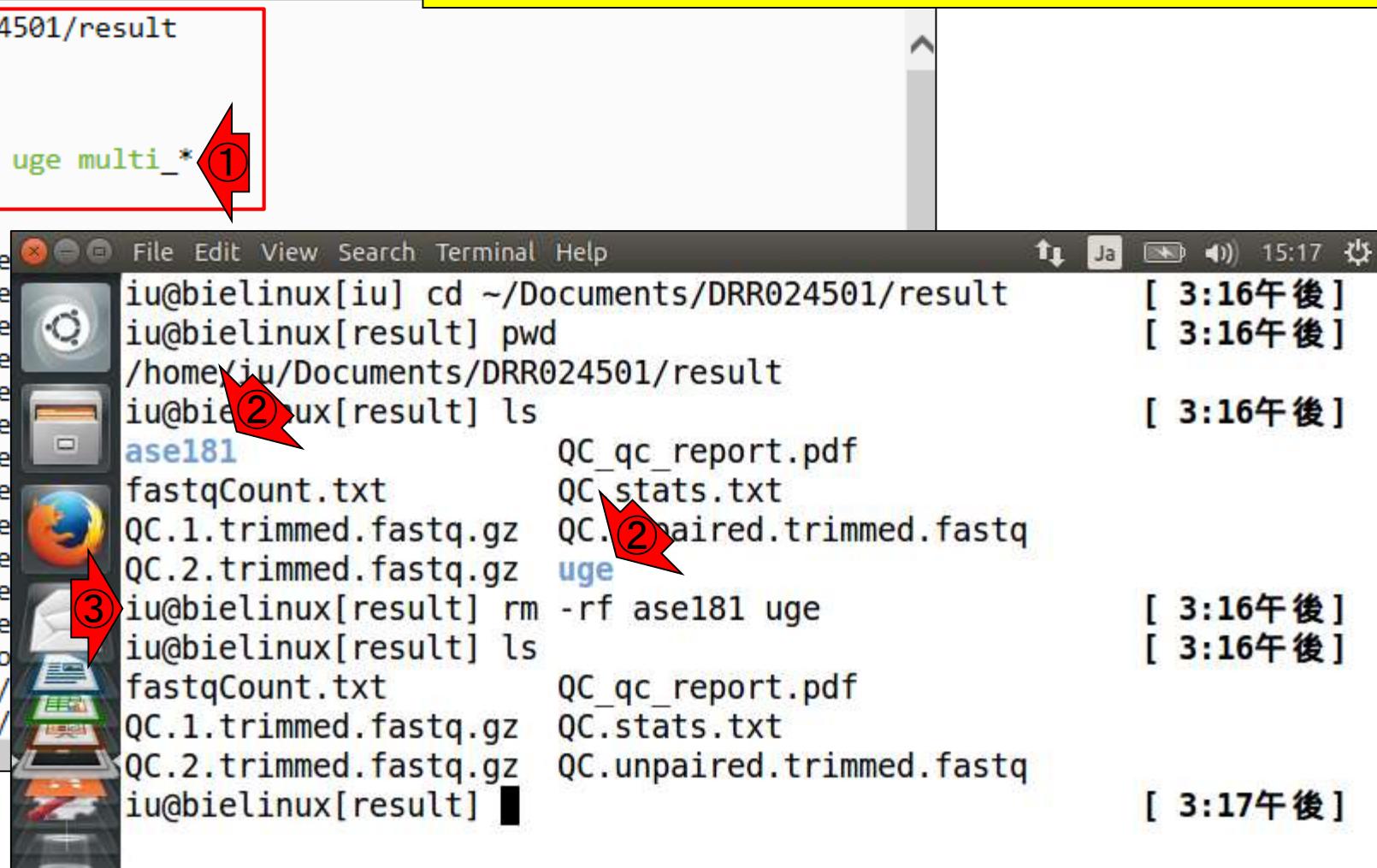
```
rm -rf ase181 mongee uge multi *
```

15

```
/home/iu/Downloads/ve  
grep -c ">" hoge_*/co  
grep -v ">" hoge_111/  
grep -v ">" hoge_121/
```

```
File Edit View Search Terminal Help
iu@bielinux[iu] cd ~/Documents/DRR024501/result
iu@bielinux[result] pwd
/home/ju/Documents/DRR024501/result
iu@bielinux[result] ls
ase181          QC_qc_report.pdf
fastqCount.txt  QC.stats.txt
QC.1.trimmed.fastq.gz  QC.2.unpaired.trimmed.fastq
QC.2.trimmed.fastq.gz  uge
iu@bielinux[result] rm -rf ase181 uge
iu@bielinux[result] ls
fastqCount.txt  QC_qc_report.pdf
QC.1.trimmed.fastq.gz  QC.stats.txt
QC.2.trimmed.fastq.gz  QC.unpaired.trimmed.fastq
iu@bielinux[result]
```

赤枠の前処理部分の注意。オリジナルのウェブ資料通りに赤枠部分をコピペしても、講習会では(mongeeやmulti_*というディレクトリを作成していないので)①のところでエラーが出て②を消せない。そのため、③のようにase181とugeディレクトリのみの削除してください



W10-5:再挑戦

- 再挑戦[W10-5]

一旦アセンブル結果を全て削除し、今度はk=111,121, 131, 151,171, 191でやってみる。

```
ls
```

```
/home/iu/Downloads/velvet_1.2.10/velveth hoge_111 111 -shortPaired -fmtf
/home/iu/Downloads/velvet_1.2.10/velveth hoge_121 121 -shortPaired -fmtf
/home/iu/Downloads/velvet_1.2.10/velveth hoge_131 131 -shortPaired -fmtf
/home/iu/Downloads/velvet_1.2.10/velveth hoge_151 151 -shortPaired -fmtf
/home/iu/Downloads/velvet_1.2.10/velveth hoge_171 171 -shortPaired -fmtf
/home/iu/Downloads/velvet_1.2.10/velveth hoge_191 191 -shortPaired -fmtf
/home/iu/Downloads/velvet_1.2.10/velvetg hoge_111
/home/iu/Downloads/velvet_1.2.10/velvetg hoge_121
/home/iu/Downloads/velvet_1.2.10/velvetg hoge_131
/home/iu/Downloads/velvet_1.2.10/velvetg hoge_151
/home/iu/Downloads/velvet_1.2.10/velvetg hoge_171
/home/iu/Downloads/velvet_1.2.10/velvetg hoge_191
grep -c ">" hoge_*/contigs.fa
grep -v ">" hoge_111/contigs.fa | wc
grep -v ">" hoge_121/contigs.fa | wc
grep -v ">" hoge_131/contigs.fa | wc
grep -v ">" hoge_151/contigs.fa | wc
grep -v ">" hoge_171/contigs.fa | wc
grep -v ">" hoge_191/contigs.fa | wc
```

15:17

/result [3:16 午後]
[3:16 午後]
[3:16 午後]
ed.fastq [3:16 午後]
[3:16 午後]
iu@bielinux[result] [3:17 午後]

W10-5:再挑戦

①kの値を大きくすると、配列数(コンティグ数)は減少傾向となる。説明はオリジナルのウェブ資料と異なります。これは、W10-2からW10-4までを省略していることに起因します

```
iu@bielinux[result] grep -c ">" hoge_*/contigs.fa [ 2:05午後]
hoge_111/contigs.fa:23761
hoge_121/contigs.fa:15776
hoge_131/contigs.fa:8398
hoge_151/contigs.fa:1306
hoge_171/contigs.fa:168
hoge_191/contigs.fa:336
iu@bielinux[result] grep -v ">" hoge_111/contigs.fa | wc
 103499 103499 5821703
iu@bielinux[result] grep -v ">" hoge_121/contigs.fa | wc
 91756 91756 4781900
iu@bielinux[result] grep -v ">" hoge_131/contigs.fa | wc
 66837 66837 3777666
iu@bielinux[result] grep -v ">" hoge_151/contigs.fa | wc
 44339 44339 2643716
iu@bielinux[result] grep -v ">" hoge_171/contigs.fa | wc
 39769 39769 2421292
iu@bielinux[result] grep -v ">" hoge_191/contigs.fa | wc
 40263 40263 2445694
iu@bielinux[result] [ 2:05午後]
```

①

W10-5:再挑戦

①ここは大まかなゲノムサイズ(総塩基数)を調べている(W8-2)。赤下線が該当部分。実際のゲノムサイズより、改行コード分だけわずかに大きな値になる

```
File Edit View Search Terminal Help ↑ ↓ Ja 14:14 [2:05午後]
iu@bielinux[result] grep -c ">" hoge_*/contigs.fa [ 2:05午後]
hoge_111/contigs.fa:23761
hoge_121/contigs.fa:15776
hoge_131/contigs.fa:8398
hoge_151/contigs.fa:1306
hoge_171/contigs.fa:168
hoge_191/contigs.fa:336
iu@bielinux[result] grep -v ">" hoge_111/contigs.fa | wc
 103499 103499 5821703
iu@bielinux[result] grep -v ">" hoge_121/contigs.fa | wc
 91756 91756 4781900
iu@bielinux[result] grep -v ">" hoge_131/contigs.fa | wc
 66837 66837 3777666
iu@bielinux[result] grep -v ">" hoge_151/contigs.fa | wc
 44339 44339 2643716
iu@bielinux[result] grep -v ">" hoge_171/contigs.fa | wc
 39769 39769 2421292
iu@bielinux[result] grep -v ">" hoge_191/contigs.fa | wc
 40263 40263 2445694
iu@bielinux[result] [ 2:05午後]
```

①

W10-6:これまでのまとめ

k-mer	コンティグ数	総塩基数	ウェブ資料
31	29502	4077679	W10-4
61	15445	3886574	W10-4
91	8583	3412266	W10-4
111	23761	5718204	W10-5
121	15776	4690144	W10-5
131	8398	3710829	W10-5
151	1306	2599377	W10-5
171	168	2381523	W10-5
181	198	2386048	W10-1
191	336	2405431	W10-5

Velvetの場合は、①k値はリード長Lの2/3程度がよいのかもと学習する(実際には、ゲノムサイズやリード数に大きく依存する)。オリジナルのリード数(約300万×2)の1/10なので、1時間程度で概要を把握可能



第6回原稿PDFのp45

尚、このデータの正解は、配列数が3 (1 chromosome + 2 plasmids)、2,400,586 bp (約 2.4MB) である⁴⁾。k値の選択の重要性がよくわかる例といえよう。

通常、Velvetを実行する場合は複数の異なるk値を用いてアセンブルを行い、それらの結果を眺める[W10]。ここでは、計10個のk値 (k=31, 61, 91, 111, 121, 131, 151, 171, 181, 191) で実行した結果を眺め、主に配列数の観点から、k=171周辺の結果が一番よさそうだと解釈する。もちろんこのデータの場合は、「真のゲノムサイズは約2.4MB」だという答えがわかった状態でアセンブリ結果の評価を行っていることになるが、実際には近縁種との比較により妥当と考えられるゲノムサイズを検討する。ここではそのような情報が得られなかつたと仮定して「ゲノムサイズ推定」を行い、アセンブリ結果の評価を行う。

ゲノムサイズ推定

ゲノムサイズの推定は、フローサイトメトリー (flow cytometry) という手法を用いて実験的に求めるやり方

一般に、Velvetを利用する場合は、複数のk値でアセンブリを行う。主観でk=171がいいと判断したところまでが今日の内容で、原稿の①のあたり。2016年08月03日(2016.08.03)の内容の予告。もちろん客観的に最もよいと思われるk値を出力してくれるプログラムも存在する。その1つであるKmerGenieのインストールから、ゲノムサイズ推定ツールとしての利用へと展開していく。明日まで「待てっ!」

①

