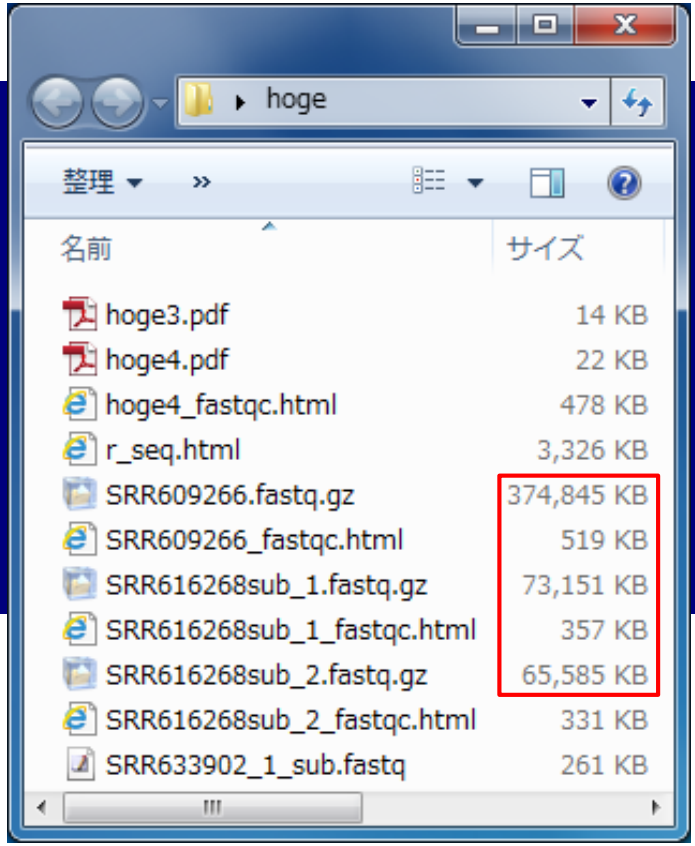
USBメモリ中のhogeフォルダをデスクトップにコピーしておいてください。コピー後のファイルサイズが同じになっているかもチェックしてください。



前回（6/16）のhogeフォルダがデスクトップに残っているかもしれないのでご注意ください。

農学生命情報科学特論I 第2回

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/

講義予定

NGSの普及により、以前は主にゲノム解析系で必要とされていた配列解析のためのスキルがトランスクリプトーム解析においても要求される時代になっています。本科目では、様々な局面で応用可能な配列解析系のスキルアップを目指し、RNA-Seqに基づくトランスクリプトーム解析を題材とした講義を行います。

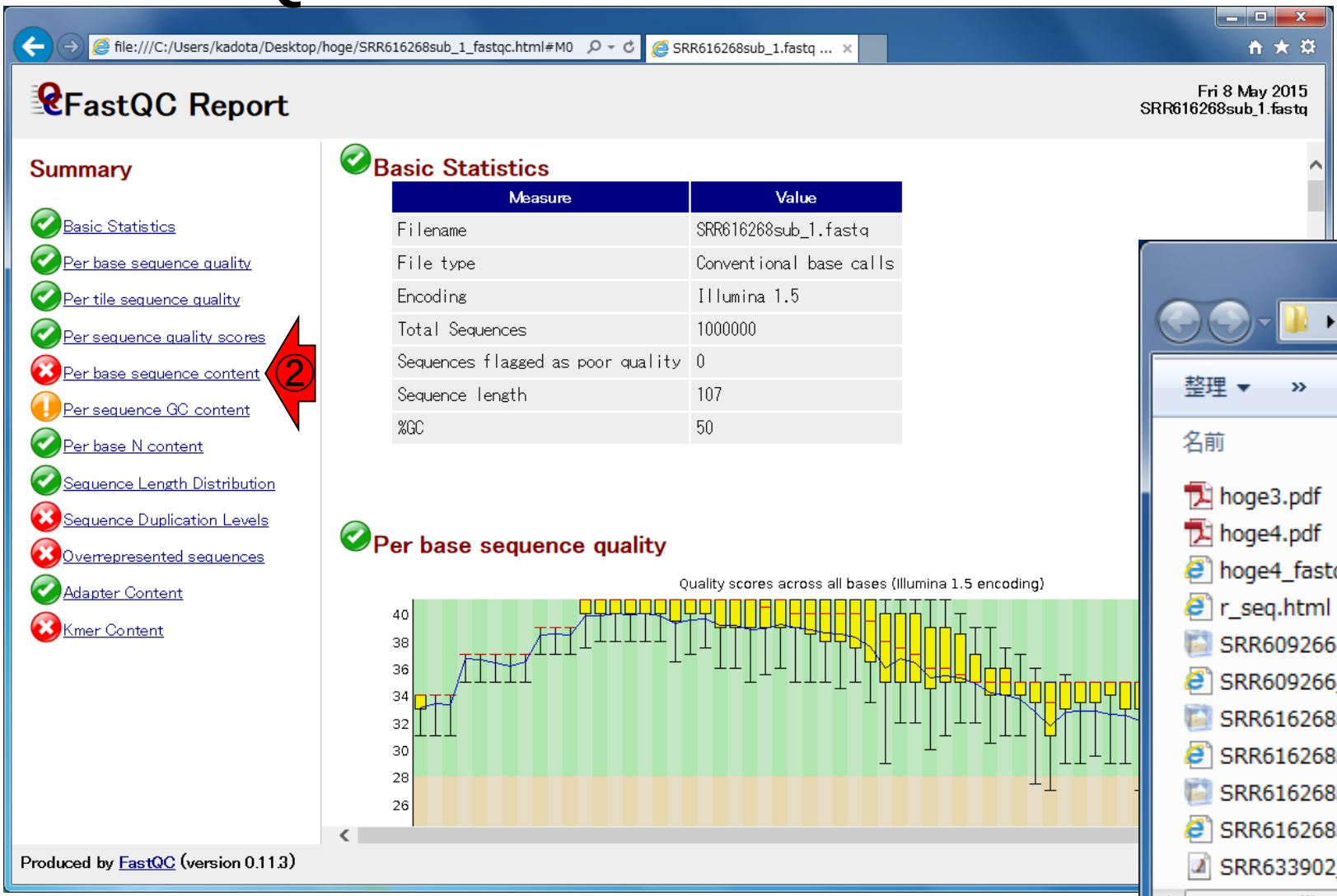
- 第1回（2015年6月16日）
  - データベース、データ取得、ファイル形式、Quality Control
  - 教科書の1.3節周辺
- 第2回（2015年6月23日）
  - Quality Control、k-mer解析、トリミング（アダプター配列除去）
- 第3回（2015年6月30日）
  - アセンブル、マッピング、カウント情報取得、クラスタリング
  - 教科書の2.3節周辺
- 第4回（2015年7月7日）
  - データ正規化、実験デザイン、分布（モデル）、発現変動解析
  - 教科書の3.3節周辺

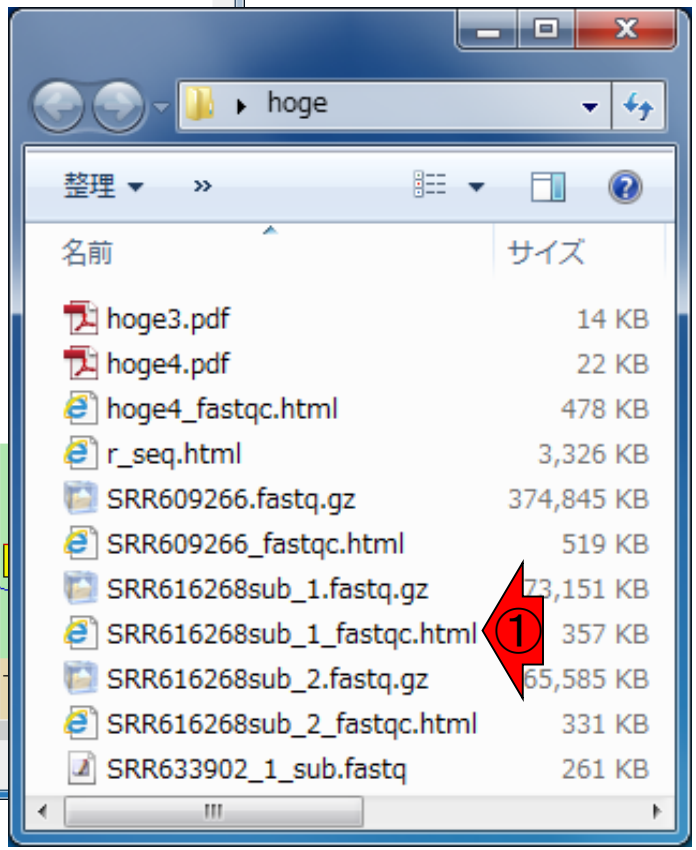
Contents

- QC(Quality Control)
  - Quality Check (FastQC):乳酸菌RNA-seqデータ
    - k-mer解析、課題1
  - Quality Check (FastQC):カイコsRNA-seqデータ
  - トリミング:カイコsRNA-seqデータ
    - 全体像を把握
    - アダプター配列除去(QuasRパッケージ)、課題2
    - Rを駆使して結果を確認
  - トリミング:乳酸菌RNA-seqデータ
    - サブセットの抽出
    - paired-endデータのアダプター配列除去(QuasRパッケージ)

FastQC結果を眺める

乳酸菌RNA-seqデータ(SRR616268sub_1.fastq)のFastQC実行結果ファイルの解説の続き





FastQC結果を眺め

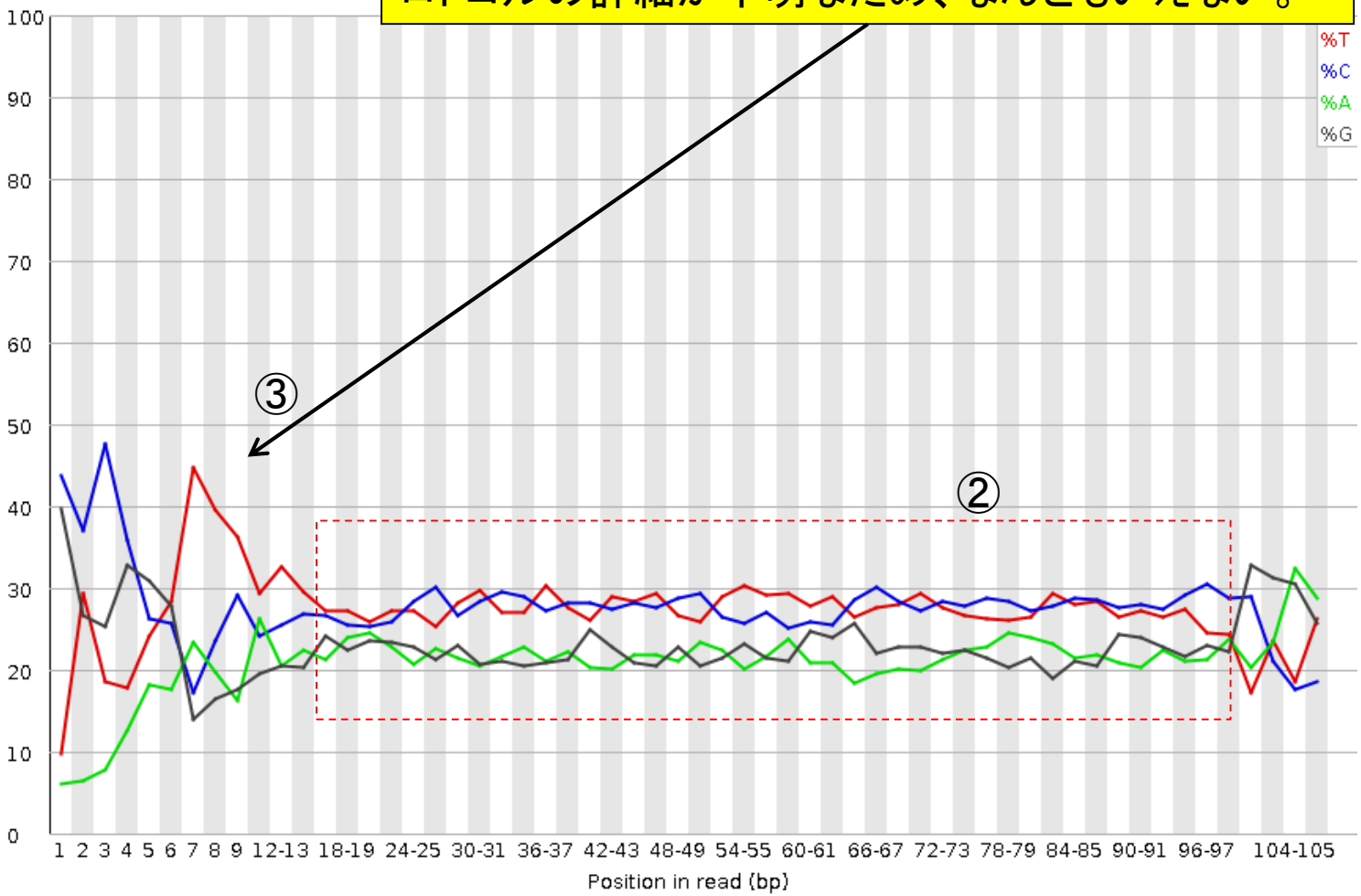
①ポジションごとの塩基の出現確率。②赤枠のような塩基ごとのプロファイルがフラットになっていれば基本的にはOK。③-15番目あたりのプロファイルは、実験プロトコル（oligo dT primer or random hexamer；DNase Iを利用した破砕）に由来するのかもしれない。このデータは、原著論文がなく実験プロトコルの詳細が不明なため、なんともいえない。

FastQC Report

Summary

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

Per base sequence cont



FastQC結果を眺める

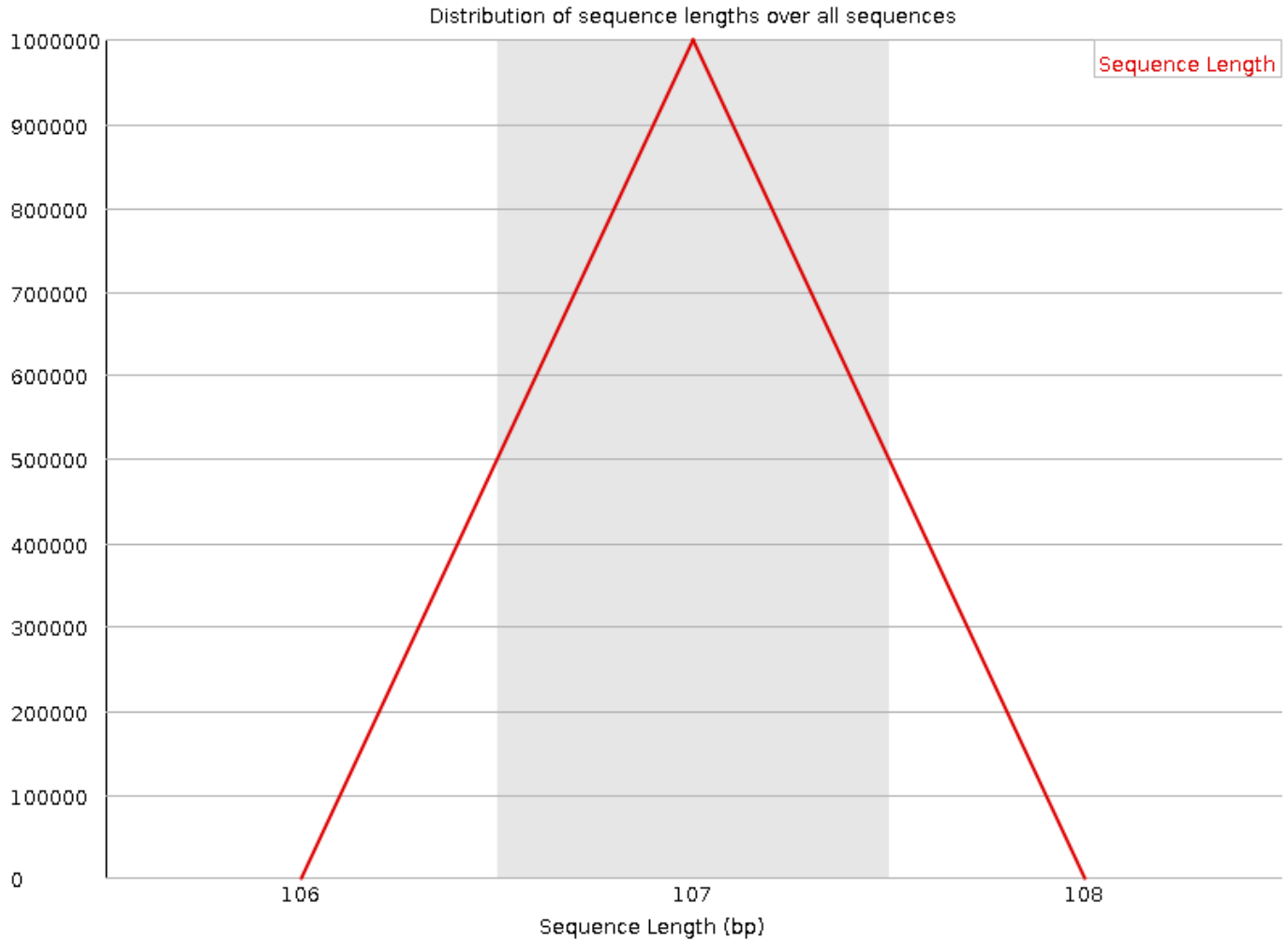
①全部で100万リードのうち、全てが107 bpだということがわかる。PacBioのデータやアダプター配列除去後のデータを入力とすればヒストグラムのようになるだろう

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



FastQC結果を眺める

①よく分かりません(爆)。**発展課題**:これについて説明せよw

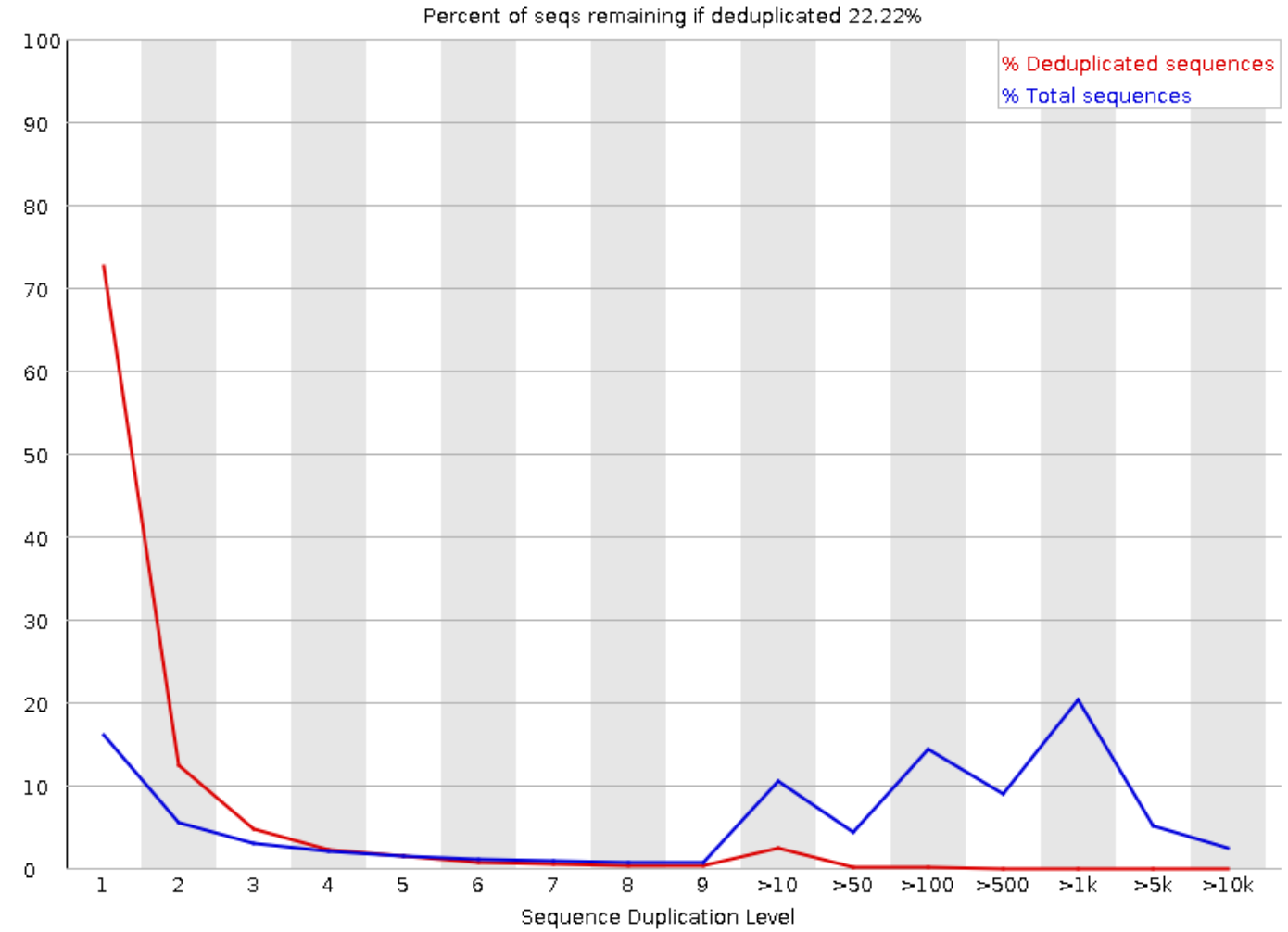
FastQC Report

Fri
SRR616268

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 ①
- Overrepresented sequences
- Adapter Content
- Kmer Content

Sequence Duplication Levels



FastQC結果を眺める

①頻出する配列をリストアップ。②トップは「CCCCGGTATA…」という50塩基の配列で14,383回出現。Percentageは1.4383%。全部で100万リードなので妥当。オリジナル107 bpのうち最初の50 bpで解析している。

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 ①
- Adapter Content
- Kmer Content

Overrepresented sequences

Sequence	Count	Percentage	Possible Source
CCCCGGTATATTTTCGGCGCAGTGCCACTCGACTAGTGAGCTATTACGCA	14383	1.4383	No Hit ②
GGCCTATTCACTGCGGCTGACCTTGCGGTCAGCACCCCTTCTTCCGAAGT	11044	1.1044	No Hit
GTGCTTTTCACCTTTCCCTCACGGTACTGGTTCACTATCGGTCACTAGGG	8892	0.8892000000000001	No Hit
CCCGGTATATTTTCGGCGCAGTGCCACTCGACTAGTGAGCTATTACGCAC	8474	0.8474	No Hit
GTCACTAGGGAGTATTTAGCCTTGGGAGATGGTCCTCCCGGATTCCGACG	8189	0.8189	No Hit
GCCGGCATTCTCACTTCTAAGCGCTCCAGCCGTCCTCACGATCGACCTTC	8132	0.8132	No Hit
GTCCAGTCCTACAACCCCGAGAAGCAAGCTTCTCGGTTTGGGCTCTTCCC	6663	0.6663	No Hit
GTCGGTTTGCGGTACGGGTAGTTTATTTCTCACTAGAAGCTTTTCTTGGC	6411	0.6411	No Hit
GGTCACTTGGTTTCGGGTCTACATCTGCTTACTCATTCGCCCTGTTCAGA	5502	0.5502	No Hit
GCCGGCATTCTCACTTCTAAGCGCTCCAGCCGTCCTCACGATCAACCTTC	4845	0.48450000000000004	No Hit
CCCTCCATCGCTTAAACAAAATAAACTAGTGCAGGAATCTCAACCTGCTT	4395	0.43949999999999995	No Hit
CCGGTATATTTTCGGCGCAGTGCCACTCGACTAGTGAGCTATTACGCACT	4385	0.4385	No Hit
CCCGCGTCTGCCGCCGGCCAGCTATGTATTCACTGACAAGCAATACACTG	4366	0.4366	No Hit
CCACAGTTTCGGTATTATGCTTAGCCCCGGTATATTTTCGGCGCAGTGCC	4314	0.4314	No Hit
CTGGGCTGTTCCCCTTTCGACAATGGACCTTATCGCTCACTGTCTGACTC	4113	0.41130000000000005	No Hit
CCGCCGTACTCAGGATCCTGGACGGAGGGTTCGACGTTTCGCTTACAGGG	4081	0.4081	No Hit
CCGGCATTCTCACTTCTAAGCGCTCCAGCCGTCCTCACGATCGACCTTCA	3846	0.3846	No Hit
GTAGGTCACTTGGTTTCGGGTCTACATCTGCTTACTCATTCGCCCTGTTC	3823	0.3823	No Hit

FastQC結果を眺める

①頻出する配列をリストアップ。②ときどき既知のアダプター配列とマッチするものが見つかる。2,415回出現した「GATCGGAA…」という50塩基の配列は、TruSeq Adapter Index 3というものと100%一致していたことを示す。このようなリード中の部分配列はトリムされるべき。

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 ①
- Adapter Content
- Kmer Content

GTTCACTATCGGTCACTAGGGAGTATTTAGCCTTGGGAGATGGTCCTCC			
GCCCCGGTATATTTTCGGCGCAGTGCCACTCGACTAGTGAGCTATTACG			
GCCCTGTTCAGACTCGCTTTCGCTACGGCTCCGACTTTTCATCTTAACC			
GCCGGTTCATTCTACAAAAGGCACGCCATTACCCGTTAACGGGCTTTGA			
GCCACATCCTTTTCCACTTAGCATAAATTTAGGGACCTTAACTGGTGATC	2486	0.2486	No Hit
CCTCCATCGCTTAAACAAAATAAACTAGTGCAGGAATCTCAACCTGCTTG	2451	0.2451	No Hit
CTTGGGAGATGGTCCTCCCGGATTCCGACGGAATTTCACGTGTTCCGCCG	2438	0.24380000000000002	No Hit
CTTGGTTTCGGGTCTACATCTGCTTACTCATTCGCCCTGTTCAGACTCGC	2430	0.243	No Hit
GTCATGGGTAGGTCACTTGGTTTCGGGTCTACATCTGCTTACTCATTCGC	2429	0.2429	No Hit
CTAGGGAGTATTTAGCCTTGGGAGATGGTCCTCCCGGATTCCGACGGAAT	2421	0.2421	No Hit ②
GATCGGAAGAGCACACGTCTGAACTCCAGTCACTTAGGCATCTCGTATGC	2415	0.2415	TruSeq Adapter, Index 3 (100% over 50bp)
CCGGCATTCTCACTTCTAAGCGCTCCAGCCGTCCTCACGATCAACCTTCA	2393	0.2393	No Hit
GGGAGTATTTAGCCTTGGGAGATGGTCCTCCCGGATTCCGACGGAATTTC	2365	0.2365	No Hit
TGGGCCTATTCACTGCGGCTGACCTTGCGGTCAGCACCCCTTCTTCCGAA	2332	0.2332	No Hit
CTCCTGCGTCCCTCCATCGCTTAAACAAAATAAACTAGTGCAGGAATCTC	2331	0.23310000000000003	No Hit
CGCCGCTACTATGGGAATCGAGTTTTCTTTCTCTTCCTGCGGGTACTGAG	2255	0.2255	No Hit
CGCCGGTTCATTCTACAAAAGGCACGCCATTACCCGTTAACGGGCTTTGA	2172	0.21719999999999998	No Hit
CCGCCGGCCAGCTATGTATTCACTGACAAGCAATACACTGATGTGTACTG	2081	0.2081	No Hit
CCCCTTTCGACAATGGACCTTATCGCTCACTGTCTGACTCCCGGAGTAAG	2070	0.207	No Hit
CTTAACTGGTGATCTGGGCTGTTCCCCTTTCGACAATGGACCTTATCGCT	2005	0.20049999999999998	No Hit

FastQC結果を眺める

他にもちらほらと既知のアダプター配列と一致するものが見つかる。

FastQC Report

Fri 8 May 2015
SRR616268sub_1.fastq

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 ①
- Adapter Content
- Kmer Content

CACGGTTTCAGGAACTGTTTCACTCCCCTTCCGGGGTGCTTTTCACCTTT	1610	0.161	No Hit
GGGCTGTTCCCCTTTCGACAATGGACCTTATCGCTCACTGTCTGACTCCC	1560	0.156	No Hit
GTCCTCTTCTGCACTCAAGTTTCCCAGTTTCCGATGCGCTTCCTCGGTTA	1559	0.1559	No Hit
GATCGGAAGAGCACACGTCTGAACTCCAGTCACCGATGTATCTCGTATGC	1539	0.1539	TruSeq Adapter, Index 2 (100% over 50bp)
CTCGCCTTAGATCCCGACTAACCCTGGGAGGACGAGCCTTCCCCAGGAAA	1499	0.14989999999999998	No Hit
GGGACCTTAACTGGTGATCTGGGCTGTTCCCCTTTCGACAATGGACCTTA	1497	0.1497	No Hit
ATTTAGCCTTGGGAGATGGTCCTCCCGGATTCCGACGGAATTTCACGTGT	1487	0.1487	No Hit
CATGGGTAGGTCACTTGGTTTCGGGTCTACATCTGCTTACTCATTCGCCC	1464	0.1464	No Hit
CCCTACTGCTGCCTCCCGTAGGAGTTTGGGCCGTGTCTCAGTCCCAATGT	1455	0.1455	No Hit
CTTTCGACAATGGACCTTATCGCTCACTGTCTGACTCCCGGAGTAAGATC	1451	0.1451	No Hit
CGCTACTCATGCCGGCATTCTCACTTCTAAGCGCTCCAGCCGTCCTCACG	1405	0.1405	No Hit
CCGGGGTGCTTTTCACCTTTCCCTCACGGTACTGGTTCACTATCGGTCAC	1361	0.1361	No Hit
CTGGTGATCTGGGCTGTTCCCCTTTCGACAATGGACCTTATCGCTCACTG	1338	0.1338	No Hit
GGGCCTATTCACTGCGGCTGACCTTGCGGTCAGCACCCCTTCTTCCGAAG	1337	0.1337	No Hit
GCCGCCGGCCAGCTATGTATTCACTGACAAGCAATACACTGATGTGTACT	1324	0.13240000000000002	No Hit
AGATCGGAAGAGCACACGTCTGAACTCCAGTCACCGATGTATCTCGTATG	1318	0.1318	TruSeq Adapter, Index 2 (100% over 49bp)
GCCGAGTTCCTTAACGAGAGTTCGCTCGCTCACCTGAGGATACTCTCCTC	1310	0.131	No Hit
CGTCCCTCCATCGCTTAAACAAAATAAACTAGTGCAGGAATCTCAACCTG	1276	0.1276	No Hit
CACACGGTTTCAGGAACTGTTTCACTCCCCTTCCGGGGTGCTTTTCACCT	1275	0.1275	No Hit

FastQC結果を眺める

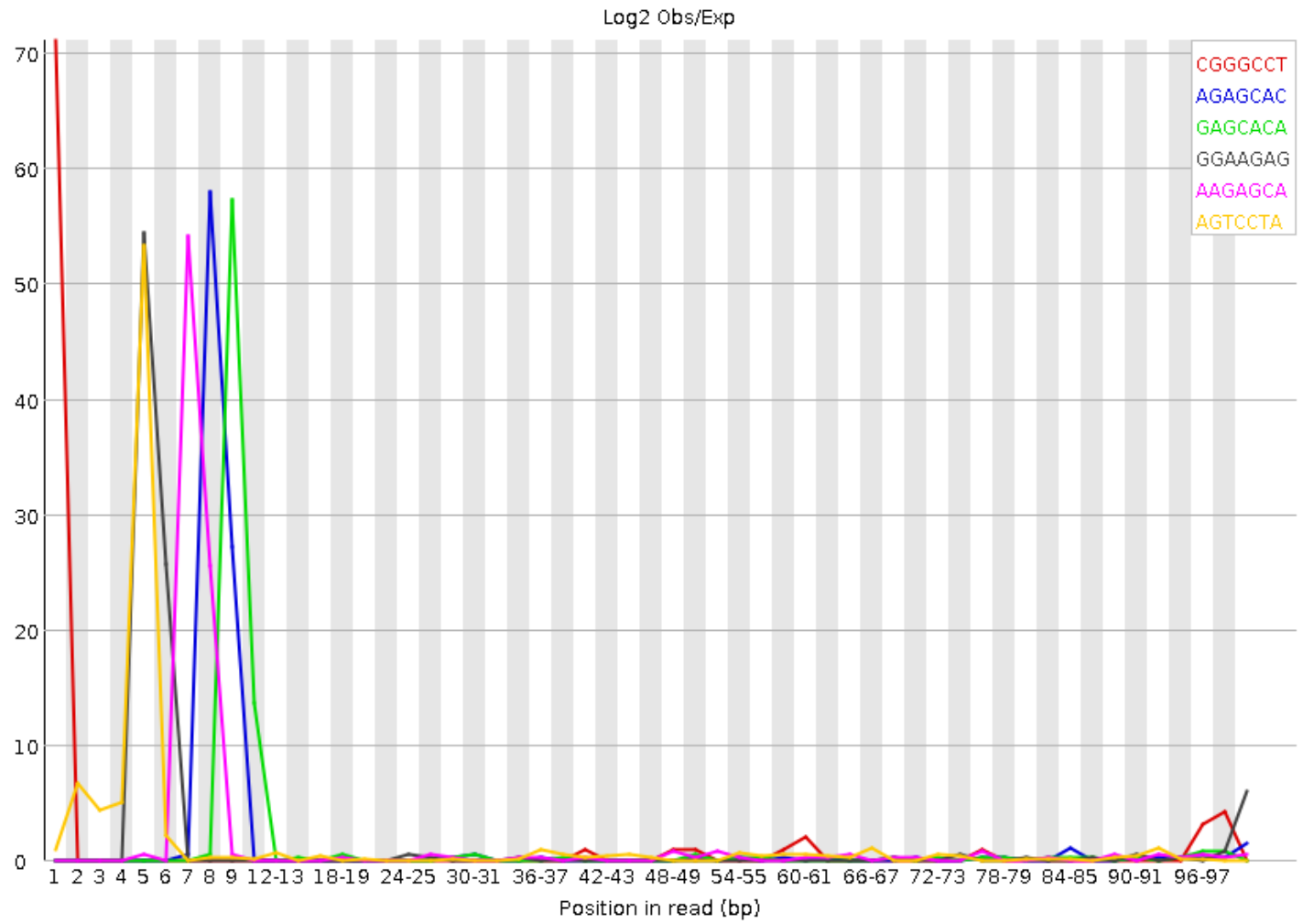
7-merの塩基配列の出現頻度解析結果。出現頻度の期待値に比べて実測値が位置特異的に極端に多いk-merの上位がリストアップされている。

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 ①

Kmer Content

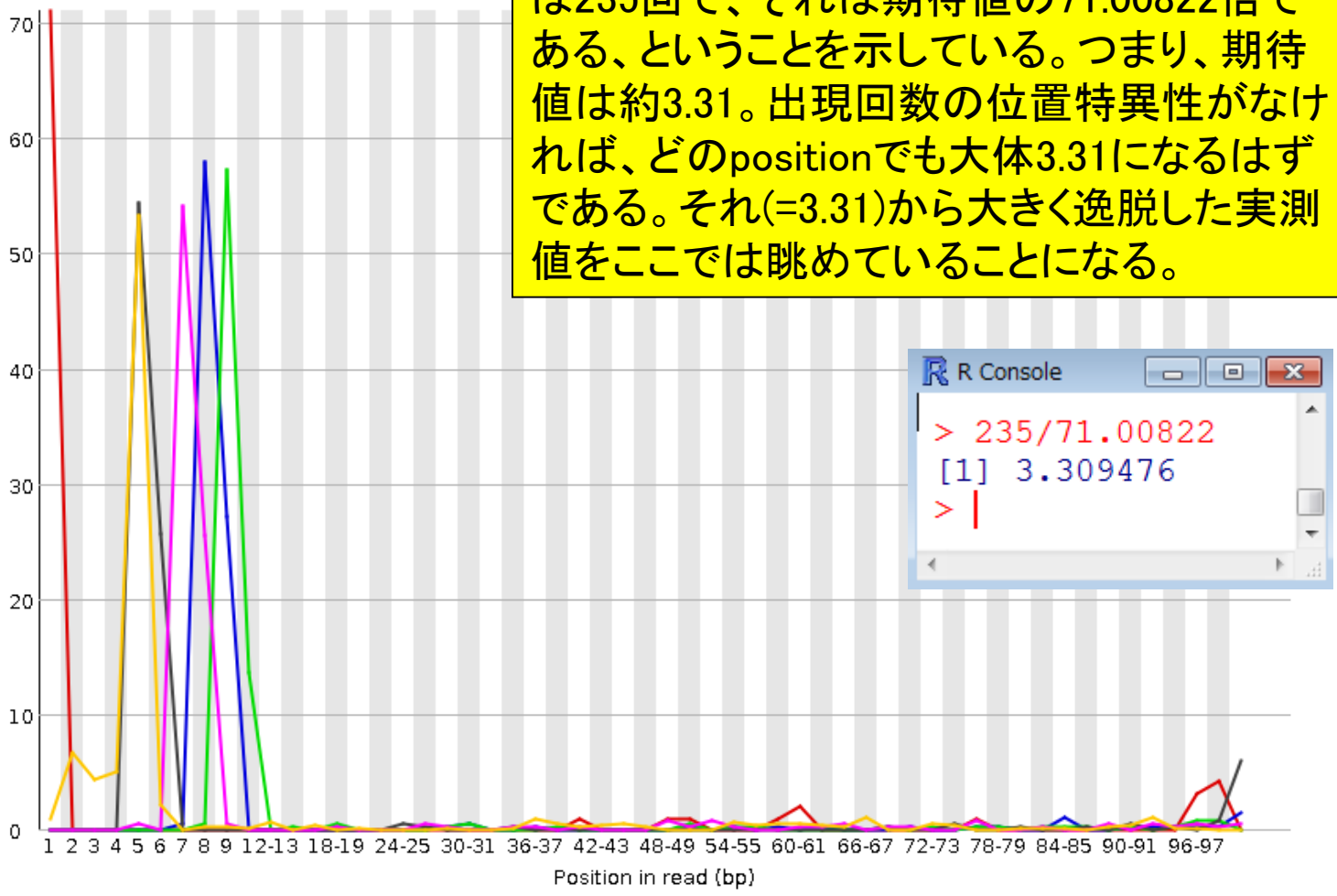


FastQC結果を眺める

ちょっと下のほうに移動。②CGGGCCTという7-merの塩基配列がposition 1で非常に多く出現する。実際のposition 1での出現回数は235回で、それは期待値の71.00822倍である、ということを示している。つまり、期待値は約3.31。出現回数の位置特異性がなければ、どのpositionでも大体3.31になるはずである。それ(=3.31)から大きく逸脱した実測値をここでは眺めていることになる。

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 ①



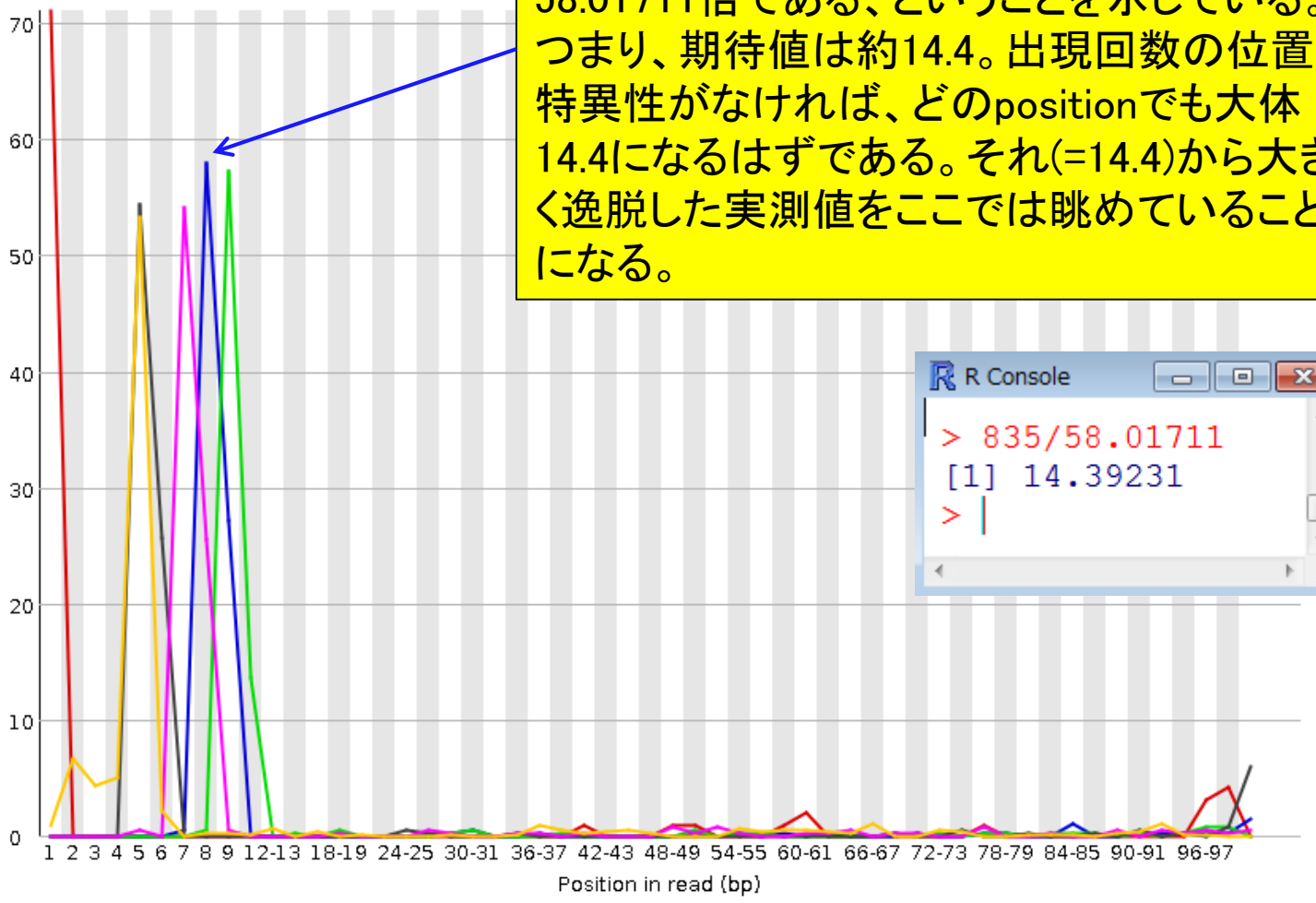
Sequence	Count	PValue	Obs/Exp Max	Max Obs/Exp Position
CGGGCCT	235	0.0	71.00822	1 ②
AGAGCAC	835	0.0	58.01711	8
GAGCACA	845	0.0	57.330517	9

FastQC結果を眺める

②次に多いのはAGAGCAC。これはposition 8で非常に多く出現する。実際のposition 8での出現回数は835回で、それは期待値の58.01711倍である、ということを示している。つまり、期待値は約14.4。出現回数の位置特異性がなければ、どのpositionでも大体14.4になるはずである。それ(=14.4)から大きく逸脱した実測値をここでは眺めていることになる。

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 ①



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

②

Contents

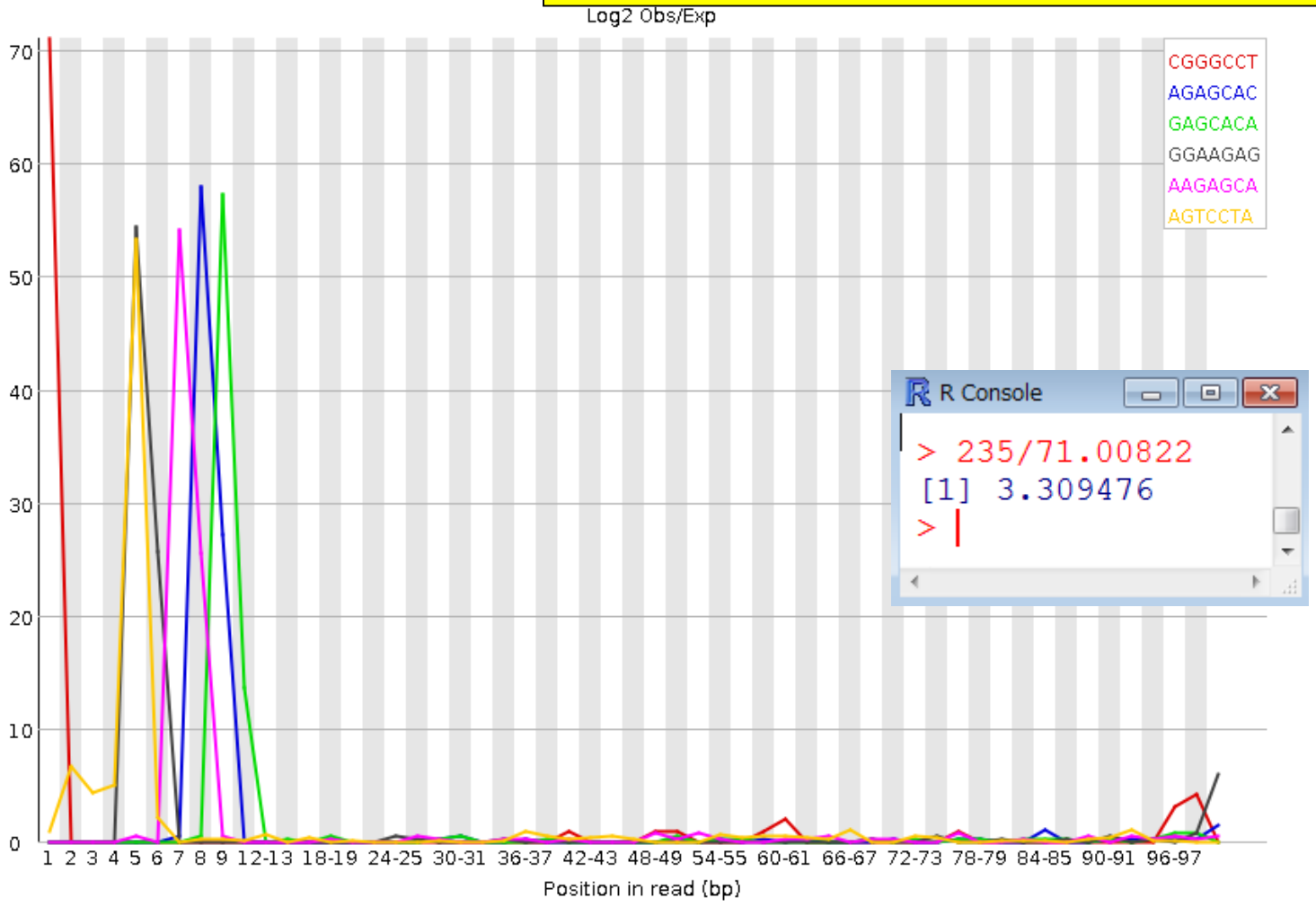
- QC(Quality Control)
  - Quality Check (FastQC):乳酸菌RNA-seqデータ
    - k-mer解析、課題1
  - Quality Check (FastQC):カイコsRNA-seqデータ
  - トリミング:カイコsRNA-seqデータ
    - 全体像を把握
    - アダプター配列除去(QuasRパッケージ)、課題2
    - Rを駆使して結果を確認
  - トリミング:乳酸菌RNA-seqデータ
    - サブセットの抽出
    - paired-endデータのアダプター配列除去(QuasRパッケージ)

k-mer解析

CGGGCCTがposition 1で235回出現しており、それが期待値(=3.31)の71.00822倍だということを、Rで確認してみよう。

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



Sequence	Count	PValue	Obs/Exp Max	Max Obs/Exp Position
CGGGCCT	235	0.0	71.00822	1 ②
AGAGCAC	835	0.0	58.01711	8
GAGCACA	845	0.0	57.330517	9

k-mer解析

CGGGCCTがposition 1で235回出現しており、それが期待値(=3.31)の71.00822倍だということを、Rで確認してみよう。

イントロ | NGS | 読み込み | FASTQ形式 | 基礎 NEW

② Sanger FASTQ形式ファイルを
します。入力、出力形式は、と
「ファイル」-「ディレクトリの変更」で

1. サンプルデータ7のFASTQ形式

SRR037439から得られるFASTQ
quality情報を除く塩基配列情報の

```
in_f <- "SRR037439.fastq"

#必要なパッケージをロード
library(Biostrings)

#入力ファイルの読み込み
fasta <- readDNAStringSet
fasta
```

2. サンプルデータ7のFASTQ形式

SRR037439から得られるFASTQ

8. FASTQ形式ファイル(SRR616268sub_1.fastq.gz)の場合:

乳酸菌RNA-seqデータSRR616268の最初の100万リード分(約73MB)です。長さは全て107 bpです。

```
in_f <- "SRR616268sub_1.fastq.gz"       #入力ファイル名を指定してin_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="fastq")#in_fで指定したファイルの読み込み
fasta                                   #確認してるだけです

#以下はおまけ(部分配列を抽出し出現頻度上位を概観)
hoge <- subseq(fasta, start=1, end=7)   #指定した始点と終点の範囲の配列を抽出
hoge                                    #確認してるだけです
head(table(hoge))                       #出現頻度を順番に表示
head(sort(table(hoge), decreasing=T))   #出現頻度上位を表示
table(hoge)["CGGGCCT"]                  #CGGGCCTの出現頻度を表示

hoge <- subseq(fasta, start=3, width=7)#指定した始点から一定範囲の配列を抽出
hoge                                    #確認してるだけです
table(hoge)["CGGGCCT"]                  #CGGGCCTの出現頻度を表示

hoge <- subseq(fasta, start=5, width=7)#指定した始点から一定範囲の配列を抽出
table(hoge)["CGGGCCT"]                  #CGGGCCTの出現頻度を表示
```

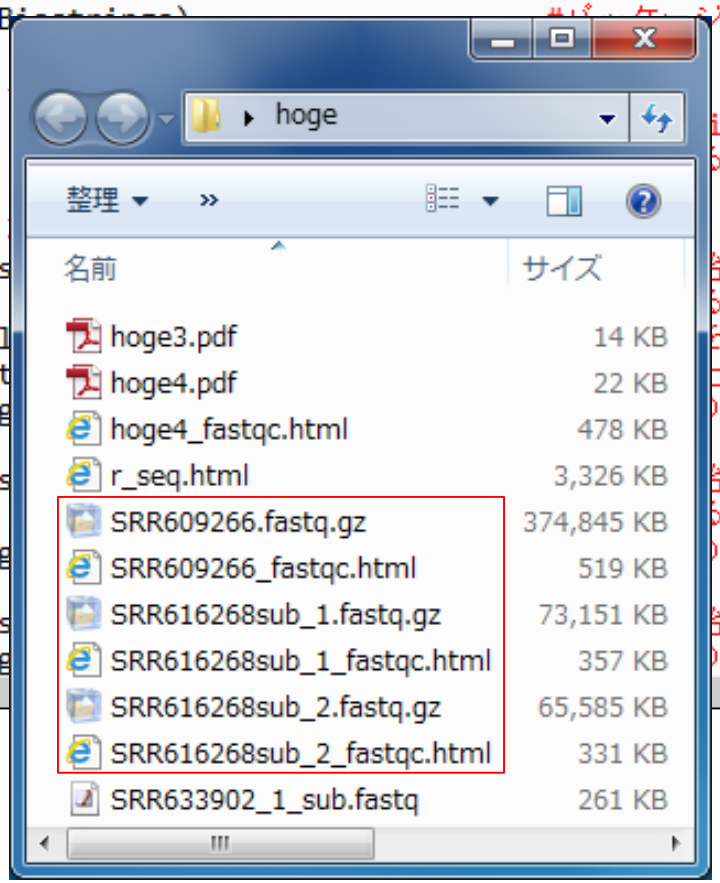
Tips：正規表現

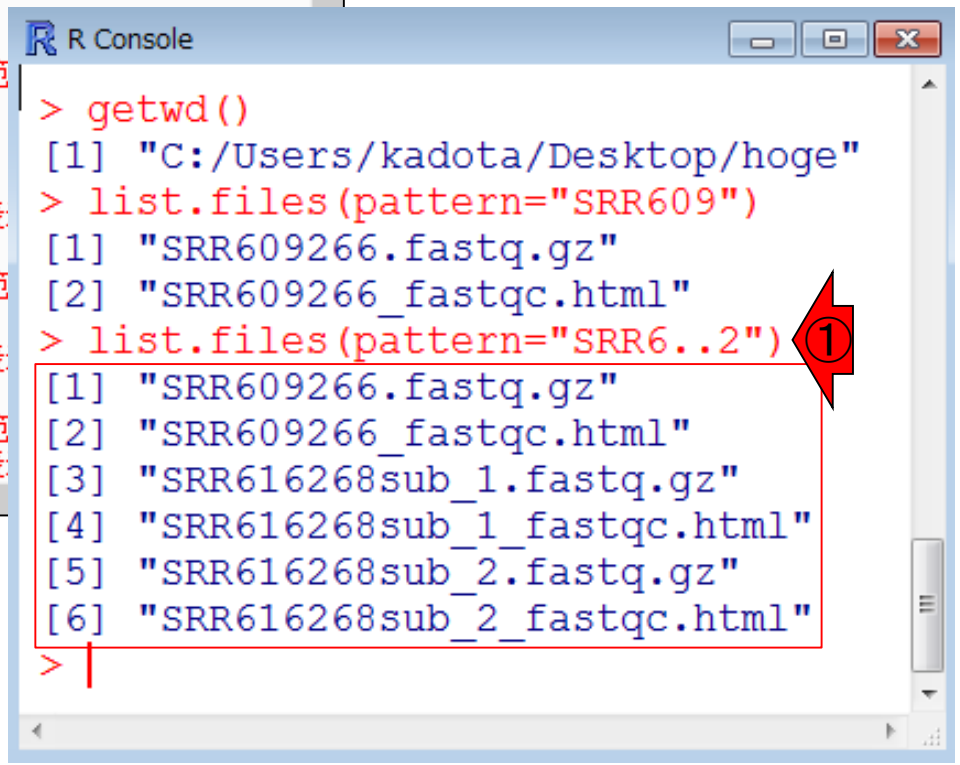
入力ファイルの存在確認がメインだが、ついでに①正規表現のテクニックを伝授。

8. FASTQ形式ファイル(SRR616268sub_1.fastq.gz)の場合:

乳酸菌RNA-seqデータSRR616268の最初の100万リード分(約73MB)です。長さは全て107 bpです。

```
in_f <- "SRR616268sub_1.fastq.gz"       #入力ファイル名を指定してin_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
fasta <-                                in_fで指定したファイルの読み込み
fasta                                   るだけです

#以下はお
hoge <- s                               始点と終点の範
hoge                                    るだけです
head(tabl                               と順番に表示
head(sort                               上位を表示
table(hog                               の出現頻度を表

hoge <- s                               始点から一定範
hoge                                    るだけです
table(hog                               の出現頻度を表

hoge <- s                               始点から一定範
table(hog                               の出現頻度を表
```





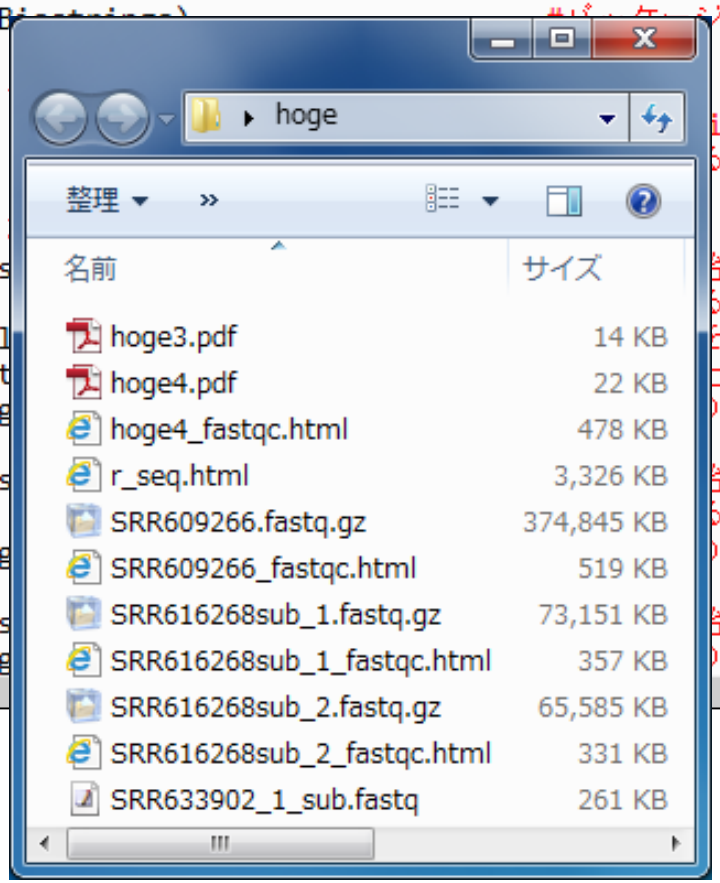
Tips：正規表現

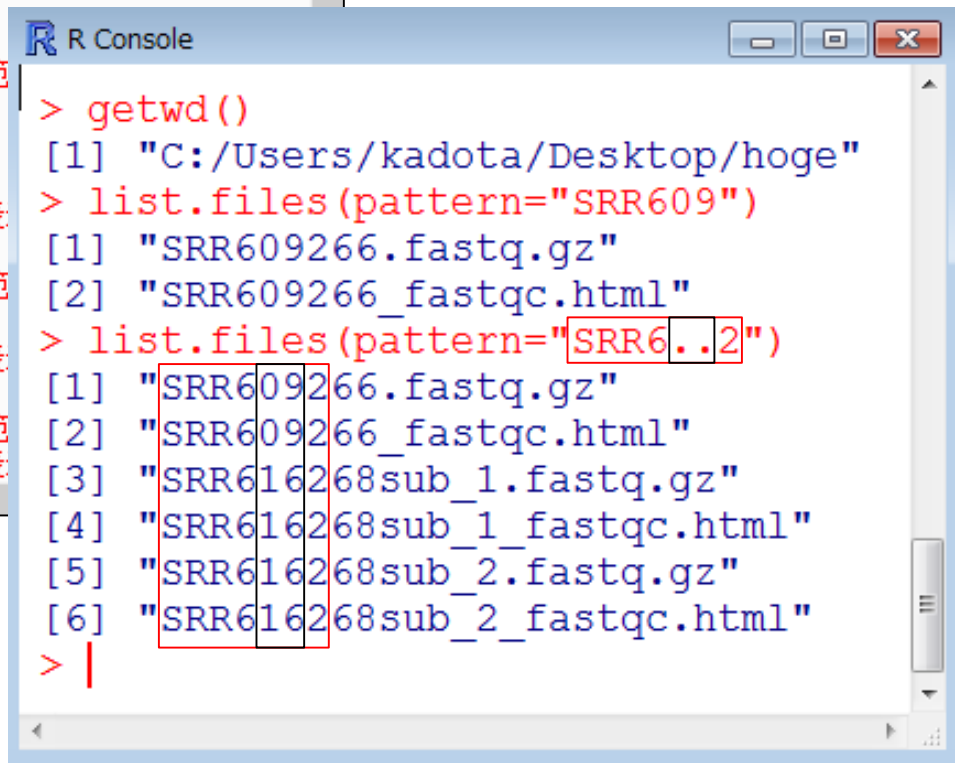
list.files関数中のpatternオプションでは、正規表現を用いた検索が可能。「. (どっと)」は任意の1文字を表す。ここでは「..」と.が2つ並んでいるので、「SRR6..2」の条件に該当する「SRR6092」と「SRR6162」を含むファイル群が表示される。アダプター配列中のIndex配列部分を曖昧にして検索したいときに便利。

8. FASTQ形式ファイル(SRR616268sub_1.fastq.gz)の場合:

乳酸菌RNA-seqデータSRR616268の最初の100万リード分(約73MB)です。長さは全

```
in_f <- "SRR616268sub_1.fastq.gz"        #入力ファイル名を指定して

#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
fasta <-                                 in_fで指定したファイルの読み込み
fasta                                    るだけです

#以下はお
hoge <- s                                始点と終点の範
hoge                                     るだけです
head(tabl                                と順番に表示
head(sort                                上位を表示
table(hog                                の出現頻度を表

hoge <- s                                始点から一定範
hoge                                     るだけです
table(hog                                の出現頻度を表

hoge <- s                                始点から一定範
table(hog                                の出現頻度を表
```





k-mer解析

Biostringsパッケージ中のreadDNAStringSet関数は、formatオプションを"fastq"にすることで、FASTQ形式ファイルを読み込める。gzip圧縮ファイルを入力として与えても自動判定して読み込んでくれる。ただし、quality score情報は読み飛ばすようだ。ここではリード塩基配列情報しか取り扱わないので特に問題ない。

8. FASTQ形式ファイル(SRR616268sub_1.fastq.gz)の場合:

乳酸菌RNA-seqデータSRR616268の最初の100万リード分(約73MB)です。長さは全

```
in_f <- "SRR616268sub_1.fastq.gz"       #入力ファイル名を指定して

#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="fastq")#in_fで指定したファイルの読み込み
fasta                                   #確認してるだけです

#以下はおまけ(部分配列を抽出し出現頻
hoge <- subseq(fasta, start=1, end
hoge
head(table(hoge))
head(sort(table(hoge), decreasing=
table(hoge)["CGGGCCT"]

hoge <- subseq(fasta, start=3, wid
hoge
table(hoge)["CGGGCCT"]

hoge <- subseq(fasta, start=5, wid
table(hoge)["CGGGCCT"]
```

R Console

```
> fasta                                   #確認してるだけです
  A DNAStringSet instance of length 1000000
          width seq                                   names
      [1]   107 AGCCCGACTTT...CTTCTCTAAC SRR616268.7 2291:...
      [2]   107 GATCTGGGCTG...ACCTCCGAAA SRR616268.20 2291...
      [3]   107 CCGGTATATTT...ACGATCTTAC SRR616268.21 2291...
      [4]   107 CTTCGATACCG...CATAAAGACC SRR616268.22 2291...
      [5]   107 CCCCGGTATAT...CAAGGTCCAT SRR616268.30 2291...
      ...   ... ...
 [999996]   107 CCCCGGTATAT...CAAGGGTTAC SRR616268.1000856...
 [999997]   107 TTCGGGTCTAC...CGGGTGAAAT SRR616268.1000857...
 [999998]   107 CGTCCATCCCG...AGCCAAGAGT SRR616268.1000858...
 [999999]   107 CTAGGGAGTAT...TTCCTGGACA SRR616268.1000859...
[1000000]   107 GCCTTGTCAAT...GACAGCAGCT SRR616268.1000860...
>
```

k-mer解析

目的：CGGGCCTがposition 1で235回出現しており、それが期待値(=3.31)の71.00822倍だということを、Rで確認。まずはposition 1から7-mer分の部分配列を切り出し、7-merの種類ごとの出現頻度を計測。

8. FASTQ形式ファイル(SRR616268sub_1.fastq.gz)の場合:

乳酸菌RNA-seqデータSRR616268の最初の100万リード分(約73MB)です。長さは全

```
in_f <- "SRR616268sub_1.fastq.gz"       #入力ファイル名を指定してin_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="fastq")#in_fで指定したファイルの読み込み
fasta                                   #確認してるだけです
```

```
#以下はおまけ(部分配列を抽出し出現頻
hoge <- subseq(fasta, start=1, end
hoge
head(table(hoge))
head(sort(table(hoge), decreasing=
table(hoge)["CGGGCCT"]

hoge <- subseq(fasta, start=3, wid
hoge
table(hoge)["CGGGCCT"]

hoge <- subseq(fasta, start=5, wid
table(hoge)["CGGGCCT"]
```

R Console

```
> fasta                                     #確認してるだけです
  A DNAStringSet instance of length 1000000
          width seq                                   names
      [1]   107 AGCCCGACTTT...CTTCTCTAAC SRR616268.7 2291:...
      [2]   107 GATCTGGGCTG...ACCTCCGAAA SRR616268.20 2291...
      [3]   107 CCGGTATATTT...ACGATCTTAC SRR616268.21 2291...
      [4]   107 CTTCGATACCG...CATAAAGACC SRR616268.22 2291...
      [5]   107 CCCCGGTATAT...CAAGGTCCAT SRR616268.30 2291...
      ...   ... ...
 [999996]   107 CCCCGGTATAT...CAAGGGTTAC SRR616268.1000856...
 [999997]   107 TTCGGGTCTAC...CGGGTGAAAT SRR616268.1000857...
 [999998]   107 CGTCCATCCCG...AGCCAAGAGT SRR616268.1000858...
 [999999]   107 CTAGGGAGTAT...TTCCTGGACA SRR616268.1000859...
[1000000]   107 GCCTTGTCAAT...GACAGCAGCT SRR616268.1000860...
>
```

k-mer解析

position 1から7-mer分の部分配列を切り出したところまで。subseq関数については、2015年4月14日の「ゲノム情報解析基礎」スライド72-75あたりを復習せよ。

8. FASTQ形式ファイル(SRR616268sub_1.fastq.gz)の場合:

乳酸菌RNA-seqデータSRR616268の最初の100万リード分(約73MB)です。長さは全て107 bpです。

```
#以下はおまけ(部分配列を抽出し出現頻度上位を概観)
hoge <- subseq(fasta, start=1, end=7)  #指定した始点と終点の範囲の配列を抽出
hoge                                   #確認してるだけです
head(table(hoge))                      #出現頻度を順番に表示
head(sort(table(hoge), decreasing=T))  #出現頻度上位を表示
table(hoge)["CGGGCCT"]                 #CGGGCCTの出現頻度を表示

hoge <- subseq(fasta, start=3, width=7)#指定した始点から一定範囲の配列を抽出
hoge                                   #確認してるだけです
table(hoge)["CGGGCCT"]

hoge <- subseq(fasta, start=5, wi
table(hoge)["CGGGCCT"]

table(subseq(fasta, start=1, widt
table(subseq(fasta, start=2, widt
table(subseq(fasta, start=3, widt
table(subseq(fasta, start=4, widt
table(subseq(fasta, start=5, widt
table(subseq(fasta, start=6, widt
table(subseq(fasta, start=7, widt
table(subseq(fasta, start=8, widt
```

R Console

```
> hoge
  A DNAStringSet instance of length 1000000
           width seq                                    names
       [1]     7 AGCCCGA                                SRR616268.7 2291:...
       [2]     7 GATCTGG                                SRR616268.20 2291...
       [3]     7 CCGGTAT                                SRR616268.21 2291...
       [4]     7 CTTCGAT                                SRR616268.22 2291...
       [5]     7 CCCCGGT                                SRR616268.30 2291...
       ...   ... ...
  [999996]     7 CCCCGGT                                SRR616268.1000856...
  [999997]     7 TTCGGGT                                SRR616268.1000857...
  [999998]     7 CGTCCAT                                SRR616268.1000858...
  [999999]     7 CTAGGGA                                SRR616268.1000859...
 [1000000]     7 GCCTTGT                                SRR616268.1000860...
> |
```

Tips：講義資料再取得

ウェブページ内での「subseq」でのキーワード検索でもいいでしょうし、「2015.04.14」でキーワード検索して講義資料PDFを再取得するのもアリでしょう。

(Rで)塩基配列解析

～NGS、RNA-seq、ゲノム、トランスクリプトーム、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

What's new?

- このウェブペ
  Macintosh20
  であるという
  (Windows20
  を体系的にま
- 参考資料(講
- 「イントロ | N
  する際に生じ
  口達洋 氏提
- Rパッケージ
  リリースに対
- Rパッケージ

- はじめに (last modified 2015/03/31)
- 参考資料(講義、講習会、本など) (last modified 2015/06/16) NEW
- 過去のお知らせ (last modified 2015/03/28) NEW
- インストール | について (last modified 2015/04/04)
- インストール | R本体 | 最新版 | Win用 (last modified 2015/03/22)推奨
- インストール | R本体 | 最新版 | Mac用 (last modified 2015/04/22)推奨
- インストール | R本体 |
- インストール | R本体 |
- インストール | Rパッケ
- インストール | Rパッケ
- インストール | Rパッケ
- インストール | Rパッケ
- インストール | Rパッケ
- (削除予定)Rのインス
- (削除予定)個別パッケ
- 基本的な利用法 (last
- サンプルデータ (last
- バイオインフォマティク
- 書籍 | トランスクリプト
- 書籍 | トランスクリプト

① ②

参考資料(講義、講習会、本など) NEW

基本的に私門田の個人ページに記載してあるものです。かなり古い講演資料などの情報をもとに勉強されている方もいらっしゃるようですので、ここでは2013年秋以降の情報を載せておくとともに、大まかな内容についても述べておきます。講演予定のものについては、資料のアップは講演当
ファイルがほしい方はお
連絡や私個人への謝辞

書籍、学会誌

- 孫建強, 三浦文
  第3回Linux環境
  2015.
  内容: 日本乳酸
  と Mac の Linux
  境の違い」とした
  料などは「書籍
  で」の項目をご
- 孫建強, 湯敏,
  手法: 第2回GU
  174, 2014.

越しください。門田が出勤していれば対応できます。2コマ(2×90 min)分。

- **門田幸二**,「講義資料」, アグリバイオインフォマティクス教育研究プログラムの大学院講義科目: ゲノム情報解析基礎, 東京大学(東京), 2015.04.21
  内容: CRANとBioconductor, BSgenomeパッケージを利用してヒトゲノム中のCpG出現確率が低いことを確認。2連続塩基の出現頻度解析。作図(box plot)。1コマ(90 min)分。
- **門田幸二**,「講義資料(Win版とMac版)」, アグリバイオインフォマティクス教育研究プログラムの大学院講義科目: ゲノム情報解析基礎, 東京大学(東京), 2015.04.14
  内容: このウェブページの基本的な使い方、ありがちなミスや警告メッセージの読み取り。コード内部の説明、関数の使用法、タブ補完、二重クォーテーション問題などのTips。multi-FASTAファイルの解析。GC含量計算など。2コマ(180 min)分。
- **門田幸二**,「ウェブページと講義資料」, アグリバイオインフォマティクス教育研究プログラムの大学院講義科目: ゲノム情報解析基礎, 東京大学(東京), 2015.04.07
  内容: 初心者向けバイオインフォマティクス全般およびゲノム情報解析系のイントロダクションの話。Rのイントロダクションやこのウェブページの簡単な使い方を含む。1コマ(90 min)分。

k-mer解析

次に、7-merの種類ごとの出現頻度をtable関数を用いて計測。①デフォルトはalphabet順に作成したk-mer (k=7)の種類ごとに出現回数を返す。②table(hoge)実行結果を降順(decreasing=T)にsortした上位6個を表示。一番出現回数が多いのはCCCCGGTの14,995回。

```
#以下はおまけ(部分配列を抽出し出現頻度上位を概観)
hoge <- subseq(fasta, start=1, end=7)   #指定した始点と終点の範囲の配
hoge                                    #確認してるだけです
head(table(hoge))                       #出現頻度を順番に表示
head(sort(table(hoge), decreasing=T))   #出現頻度上位を表示
table(hoge)["CGGGCCT"]                  #CGGGCCTの出現頻度を表示

hoge <- subseq(fasta, start=3, width=7)#指定した始点から一定範囲の配列を抽出
hoge                                    #確認してるだけです
table(hoge)["CGGGCCT"]                  #CGGGCCTの出現頻度を表示

hoge <- subseq(fasta, start=5, width=7)#指定した始
table(hoge)["CGGGCCT"]                  #CGGGCCTの

table(subseq(fasta, start=1, width=7))["CGGGCCT"]
table(subseq(fasta, start=2, width=7))["CGGGCCT"]
table(subseq(fasta, start=3, width=7))["CGGGCCT"]
table(subseq(fasta, start=4, width=7))["CGGGCCT"]
table(subseq(fasta, start=5, width=7))["CGGGCCT"]
table(subseq(fasta, start=6, width=7))["CGGGCCT"]
table(subseq(fasta, start=7, width=7))["CGGGCCT"]
table(subseq(fasta, start=8, width=7))["CGGGCCT"]
```

①

②

R Console

```
> head(table(hoge))                          #出現$
hoge
AAAAAAA AAAAAAC AAAAAAG AAAAAAT AAAAACA AAAAACG
    238      26       3       1       1       1
> head(sort(table(hoge), decreasing=T))   #出現$
hoge
CCCCGGT GCCGGCA GGCCTAT GTGCTTT CCCGGTA GTCACTA
  14995   13663   11403    9206    8811    8451
> table(hoge)["CGGGCCT"]                     #CGGGC$
CGGGCCT
   1804
> length(table(hoge))
[1] 15639
> 4^7
[1] 16384
>
```

k-mer解析

③ピンポイントで目的のCGGGCCTの出現回数(=1,804)を表示。FastQC内部の計算事情を知らないヒトは、この段階でオカシイと思わなければいけない。

```
#以下はおまけ(部分配列を抽出し出現頻度上位を概観)
hoge <- subseq(fasta, start=1, end=7)  #指定した始点と終点の範囲の配
hoge                                   #確認してるだけです
head(table(hoge))                      #出現頻度を順番に表示
head(sort(table(hoge), decreasing=T))  #出現頻度上位を表示
table(hoge)["CGGGCCT"]                 #CGGGCCTの出現頻度を表示

hoge <- subseq(fasta, start=3, width=7)#指定した始点から一定範囲の配列を抽出
hoge                                   #確認してるだけです
table(hoge)["CGGGCCT"]                 #CGGGCCTの出現頻度を表示

hoge <- subseq(fasta, start=5, width=7)#指定した始
table(hoge)["CGGGCCT"]                 #CGGGCCTの

table(subseq(fasta, start=1, width=7))["CGGGCCT"]
table(subseq(fasta, start=2, width=7))["CGGGCCT"]
table(subseq(fasta, start=3, width=7))["CGGGCCT"]
table(subseq(fasta, start=4, width=7))["CGGGCCT"]
table(subseq(fasta, start=5, width=7))["CGGGCCT"]
table(subseq(fasta, start=6, width=7))["CGGGCCT"]
table(subseq(fasta, start=7, width=7))["CGGGCCT"]
table(subseq(fasta, start=8, width=7))["CGGGCCT"]
```

R Console

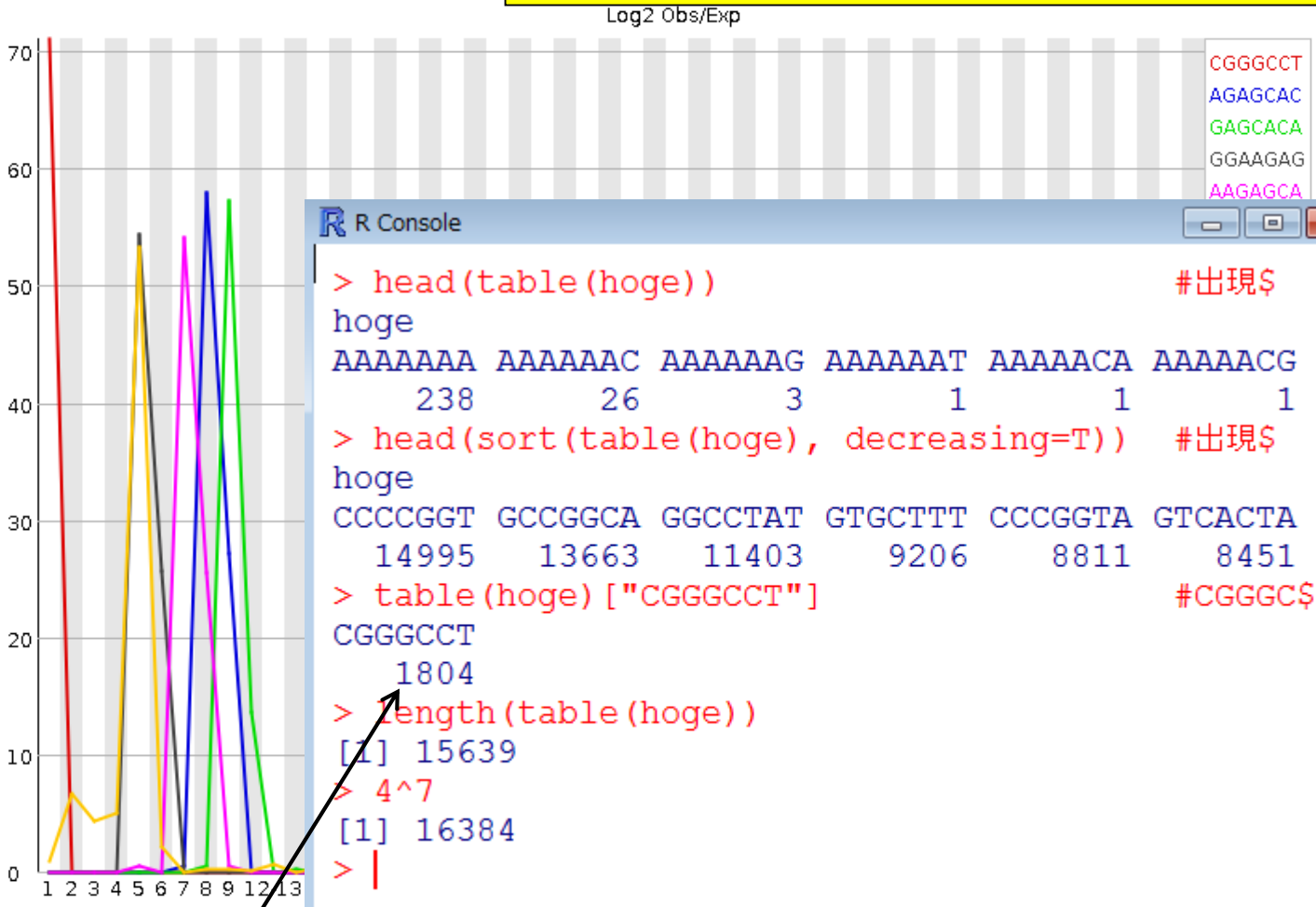
```
> head(table(hoge))                                #出現$
hoge
AAAAAAA AAAAAAC AAAAAAG AAAAAAT AAAAACA AAAAACG
    238      26       3       1       1       1
> head(sort(table(hoge), decreasing=T))  #出現$
hoge
CCCCGGT GCCGGCA GGCCTAT GTGCTTT CCCGGTA GTCACTA
  14995   13663   11403    9206    8811    8451
> table(hoge)["CGGGCCT"]                           #CGGGC$
CGGGCCT
   1804
> length(table(hoge))
[1] 15639
> 4^7
[1] 16384
>
```

k-mer解析

理由は、FastQC実行結果では、「CGGGCCTがposition 1で235回出現しており、それが期待値(=3.31)の71.00822倍」だったからである

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



R Console

```
> head(table(hoge))                          #出現$
hoge
AAAAAAA AAAAAAC AAAAAAG AAAAAAT AAAAACA AAAAACG
    238      26       3       1       1       1
> head(sort(table(hoge), decreasing=T))   #出現$
hoge
CCCCGGT GCCGGCA GGCCTAT GTGCTTT CCCGGTA GTCACTA
  14995   13663   11403    9206    8811    8451
> table(hoge)["CGGGCCT"]                        #CGGGC$
CGGGCCT
   1804
> length(table(hoge))
[1] 15639
> 4^7
[1] 16384
>
```

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

FastQCは一部を計算

FastQCはプログラム名通り（半分冗談半分本気）、リーズナブルな計算時間でQC結果を返すことに重きを置いている。全リードの2%のみで解析しているため、Rの結果とは合わない。

Babraham Bioinformatics

About | People | Services | Projects | Training | Publications

FastQC

Documentation

A copy of the FastQC documentation is available for you to try before you buy (well download..).

Example Repo

- Good Illu
- Bad Illumi
- Adapter d
- Small RN
- Reduced
- PacBio
- 454

① ②

Index of /projects/fastqc/Help

Name

Parent Directory

1 Introduction/ 25

2 Basic Operations/ 25

3 Analysis Modules/ 25

Apache/2.2.15 (CentOS) Se

Index of /projects/fastqc/Help/3 Analysis Modules

Name	Last modified	Size	Description
Parent Directory		-	
1 Basic Statistics.html	25-Mar-2015 09:40	1.8K	
2 Per Base Sequence Quality.html	25-Mar-2015 09:40	3.6K	
3 Per Sequence Quality Scores.html	25-Mar-2015 09:40	1.7K	
4 Per Base Se			
5 Per Sequenc			
6 Per Base N			
7 Sequence L			
8 Duplicate Se			
9 Overrepresented Sequences.html	25-Mar-2015 09:40	2.4K	
10 Adapter Content.html	25-Mar-2015 09:40	2.4K	
11 Kmer Content.html	25-Mar-2015 09:40	2.5K	
12 Per Tile Sequence Quality.html	25-Mar-2015 09:40	2.2K	
duplication_levels.png	25-Mar-2015 09:40	20K	

③

To allow this module to run in a reasonable time only 2% of the whole library is analysed and the results are extrapolated to the rest of the library. Sequences longer than 500bp are truncated to 500bp for this analysis.

k-mer解析

```
#以下はおまけ(部分配列を抽出し出現頻度上位を概観)
hoge <- subseq(fasta, start=1, end=7)   #指定した始点と終点の範囲の
hoge                                    #確認してるだけです
head(table(hoge))                       #出現頻度を順番に表示
head(sort(table(hoge), decreasing=T))   #出現頻度上位を表示
table(hoge)["CGGGCCT"]                  #CGGGCCTの出現頻度を表示

hoge <- subseq(fasta, start=3, width=7)#指定した始点から一定範囲の配列を抽出
hoge                                    #確認してるだけです
table(hoge)["CGGGCCT"]                  #CGGGCCTの出現頻度を表示

hoge <- subseq(fasta, start=5, width=7)#指定した始
table(hoge)["CGGGCCT"]                  #CGGGCCTの

table(subseq(fasta, start=1, width=7))["CGGGCCT"]
table(subseq(fasta, start=2, width=7))["CGGGCCT"]
table(subseq(fasta, start=3, width=7))["CGGGCCT"]
table(subseq(fasta, start=4, width=7))["CGGGCCT"]
table(subseq(fasta, start=5, width=7))["CGGGCCT"]
table(subseq(fasta, start=6, width=7))["CGGGCCT"]
table(subseq(fasta, start=7, width=7))["CGGGCCT"]
table(subseq(fasta, start=8, width=7))["CGGGCCT"]
```

①table(hoge)実行結果の要素数は15,639。それに対し、②理論上の7-merの種類数は4^7 = 16,384。この結果より、出現回数が0個の7-merが(16,384 − 15,639) = 745個あったのだろうと推測する。

R Console

```
> head(table(hoge))                                  #出現$
hoge
AAAAAAA AAAAAAC AAAAAAG AAAAAAT AAAAACA AAAAACG
    238      26       3       1       1       1
> head(sort(table(hoge), decreasing=T))   #出現$
hoge
CCCCGGT GCCGGCA GGCCTAT GTGCTTT CCCGGTA GTCACTA
  14995   13663   11403    9206    8811    8451
> table(hoge)["CGGGCCT"]                             #CGGGC$
CGGGCCT
   1804
> length(table(hoge))
[1] 15639
> 4^7
[1] 16384
>
```

①

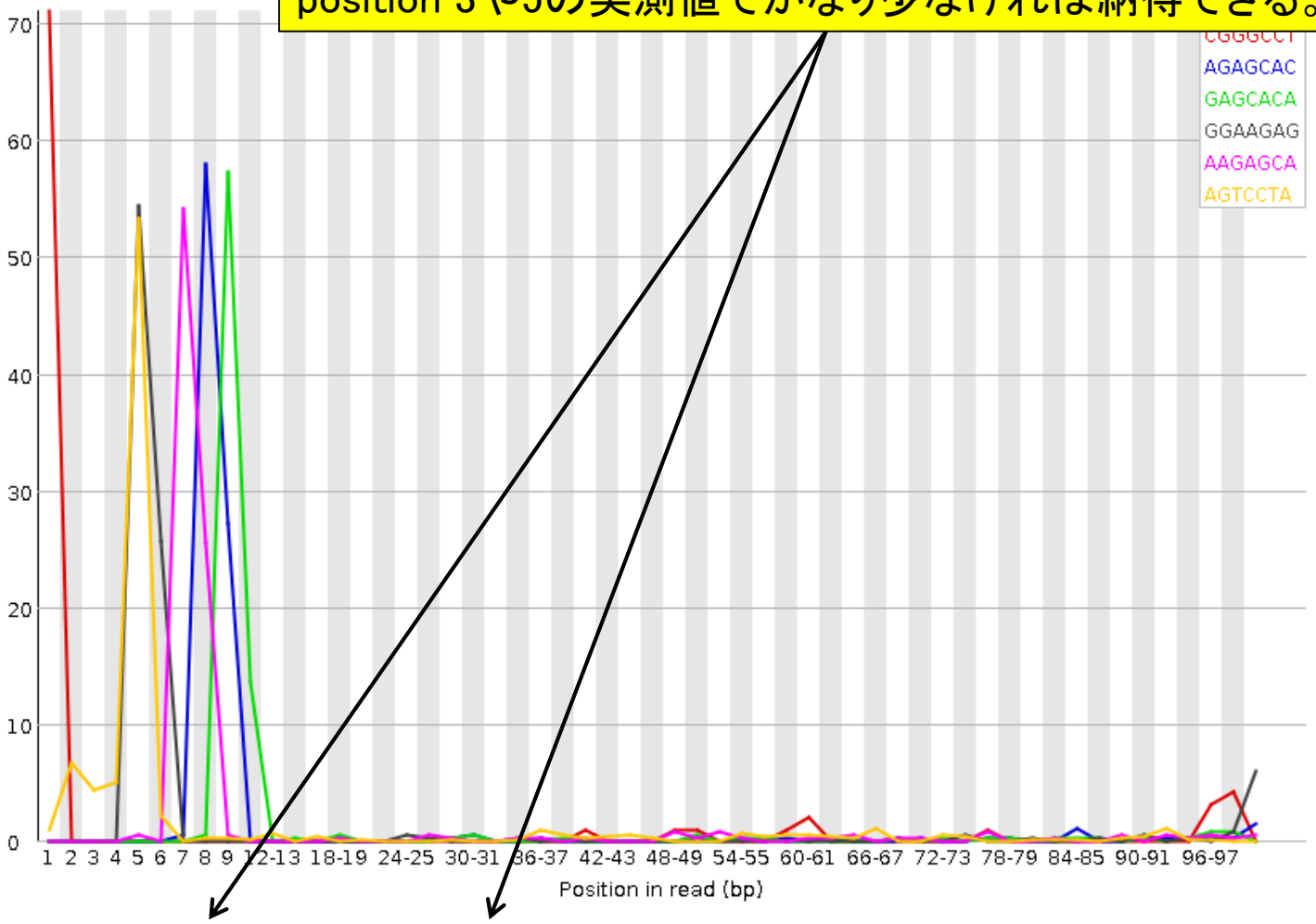
②

k-mer解析

今ここで調べたいのは、「CGGGCCTがposition 1で235回と、期待値の約71倍多く出現していた」こと。Rの全データを用いた結果はposition 1で1,804回であったことから、position 3や5の実測値でかなり少なければ納得できる。

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



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

k-mer解析

position 3におけるCGGGCCTの実測値を得るためには、赤枠部分の出現頻度を調べればよい。

8. FASTQ形式ファイル(SRR616268sub_1.fastq.gz)の場合:

乳酸菌RNA-seqデータSRR616268の最初の100万リード分(約73MB)です。長さは全て107 bpです。

```
in_f <- "SRR616268sub_1.fastq.gz"       #入力ファイル名を指定してin_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="fastq")#in_fで指定したファイルの読み込み
fasta                                   #確認してるだけです

#以下はおまけ(部分配列を抽出し出現頻
hoge <- subseq(fasta, start=1, end
hoge
head(table(hoge))
head(sort(table(hoge), decreasing=
table(hoge)["CGGGCCT"]

hoge <- subseq(fasta, start=3, wi
hoge
table(hoge)["CGGGCCT"]

hoge <- subseq(fasta, start=5, wi
table(hoge)["CGGGCCT"]
```

R Console

```
> fasta                                          #確認してるだけです
  A DNAStringSet instance of length 1000000
          width seq                                    names
      [1]   107 AGCCCGACTTT...CTTCTCTAAC SRR616268.7 2291:...
      [2]   107 GATCTGGGCTG...ACCTCCGAAA SRR616268.20 2291...
      [3]   107 CCGGTATATTT...ACGATCTTAC SRR616268.21 2291...
      [4]   107 CTTCGATACCG...CATAAAGACC SRR616268.22 2291...
      [5]   107 CCCCGGTATAT...CAAGGTCCAT SRR616268.30 2291...
      ...   ... ...
 [999996]   107 CCCCGGTATAT...CAAGGGTTAC SRR616268.1000856...
 [999997]   107 TTCGGGTCTAC...CGGGTGAAAT SRR616268.1000857...
 [999998]   107 CGTCCATCCCG...AGCCAAGAGT SRR616268.1000858...
 [999999]   107 CTAGGGAGTAT...TTCCTGGACA SRR616268.1000859...
[1000000]   107 GCCTTGTCAAT...GACAGCAGCT SRR616268.1000860...
>
```

k-mer解析

```
hoge <- subseq(fasta, start=3, width=7)#指定した始点から一定範囲の配列を抽
hoge                                   #確認してるだけです
table(hoge)["CGGGCCT"]                 #CGGGCCTの出現頻度を表示

hoge <- subseq(fasta, start=5, width=7)#指定した始点から一定範囲の配列を抽
table(hoge)["CGGGCCT"]                 #CGGGCCTの出現頻度を表示

table(subseq(fasta, start=1, width=
table(subseq(fasta, start=2, width=
table(subseq(fasta, start=3, width=
table(subseq(fasta, start=4, width=
table(subseq(fasta, start=5, width=
table(subseq(fasta, start=6, width=
table(subseq(fasta, start=7, width=
table(subseq(fasta, start=8, width=
table(subseq(fasta, start=9, width=
table(subseq(fasta, start=10, width=
table(subseq(fasta, start=11, width=
table(subseq(fasta, start=12, width=
table(subseq(fasta, start=13, width=
table(subseq(fasta, start=14, width=
table(subseq(fasta, start=15, width=
```

k=7と部分配列長がfixされている場合は、subseq関数利用時にwidthオプションを用いるほうがシンプル。position 3におけるCGGGCCTの出現回数は41回。position 1の1,804回に比べて大幅に少ない。

R Console

```
> hoge <- subseq(fasta, start=3, width=7)#指定した始点から$
> hoge                                    #確認してるだけで$
  A DNAStringSet instance of length 1000000
           width seq                     names
       [1]     7 CCCGACT                 SRR616268.7 2291:...
       [2]     7 TCTGGGC                 SRR616268.20 2291...
       [3]     7 GGTATAT                 SRR616268.21 2291...
       [4]     7 TCGATAC                 SRR616268.22 2291...
       [5]     7 CCGGTAT                 SRR616268.30 2291...
       ...   ... ...
  [999996]     7 CCGGTAT                 SRR616268.1000856...
  [999997]     7 CGGGTCT                 SRR616268.1000857...
  [999998]     7 TCCATCC                 SRR616268.1000858...
  [999999]     7 AGGGAGT                 SRR616268.1000859...
 [1000000]     7 CTTGTCA                 SRR616268.1000860...
> table(hoge)["CGGGCCT"]                  #CGGGCCTの出現頻$
CGGGCCT
     41
>
```

k-mer解析

position 5におけるCGGGCCTの出現回数は3回。position 1の1,804回に比べて大幅に少ない。このように動作確認をしつつ、徐々に必要最小限の記述に進化させていく。

```
hoge <- subseq(fasta, start=5, width=7)#指定した始点から一定範囲の配列を抽
table(hoge)["CGGGCCT"]                 #CGGGCCTの出現頻度を表示

table(subseq(fasta, start=1, width=7))["CGGGCCT"]#一気に計算
table(subseq(fasta, start=2, width=7))["CGGGCCT"]#一気に計算
table(subseq(fasta, start=3, width=7))["CGGGCCT"]#一気に計算
table(subseq(fasta, start=4, width=7))["CGGGCCT"]#一気に計算
table(subseq(fasta, start=5, width=
table(subseq(fasta, start=6, width=
table(subseq(fasta, start=7, width=
table(subseq(fasta, start=8, width=
table(subseq(fasta, start=9, width=
table(subseq(fasta, start=10, width=
table(subseq(fasta, start=11, width=
table(subseq(fasta, start=12, width=
table(subseq(fasta, start=13, width=
table(subseq(fasta, start=14, width=
table(subseq(fasta, start=15, width=
table(subseq(fasta, start=16, width=
table(subseq(fasta, start=17, width=
table(subseq(fasta, start=18, width=
table(subseq(fasta, start=19, width=
```

R Console

```
      [1]      7 CCCGACT                    SRR616268.7 2291:...
      [2]      7 TCTGGGC                    SRR616268.20 2291...
      [3]      7 GGTATAT                    SRR616268.21 2291...
      [4]      7 TCGATAC                    SRR616268.22 2291...
      [5]      7 CCGGTAT                    SRR616268.30 2291...
      ...    ... ...
 [999996]      7 CCGGTAT                    SRR616268.1000856...
 [999997]      7 CGGGTCT                    SRR616268.1000857...
 [999998]      7 TCCATCC                    SRR616268.1000858...
 [999999]      7 AGGGAGT                    SRR616268.1000859...
[1000000]      7 CTTGTCA                    SRR616268.1000860...
> table(hoge)["CGGGCCT"]                     #CGGGCCTの出現頻$
CGGGCCT
     41
> hoge <- subseq(fasta, start=5, width=7)#指定した始点から$
> table(hoge)["CGGGCCT"]                     #CGGGCCTの出現頻$
CGGGCCT
     3
>
```

k-mer解析

一気にコピペで計算。start=1, …, 35までのCGGGCCTの出現頻度。start=1-3あたりまでは高いがそれ以降は10前後になっていることがわかる。

R Console

```
> table(subseq(fasta, start=29, width=7))["CGGGCCT"]$
CGGGCCT
      2
> table(subseq(fasta, start=30, width=7))["CGGGCCT"]$
CGGGCCT
      3
> table(subseq(fasta, start=31, width=7))["CGGGCCT"]$
CGGGCCT
      7
> table(subseq(fasta, start=32, width=7))["CGGGCCT"]$
CGGGCCT
      6
> table(subseq(fasta, start=33, width=7))["CGGGCCT"]$
CGGGCCT
      1
> table(subseq(fasta, start=34, width=7))["CGGGCCT"]$
CGGGCCT
     10
> table(subseq(fasta, start=35, width=7))["CGGGCCT"]$
CGGGCCT
     13
>
```

k-mer解析

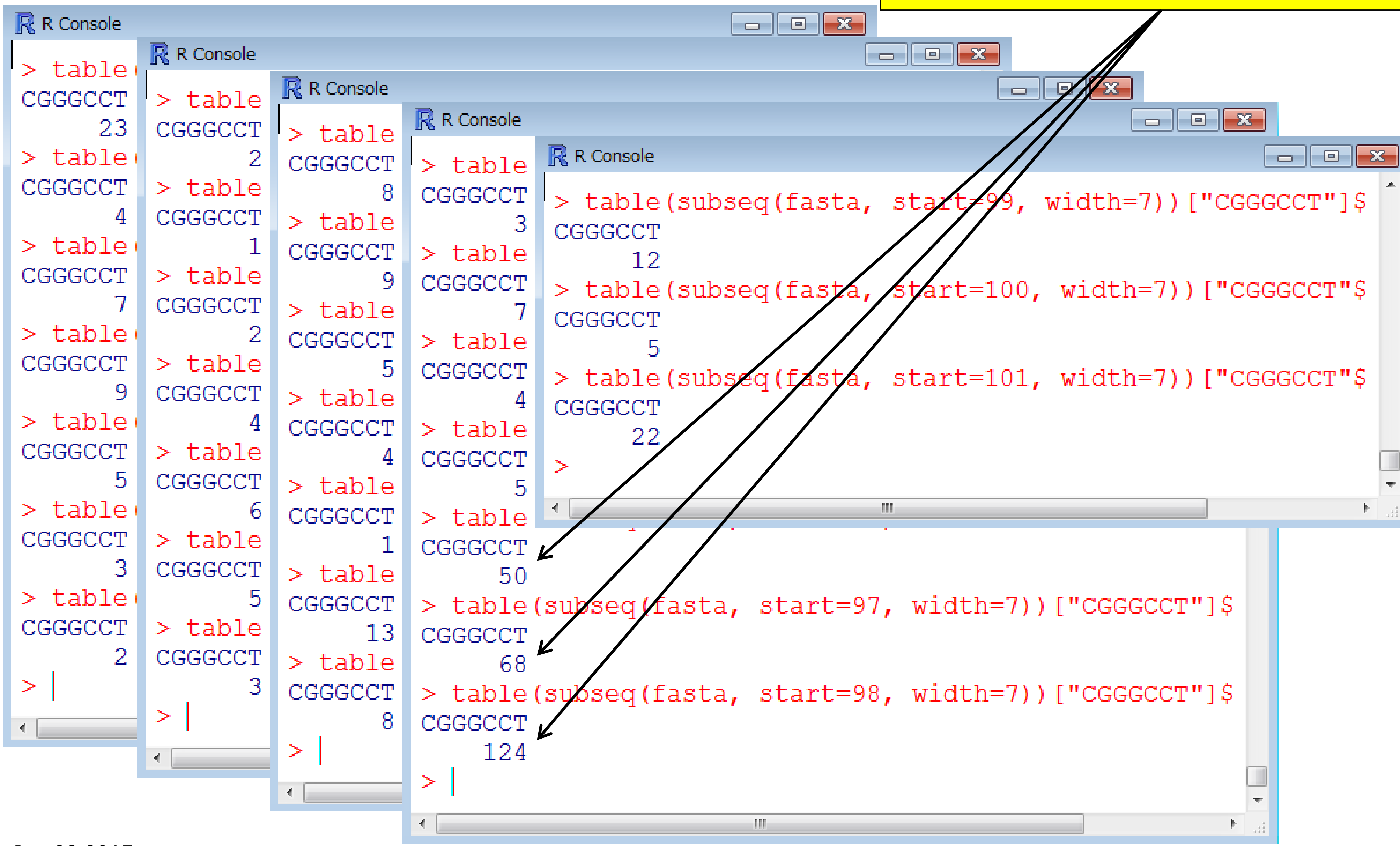
start=36, …, 70までのCGGGCCTの出現頻度。このあたりも10前後と低い出現回数。

R Console

```
> table(subseq(fasta, start=64, width=7))["CGGGCCT"]$
CGGGCCT
     14
> table(subseq(fasta, start=65, width=7))["CGGGCCT"]$
CGGGCCT
     12
> table(subseq(fasta, start=66, width=7))["CGGGCCT"]$
CGGGCCT
      6
> table(subseq(fasta, start=67, width=7))["CGGGCCT"]$
CGGGCCT
      6
> table(subseq(fasta, start=68, width=7))["CGGGCCT"]$
CGGGCCT
      8
> table(subseq(fasta, start=69, width=7))["CGGGCCT"]$
CGGGCCT
      8
> table(subseq(fasta, start=70, width=7))["CGGGCCT"]$
CGGGCCT
      7
>
```

k-mer解析

start=71, …, 101までのCGGGCCTの出現頻度。start=96-98あたりで出現頻度が増加していることがわかる。



k-mer解析

position 1, 2, …, 101までのCGGGCCTの実測した出現頻度をベクトルとしてObsというオブジェクトに格納。①head関数でベクトルObsの最初の6個の要素を表示。②length関数でベクトルObsの要素数(=101)を表示。

R Console

```
> Obs <- c(
+ 1804, 69, 41, 9, 3, 1, 1, # position1-7
+ 3, 6, 4, 11, 7, 18, 8,    # position8-14
+ 6, 7, 12, 4, 4, 15, 4,    # position15-21
+ 8, 2, 9, 3, 3, 1, 6,      # position22-28
+ 2, 3, 7, 6, 1, 10, 13,    # position29-35
+ 12, 9, 12, 6, 8, 17, 3,   # position36-42
+ 4, 13, 4, 5, 2, 7, 10,    # position43-49
+ 9, 10, 3, 2, 6, 6, 9,     # position50-56
+ 7, 14, 14, 45, 22, 16, 16,# position57-63
+ 14, 12, 6, 6, 8, 8, 7,    # position64-70
+ 23, 4, 7, 9, 5, 3, 2,     # position71-77
+ 2, 1, 2, 4, 6, 5, 3,      # position78-84
+ 8, 9, 5, 4, 1, 13, 8,     # position85-91
+ 3, 7, 4, 5, 50, 68, 124,  # position92-98
+ 12, 5, 22)                # position99-101
> head(Obs) ①
[1] 1804   69   41    9    3    1
> length(Obs)②
[1] 101
>
```

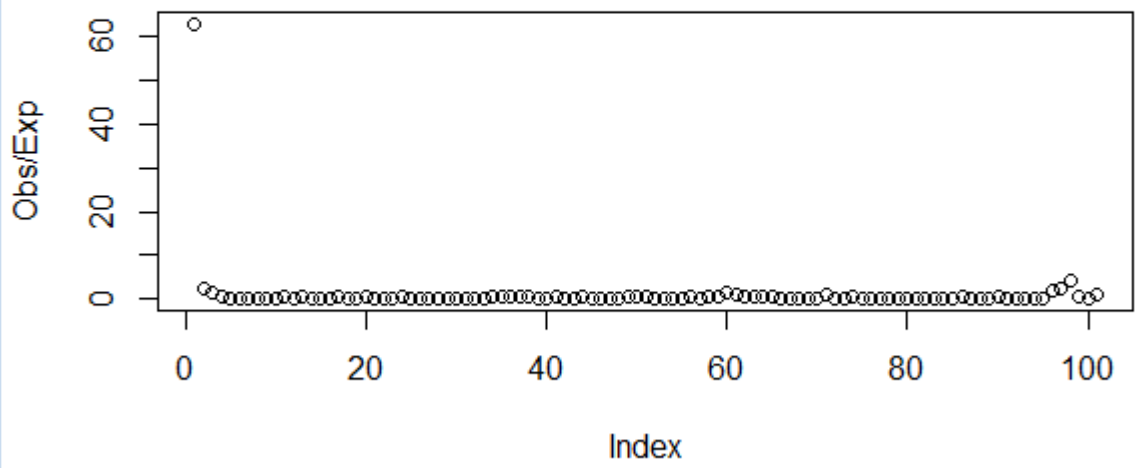
k-mer解析

①平均出現頻度(=期待値; ②Exp)は25.58、③position 1の実測値(=1,804)をExpで割ったものがFastQCのKmer_Contentという項目。ここでは63.11という値が得られている。④R上ではObs/Expとやるだけでよい。⑤plotの基本形で全体像を把握。

R Console

```
> head(Obs)
[1] 1804   69   41    9    3    1
> length(Obs)
[1] 101
> mean(Obs) ①
[1] 28.58416
> Exp <- mean(Obs) ②
> 1804/Exp  ③
[1] 63.11188
> head(Obs)
[1] 1804   69   41    9    3    1
> head(Obs/Exp) ④
[1] 63.11188085  2.41392449  1.43436
[4]  0.31485972  0.10495324  0.03498
> plot(Obs/Exp) ⑤
>
```

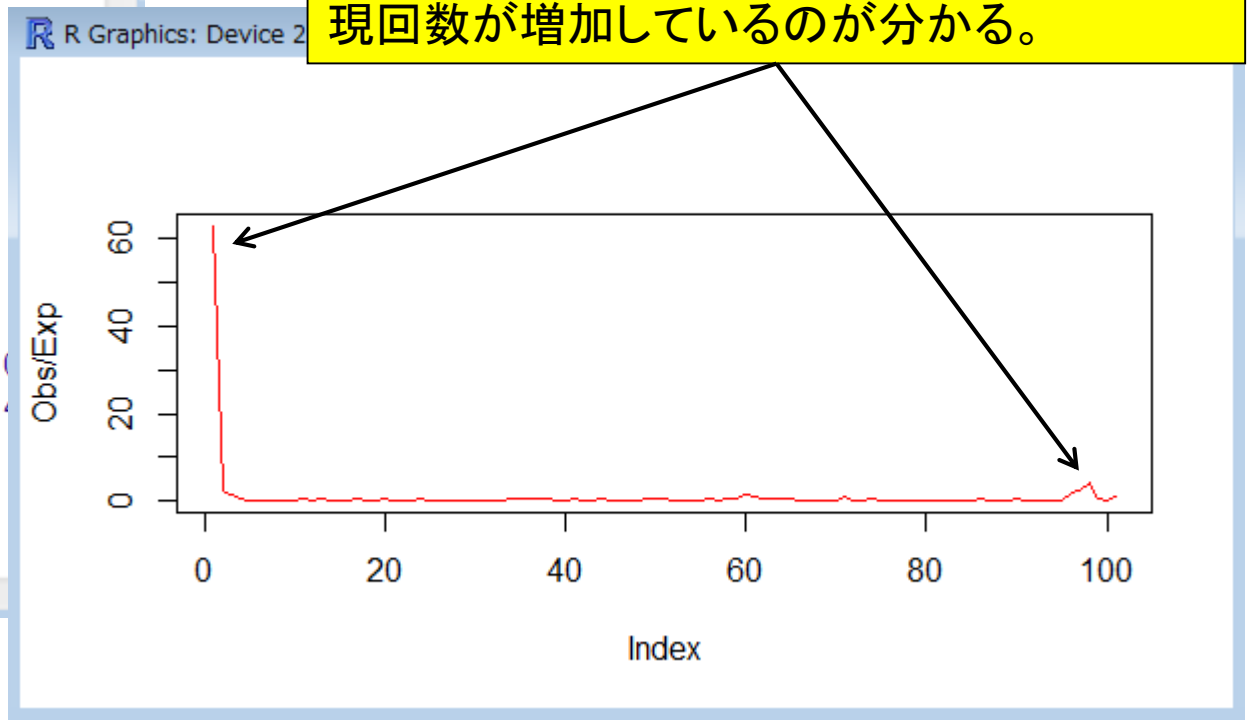
R Graphics: Device 2 (ACTIVE)



k-mer解析

```
R Console
> head(Obs)
[1] 1804   69   41    9    3    1
> length(Obs)
[1] 101
> mean(Obs) ①
[1] 28.58416
> Exp <- mean(Obs) ②
> 1804/Exp  ③
[1] 63.11188
> head(Obs)
[1] 1804   69   41    9    3    1
> head(Obs/Exp) ④
[1] 63.11188085  2.41392449  1.43436
[4]  0.31485972  0.10495324  0.03498
> plot(Obs/Exp) ⑤
⑥ plot(Obs/Exp, type="l", col="red")
>
```

①平均出現頻度(=期待値; ②Exp)は25.58、③position 1の実測値(=1,804)をExpで割ったものがFastQCのKmer_Contentという項目。ここでは63.11という値が得られている。④R上ではObs/Expとやるだけでよい。⑤plotの基本形で全体像を把握。⑥赤の折れ線グラフで再度プロット。position 1と96-98周辺で出現回数が増加しているのが分かる。

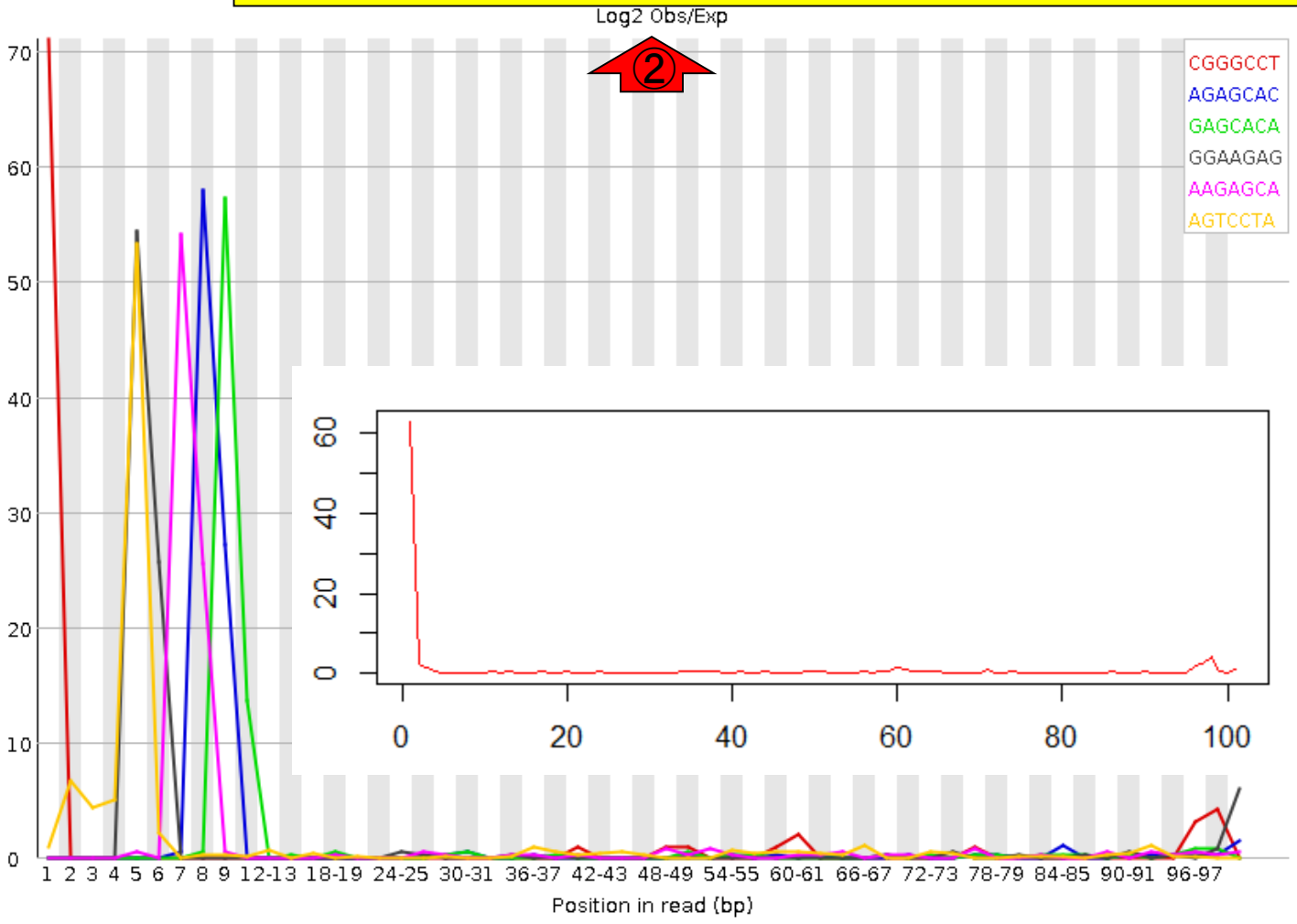


k-mer解析

①FastQCのKmer_Content中のCGGGCCTの結果とよく似ていることが分かります。②FastQC中では「Log2 Obs/Exp」となっていますが、おそらくこれは「Obs/Exp」が正解です。

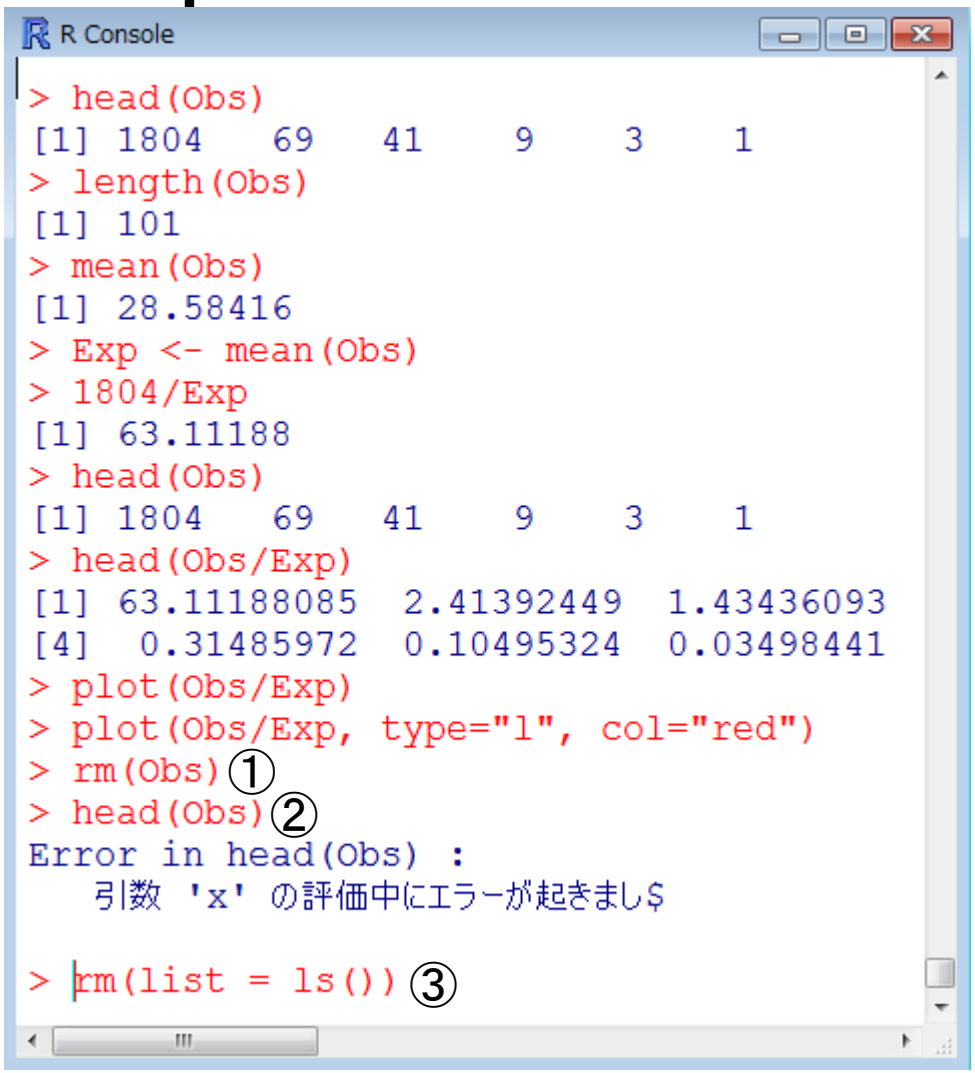
Summary

- 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 ①



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

Tips:オブジェクトの消去



①オブジェクトの消去はrm関数を用いる。rm(Obs)実行後はObsが存在しないので、②head(Obs)をやっても正しくErrorが出る。③このR Console画面上で利用している全オブジェクト(in_f, Exp, fasta, hogeなど)を一度に消去する場合は「rm(list = ls())」。ここでは後の作業に支障がでるのでやらないで。

孫 建強氏提供情報。Thanks to preDr. Sun J.

Tips：forループ

forループで一気にCGGGCCTのposition 1-101の出現回数を計算することもできる。これはforループの基本形（初級）

- 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
- イントロ | NGS | 読み込み | BSgenome | 基本情報を取得 (last modified 2015/06/10)
- イントロ | NGS | 読み込み | FASTA形式 | 基本情報を取得 (last modified 2014/08/18
- イントロ | NGS | 読み込み | FASTA形式 | descrip…nfの記述を整形 (last modified
- イントロ | NGS | 読み込み | FASTQ形式 | 基礎 ①…modified 2015/06/19) NEW
- イントロ | NGS | 読み込み | FASTQ形式 | 応用 (…t modified 2015/06/18) NEW

イントロ | NGS | 読み込み | FASTQ形式 | 基礎 NEW

②

8. FASTQ形式ファイル(SRR616268sub_1.fastq.gz)の場合:

乳酸菌RNA-seqデータSRR616268の最初の100万リード分(約73MB)です。長さは全て107 bpです。

```
in_f <- "SRR616268sub_1.fastq.gz"      #入力ファイル名を指定してin_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
fasta <- readDNAStringSet(in_f, "fastq")
fasta

#以下はおまけ
hoge <- subseq(fasta, start=1, width=7)
hoge
head(table(hoge))
head(sort(table(hoge), decreasing=T))
table(hoge)["CGGGCCT"]

hoge <- subseq(fasta, start=2, width=7)
hoge
table(hoge)["CGGGCCT"]

hoge <- subseq(fasta, start=3, width=7)
table(hoge)["CGGGCCT"]
```

```
table(subseq(fasta, start=99, width=7))["CGGGCCT"]#一気に計算
table(subseq(fasta, start=100, width=7))["CGGGCCT"]#一気に計算
table(subseq(fasta, start=101, width=7))["CGGGCCT"]#一気に計算

#以下はおまけ(forループを用いて美しく...初級)
param_kmer <- 7                        #k-merのkの値を指定
param_obj <- "CGGGCCT"                 #調べたいk-merを指定
Obs <- NULL                            #おまじない
for(i in 1:101){                       #ループを回す
    Obs <- c(Obs, table(subseq(fasta, start=i, width=param_kmer))[param_obj])
}
head(Obs)
mean(Obs)
Exp <- mean(Obs)
head(Obs/Exp)
plot(Obs/Exp, type="l", col="red")
```

Tips：forループ

```
table(subseq(fasta, start=99, width=7))["CGGGCCT"]#一気に計算
table(subseq(fasta, start=100, width=7))["CGGGCCT"]#一気に計算
table(subseq(fasta, start=101, width=7))["CGGGCCT"]#一気に計算

#以下はおまけ(forループを用いて美しく...初級)
param_kmer <- 7                          #k-merのkの値を指定
param_obj <- "CGGGCCT"                   #調べたいk-merを指定
Obs <- NULL                              #おまじない
for(i in 1:101){                         #ループを回す
    Obs <- c(Obs, table(subseq(fasta, start=i, width=param_km
}
head(Obs)
mean(Obs)
Exp <- mean(Obs)
head(Obs/Exp)
plot(Obs/Exp, type="l", col="red")
```

```
#以下はおまけ(forループを用いて美しく...中級)
param_obj <- "CGGGCCT"                   #調べたいk-merを指定
Obs <- NULL                              #おまじない
for(i in 1:101){                         #ループを回す
    Obs <- c(Obs, table(subseq(fasta, start=i, width=nchar(pa
```

「手計算?!で得られた結果」と「forループを用いて得られた結果」の値が同じであることを頼りにこのような記述形式であっていることを確認。tail(Obs)もやったほうがいいかもしれない。

R Console

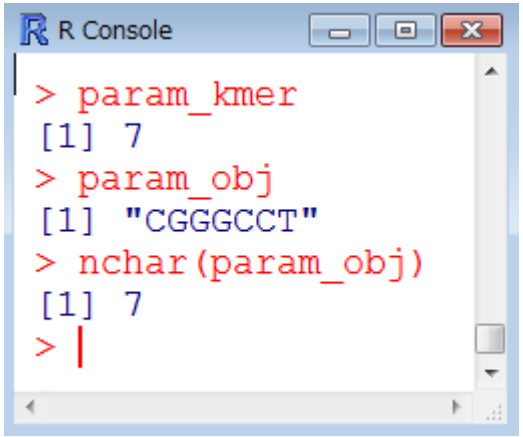
```
> #以下はおまけ(forループを用いて美しく$
> param_kmer <- 7                        $
> param_obj <- "CGGGCCT"                 $
> Obs <- NULL                            $
> for(i in 1:101){                       $
+     Obs <- c(Obs, table(subseq(fasta,$
+ }
> head(Obs)
CGGGCCT CGGGCCT CGGGCCT CGGGCCT CGGGCCT
   1804      69      41       9       3
CGGGCCT
      1
> mean(Obs)
[1] 28.58416
> Exp <- mean(Obs)
> head(Obs/Exp)
    CGGGCCT     CGGGCCT     CGGGCCT
63.11188085  2.41392449  1.43436093
    CGGGCCT     CGGGCCT     CGGGCCT
 0.31485972  0.10495324  0.03498441
> plot(Obs/Exp, type="l", col="red")
>
```

 参考

Tips：スキルアップ

初級 → 中級。なるべく少ない入力項目数で実行。

```
#以下はおまけ(forループを用いて美しく...初級)
param_kmer <- 7                          #k-merのkの値を指定
param_obj <- "CGGGCCT"                   #調べたいk-merを指定
Obs <- NULL                              #おまじない
for(i in 1:101){                         #ループを回す
    Obs <- c(Obs, table(subseq(fasta, start=i, width=param_kmer))[param_obj])
}
head(Obs)
mean(Obs)
Exp <- mean(Obs)
head(Obs/Exp)
plot(Obs/Exp, type="l", col="red")

#以下はおまけ(forループを用いて美しく...中級)
param_obj <- "CGGGCCT"                   #調べたいk-merを指定
Obs <- NULL                              #おまじない
for(i in 1:101){                         #ループを回す
    Obs <- c(Obs, table(subseq(fasta, start=i, width=nchar(param_obj)))[param
}
head(Obs)
mean(Obs)
Exp <- mean(Obs)
```

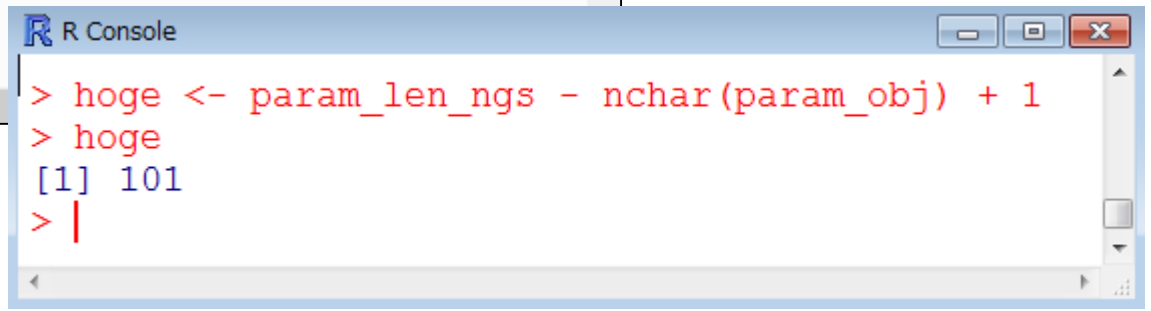


参考

Tips：スキルアップ

```
#以下はおまけ(forループを用いて美しく...中級)
param_obj <- "CGGGCCT"                    #調べたいk-merを指定
Obs <- NULL                               #おまじない
for(i in 1:101){                          #ループを回す
    Obs <- c(Obs, table(subseq(fasta, start=i, width=nchar(par
}
head(Obs)
mean(Obs)
Exp <- mean(Obs)
head(Obs/Exp)
plot(Obs/Exp, type="l", col="red")

#以下はおまけ(forループを用いて美しく...上級1)
param_len_ngs <- 107                      #リード長を指定
param_obj <- "CGGGCCT"                    #調べたいk-merを指定
Obs <- NULL                               #おまじない
hoge <- param_len_ngs - nchar(param_obj) + 1#positionの右端の値を計算してhogeに
for(i in 1:hoge){                          #ループを回す
    Obs <- c(Obs, table(subseq(fasta, start=i, width=nchar(param_obj)))[param
}
head(Obs)
mean(Obs)
```

中級 → 上級1。なるべくプログラム本体を気にすることなく、パラメータ部分をいじるだけで実行させることを心掛ける。中級は、k=7以外のときに101の数値を変更しなければいけないという問題点がある。この101という数値は、「107 − 7 + 1」として計算できる。つまり、リード長Lとkの値が決まればおのずと「L − k + 1」で決まるということ。一見ややこしいが、このほうがパラメータ部分のみしか気にしなくていいので後々心穏やかになれる。



Tips：スキルアップ

上級1 ⇔ 上級2。107 bpというリード長はfastaオブジェクトのwidth列、つまりwidth(fasta)部分で取得可能。全リードの長さは一定という前提条件のもと、ここではwidth(fasta)[1]として、最初のリード長を代表値として採用している。この前提条件を満たさない場合にエラーが出る。それゆえ、上級1程度にしておくほうが門田好み（蓼食う虫も好き好き; some prefer nettles）。

```
#以下はおまけ(forループを用いて美しく...上級1)
param_len_ngs <- 107                    #リード長
param_obj <- "CGGGCCT"                  #調べたい
Obs <- NULL                             #おまじな
hoge <- param_len_ngs - nchar(param_obj) + 1#pos
for(i in 1:hoge){
    Obs <- c(Obs, table(subseq(fasta, st
}
head(Obs)
mean(Obs)
Exp <- mean(Obs)
head(Obs/Exp)
plot(Obs/Exp, type="l", col="red")

#以下はおまけ(forループを用いて美しく...上級
param_obj <- "CGGGCCT"                  #
Obs <- NULL                             #
hoge <- width(fasta)[1] - nchar(param_ob
for(i in 1:hoge){
    Obs <- c(Obs, table(subseq(fasta, st
}
head(Obs)
```

R Conso

```
> fasta
  A DNAStringSet instance of length 1000000
          width seq                     names
      [1]   107 AGCCCGA...CTCTAAC SRR616268.7 2291:...
      [2]   107 GATCTGG...TCCGAAA SRR616268.20 2291...
      [3]   107 CCGGTAT...ATCTTAC SRR616268.21 2291...
      [4]   107 CTTCGAT...AAAGACC SRR616268.22 2291...
      [5]   107 CCCCGGT...GGTCCAT SRR616268.30 2291...
      ...   ... ...
 [999996]   107 CCCCGGT...GGGTTAC SRR616268.1000856...
 [999997]   107 TTCGGGT...GTGAAAT SRR616268.1000857...
 [999998]   107 CGTCCAT...CAAGAGT SRR616268.1000858...
 [999999]   107 CTAGGGA...CTGGACA SRR616268.1000859...
[1000000]   107 GCCTTGT...AGCAGCT SRR616268.1000860...
> width(fasta)[1]
[1] 107
> head(width(fasta))
[1] 107 107 107 107 107 107
> median(width(fasta))
[1] 107
>
```

k-mer解析

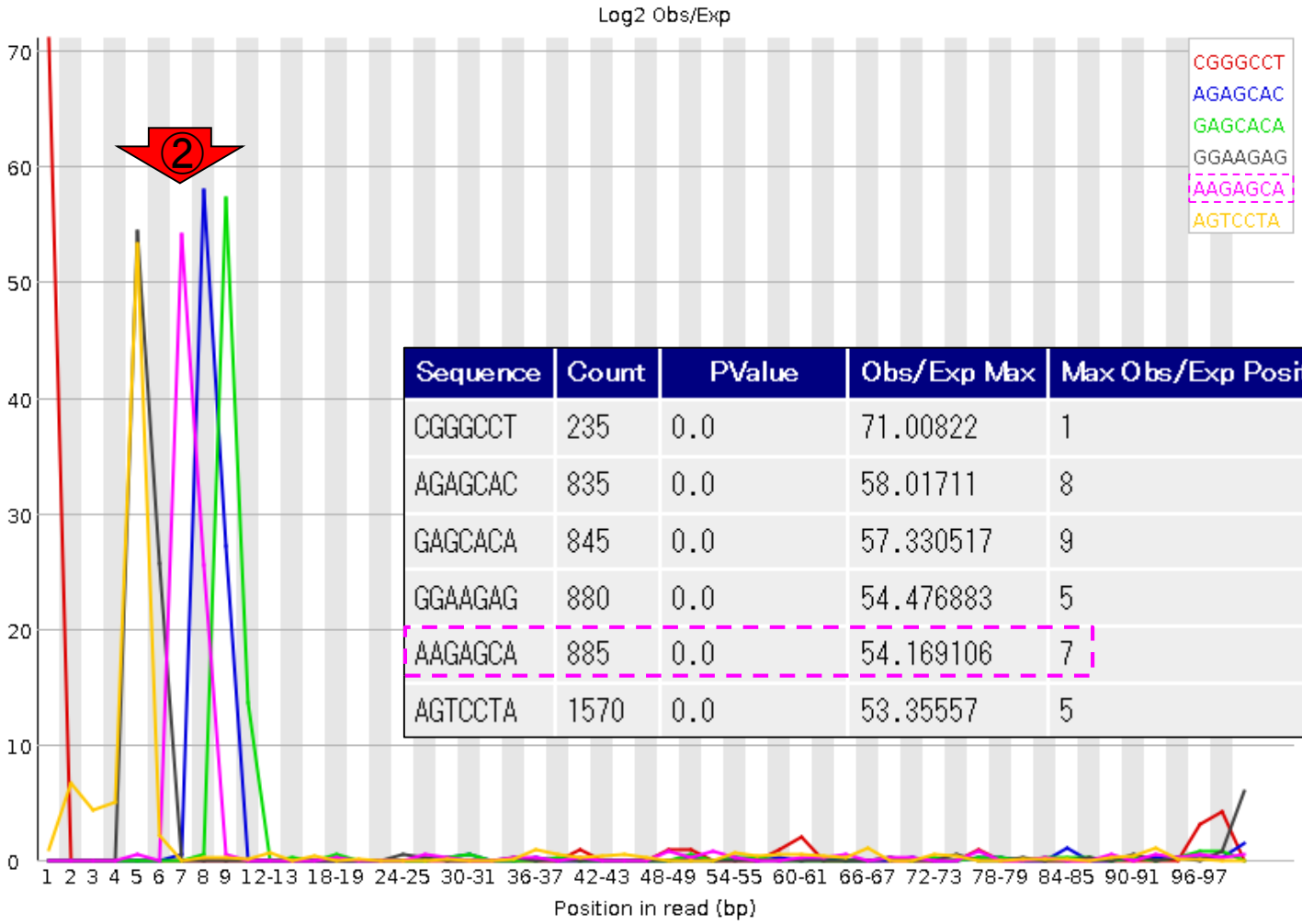
「上級1」のコードをテンプレートにして、Position 7で特異的に出現しているAAGAGCAをRで確認してみよう。

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

②



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

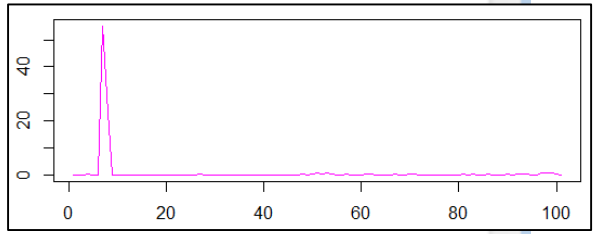
k-mer解析

「上級1」のコードをテンプレートにして、Position 7で特異的に出現しているAAGAGCAをRで確認してみよう。Head関数出力結果は最初の6個（つまりposition 1-6）までしか表示させていないので、ここではn=8としてposition 1-8まで表示させるようにしている。たしかにposition 7で期待値(Exp = 90.09)の55.15倍の出現回数になっていることが分かる。

```
#以下はおまけ(forループを用いて美しく...上級1)
param_len_ngs <- 107                    #リ
param_obj <- "CGGGCCT"                  #調
Obs <- NULL                             #お
hoge <- param_len_ngs - nchar(param_obj) +
for(i in 1:hoge){
    Obs <- c(Obs, table(subseq(fasta,
}
head(Obs)
mean(Obs)
Exp <- mean(Obs)
head(Obs/Exp)
plot(Obs/Exp, type="l", col="red")
```

R Console

```
> param_len_ngs <- 107                    #リード長を$
> param_obj <- "AAGAGCA"                  #調べたいk-m$
> Obs <- NULL                             #おまじない
> hoge <- param_len_ngs - nchar(param_obj) + 1#positi$
> for(i in 1:hoge){                        #ループを回$
+     Obs <- c(Obs, table(subseq(fasta, start=i, widt$
+ }
> head(Obs, n=8)
AAGAGCA AAGAGCA AAGAGCA AAGAGCA AAGAGCA AAGAGCA
      3       6      11      27      11      21
AAGAGCA AAGAGCA
   4968    2096
> mean(Obs)
[1] 90.08911
> Exp <- mean(Obs)
> head(Obs/Exp, n=8)
    AAGAGCA     AAGAGCA     AAGAGCA     AAGAGCA
 0.03330036  0.06660073  0.12210133  0.29970326
    AAGAGCA     AAGAGCA     AAGAGCA     AAGAGCA
 0.12210133  0.23310254 55.14540059 23.26585339
> plot(Obs/Exp, type="l", col="magenta")
>
```

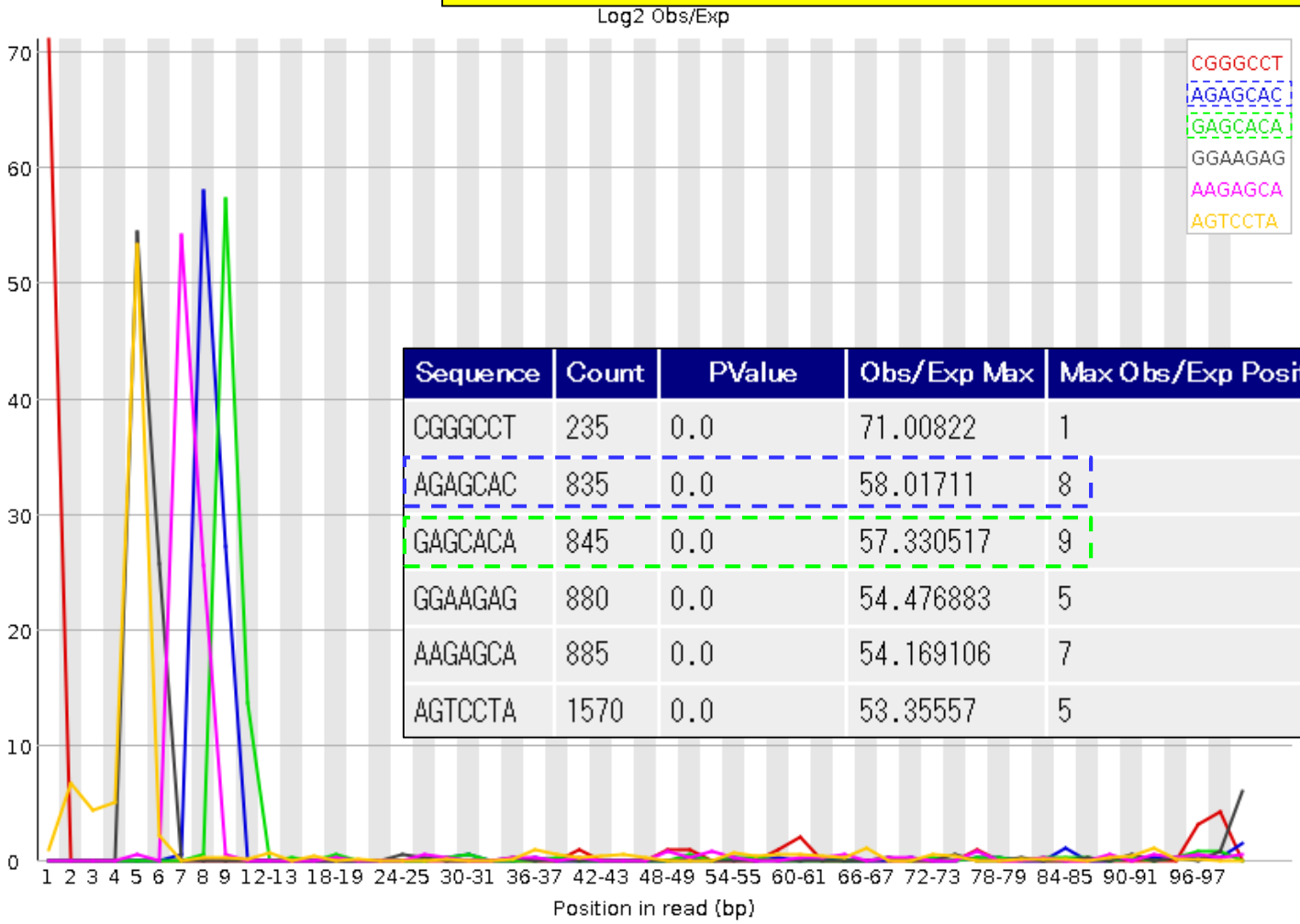


課題1

「上級1」のコードをテンプレートにして、①AGAGCAC、②GAGCACA、③長さを含めて任意、のk-mer解析を行い、得られた結果を示せ。

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



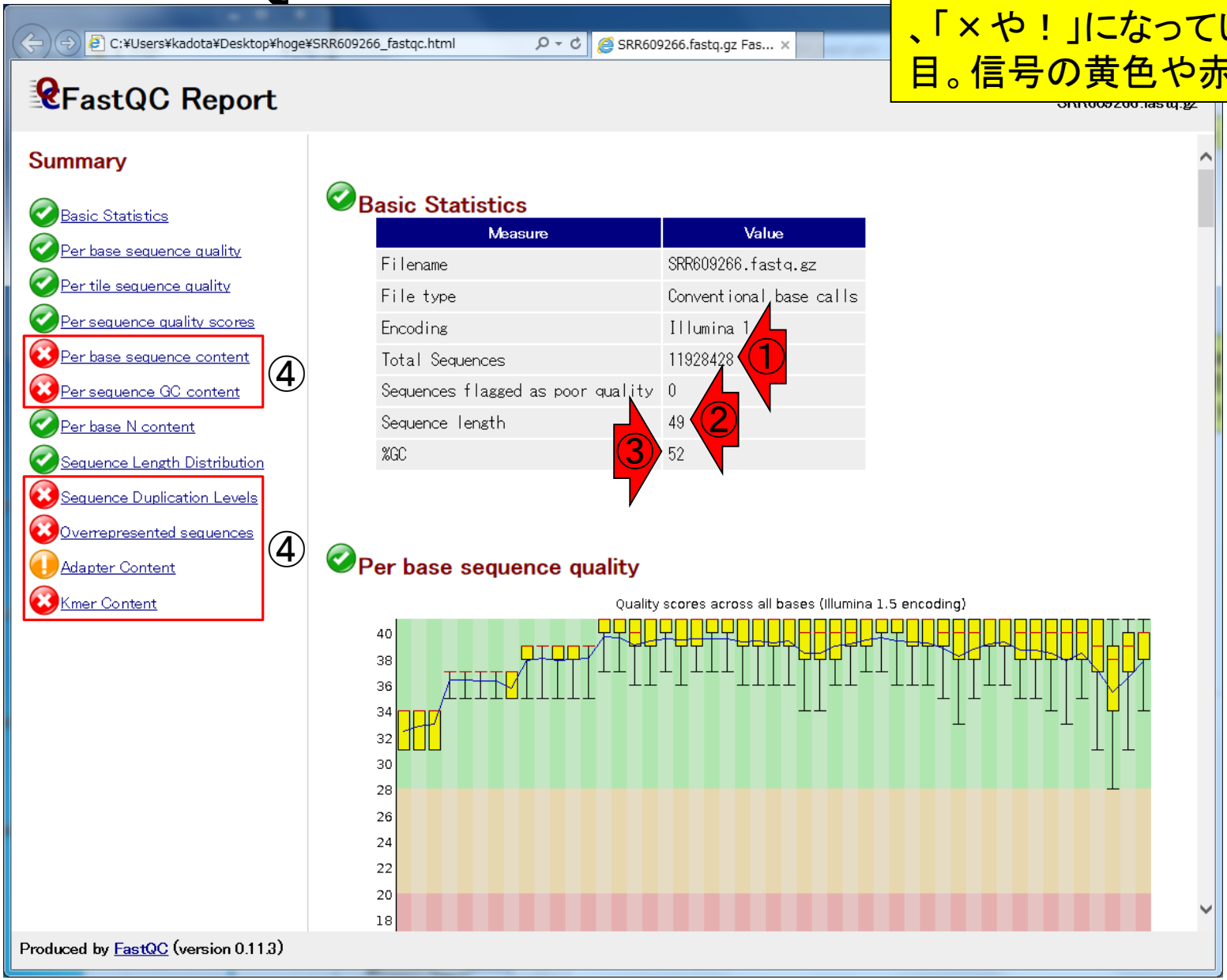
Sequence	Count	PValue	Obs/Exp Max	Max Obs/Exp Position
CGGGCCT	235	0.0	71.00822	1
AGAGCAC	835	0.0	58.01711	8
GAGCACA	845	0.0	57.330517	9
GGAAGAG	880	0.0	54.476883	5
AAGAGCA	885	0.0	54.169106	7
AGTCCTA	1570	0.0	53.35557	5

Contents

- QC(Quality Control)
  - Quality Check (FastQC):乳酸菌RNA-seqデータ
    - k-mer解析、課題1
  - Quality Check (FastQC):カイコsRNA-seqデータ
  - トリミング:カイコsRNA-seqデータ
    - 全体像を把握
    - アダプター配列除去(QuasRパッケージ)、課題2
    - Rを駆使して結果を確認
  - トリミング:乳酸菌RNA-seqデータ
    - サブセットの抽出
    - paired-endデータのアダプター配列除去(QuasRパッケージ)

FastQC結果を眺める

カイコsRNA-seqデータのFastQC結果。①総リード数は11,928,428（約1,200万）。②リードの長さは49 bp。③GC含量は52%。④赤枠の、「×や！」になっているものが要チェック項目。信号の黄色や赤色をイメージすればよい



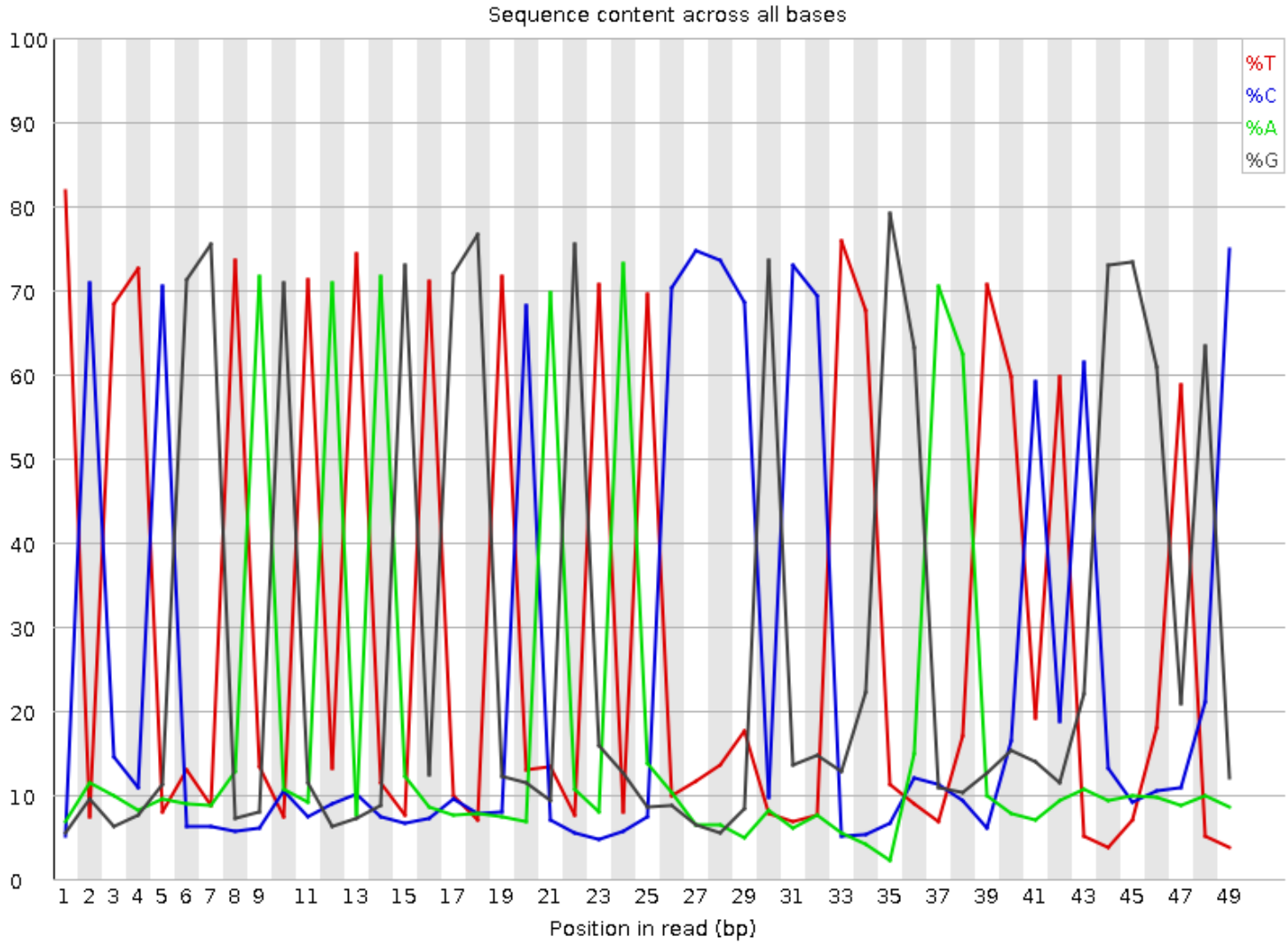
FastQC結果を眺める

①ポジションごとの塩基の出現確率。明らかに変。あたかも全く同じ塩基配列のリード(TCTTCGGTAG…)ばかりが読まれているようなパターンに見える。

Summary

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

Per base sequence content



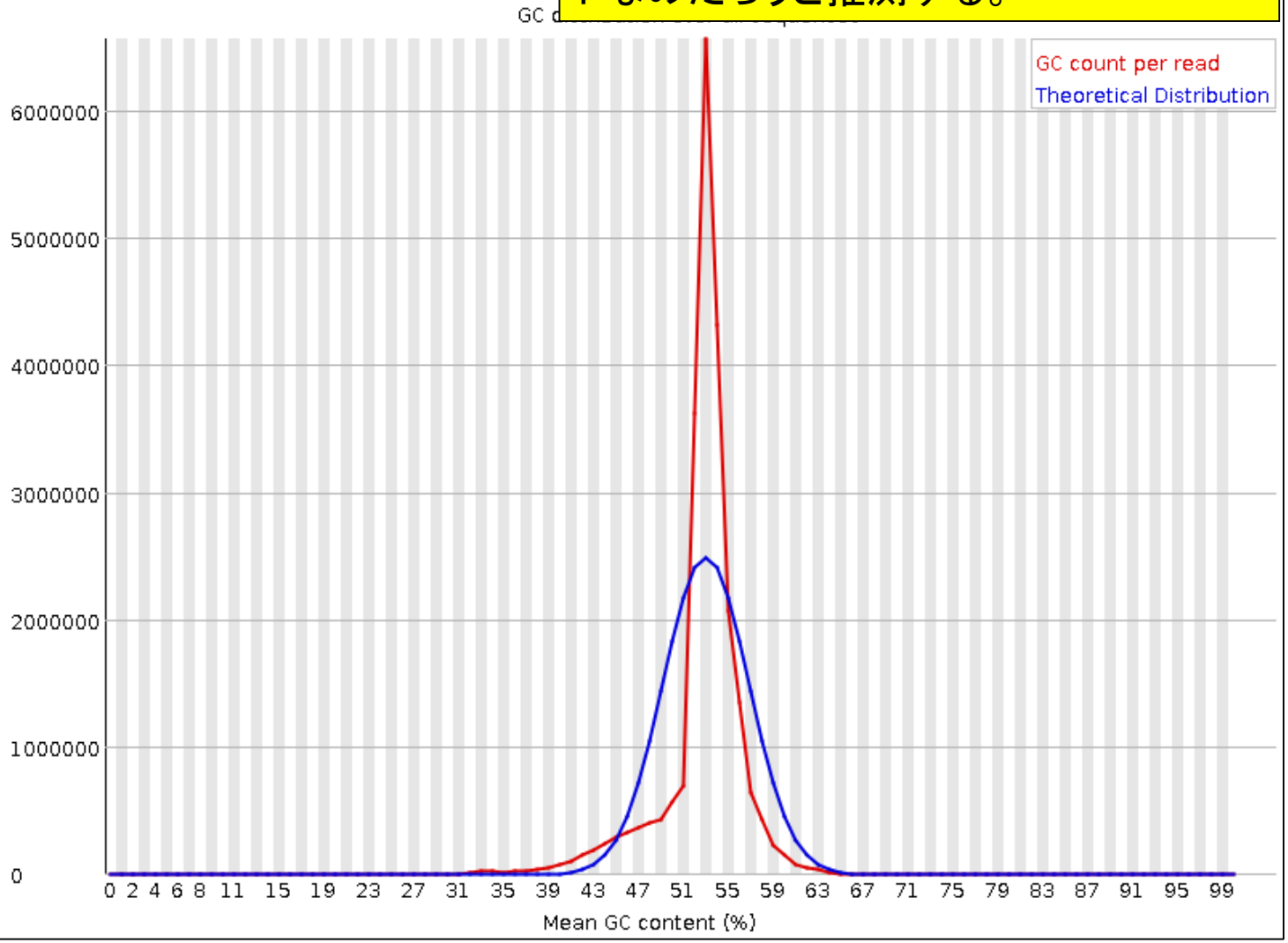
FastQC結果を眺める

おそらく赤色の実測値のピークは52%のところ。TCTTCGGTAG…ばかりが読まれているであろうこと、全部で49 bpであることから25 bp分がG or Cのリードなのだろうと推測する。

Summary

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

Per sequence GC content



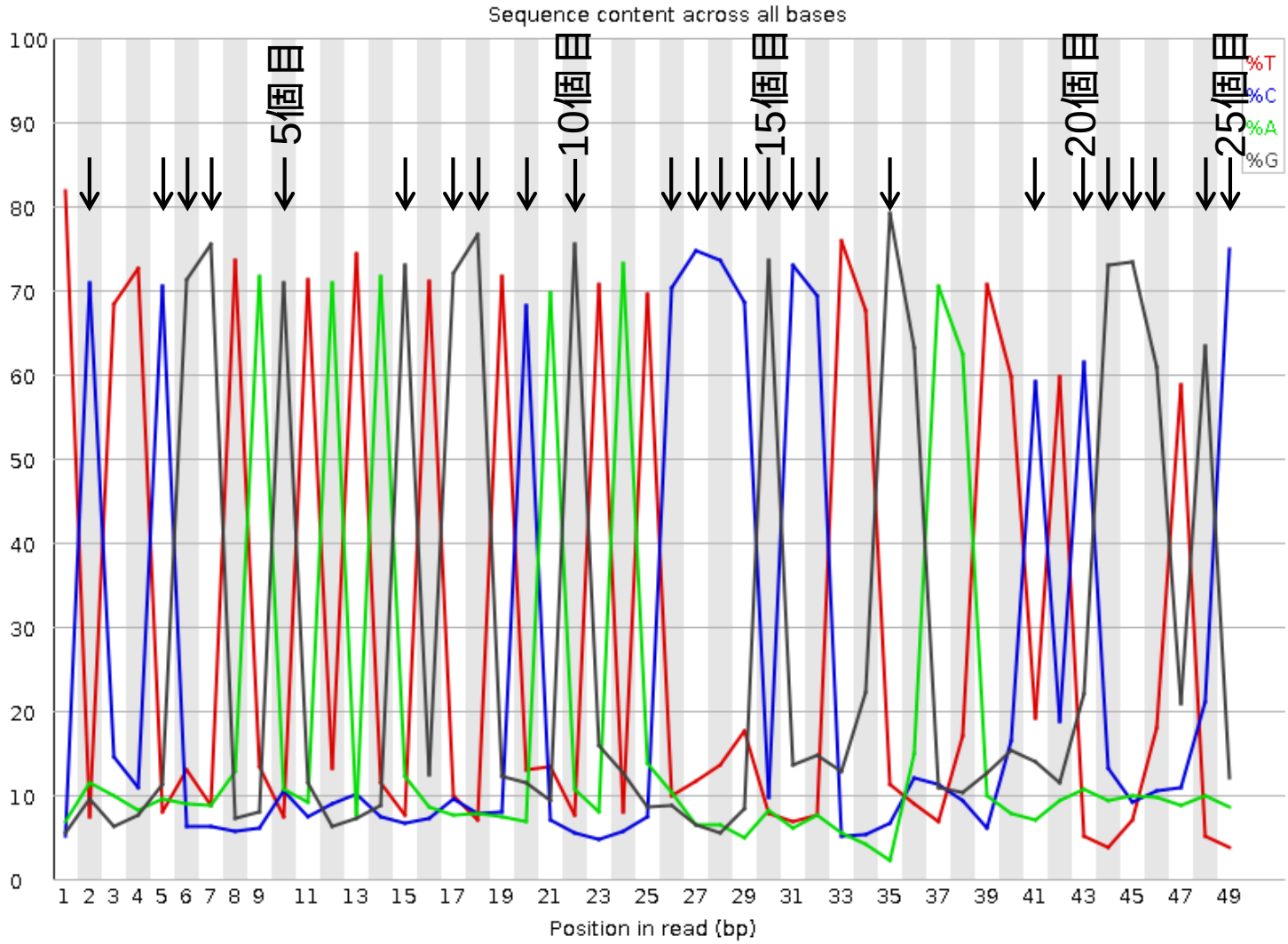
ほらね

FastQC結果を眺める

Summary

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

Per base sequence content



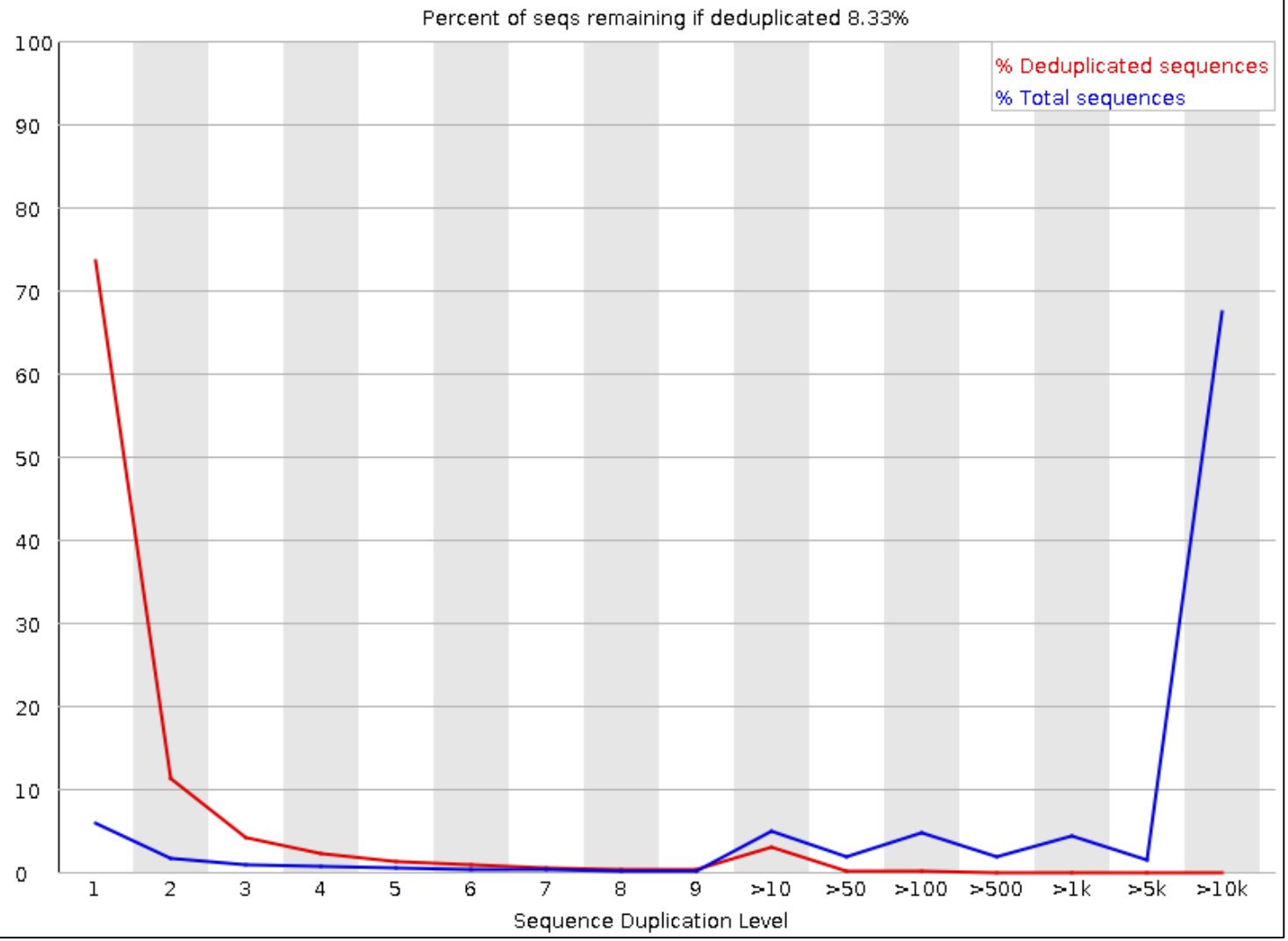
FastQC結果を眺める

私はこの段階でもまだここの解釈が困難です(のちに% Total sequencesは納得した)。
発展課題:% Deduplicated sequencesの意味を説明せよ。

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 ①
- Overrepresented sequences
- Adapter Content
- Kmer Content

Sequence Duplication Levels



FastQC結果を眺める

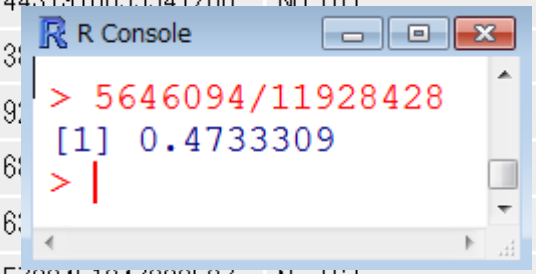
①Overrepresented sequences。総リード数11,928,428のうち、②赤枠の塩基配列からなるリードが5,646,094個(47.3%)を占める。

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 ①
- Adapter Content
- Kmer Content

Overrepresented sequences

Sequence	Count	Percentage	Possible Source
TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTTGGAATTCTCGGGTGC	5646094	47.333093681749176	No Hit ②
TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTGGAATTCTCGGGTGCC	1201120	10.06939053494727	No Hit
GGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	172150	1.443191005554126	No Hit
TGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	164982	1.3	
TCCTCGGTAGTATAGTGGTGAGTATGCACGCCTTGGAATTCTCGGGTGC	110129	0.9	
TCCTCGGTAGTATAGTGGTGAGTATGCTCGCCTGGAATTCTCGGGTGCC	82246	0.6	
GCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGCC	75176	0.6	
TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCGTGGAATTCTCGGGTGC	68069	0.5706451847636587	No Hit
TGGAATTCTCGGGTGCCAAGGAACTCCAGTCACATCACGATCTCGTATG	58409	0.4896621750996862 4	RNA PCR Primer, Index 1 (100% over 49bp)
AAATCATTACCCTGGACGGTGGATCACTGGAATTCTCGGGTGCCAAGGA	38416	0.3220541717651312	RNA PCR Primer, Index 1 (100% over 22bp)
TCCTCGGTAGTATAGTGGTGAGTATGCTCGCCTTGGAATTCTCGGGTGC	35680	0.299117369028006	No Hit
TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTGTGGAATTCTCGGGTG	28678	0.2404172620231266	No Hit
TCCTCGGTAGTATAGTGGTGAGTATGCACGCCTGGAATTCTCGGGTGCC	28671	0.24035857868279037	No Hit
AAATCATTACCCTGGACGGTGGATCACTTGGCTCGCGTGGAATTCTCGG	27757	0.23269621110174787	No Hit
CGGGCAACCCGCTGAAACTCCTTCGTGCTGGGGATTGTGGAATTCTCGG	22712	0.19040228938800652	No Hit
TCCCTGGTTGATCCTGCCTGGAATTCTCGGGTGCCAAGGAACTCCAGTC	19997	0.16764153667189002	RNA PCR Primer, Index 1 (100% over 31bp)



FastQC結果を眺める

もう少し下のほうまで眺める。10,000回以上出現しているリードが結構あることに気づく。そして「10,000は10kと表現」することを思い出し、「Sequence Duplication Levels」の項目の意味についてひらめく。

TCCCTGGTTGATCCTGCCTGGAATTCTCGGGTGCCAAGGAACTCCAGTC	19997	0.1676415366	
TCCTCGGTAGTATAGTGGTGAGTATGCCCGCCTTGGAATTCTCGGGTGC	19932	0.1670966199	
AAATCATTACCCTGGACGGTGGATCACTTGGCTCGCGGTGGAATTCTCG	19650	0.16473251965808067	No Hit
GTCAGTCGATCCTAAGCTCAAGGAGAGATGGAATTCTCGGGTGCCAAGG	19014	0.15940071902181913	Illumina Small RNA Adapter 2 (100% over 21bp)
GACTGGGAGCGTGGCGTCTCCTGTAATTCGGCTATGGAATTCTCGGGTG	17862	0.14974311786934538	No Hit
GTTGGCCTCAGATCAGGGAGGATCACCCGCCGAATTTAAGCTGGAATTC	16808	0.14090708348157863	No Hit
TGGACGGAGAACTGATAAGGGCTGGAATTCTCGGGTGCCAAGGAACTCC	16488	0.13822441649478037	RNA PCR Primer, Index 1 (100% over 27bp)
GGCCGTGATCGTCTAGTGGTTAGGCCCCTACGTTGGAATTCTCGGGTGC	15993	0.13407466599957682	No Hit
CATTTGGATCGCGGAGATCTGGAATTCTCGGGTGCCAAGGAACTCCAGT	15322	0.1284494486616342	RNA PCR Primer, Index 1 (100% over 30bp)
CTTCGGTAGTATAGTGGTCAGTATCCCCGCCTTGGAATTCTCGGGTGCC	15147	0.12698236515322892	No Hit
TGCCGTGATCGTCTAGTGGTTAGGCCCCTACGTTGGAATTCTCGGGTGC	14993	0.12569133166583224	No Hit
AGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	14895	0.12486976490112528	No Hit
TGAGATCATTGTGAAAGCTAATTGGAATTCTCGGGTGCCAAGGAACTCC	14165	0.11874993083749175	RNA PCR Primer, Index 1 (100% over 27bp)
TATTATGATGACAAACAAACTAAGGAACCACTGATTGCATTGGAATTCT	13934	0.11681338060639676	No Hit
TCCTCGGTAGTATAGTGGTCAGTATCCCCGCCTTGGAATTCTCGGGTGC	13352	0.11193428002415741	No Hit
GTTGGCCTCAGATCAGGGAGGATCACCCGCCGAATTTAAGTGGAATTCT	13152	0.11025761315740851	No Hit
AAATCATTACCCTGGACGGTGGATCACTTGGCTCGCGGGTGGAATTCTC	12713	0.10657732938489464	No Hit

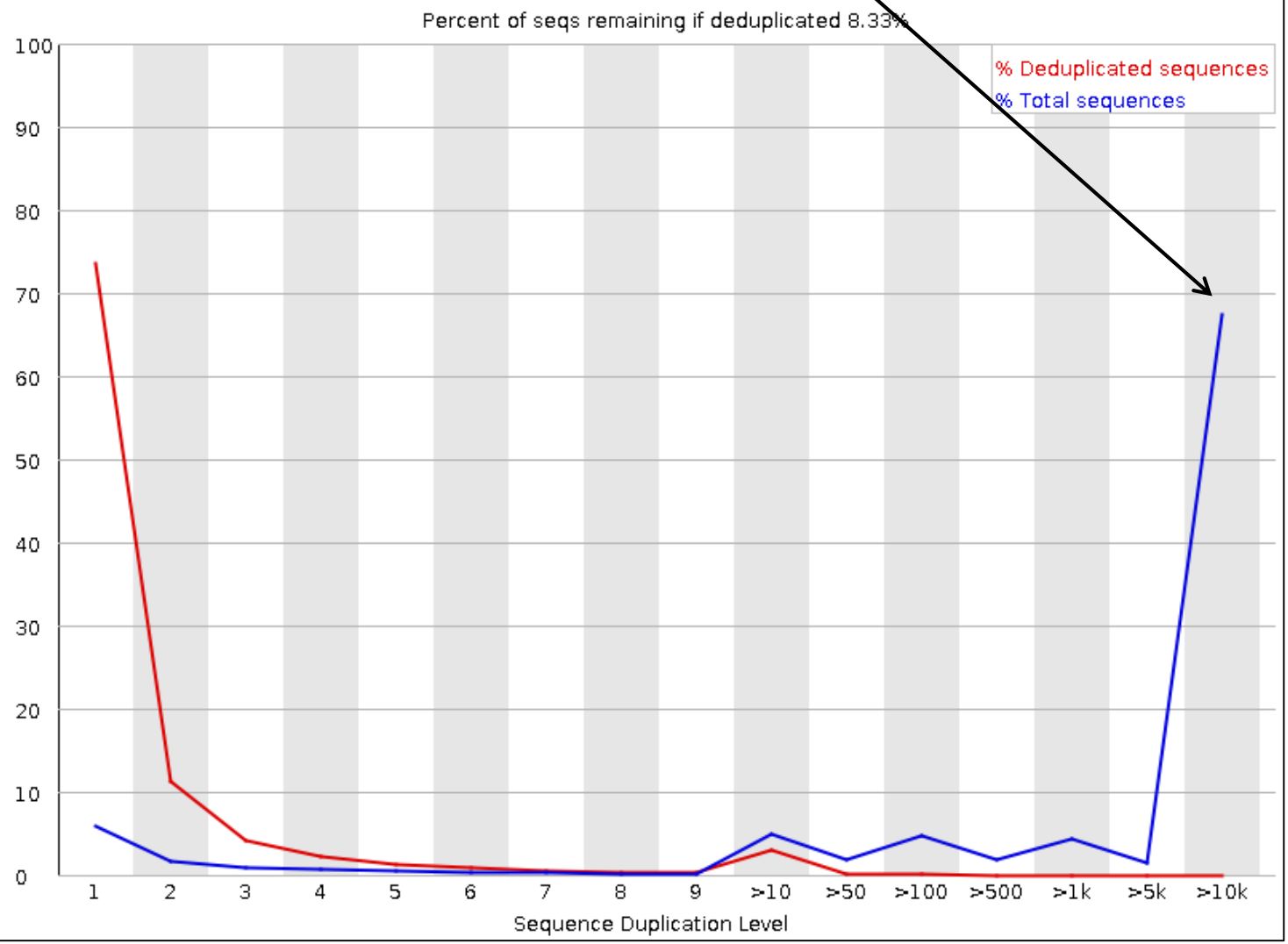
FastQC結果を眺める

つまり、「Sequence Duplication Levelが10kより大きいものが全体(Total sequences; 11,928,428)の70%弱を占めるということを言っているのだろうと予想を立てる。

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 ①
- Overrepresented sequences
- Adapter Content
- Kmer Content

Sequence Duplication Lev



FastQC結果を眺める

FastQCのhtmlレポートで見られるものはCount列が12713個までだが、それを全部足すと8,033,706個(67.35%)。残りの10,000回出現するリードをまで足したとしても、たかが知れている。つまり、70%は超えることはなく、せいぜい68%程度だと考えればリーズナブル。

TCCCTGGTTGATCCTGCCTGGAATTCTCGGGTGCCAAGGAACTCCAGTC	19997	0.1676415	
TCCTCGGTAGTATAGTGGTGAGTATGCCCGCCTTGGAATTCTCGGGTGC	19932	0.1670966	
AAATCATTACCCTGGACGGTGGATCACTTGGCTCGCGGTGGAATTCTCG	19650	0.1647325	
GTCAGTCGATCCTAAGCTCAAGGAGAGATGGAATTCTCGGGTGCCAAGG	19014	0.15940071902181913	Illumina Small RNA Adapter 2 (100% over 21bp)
GACTGGGAGCGTGGCGTCTCCTGTAATTCGGCTATGGAATTCTCGGGTG	17862	0.14974311786934538	No Hit
GTTGGCCTCAGATCAGGGAGGATCACCCGCCGAATTTAAGCTGGAATTC	16808	0.14090708348157863	No Hit
TGGACGGAGAACTGATAAGGGCTGGAATTCTCGGGTGCCAAGGAACTCC	16488	0.1382244164947803?	RNA PCR Primer, Index 1
GGCCGTGATCGTCTAGTGGTTAGGCCCCTACGTTGGAATTCTCGGGTGC	15993	0.1340	
CATTTGGATCGCGGAGATCTGGAATTCTCGGGTGCCAAGGAACTCCAGT	15322	0.1284	
CTTCGGTAGTATAGTGGTCAGTATCCCCGCCTTGGAATTCTCGGGTGCC	15147	0.1269	
TGCCGTGATCGTCTAGTGGTTAGGCCCCTACGTTGGAATTCTCGGGTGC	14993	0.1256	
AGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	14895	0.1248	
TGAGATCATTGTGAAAGCTAATTGGAATTCTCGGGTGCCAAGGAACTCC	14165	0.1187	
TATTATGATGACAAACAAACTAAGGAACCACTGATTGCATTGGAATTCT	13934	0.1168	
TCCTCGGTAGTATAGTGGTCAGTATCCCCGCCTTGGAATTCTCGGGTGC	13352	0.1119	
GTTGGCCTCAGATCAGGGAGGATCACCCGCCGAATTTAAGTGGAATTCT	13152	0.1102	
AAATCATTACCCTGGACGGTGGATCACTTGGCTCGCGGGTGGAATTCTC	12713	0.10657732938489464	No Hit

R Console

```
> 5646094 + 1201120 + 172150 + 164982 +
+ 110129 + 82246 + 75176 + 68069 +
+ 58409 + 38416 + 35680 + 28678 +
+ 28671 + 27757 + 22712 + 19997 +
+ 19932 + 19650 + 19014 + 17862 +
+ 16808 + 16488 + 15993 + 15322 +
+ 15147 + 14993 + 14895 + 14165 +
+ 13934 + 13352 + 13152 + 12713
[1] 8033706
> 8033706/11928428
[1] 0.6734924
>
```

Contents

- QC(Quality Control)
  - Quality Check (FastQC):乳酸菌RNA-seqデータ
    - k-mer解析、課題1
  - Quality Check (FastQC):カイコsRNA-seqデータ
  - トリミング:カイコsRNA-seqデータ
    - 全体像を把握
    - アダプター配列除去(QuasRパッケージ)、課題2
    - Rを駆使して結果を確認
  - トリミング:乳酸菌RNA-seqデータ
    - サブセットの抽出
    - paired-endデータのアダプター配列除去(QuasRパッケージ)

トリミング(Trimming)

原著論文(Nie et al., BMC Genomics, 2013)では、IlluminaのadaptersやsmallRNA primer setを使ったと記載。これとPossible Sourceを比較

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 ①
- Adapter Content
- Kmer Content

Overrepresented sequences

Sequence	Count	Percentage	Possible Source
TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTTGGAATTCTCGGGTGC	5646094	47.333093681749176	No Hit
TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTGGAATTCTCGGGTGCC	1201120	10.06939053494727	No Hit
GGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	172150	1.4431910055541266	No Hit
TGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	164982	1.383099265049845 5	No Hit
TCCTCGGTAGTATAGTGGTGAGTATGCACGCCTTGGAATTCTCGGGTGC	110129	0.923248226840955	No Hit
TCCTCGGTAGTATAGTGGTGAGTATGCTCGCCTGGAATTCTCGGGTGCC	82246	0.6894957156131554	No Hit
GCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGCC	75176	0.6302255418735813	No Hit
TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCGTGGAATTCTCGGGTGC	68069	0.5706451847636587	No Hit
TGGAATTCTCGGGTGCCAAGGAACTCCAGTCACATCACGATCTCGTATG	58409	0.4896621750996862 4	RNA PCR Primer, Index 1 (100% over 49bp)
AAATCATTACCCTGGACGGTGGATCACTGGAATTCTCGGGTGCCAAGGA	38416	0.3220541717651312	RNA PCR Primer, Index 1 (100% over 22bp)
TCCT		8006	No Hit
TCTT		31266	No Hit
TCCT		279037	No Hit
AAAT		174787	No Hit
CGGG		800652	No Hit
TCCC		189002	RNA PCR Primer, Index 1 (100% over 31bp)

RNA-seq

A small RNA library was constructed using the total RNAs extracted above. Briefly, the RNAs were size-fractionated on a 15% polyacrylamide gel, and the 18-50nt fraction was collected. The collected small RNAs were ligated with 5′ and 3′ Illumina adaptors and subsequently used as a template to synthesize first-strand cDNA. The cDNA was amplified by PCR with the Illumina small RNA primer set and sequenced on the

トリミング(Trimming)

①赤下線の配列は「100% over 49bp」という記載から、RNA PCR Primer Index 1という配列の一部なのだろうと判断。

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

Overrepresented sequences

Sequence	Count	Percentage	Possible Source
TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTTGGAATTCTCGGGTGC	5646094	47.333093681749176	No Hit
TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTGGAATTCTCGGGTGCC	1201120	10.06939053494727	No Hit
GGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	172150	1.4431910055541266	No Hit
TGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	164982	1.3830992650498455	No Hit
TCCTCGGTAGTATAGTGGTGAGTATGCACGCCTTGGAATTCTCGGGTGC	110129	0.923248226840955	No Hit
TCCTCGGTAGTATAGTGGTGAGTATGCTCGCCTGGAATTCTCGGGTGCC	82246	0.6894957156131554	No Hit
GCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGCC	75176	0.6302255418735813	No Hit
TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCGTGGAATTCTCGGGTGC	68069	0.5706451847636587	No Hit
① TGGAATTCTCGGGTGCCAAGGAACTCCAGTCACATCACGATCTCGTATG	58409	0.4896621750996862 4	RNA PCR Primer, Index 1 (100% over 49bp)
AAATCATTACCCTGGACGGTGGATCACTGGAATTCTCGGGTGCCAAGGA	38416	0.3220541717651312	RNA PCR Primer, Index 1 (100% over 22bp)
TCCTCGGTAGTATAGTGGTGAGTATGCTCGCCTTGGAATTCTCGGGTGC	35680	0.299117369028006	No Hit
TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTGTGGAATTCTCGGGTG	28678	0.2404172620231266	No Hit
TCCTCGGTAGTATAGTGGTGAGTATGCACGCCTGGAATTCTCGGGTGCC	28671	0.24035857868279037	No Hit
AAATCATTACCCTGGACGGTGGATCACTTGGCTCGCGTGGAATTCTCGG	27757	0.23269621110174787	No Hit
CGGGCAACCCGCTGAAACTCCTTCGTGCTGGGGATTGTGGAATTCTCGG	22712	0.19040228938800652	No Hit
TCCCTGGTTGATCCTGCCTGGAATTCTCGGGTGCCAAGGAACTCCAGTC	19997	0.16764153667189002	RNA PCR Primer, Index 1 (100% over 31bp)

トリミング(Trimming)

②のリードは「100% over 22bp」という記載から、リード中の22 bp分がRNA PCR Primer Index 1という配列の一部と100%一致なのだろうと判断。つまり、②のリードの黒下線部分以外がsmall RNA配列だろうと判断。

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

Overrepresented sequences

	Sequence			
	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTTGGAATTCTCGGGTGC	5646094	47.333093681749176	No Hit
	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTGGAATTCTCGGGTGCC	1201120	10.06939053494727	No Hit
	GGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	172150	1.4431910055541266	No Hit
	TGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	164982	1.3830992650498455	No Hit
	TCCTCGGTAGTATAGTGGTGAGTATGCACGCCTTGGAATTCTCGGGTGC	110129	0.923248226840955	No Hit
	TCCTCGGTAGTATAGTGGTGAGTATGCTCGCCTGGAATTCTCGGGTGCC	82246	0.6894957156131554	No Hit
	GCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGCC	75176	0.6302255418735813	No Hit
	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCGTGGAATTCTCGGGTGC	68069	0.5706451847636587	No Hit
①	TGGAATTCTCGGGTGCCAAGGAACTCCAGTCACATCACGATCTCGTATG	58409	0.4896621750996862 4	RNA PCR Primer, Index 1 (100% over 49bp)
②	AAATCATTACCCTGGACGGTGGATCACTGGAATTCTCGGGTGCCAAGGA	38416	0.3220541717651312	RNA PCR Primer, Index 1 (100% over 22bp)
	TCCTCGGTAGTATAGTGGTGAGTATGCTCGCCTTGGAATTCTCGGGTGC	35680	0.299117369028006	No Hit
	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTGTGGAATTCTCGGGTG	28678	0.2404172620231266	No Hit
	TCCTCGGTAGTATAGTGGTGAGTATGCACGCCTGGAATTCTCGGGTGCC	28671	0.24035857868279037	No Hit
	AAATCATTACCCTGGACGGTGGATCACTTGGCTCGCGTGGAATTCTCGG	27757	0.23269621110174787	No Hit
	CGGGCAACCCGCTGAAACTCCTTCGTGCTGGGGATTGTGGAATTCTCGG	22712	0.19040228938800652	No Hit
	TCCCTGGTTGATCCTGCCTGGAATTCTCGGGTGCCAAGGAACTCCAGTC	19997	0.16764153667189002	RNA PCR Primer, Index 1 (100% over 31bp)

トリミング(Trimming)

③のリードについても同じ流れで黒下線部分の22 bpの配列を同定可能。「100% over 31bp」と書いてあるので、黒下線右側を①のリードと見比べれば理解しやすい。

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

Overrepresented sequences

	Sequence	Count	Percentage	Possible Source
	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTTGGAATTCTCGGGTGC	5646094	47.333093681749176	No Hit
	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTGGAATTCTCGGGTGCC	1201120	10.06939053494727	No Hit
	GGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	172150	1.4431910055541266	No Hit
	TGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	164982	1.3830992650498455	No Hit
	TCCTCGGTAGTATAGTGGTGAGTATGCACGCCTTGGAATTCTCGGGTGC	110129	0.923248226840955	No Hit
	TCCTCGGTAGTATAGTGGTGAGTATGCTCGCCTGGAATTCTCGGGTGCC	82246	0.6894957156131554	No Hit
	GCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGCC	75176	0.6302255418735813	No Hit
	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCGTGGAATTCTCGGGTGC	68069	0.5706451847636587	No Hit
①	TGGAATTCTCGGGTGCCAAGGAACTCCAGTCACATCACGATCTCGTATG	58409	0.4896621750996862 4	RNA PCR Primer, Index 1 (100% over 49bp)
②	AAATCATTACCCTGGACGGTGGATCACTGGAATTCTCGGGTGCCAAGGA	38416	0.3220541717651312	RNA PCR Primer, Index 1 (100% over 22bp)
	TCCTCGGTAGTATAGTGGTGAGTATGCTCGCCTTGGAATTCTCGGGTGC	35680	0.299117369028006	No Hit
	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTGTGGAATTCTCGGGTG	28678	0.2404172620231266	No Hit
	TCCTCGGTAGTATAGTGGTGAGTATGCACGCCTGGAATTCTCGGGTGCC	28671	0.24035857868279037	No Hit
	AAATCATTACCCTGGACGGTGGATCACTTGGCTCGCGTGGAATTCTCGG	27757	0.23269621110174787	No Hit
	CGGGCAACCCGCTGAAACTCCTTCGTGCTGGGGATTGTGGAATTCTCGG	22712	0.19040228938800652	No Hit
③	TCCCTGGTTGATCCTGCCTGGAATTCTCGGGTGCCAAGGAACTCCAGTC	19997	0.16764153667189002	RNA PCR Primer, Index 1 (100% over 31bp)

トリミング(Trimming)

RNA PCR Primer Index 1の最初の22 bpをコピーして文字列検索。

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

Overrepresented sequences

	Sequence	Count	Percentage	Possible Source
	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTTGGAATTCTCGGGTGC	5646094	47.333093681749176	No Hit
	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTGGAATTCTCGGGTGCC	1201120	10.06939053494727	No Hit
	GGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	172150	1.4431910055541266	No Hit
	TGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	164982	1.3830992650498455	No Hit
	TCCTCGGTAGTATAGTGGTGAGTATGCACGCCTTGGAATTCTCGGGTGC	110129	0.923248226840955	No Hit
	TCCTCGGTAGTATAGTGGTGAGTATGCTCGCCTGGAATTCTCGGGTGCC	82246	0.6894957156131554	No Hit
	GCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGCC	75176	0.6302255418735813	No Hit
	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCGTGGAATTCTCGGGTGC	68069	0.5706451847636587	No Hit
①	TGGAATTCTCGGGTGCCAAGGAACTCCAGTCACATCACGATCTCGTATG	58409	0.4896621750996862 4	RNA PCR Primer, Index 1 (100% over 49bp)
②	AAATCATTACCCTGGACGGTGGATCACTGGAATTCTCGGGTGCCAAGGA	38416	0.3220541717651312	RNA PCR Primer, Index 1 (100% over 22bp)
	TCCTCGGTAGTATAGTGGTGAGTATGCTCGCCTTGGAATTCTCGGGTGC	35680	0.299117369028006	No Hit
	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTGTGGAATTCTCGGGTG	28678	0.2404172620231266	No Hit
	TCCTCGGTAGTATAGTGGTGAGTATGCACGCCTGGAATTCTCGGGTGCC	28671	0.24035857868279037	No Hit
	AAATCATTACCCTGGACGGTGGATCACTTGGCTCGCGTGGAATTCTCGG	27757	0.23269621110174787	No Hit
	CGGGCAACCCGCTGAAACTCCTTCGTGCTGGGGATTGTGGAATTCTCGG	22712	0.19040228938800652	No Hit
③	TCCCTGGTTGATCCTGCCTGGAATTCTCGGGTGCCAAGGAACTCCAGTC	19997	0.16764153667189002	RNA PCR Primer, Index 1 (100% over 31bp)

トリミング(Trimming)

RNA PCR Primer Index 1の最初の22 bpをコピーして文字列検索。④⑤⑥の結果ともに妥当。

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

③	TCCTCGGTAGTATAGTGGTGAGTATGCCCGCCTTGGAATTCTCGGGTGC	19932	0.16709661994019664	No Hit
	AAATCATTACCCTGGACGGTGGATCACTTGGCTCGCGGTGGAATTCTCG	19650	0.16473251965808067	No Hit
	GTCAGTCGATCCTAAGCTCAAGGAGAGATGGAATTCTCGGGTGCCAAGG	19014	0.15940071902181913	Illumina Small RNA Adapter 2 (100% over 21bp)
	GACTGGGAGCGTGGCGTCTCCTGTAATTCGGCTATGGAATTCTCGGGTG	17862	0.14974311786934538	No Hit
	GTTGGCCTCAGATCAGGGAGGATCACCCGCCGAATTTAAGCTGGAATTC	16808	0.14090708348157863	No Hit
④	TGGACGGAGAACTGATAAGGGCTGGAATTCTCGGGTGCCAAGGAACTCC	16488	0.13822441649478037	RNA PCR Primer, Index 1 (100% over 27bp)
	GGCCGTGATCGTCTAGTGGTTAGGCCCCTACGTTGGAATTCTCGGGTGC	15993	0.13407466599957682	No Hit
⑤	CATTTGGATCGCGGAGATCTGGAATTCTCGGGTGCCAAGGAACTCCAGT	15322	0.1284494486616342	RNA PCR Primer, Index 1 (100% over 30bp)
	CTTCGGTAGTATAGTGGTCAGTATCCCCGCCTTGGAATTCTCGGGTGCC	15147	0.12698236515322892	No Hit
	TGCCGTGATCGTCTAGTGGTTAGGCCCCTACGTTGGAATTCTCGGGTGC	14993	0.12569133166583224	No Hit
	AGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	14895	0.12486976490112528	No Hit
⑥	TGAGATCATTGTGAAAGCTAATTGGAATTCTCGGGTGCCAAGGAACTCC	14165	0.11874993083749175	RNA PCR Primer, Index 1 (100% over 27bp)
	TATTATGATGACAAACAAACTAAGGAACCACTGATTGCATTGGAATTCT	13934	0.11681338060639676	No Hit
	TCCTCGGTAGTATAGTGGTCAGTATCCCCGCCTTGGAATTCTCGGGTGC	13352	0.11193428002415741	No Hit
	GTTGGCCTCAGATCAGGGAGGATCACCCGCCGAATTTAAGTGGAATTCT	13152	0.11025761315740851	No Hit
	AAATCATTACCCTGGACGGTGGATCACTTGGCTCGCGGGTGGAATTCTC	12713	0.10657732938489464	No Hit

トリミング(Trimmin

RNA PCR Primer Index 1の最初の22 bpをコピーして文字列検索。⑦はこれまでと違ってIllumina Small RNA Adapter 2 (100% over 21bp)という記述。しかし、⑦のリードの右側21 bpは検索文字列の左側21 bpと全く同じことに気づく。Illumina Small RNA Adapter 2というのがどこにあるかは(ちゃんと調べてないので)不明。

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

	TCCTCGGTAGTATAGTGGTC			
	AAATCATTACCCTGGACGGT			
⑦	GTCAGTCGATCCTAAGCTCAAGGAGAGATGGAATTCTCGGGTGCCAAGG	19014	0.15940071902181913	Illumina Small RNA Adapter 2 (100% over 21bp)
	GACTGGGAGCGTGGCGTCTCCTGTAATTCGGCTATGGAATTCTCGGGTG	17862	0.14974311786934538	No Hit
	GTTGGCCTCAGATCAGGGAGGATCACCCGCCGAATTTAAGCTGGAATTC	16808	0.14090708348157863	No Hit
	TGGACGGAGAACTGATAAGGGCTGGAATTCTCGGGTGCCAAGGAACTCC	16488	0.13822441649478037	RNA PCR Primer, Index 1 (100% over 27bp)
	GGCCGTGATCGTCTAGTGGTTAGGCCCCTACGTTGGAATTCTCGGGTGC	15993	0.13407466599957682	No Hit
	CATTTGGATCGCGGAGATCTGGAATTCTCGGGTGCCAAGGAACTCCAGT	15322	0.1284494486616342	RNA PCR Primer, Index 1 (100% over 30bp)
	CTTCGGTAGTATAGTGGTCAGTATCCCCGCCTTGGAATTCTCGGGTGCC	15147	0.12698236515322892	No Hit
	TGCCGTGATCGTCTAGTGGTTAGGCCCCTACGTTGGAATTCTCGGGTGC	14993	0.12569133166583224	No Hit
	AGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	14895	0.12486976490112528	No Hit
	TGAGATCATTGTGAAAGCTAATTGGAATTCTCGGGTGCCAAGGAACTCC	14165	0.11874993083749175	RNA PCR Primer, Index 1 (100% over 27bp)
	TATTATGATGACAAACAAACTAAGGAACCACTGATTGCATTGGAATTCT	13934	0.11681338060639676	No Hit
	TCCTCGGTAGTATAGTGGTCAGTATCCCCGCCTTGGAATTCTCGGGTGC	13352	0.11193428002415741	No Hit
	GTTGGCCTCAGATCAGGGAGGATCACCCGCCGAATTTAAGTGGAATTCT	13152	0.11025761315740851	No Hit
	AAATCATTACCCTGGACGGTGGATCACTTGGCTCGCGGGTGGAATTCTC	12713	0.10657732938489464	No Hit

トリミング(Trimmi

⑧よく見ると、22 bp分の長さの黒下線より短い「RNA PCR Primer Index 1の左側の部分配列」がOverrepresented sequencesの上位の右側に存在することがわかる。これらはおそらく短いためにNo Hitとなったのであろう。

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

Overrepresente

	Sequence	Count	Percentage	Possible Source
⑧	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTTGGAATTCTCGGGTGC	5646094	47.333093681749176	No Hit
	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTGGAATTCTCGGGTGCC	1201120	10.06939053494727	No Hit
	GGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	172150	1.4431910055541266	No Hit
	TGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	164982	1.3830992650498455	No Hit
	TCCTCGGTAGTATAGTGGTGAGTATGCACGCCTTGGAATTCTCGGGTGC	110129	0.923248226840955	No Hit
	TCCTCGGTAGTATAGTGGTGAGTATGCTCGCCTGGAATTCTCGGGTGCC	82246	0.6894957156131554	No Hit
	GCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGCC	75176	0.6302255418735813	No Hit
	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCGTGGAATTCTCGGGTGC	68069	0.5706451847636587	No Hit
①	TGGAATTCTCGGGTGCCAAGGAACTCCAGTCACATCACGATCTCGTATG	58409	0.4896621750996862 4	RNA PCR Primer, Index 1 (100% over 49bp)
②	AAATCATTACCCTGGACGGTGGATCACTGGAATTCTCGGGTGCCAAGGA	38416	0.3220541717651312	RNA PCR Primer, Index 1 (100% over 22bp)
	TCCTCGGTAGTATAGTGGTGAGTATGCTCGCCTTGGAATTCTCGGGTGC	35680	0.299117369028006	No Hit
	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTGTGGAATTCTCGGGTG	28678	0.2404172620231266	No Hit
	TCCTCGGTAGTATAGTGGTGAGTATGCACGCCTGGAATTCTCGGGTGCC	28671	0.24035857868279037	No Hit
	AAATCATTACCCTGGACGGTGGATCACTTGGCTCGCGTGGAATTCTCGG	27757	0.23269621110174787	No Hit
	CGGGCAACCCGCTGAAACTCCTTCGTGCTGGGGATTGTGGAATTCTCGG	22712	0.19040228938800652	No Hit
③	TCCCTGGTTGATCCTGCCTGGAATTCTCGGGTGCCAAGGAACTCCAGTC	19997	0.16764153667189002	RNA PCR Primer, Index 1 (100% over 31bp)

トリミング(Trimmi

これらの部分文字列をハイライトさせるべく、「黒よりも短く、赤枠をハイライトさせられる程度の長さの部分文字列を用いて調べる。①の左側の一部をコピーして検索。黄色でハイライトされている部分より右側でもきれいに一致していることがわかる。

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

Overrepresent

⑧	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTTGGAATTCTCGGGTGC	5646094	47.333093681749176	No Hit
	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTGGAATTCTCGGGTGCC	1201120	10.06939053494727	No Hit
	GGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	172150	1.4431910055541266	No Hit
	TGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	164982	1.3830992650498455	No Hit
	TCCTCGGTAGTATAGTGGTGAGTATGCACGCCTTGGAATTCTCGGGTGC	110129	0.923248226840955	No Hit
	TCCTCGGTAGTATAGTGGTGAGTATGCTCGCCTGGAATTCTCGGGTGCC	82246	0.6894957156131554	No Hit
	GCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGCC	75176	0.6302255418735813	No Hit
	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCGTGGAATTCTCGGGTGC	68069	0.5706451847636587	No Hit
①	TGGAATTCTCGGGTGCCAAGGAACTCCAGTCACATCACGATCTCGTATG	58409	0.4896621750996862 4	RNA PCR Primer, Index 1 (100% over 49bp)
	AAATCATTACCCTGGACGGTGGATCACTGGAATTCTCGGGTGCCAAGGA	38416	0.3220541717651312	RNA PCR Primer, Index 1 (100% over 22bp)
	TCCTCGGTAGTATAGTGGTGAGTATGCTCGCCTTGGAATTCTCGGGTGC	35680	0.299117369028006	No Hit
	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTGTGGAATTCTCGGGTG	28678	0.2404172620231266	No Hit
	TCCTCGGTAGTATAGTGGTGAGTATGCACGCCTGGAATTCTCGGGTGCC	28671	0.24035857868279037	No Hit
	AAATCATTACCCTGGACGGTGGATCACTTGGCTCGCGTGGAATTCTCGG	27757	0.23269621110174787	No Hit
	CGGGCAACCCGCTGAAACTCCTTCGTGCTGGGGATTGTGGAATTCTCGG	22712	0.19040228938800652	No Hit
	TCCCTGGTTGATCCTGCCTGGAATTCTCGGGTGCCAAGGAACTCCAGTC	19997	0.16764153667189002	RNA PCR Primer, Index 1 (100% over 31bp)

トリミング(Trimming)

出現回数下位を眺めている。黒下線で示すように、ハイライトされていないNo Hitのリード中にも、さらに短い「RNA PCR Primer Index 1の左側の塩基配列」を含んでいることがわかる。

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

TCCCTGGTTGATCCTGCCTGGAATTCTCGG			31bp)
TCCTCGGTAGTATAGTGGTGAGTATGCCCGCCTTGGAATTCTCGGGTGC	19932	0.16709661994019664	No Hit
AAATCATTACCCTGGACGGTGGATCACTTGGCTCGCGGTGGAATTCTCG	19650	0.16473251965808067	No Hit
GTCAGTCGATCCTAAGCTCAAGGAGAGATGGAATTCTCGGGTGCCAAGG	19014	0.15940071902181913	Illumina Small RNA Adapter 2 (100% over 21bp)
GACTGGGAGCGTGGCGTCTCCTGTAATTCGGCTATGGAATTCTCGGGTG	17862	0.14974311786934538	No Hit
GTTGGCCTCAGATCAGGGAGGATCACCCGCCGAATTTAAGCTGGAATTC	16808	0.14090708348157863	No Hit
TGGACGGAGAACTGATAAGGGCTGGAATTCTCGGGTGCCAAGGAACTCC	16488	0.13822441649478037	RNA PCR Primer, Index 1 (100% over 27bp)
GGCCGTGATCGTCTAGTGGTTAGGCCCCTACGTTGGAATTCTCGGGTGC	15993	0.13407466599957682	No Hit
CATTTGGATCGCGGAGATCTGGAATTCTCGGGTGCCAAGGAACTCCAGT	15322	0.1284494486616342	RNA PCR Primer, Index 1 (100% over 30bp)
CTTCGGTAGTATAGTGGTCAGTATCCCCGCCTTGGAATTCTCGGGTGCC	15147	0.12698236515322892	No Hit
TGCCGTGATCGTCTAGTGGTTAGGCCCCTACGTTGGAATTCTCGGGTGC	14993	0.12569133166583224	No Hit
AGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	14895	0.12486976490112528	No Hit
TGAGATCATTGTGAAAGCTAATTGGAATTCTCGGGTGCCAAGGAACTCC	14165	0.11874993083749175	RNA PCR Primer, Index 1 (100% over 27bp)
TATTATGATGACAAACAAACTAAGGAACCACTGATTGCATTGGAATTCT	13934	0.11681338060639676	No Hit
TCCTCGGTAGTATAGTGGTCAGTATCCCCGCCTTGGAATTCTCGGGTGC	13352	0.11193428002415741	No Hit
GTTGGCCTCAGATCAGGGAGGATCACCCGCCGAATTTAAGTGGAATTCT	13152	0.11025761315740851	No Hit
AAATCATTACCCTGGACGGTGGATCACTTGGCTCGCGGGTGGAATTCTC	12713	0.10657732938489464	No Hit

トリミング(Trimming)

これらのハイライトされた部分配列や黒下線部分のアダプター配列(adapter sequences)は、予め取り除かれるべき(このデータはカイコのsmall RNAなのでカイコ中に本来存在せず、マップもされない)。

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

TCCCTGGTTGATCCTGCCTGGAATTC			31bp)
TCCTCGGTAGTATAGTGGTGAGTATGCCCGCCTTGGAATTCTCGGGTGC	19932	0.16709661994019664	No Hit
AAATCATTACCCTGGACGGTGGATCACTTGGCTCGCGGTGGAATTCTCG	19650	0.16473251965808067	No Hit
GTCAGTCGATCCTAAGCTCAAGGAGAGATGGAATTCTCGGGTGCCAAGG	19014	0.15940071902181913	Illumina Small RNA Adapter 2 (100% over 21bp)
GACTGGGAGCGTGGCGTCTCCTGTAATTCGGCTATGGAATTCTCGGGTG	17862	0.14974311786934538	No Hit
GTTGGCCTCAGATCAGGGAGGATCACCCGCCGAATTTAAGCTGGAATTC	16808	0.14090708348157863	No Hit
TGGACGGAGAACTGATAAGGGCTGGAATTCTCGGGTGCCAAGGAACTCC	16488	0.13822441649478037	RNA PCR Primer, Index 1 (100% over 27bp)
GGCCGTGATCGTCTAGTGGTTAGGCCCCTACGTTGGAATTCTCGGGTGC	15993	0.13407466599957682	No Hit
CATTTGGATCGCGGAGATCTGGAATTCTCGGGTGCCAAGGAACTCCAGT	15322	0.1284494486616342	RNA PCR Primer, Index 1 (100% over 30bp)
CTTCGGTAGTATAGTGGTCAGTATCCCCGCCTTGGAATTCTCGGGTGCC	15147	0.12698236515322892	No Hit
TGCCGTGATCGTCTAGTGGTTAGGCCCCTACGTTGGAATTCTCGGGTGC	14993	0.12569133166583224	No Hit
AGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	14895	0.12486976490112528	No Hit
TGAGATCATTGTGAAAGCTAATTGGAATTCTCGGGTGCCAAGGAACTCC	14165	0.11874993083749175	RNA PCR Primer, Index 1 (100% over 27bp)
TATTATGATGACAAACAAACTAAGGAACCACTGATTGCATTGGAATTCT	13934	0.11681338060639676	No Hit
TCCTCGGTAGTATAGTGGTCAGTATCCCCGCCTTGGAATTCTCGGGTGC	13352	0.11193428002415741	No Hit
GTTGGCCTCAGATCAGGGAGGATCACCCGCCGAATTTAAGTGGAATTCT	13152	0.11025761315740851	No Hit
AAATCATTACCCTGGACGGTGGATCACTTGGCTCGCGGGTGGAATTCTC	12713	0.10657732938489464	No Hit

トリミング(Trimming)

このデータ中のアダプター配列除去を行う場合は、RNA PCR Primer, Index 1の実際の配列長が49 bp以上であろうがなかろうが、①の塩基配列をそのままアダプター配列として与えてもうまくトリムできるだろうと判断。

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

Overrepresented sequences

Sequence			
TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTTGGAATTCTCGGGTGC	5646094	47.333093681749176	No Hit
TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTGGAATTCTCGGGTGCC	1201120	10.06939053494727	No Hit
GGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	172150	1.4431910055541266	No Hit
TGCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGC	164982	1.3830992650498455	No Hit
TCCTCGGTAGTATAGTGGTGAGTATGCACGCCTTGGAATTCTCGGGTGC	110129	0.923248226840955	No Hit
TCCTCGGTAGTATAGTGGTGAGTATGCTCGCCTGGAATTCTCGGGTGCC	82246	0.6894957156131554	No Hit
GCCGTGATCGTCTAGTGGTTAGGACCCTACGTTGGAATTCTCGGGTGCC	75176	0.6302255418735813	No Hit
TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCGTGGAATTCTCGGGTGC	68069	0.5706451847636587	No Hit
① TGGAATTCTCGGGTGCCAAGGAACTCCAGTCACATCACGATCTCGTATG	58409	0.4896621750996862 4	RNA PCR Primer, Index 1 (100% over 49bp)
AAATCATTACCCTGGACGGTGGATCACTGGAATTCTCGGGTGCCAAGGA	38416	0.3220541717651312	RNA PCR Primer, Index 1 (100% over 22bp)
TCCTCGGTAGTATAGTGGTGAGTATGCTCGCCTTGGAATTCTCGGGTGC	35680	0.299117369028006	No Hit
TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTGTGGAATTCTCGGGTG	28678	0.2404172620231266	No Hit
TCCTCGGTAGTATAGTGGTGAGTATGCACGCCTGGAATTCTCGGGTGCC	28671	0.24035857868279037	No Hit
AAATCATTACCCTGGACGGTGGATCACTTGGCTCGCGTGGAATTCTCGG	27757	0.23269621110174787	No Hit
CGGGCAACCCGCTGAAACTCCTTCGTGCTGGGGATTGTGGAATTCTCGG	22712	0.19040228938800652	No Hit
TCCCTGGTTGATCCTGCCTGGAATTCTCGGGTGCCAAGGAACTCCAGTC	19997	0.16764153667189002	RNA PCR Primer, Index 1 (100% over 31bp)

Contents

- QC(Quality Control)
  - Quality Check (FastQC):乳酸菌RNA-seqデータ
    - k-mer解析、課題1
  - Quality Check (FastQC):カイコsRNA-seqデータ
  - トリミング:カイコsRNA-seqデータ
    - 全体像を把握
    - アダプター配列除去(QuasRパッケージ)、課題2
    - Rを駆使して結果を確認
  - トリミング:乳酸菌RNA-seqデータ
    - サブセットの抽出
    - paired-endデータのアダプター配列除去(QuasRパッケージ)

アダプター配列除去

QuasRパッケージを用いてアダプター配列除去の基本形をやってみよう。約5分

- 前処理 | フィルタリング | GFF/GTF形式ファイル (last modified 2013/10/10)
- 前処理 | フィルタリング | 組合せ | ACGTのみ & 指定した長さの範囲の配列 (last
- 前処理 | トリミング | ポリA配列除去 | ShortRead(Morgan_2009) (last modif
- 前処理 | トリミング | アダプター配列除去(基礎) | QuasR(Gaidatzis_2015) (mod
- 前処理 | トリミング | アダプター配列除去(基礎) | girafe(Toedling_2010) (las modif
- 前処理 | トリミング |
- 前処理 | トリミング |
- 前処理 | トリミング |
- 前処理 | トリミング |
- アセンブル | について
- アセンブル | ゲノム用
- アセンブル | トランス
- マッピング | について
- マッピング | basic alig
- マッピング | splice-aw
- マッピング | Bisulfite
- マッピング | (ESTレベ
- マッピング | 基礎 (las
- マッピング | single-en

① ②

前処理 | トリミング | アダプター配列除去(基礎) | QuasR(Gaidatzis_2015) NEW

QuasRパッケージを用いたアダプター配列除去の基本形を示します。param_nrecオプションは、一度に処理するリード数を指定しているところですが、基本的に無視で構いません。デフォルトの1000000のときに、メモリ不足でフリーズしたので、デフォルトの
です。それでも
「ファイル」-「ディレク

1. gzip圧縮状態のF

small RNA-seqデー
中の記述から GSE
(Zhu_2013)の7を実
は"TGGAATTCTC
以外はデフォルトで
り、右側のアダプタ
いる理由です。約5

```
in_f <- "SRR60
out_f <- "hoge
param_adapter
param_nrec <-

#必要なパッケー
library(QuasR)

#本番(アダプター
res <- preproc
```

4. gzip圧縮状態のFASTQ形式ファイル(SRR609266.fastq.gz)の場合:

small RNA-seqデータ(ファイルサイズは400Mb弱、11928428リード)です。このファイルに対するFastQC実行結果として「RNA PCR Primer, Index 1」(RPI1)が含まれているとレポートされた49 bpをアダプター配列として入力しています。ここでは、アダプター配列以外はデフォルトで実行しています。アダプター配列の位置は5'側(左側)ではなく3'側(右側)にあるという前提であり、右側のアダプター配列しかトリムしないやり方です。それが、preprocessReads関数実行時にRpatternのみ記載している理由です。約5分。

```
in_f <- "SRR609266.fastq.gz"          #入力ファイル名を指定してin_fに格納
out_f <- "hoge4.fastq.gz"             #出力ファイル名を指定してout_fに格納
param_adapter <- "TGGAATTCTCGGGTGCCAAGGAACTCCAGTCACATCACGATCTCGTATG"#アダプター配
param_nrec <- 500000                  #一度に処理するリード数を指定

#必要なパッケージをロード
library(QuasR)

#本番(アダプター配列除去)
res <- preprocessReads(filename=in_f,
                       outputFilename=out_f,
                       Rpattern=param_adapter)
res
file.size(in_f)
file.size(out_f)
```

R Console

```
> getwd()
[1] "C:/Users/kadota/Desktop/hoge"
> list.files(pattern="609266.")
[1] "SRR609266.fastq.gz"
[2] "SRR609266_fastqc.html"
>
```

アダプター配列除去

この状態がしばらく続きます。
「応答なし」的な状態になっても
、しばらく放置すると復活します

4. gzip圧縮状態のFASTQ形式ファイル(SRR609266.fastq.gz)の場合:

small RNA-seqデータ(ファイルサイズは400Mb弱、11928428リード)です。このファイルに対するFastQC実行結果として「RNA PCR Primer, Index 1」(RPI1)が含まれているとレポートされた49 bpをアダプター配列として入力しています。ここでは、アダプター配列以外はデフォルトで実行しています。アダプター配列の位置は5'側(左側)ではなく3'側(右側)にあるという前提であり、右側のアダプター配列しかトリムしないやり方です。それが、preprocessReads関数実行時にRpatternのみ記載している理由です。約5分。

```
in_f <- "SRR609266.fastq.gz"              #入力ファイル名を指定してin_fに格納
out_f <- "hoge4.fastq.gz"                 #出力ファイル名を指定してout_fに格納
param_adapter <- "TGGAATTCTCGGGTGCCAAGGAACTCCAGTCACATCACGATCTCGTATG"#アダプター配
param_nrec <- 500000                      #一度に処理するリード数を指定

#必要なパッケージをロード
library(QuasR)                            #パッケー

#本番(アダプター配列除去)
res <- preprocessReads(filename=in_f,     #アダプタ
             outputFilename=out_f,        #アダプタ
             Rpattern=param_adapter)      #アダプタ
res                                       #確認して
file.size(in_f)                           #入力ファ
file.size(out_f)                          #出力ファ
```

R Console

```
    tapply, union, unique, unlist, unsplit

 要求されたパッケージ S4Vectors をロード中です
 要求されたパッケージ stats4 をロード中です
 要求されたパッケージ IRanges をロード中です
 要求されたパッケージ GenomeInfoDb をロード中です
 要求されたパッケージ Rbowtie をロード中です
>
> #本番(アダプター配列除去)
> res <- preprocessReads(filename=in_f,   #アダプ$
+              outputFilename=out_f,      #アダプ$
+              Rpattern=param_adapter)    #アダプ$
  filtering SRR609266.fastq.gz
```

アダプター配列除去

これが正常終了時の画面。アダプター配列除去後のリードが短いもの（tooShort）が90,242個、Nを多く含むもの(tooManyN)が875個あったことがわかります。①これらを除いた残りの11,837,311リードが出力されているようです。②hoge4.fastq.gzのファイルサイズは281MB。

4. gzip圧縮状態のFASTQ形式ファイル(SRR609266.fastq.gz)の場合:

small RNA-seqデータ(ファイルサイズは400Mb弱、11928428リード)です。このフ
実行結果として「RNA PCR Primer, Index 1」(RPI1)が含まれているとレポートされ
配列として入力しています。ここでは、アダプター配列以外はデフォルトで実行し
配列の位置は5'側(左側)ではなく3'側(右側)にあるという前提であり、右側のアダプター配列しかトリム
しないやり方です。それが、preprocessReads関数実行時にRpatternのみ記載している理由です。約5分。

```
in_f <- "SRR609266.fastq.gz"              #入力ファイル名を指定してin_fに格納
out_f <- "hoge4.fastq.gz"                 #出力ファイル名を指定してout_fに格納
param_adapter <- "TGGAATTCTCGGGTGCCAAGGAACTCCAG
param_nrec <- 500000                      #一度に処

#必要なパッケージをロード
library(QuasR)                            #パッケー

#本番(アダプター配列除去)
res <- preprocessReads(filename=in_f,     #アダプタ
             outputFilename=out_f,        #アダプタ
             Rpattern=param_adapter)      #アダプタ
res                                       #確認して
file.size(in_f)                           #入力ファ
file.size(out_f)                          #出力ファ
```

R Console

```
+                outputFilename=out_f,      #アダプ$
+                Rpattern=param_adapter)    #アダプ$
  filtering SRR609266.fastq.gz
> res                                      #確認し$
                 SRR609266.fastq.gz
totalSequences              11928428
matchTo5pAdapter                   0
matchTo3pAdapter            11839903
tooShort                       90242
tooManyN                         875
lowComplexity                      0
totalPassed                 11837311   ①
> file.size(in_f)                        #入力フ$
[1] 383840833
> file.size(out_f)                       #出力フ$
[1] 281324281
>
```

②

トリム後のFastQC結果

hoge4.fastq.gzに対するFastQC実行結果ファイル(hoge4_fastqc.html)を眺める。hogeフォルダ中にあります。
①リード数はQuasR実行結果と同じ11,837,311個。

Summary

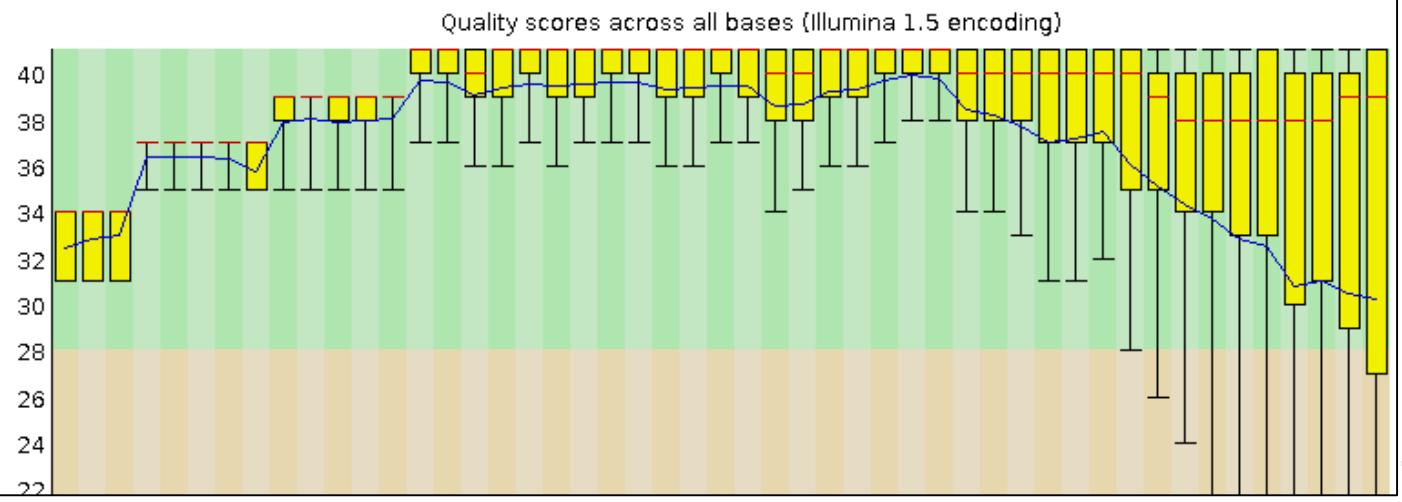
- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content
- Kmer Content

Basic Statistics

Measure	V
Filename	hoge4.fastq.gz
File type	Conventional base calls
Encoding	Illumina 1.5
Total Sequences	11837311
Sequences flagged as poor quality	0
Sequence length	14-49
%GC	50

①

Per base sequence quality



トリム後のFastQC結果

①配列長の範囲は14-49。このことから、②QuasR実行結果のtooShortで落とされた90,242リードは、アダプター配列除去後に14塩基未満になったものたちだろうと推測。

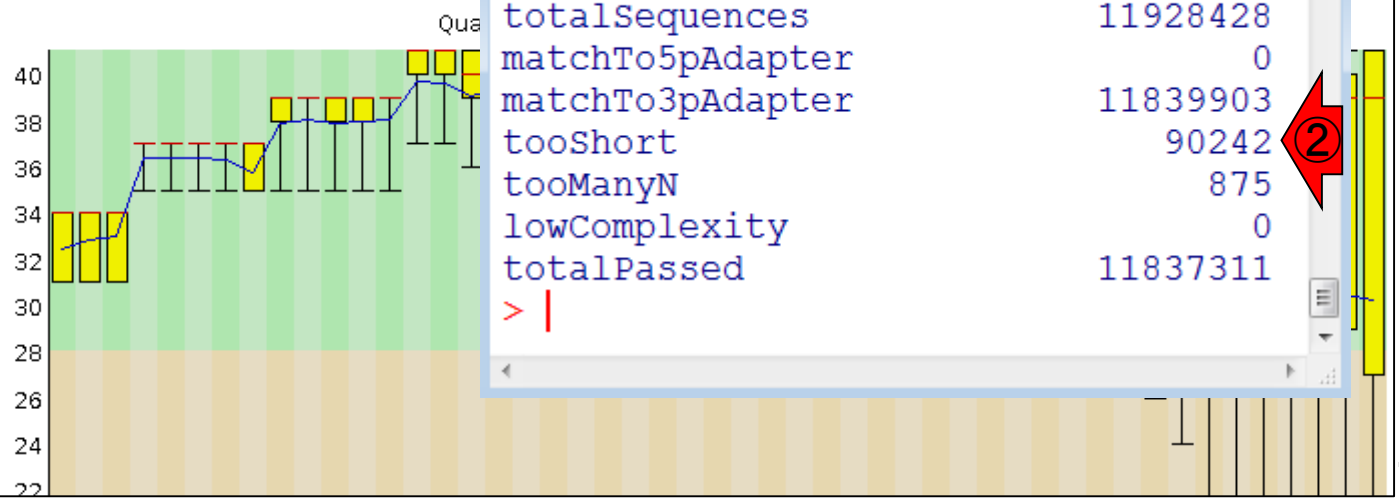
Summary

- 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	V
Filename	hoge4.fastq.gz
File type	Conventional base calls
Encoding	Illumina 1.5
Total Sequences	11837311
Sequences flagged as poor quality	0
Sequence length	14-49 ①
%GC	50

Per base sequence quality



R Console

```
> res
                        SRR609266.fastq.gz
totalSequences                     11928428
matchTo5pAdapter                         0
matchTo3pAdapter                  11839903
tooShort                             90242 ②
tooManyN                               875
lowComplexity                            0
totalPassed                       11837311
> |
```

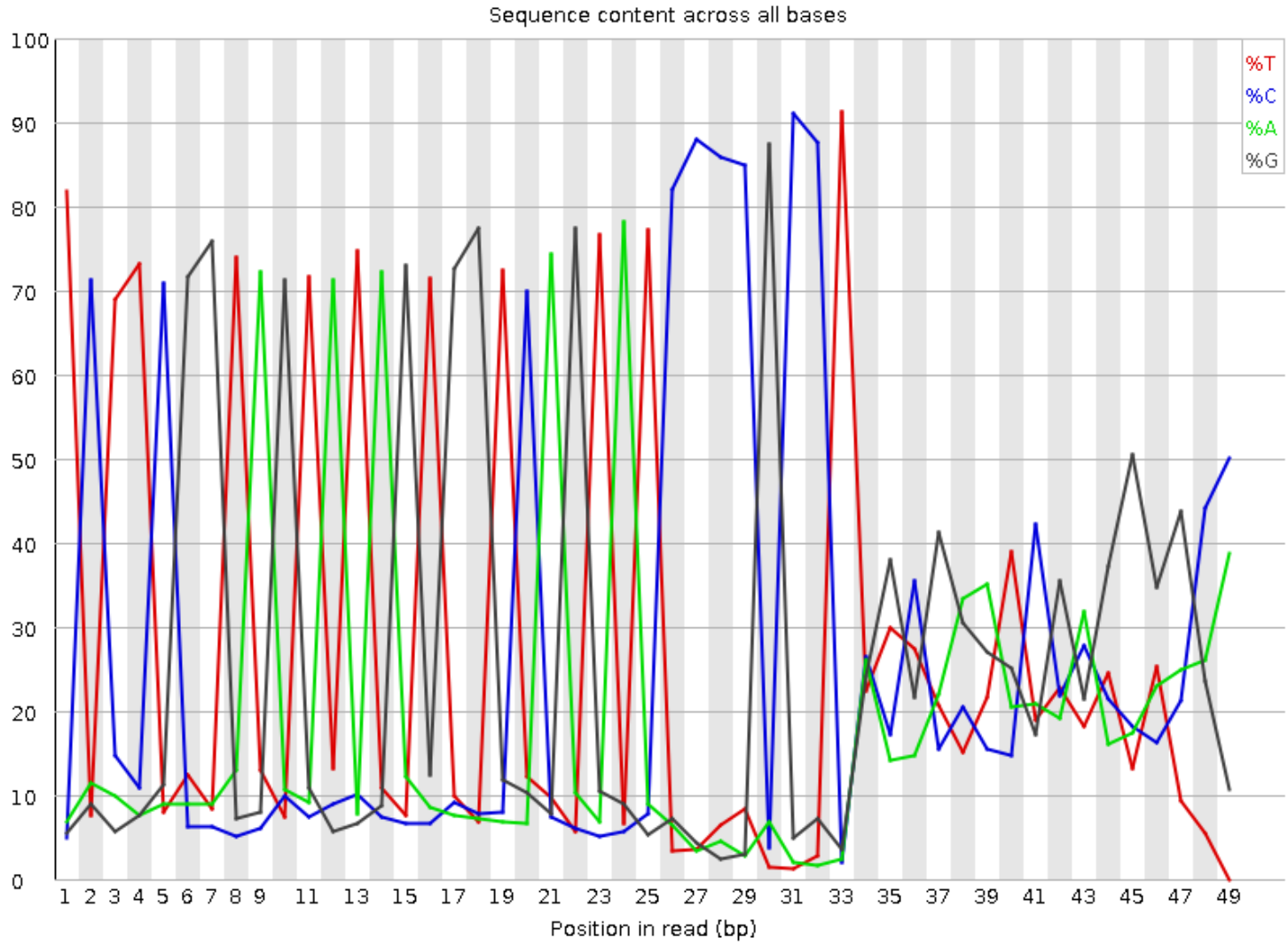
トリム後のFastQC結果

課題2:なぜアダプター配列除去前後で34 bp以降のプロファイルが劇的に変わったのか述べよ。

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

Per base sequence content



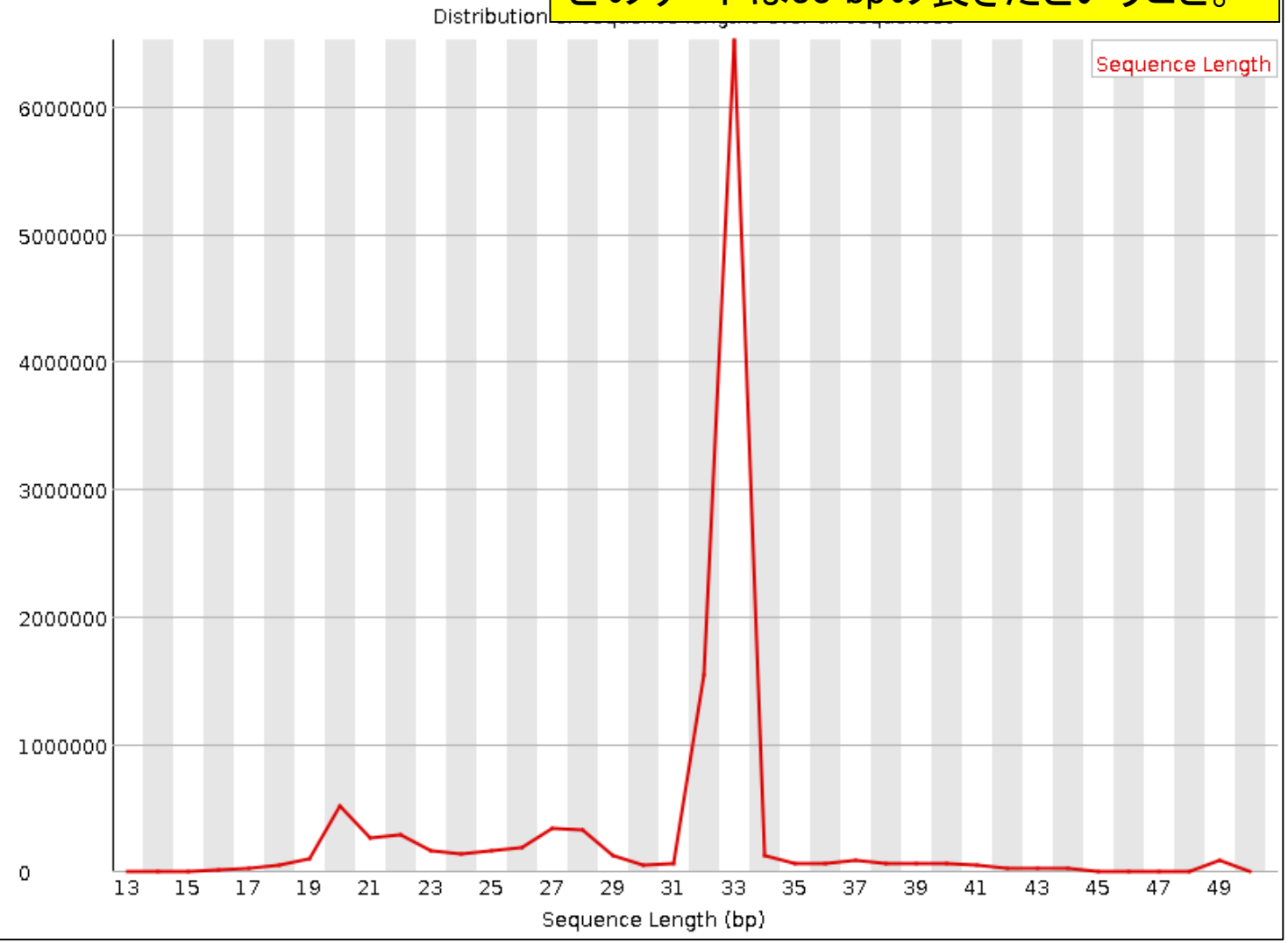
トリム後のFastQC結果

アダプター配列除去前は49 bpのところに一本のピークが見えていた(つまり全リードの長さが49 bp)。除去後は33 bpのところにピークがある。つまり、ほとんどのリードは33 bpの長さだということ。

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 Length Distribution



トリム後のFastQC結果

Possible Sourceのところが全てNo Hitになっていることが分かる。QuasRを用いたアダプター配列除去手順の正しさは、この結果からもわかる。

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

Overrepresented sequences

Sequence	Count	Percentage	Possible Source
TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCT	5711563	48.25051061005325	No Hit
TCTTCGGTAGTATAGTGGTCAGTATCCCCGCC	1224329	10.34296556033714	No Hit
GGCCGTGATCGTCTAGTGGTTAGGACCCTACGT	174048	1.4703339297244111	No Hit
TGCCGTGATCGTCTAGTGGTTAGGACCCTACGT	166820	1.4092727647351666	No Hit
TCCTCGGTAGTATAGTGGTGAGTATGCACGCCT	111334	0.9405345521461759	No Hit
TCCTCGGTAGTATAGTGGTGAGTATGCTCGCC	83415	0.7046786216903485	No Hit
GCCGTGATCGTCTAGTGGTTAGGACCCTACGT	76203	0.6437526225339522	No Hit
TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCG	68748	0.5807737922911715	No Hit
AAATCATTACCCTGGACGGTGGATCAC	39237	0.33146886146693283	No Hit
TCCTCGGTAGTATAGTGGTGAGTATGCTCGCCT	36078	0.30478205734393565	No Hit
TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTG	29252	0.24711693390500597	No Hit
TCCTCGGTAGTATAGTGGTGAGTATGCACGCC	29165	0.246381969688893	No Hit
AAATCATTACCCTGGACGGTGGATCACTTGGCTCGCG	28441	0.24026571575250494	No Hit
CGGGCAACCCGCTGAAACTCCTTCGTGCTGGGGATTG	23202	0.19600735335922154	No Hit
TCCCTGGTTGATCCTGCC	20710	0.17495527489309015	No Hit
TCCTCGGTAGTATAGTGGTGAGTATGCCCGCCT	20141	0.17014843996242052	No Hit
AAATCATTACCCTGGACGGTGGATCACTTGGCTCGCGG	19867	0.16783372507489241	No Hit
GTCAGTCGATCCTAAGCTCAAGGAGAGA	19382	0.16373651076667667	No Hit
GACTGGGAGCGTGGCGTCTCCTGTAATTCGGCTA	18124	0.1531090971589747	No Hit
TGGACGGAGAACTGATAAGGGC	17985	0.15193484398610463	No Hit

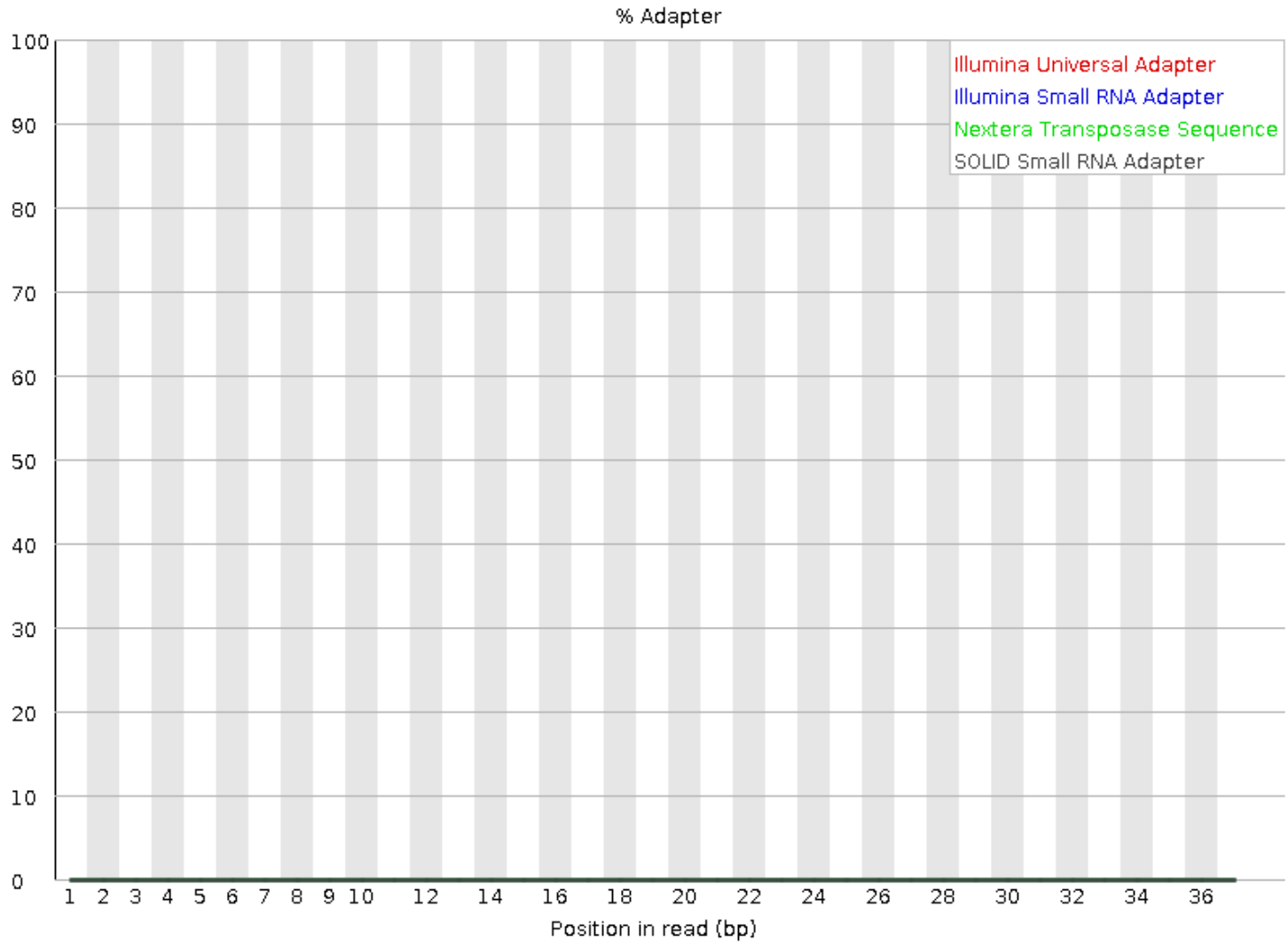
トリム後のFastQC結果

アダプター配列除去前はIllumina Small RNA Adapterの割合が20bp以降で上昇していたが、除去後は消えていることがわかる。

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



Contents

- QC(Quality Control)
  - Quality Check (FastQC):乳酸菌RNA-seqデータ
    - k-mer解析、課題1
  - Quality Check (FastQC):カイコsRNA-seqデータ
  - トリミング:カイコsRNA-seqデータ
    - 全体像を把握
    - アダプター配列除去(QuasRパッケージ)、課題2
    - Rを駆使して結果を確認
  - トリミング:乳酸菌RNA-seqデータ
    - サブセットの抽出
    - paired-endデータのアダプター配列除去(QuasRパッケージ)

参考

Sequence logos

FastQCの項目「Per base sequence content」と似たような結果は、sequence logosでも得ることができます。

- 解析 | 一般 | アラインメント(ペアワイズ;応用) (last modified 2010/6/8)
- 解析 | 一般 | パターンマッチング (last modified 2013/06/19)
- 解析 | 一般 | GC含量(GC contents) (last modified 2015/04/18)
- 解析 | 一般 | Sequence logos(Schneider_1990) (last modified 2014/07/23)
- 解析 | 一般 | 上流配列解析 | LDSS(Yamamoto_2007) (last modified 2015/02/19)
- 解析 | 一般 | 上流配列解析 | Relative Appearance Ratio(Yamamoto_2011) (last m
- 解析 | 基礎 | k-mer | ゲノム
- 解析 | 基礎 | 平均-分散プロ
- 解析 | 基礎 | 平均-分散プロ
- 解析 | 新規転写物同定(ゲ
- 解析 | 発現量推定(トランス
- 解析 | クラスタリング | につい
- 解析 | クラスタリング | サン
- 解析 | クラスタリング | サン
- 解析 | クラスタリング | 遺伝
- 解析 | クラスタリング | 遺伝
- 解析 | シミュレーションカウン
- 解析 | シミュレーションカウン

①

解析 | 一般 | Sequence logos(Schneider_1990)

seqLogoパッケージを用いてsequen
します。ここでは、multi-FASTAフ
利用します。上流-35 bpにTATA box
「ファイル」-「ディレクトリの変更」で解析
ビベ。

②

1. 入力ファイルがmulti-FASTA形式の

```
in_f <- "test1.fasta"

#必要なパッケージをロード
library(Biostrings)
library(seqLogo)

#入力ファイルの読み込み
fasta <- readDNAStringSet(in

#本番(sequence logoを実行)
hoge <- consensusMatrix(fast
out <- makePWM(hoge[1:4,])
seqLogo(out)
```

9. FASTQ形式ファイル(hoge4.fastq.gz)の場合:

small RNA-seqデータ(280Mb弱、11,928,428リード)です。原著論文(Nie et al., BMC Genomics, 2013)中の記述からGSE41841を頼りに、SRP016842にたどりつき、前処理|トリミング|アダプター配列除去(応用)|ShortRead(Morgan_2009)の4を実行して得られたものが入力ファイルです。アダプター配列除去後のデータなので、リードごとに配列長が異なる場合でも読み込めるShortReadパッケージ中のreadFastq関数を用いています。

```
in_f <- "hoge4.fastq.gz"                    #入力ファイル名を指定してin_fに格納
out_f <- "hoge9.png"                        #出力ファイル名を指定してout_fに格納
param_fig <- c(787, 370)                    #ファイル出力時の横幅と縦幅を指定(単

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

#入力ファイルの読み込み
fastq <- readFastq(in_f)                    #in_fで指定したファイルの読み込み
fasta <- sread(fastq)                       #リード塩基配列情報をfastaに格納

#本番(sequence logoを実行)
hoge <- consensusMatrix(fasta, as.prob=T, baseOnly=T)#各ポジションの塩基組
out <- makePWM(hoge[1:4,])                  #hogeはACGT以外の塩基(例えばN)のpr

#ファイルに保存
png(out_f, pointsize=13, width=param_fig[1], height=param_fig[2])#出力フ
seqLogo(out)                                #塩基組成やicの情報を含むoutを入力と
dev.off()                                   #おまじない
```

参考

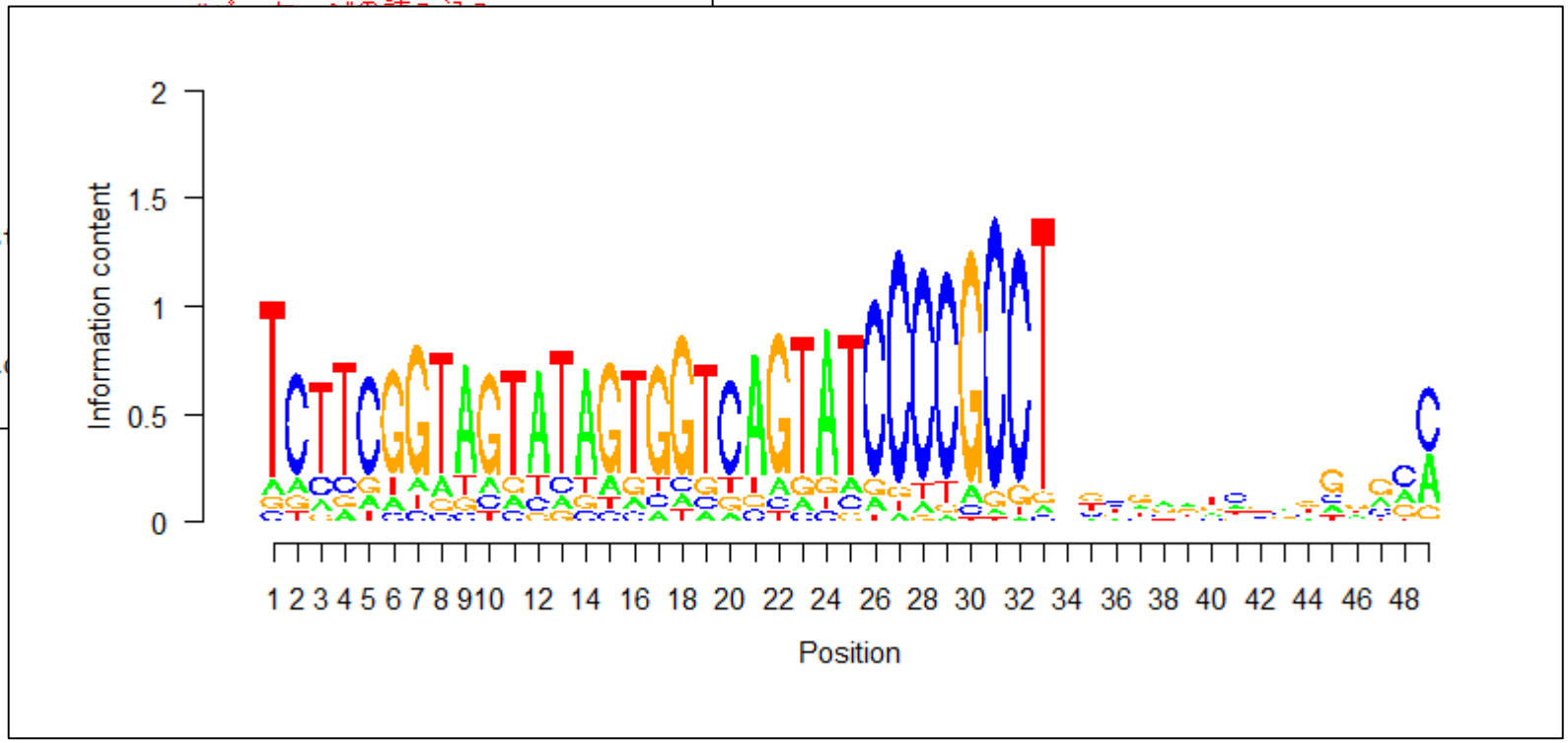
Sequence logos

塩基ごとの色の使い分けもG以外は同じ（A: green, C: blue, and T: red）なので分かりやすいと思います。

9. FASTQ形式ファイル(hoge4.fastq.gz)の場合:

small RNA-seqデータ(280Mb弱、11,928,428リード)です。原著論文(Nie et al., BMC Genomics, 2013)中の記述からGSE41841を頼りに、SRP016842にたどりつき、前処理|トリミング|アダプター配列除去(応用)|ShortRead(Morgan 2009)の4を実行して得られたものが入力ファイルです。アダプター配列除去後のデータなので、リードごとに配列長が異なる場合でも読み込めるShortReadパッケージ中の readFastq関数を用いています。

```
in_f <- "hoge4.fastq.gz"                #入力ファイル名を指定してin_fに格納
out_f <- "hoge9.png"                    #出力ファイル名を指定してout_fに格納
param_fig <- c(787, 370)                #ファイル出力時の横幅と縦幅を指定(単

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

#入力ファイルの読み込み
fastq <- readFastq(in_f)
fasta <- sread(fastq)

#本番(sequence logoを実行)
hoge <- consensusMatrix(fas
out <- makePWM(hoge[1:4,])

#ファイルに保存
png(out_f, pointsize=13, wi
seqLogo(out)
dev.off()
```



配列長分布

FastQCの項目「Sequence Length Distribution」と似たような結果も得ることができます。

- 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 (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 ① modified 2015/06/22) NEW
- 前処理 | クオリティチェック | Overrepresented sequences | ShortRead(Morgan 2009) (last modified 2015/0
- 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
- 前処理 | フィルタリン
- 前処理 | フィルタリン
- 前処理 | フィルタリン
- 前処理 | フィルタリン
- 前処理 | フィルタリン
- 前処理 | フィルタリン
- 前処理 | フィルタリン
- 前処理 | フィルタリン
- 前処理 | フィルタリン
- 前処理 | フィルタリン
- 前処理 | フィルタリン
- 前処理 | トリミング |

前処理 | クオリティチェック | 配列長分布を調べる NEW

FASTAまたはFASTQ形式ファイルを読み込んで配列長分布を得るやり方を示します。
「ファイル」ー「ディレクトリ

②

1. FASTA形式ファイル(sa

```
in_f <- "sample2.fast
out_f <- "hoge1.txt"

#必要なパッケージをロー
library(Biostrings)

#入力ファイルの読み込み
fasta <- readDNAStrin
fasta

#本番
out <- table(width(fa
out

#ファイルに保存
write.table(out, out_
```

5. gzip圧縮状態のFASTQ形式ファイル(hoge4.fastq.gz)の場合:

配列長の異なる架空のファイルです。param_nbinsで50と指定すると、1 bpおきに、1-50 bpまでのリード数が格納されます。QuasRパッケージ(Gaidatzis et al., 2015)マニュアル中のbarplotを用いた作図形式に似せています。

```
in_f <- "hoge4.fastq.gz"                #入力ファイル名を指定してin_fに格納
out_f1 <- "hoge5.txt"                   #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge5.png"                   #出力ファイル名を指定してout_f2に格納
param_nbins <- 50                       #分割数(binの数)を指定
param_fig <- c(700, 400)                #ファイル出力時の横幅と縦幅を指定(単位はピク

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

#入力ファイルの読み込み
fastq <- readFastq(in_f)                #in_fで指定したファイルの読み込み
sread(fastq)                            #配列情報を表示

#本番
out <- tabulate(width(fastq), param_nbins)#出現頻度情報を得た結果をoutに格納
out                                     #確認してるだけです

#ファイルに保存(テキストファイル)
names(out) <- 1:param_nbins             #x軸の数値情報取得
tmp <- cbind(names(out), out)           #保存したい情報をtmpに格納
write.table(tmp, out_f1, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指
```

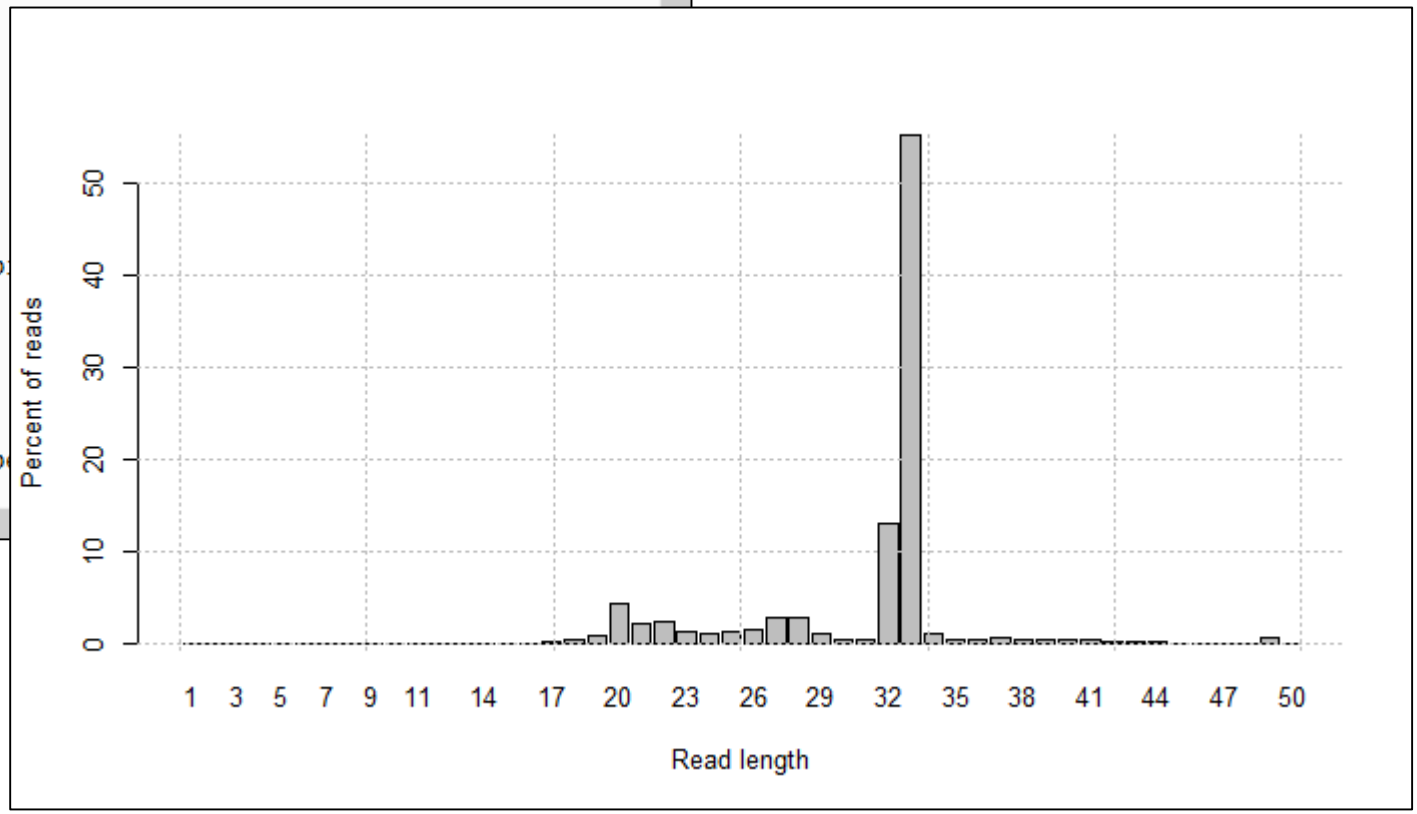
配列長分布

QuasRパッケージマニュアル中の記述形式と似せています。

5. gzip圧縮状態のFASTQ形式ファイル(hoge4.fastq.gz)の場合:

配列長の異なる架空のファイルです。param_nbinsで50と指定すると、1 bpおきに、1-50 bpまでのリード数が格納されます。QuasRパッケージ(Gaidatzis et al., 2015)マニュアル中のbarplotを用いた作図形式に似せています。

```
in_f <- "hoge4.fastq.gz"                #入力ファイル名を指定してin_fに格納
out_f1 <- "hoge5.txt"                   #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge5.png"                   #出力ファイル名を指定してout_f2に格納
param_nbins <- 50                       #分割数(binの数)を指定
param_fig <- c(700, 400)                #ファイル出力時の横幅と縦幅を指定(単位はピク

#必要なパッケージをロード
library(ShortRead)

#入力ファイルの読み込み
fastq <- readFastq(in_f)
sread(fastq)

#本番
out <- tabulate(width(fastq), param_nb
out

#ファイルに保存(テキストファイル)
names(out) <- 1:param_nbins
tmp <- cbind(names(out), out)
write.table(tmp, out_f1, sep="\t", app
```



リードごとの出現回数

FastQCの項目「Overrepresented sequences」と似たような結果も得ることができます。

- 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) NEW
- 前処理 | クオリティチェック | Overrepresented sequences | ShortRead(Morgan_2009) ①
- 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 2014/0

前処理 | クオリティチェック | Overrepresented sequences | ShortRead(Morgan_2009) NEW

ShortReadパッケージを用いてリードの種類ごとの出現回数をを得るやり方を示します。FastQCの Overrepresented sequencesの項目と同じような結果が得られます。前処理 | フィルタリング | 重複のない配列セットを作成も基本的にやっていることは同じです。
「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。

1. gzip圧縮状態のFAS

small RNA-seqデータ(ファ

```
in_f <- "SRR609266.
out_f <- "hoge1.txt

#必要なパッケージをロ
library(ShortRead)

#入力ファイルの読み込
fastq <- readFastq(
sread(fastq)

#本番
out <- table(sread(
out <- sort(out, de
head(out)

#ファイルに保存
tmp <- cbind(names(
write.table(tmp, ou
```

② 2. gzip圧縮状態のFASTQ形式ファイル(hoge4.fastq.gz)の場合:

配列長の 異なる架空のファイルです。

```
in_f <- "hoge4.fastq.gz"                #入力ファイル名を指定してin_fに格納
out_f <- "hoge2.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(ShortRead)                      #パッケージの読み込み

#入力ファイルの読み込み
fastq <- readFastq(in_f)                #in_fで指定したファイルの読み込み
sread(fastq)                            #配列情報を表示

#本番
out <- table(sread(fastq))              #リードごとの出現頻度情報を得た結果をoutに格納
out <- sort(out, decreasing=T)          #出現頻度を降順にソートした結果をoutに格納
head(out)                               #確認してるだけです

#ファイルに保存
tmp <- cbind(names(out), out)           #保存したい情報をtmpに格納
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F)#outの中身を指定し
```

リードごとの出現回数

FastQCの項目「Overrepresented sequences」と似たような結果も得ることができます。

2. gzip圧縮状態のFASTQ形式ファイル(hoge4.fastq.gz)の場合:

配列長の異なる架空のファイルです。

```
in_f <- "hoge4.fastq.gz"                #入力ファイル名を指定してin_fに格納
out_f <- "hoge2.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(ShortRead)                      #パッ

#入力ファイルの読み込み
fastq <- readFastq(in_f)                #in
sread(fastq)                            #配列

#本番
out <- table(sread(fastq))              #リ
out <- sort(out, decreasing=T)          #出現
head(out)                               #確認

#ファイルに保存
tmp <- cbind(names(out), out)           #保存
write.table(tmp, out_f, sep="\t", append=F,
```

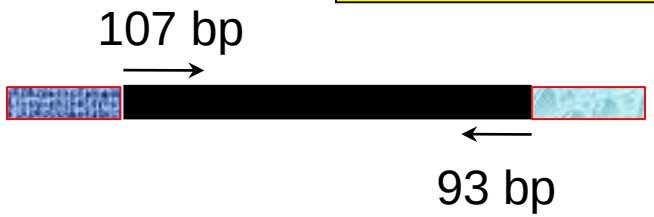
	A	B
1	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCT	5711563
2	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCC	1224329
3	GGCCGTGATCGTCTAGTGGTTAGGACCCTACGT	174048
4	TGCCGTGATCGTCTAGTGGTTAGGACCCTACGT	166820
5	TCCTCGGTAGTATAGTGGTGAGTATGCACGCCT	111334
6	TCCTCGGTAGTATAGTGGTGAGTATGCTCGCC	83415
7	GCCGTGATCGTCTAGTGGTTAGGACCCTACGT	76203
8	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCG	68748
9	AAATCATTACCCTGGACGGTGGATCAC	39237
10	TCCTCGGTAGTATAGTGGTGAGTATGCTCGCCT	36078
11	TCTTCGGTAGTATAGTGGTCAGTATCCCCGCCTG	29252
12	TCCTCGGTAGTATAGTGGTGAGTATGCACGCC	29165
13	AAATCATTACCCTGGACGGTGGATCACTTGGCTCGCG	28441
14	CGGGCAACCCGCTGAAACTCCTTCGTGCTGGGGATTG	23202
15	TCCCTGGTTGATCCTGCC	20710
16	TCCTCGGTAGTATAGTGGTGAGTATGCCCGCCT	20141
17	AAATCATTACCCTGGACGGTGGATCACTTGGCTCGCGG	19867
18	GTCAGTCGATCCTAAGCTCAAGGAGAGA	19382
19	GACTGGGAGCGTGGCGTCTCCTGTAATTCGGCTA	18124

Contents

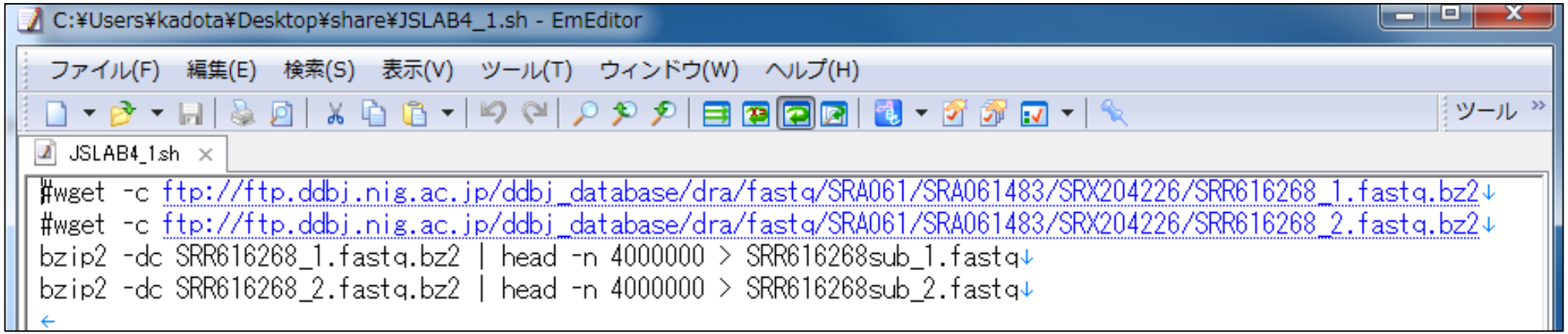
- QC(Quality Control)
  - Quality Check (FastQC):乳酸菌RNA-seqデータ
    - k-mer解析、課題1
  - Quality Check (FastQC):カイコsRNA-seqデータ
  - トリミング:カイコsRNA-seqデータ
    - 全体像を把握
    - アダプター配列除去(QuasRパッケージ)、課題2
    - Rを駆使して結果を確認
  - トリミング:乳酸菌RNA-seqデータ
    - サブセットの抽出
    - paired-endデータのアダプター配列除去(QuasRパッケージ)

一番下はLinuxコマンド。Rではどうやるのか？

乳酸菌RNA-seqデータ



- paired-endのオリジナルデータ
  - 134,755,996リード
  - SRR616268_1.fastq.bz2：107 bp, 7,662,128,101 bytes (約7.7GB)
  - SRR616268_2.fastq.bz2：93 bp, 7,017,031,734 bytes (約7.0GB)
- Linuxコマンドを用いて最初の1,000,000リード分からなるサブセットを抽出
  - 1,000,000リード(one million reads)
  - SRR616268sub_1.fastq.gz：107 bp, 74,906,576 bytes (約75MB)
  - SRR616268sub_2.fastq.gz：93 bp, 67,158,462 bytes (約67MB)



サブセットの抽出

- イントロ | NGS | 読み込み | FASTA形式 | 基本情報を取得 (last modified 2014/08/18)
- 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 2015/06/20) NEW
- イントロ | NGS | 読み込み | FASTQ形式 | 応用 (last modified 2015/06/18) NEW
- 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の*_qseq.txt (last modified 2013/06/17)
- イントロ | ファイル形式の変換 | について (last modified 2014/06/09)
- イントロ | ファイル形式の変換 | BAM --> BED (last modified 2014/06/21)
- イントロ | ファイル形式の変換 | FASTQ --> FASTA (last modified 2015/06/15) NEW
- イントロ | ファイル形式の変換 | Genbank --> FASTA (last modified 2014/03/10)
- イントロ | ファイル形式の変換 | qseq --> FASTA (last modified 2013/06/17)
- イントロ | ファイル形式の変換 | qseq --> Illumina FASTQ (last modified 2013/06/17)
- イントロ | ファイル形式の変換 | qseq --> Sanger FASTQ (last modified 2013/08/19)
- 前処理 | クオリティコントロール | について (last modified 2015/05/04)
- 前処理 | クオリティチェック | QuasR(Gaidatzis 2015) (last modified 2015/06/15) NEW
- 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) NEW
- 前処理 | クオリティチェック | Overrepresented sequences | ShortRead(Morgan 2009) (last modified 2015/06/22)
- 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 2014/08/04)
- 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 2013/09/27)
- 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)
- 前処理 | フィルタリング | 任意のIDを含む配列を抽出 (last modified 2013/06/18)
- 前処理 | フィルタリング | Illuminaのpass filtering (last modified 2013/06/19)
- 前処理 | フィルタリング | GFF/GTF形式ファイル (last modified 2013/10/10)
- 前処理 | フィルタリング | 組合せ | ACGTのみ & 指定した長さの範囲の配列 (last modified 2014/06/11)

①

項目としては、感覚的には黒矢印部分だが、例題数が多すぎると思ったので、とりあえず①のところに書いてます。

サブセットの抽出

イントロ | NGS | 読み込み | FASTQ形式 | 応用 NEW

FASTQ形式ファイルを読み込むやり方を示します。ファイルサイズが数GBというレベルになってきてますので、圧縮ファイルでの読み込みが基本となりつつあります。しかし、一度に圧縮ファイル中の中身を一旦全て読み込むことも厳しい、あるいはもはやできない状態になってきています。この背景を踏まえ、「応用」では、圧縮FASTQファイルを入力として、メモリオーバーフロー（スタックオーバーフロー; stack overflow）にならないように一部だけを読み込む手順を示します。まだ途中段階ですが、野間口達洋 氏提供情報をもとにいくつか試しています。
「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。

1. small RNA-seqのgzip圧縮FASTQ形式ファイル(SRR609266.fastq.gz)の場合：

イントロ | NGS | 配列取得 | FASTQ or SRA | SRAdb(Zhu_2013)の7を実行して得られたカイコsmall RNA-seqデータ(Nie et al., BMC Genomics, 2013)です。入力ファイルサイズは400Mb弱、11,928,428リードです。この中から最初の100000リード分をgzip圧縮ファイルとして出力しています。

```
in_f <- "SRR609266.fastq.gz"            #入力ファイル名を指定してin_fに格納
out_f <- "hoge1.fasta.gz"               #出力ファイル名を指定してout_fに格納
param <- 100000                         #抽出したいリード数を指定

#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="fastq",#in_fで指定したファイルの読み込み
           skip=0, nrec=param)          #in_fで指定したファイルの読み込み
fasta                                   #確認してるだけです

#ファイルに保存
writeXStringSet(fasta, file=out_f,      #fastaの中身を指定したファイル名で保存
    format="fasta", compress=T, width=50)#fastaの中身を指定したファイル名で保存
```

2. small RNA-seqのgzip圧縮FASTQ形式ファイル(SRR609266.fastq.gz)の場合：

イントロ | NGS | 配列取得 | FASTQ or SRA | SRAdb(Zhu_2013)の7を実行して得られた 2015年6月18日現在(R ver. 3.2.0; Biostrings ver. 2.36.1)、エラーを吐かずにFASTQ形式のgzip圧縮ファイルは得られるのですが、、、**クオリティスコア部分が変**です。つまり、このやり方はダメです。おそらくreadDNAStringSet関数で読み込んだ情報の中にクオリティスコアが含まれていないため、ダミーの「J」というスコアがそのまま表示されているだけだと思います。readDNAStringSet関数でクオリティスコアを読み込むやり方（あるいは取り出し方）が分かった方は教えて下さいm(_ _)m

```
in_f <- "SRR609266.fastq.gz"            #入力ファイル名を指定してin_fに格納
out_f <- "hoge2.fastq.gz"               #出力ファイル名を指定してout_fに格納
param <- 100000                         #抽出したいリード数を指定

#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
fastq <- readDNAStringSet(in_f, format="fastq",#in_fで指定したファイルの読み込み
           skip=0, nrec=param)          #in_fで指定したファイルの読み込み
fastq                                   #確認してるだけです

#ファイルに保存
writeXStringSet(fastq, file=out_f,      #fastqの中身を指定したファイル名で保存
    format="fastq", compress=T)         #fastqの中身を指定したファイル名で保存
```

3. small RNA-seqのgzip圧縮FASTQ形式ファイル(SRR609266.fastq.gz)の場合：

原始的なやり方ですが、とりあえずうまく動きます。

```
in_f <- "SRR609266.fastq.gz"            #入力ファイル名を指定してin_fに格納
out_f <- "hoge3.fastq"                  #出力ファイル名を指定してout_fに格納
param <- 100000                         #抽出したいリード数を指定

#入力ファイルの読み込み
fastq <- readLines(in_f, n=param*4)     #in_fで指定したファイルの読み込み

#ファイルに保存
writeLines(fastq, con=out_f)            #fastqの中身を指定したファイル名で保存
```

①は出力がFASTA形式、②はFASTQ形式で出力しようとして失敗中（爆）。現状では、最もシンプル且つ無難な③がおススメ。但し、出力がFASTQ形式だがgzip圧縮ファイルにはなっていません。**発展課題**：例題4として掲載できる、gzip圧縮FASTQファイルとして5000番目から10000番目までのリードを出力するようなコードをレポートせよ。**おまけ**：QuasRパッケージはbzip2圧縮ファイル(.bz2)も入力として受け付けてくれるようです。

野間口達洋氏提供情報。Thanks to preDr. Nomaguchi T.

Contents

- QC(Quality Control)
  - Quality Check (FastQC):乳酸菌RNA-seqデータ
    - k-mer解析、課題1
  - Quality Check (FastQC):カイコsRNA-seqデータ
  - トリミング:カイコsRNA-seqデータ
    - 全体像を把握
    - アダプター配列除去(QuasRパッケージ)、課題2
    - Rを駆使して結果を確認
  - トリミング:乳酸菌RNA-seqデータ
    - サブセットの抽出
    - paired-endデータのアダプター配列除去(QuasRパッケージ)

Paired-endの取扱い

Paired-endデータに対応できるパッケージは、まだそれほど多くないという例を示します。

- 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 2015/06/22)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 2015/06/23)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 2014/06/18)
- 前処理 | トリミング | 指定した末端塩基数は除去 (last modified 2013/06/11)
- アセンブル | につい
- アセンブル | ゲノム
- アセンブル | トラン
- マッピング | につい
- マッピング | basic a
- マッピング | splice-
- マッピング | Bisulfi
- マッピング | (ESTレ
- マッピング | 基礎 (
- マッピング | single-
- マッピング | single-

①

前処理 | トリミング | アダプター配列除去(応用) | QuasR(Gaidatzis_2015) NEW

②

QuasRパッケージを
生してR Guiが終了
6月21日にparam_nre
構いません。。デフォル
す。処理時間は長くなり
「ファイル」-「ディレクトリ

1. gzip圧縮状態のFAST

small RNA-seqデータ(
2013)中の記述から GS
SRAdb(Zhu_2013)のア
の低いリードを除去した
ませんでしたが、Table
(例:"GCAGTCGTGG
列ということになります
は"TGGAATTCTCGG
チまで許容して(推定)
が18nt以上のものをフ
あるという前提であり、
RpatternとmaxRmisma

```
in_f <- "SRR6092
out_f <- "hoge1.
param_adapter <-
```

2. gzip圧縮FASTQ形式ファイル(SRR616268sub_1.fastq.gzとSRR616268sub_2.fastq.gz)の場合:

乳酸菌RNA-seqデータSRR616268の最初の100万リード分です。paired-endデータです。SRR616268sub_1.fastq.gzは、約75MB、全リード107 bpです。SRR616268sub_2.fastq.gzは、約67MB、全リード93 bpです。FastQC実行結果として「TruSeq Adapter, Index 3」が含まれているとレポートされました。これでググると塩基配列情報は"GATCGGAAGAGCACACGTCTGAACTCCAGTCACTTAGGCATCTCGTATGCCGTCTTCTGCTTG"と書いてあったので、これを入力として与えます。アダプター配列の位置は5'側(左側)にあるという前提であり、左側のアダプター配列しかトリムしないやり方です。それが、preprocessReads関数実行時にLpatternのみ記載している理由です。残念ながら、QuasR (ver. 1.8.2)では「Removing adapters from paired-end samples is not yet supported」と出ます。2015年6月23日に開発者に実装を心待ちにしているとメールをしておきました。

```
in_f1 <- "SRR616268sub_1.fastq.gz"
in_f2 <- "SRR616268sub_2.fastq.gz"
out_f1 <- "hoge1_1.fastq.gz"
out_f2 <- "hoge1_2.fastq.gz"
param_adapter <- "GATCGGAAGAGCACACG
param_nrec <- 500000

#必要なパッケージをロード
library(QuasR)

#本番(前処理)
res <- preprocessReads(filename=in_
                       filenameMate=in_f2,
                       outputFilename=out_f1,
                       outputFilenameMate=out_f2,#前処理を実行
                       Lpattern=param_adapter)    #前処理を実行
res                                               #確認してるだけです
```

R Console

```
> getwd()
[1] "C:/Users/kadota/Desktop/hoge"
> list.files(pattern="sub_..fastq")
[1] "SRR616268sub_1.fastq.gz"
[2] "SRR616268sub_1_fastqc.html"
[3] "SRR616268sub_2.fastq.gz"
[4] "SRR616268sub_2_fastqc.html"
>
```

Paired-endの取扱い

QuasR (ver. 1.8.2)では、まだpaired-endデータのアダプター配列除去には対応できていないようです。

2. gzip圧縮FASTQ形式ファイル(SRR616268sub_1.fastq.gzとSRR616268sub_2.fastq.gz)の場合:

乳酸菌RNA-seqデータSRR616268の最初の100万リード分です。paired-endデータです。SRR616268sub_1.fastq.gzは、約75MB、全リード107 bpです。SRR616268sub_2.fastq.gzは、約67MB、全リード93 bpです。FastQC実行結果として「TruSeq Adapter, Index 3」が含まれているとレポートされました。これでググると塩基配列情報は"GATCGGAAGAGCACACGTCTGAACTCCAGTCACTTAGGCATCTCGTATGCCGTCTTCTGCTTG"と書いてあったので、これを入力として与えます。アダプター配列の位置は5'側(左側)にあるという前提であり、左側のアダプター配列しかトリムしないやり方です。それが、preprocessReads関数実行時にLpatternのみ記載している理由です。残念ながら、QuasR (ver. 1.8.2)では「Removing adapters from paired-end samples is not yet supported」と出ます。2015年6月23日に開発者に実装を心待ちにしているとメールをしておきました。

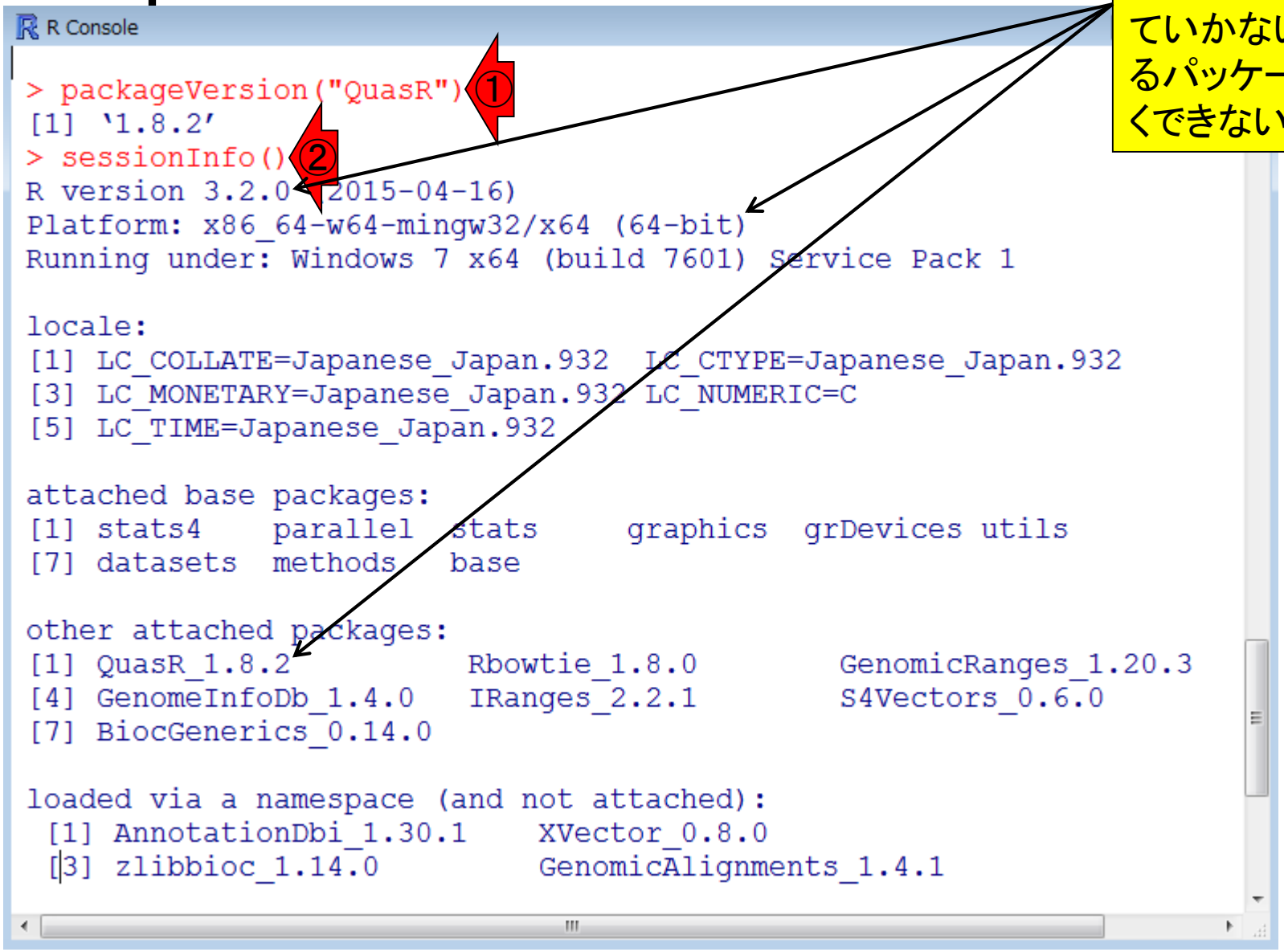
```
in_f1 <- "SRR616268sub_1.fastq.gz"      #入力ファイル名を指定してin_f1に格納
in_f2 <- "SRR616268sub_2.fastq.gz"      #入力ファイル名を指定してin_f2に格納
out_f1 <- "hoge1_1.fastq.gz"            #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge1_2.fastq.gz"            #出力ファイル名を指定してout_f2に格納
param_adapter <- "GATCGGAAG
param_nrec <- 500000

#必要なパッケージをロード
library(QuasR)

#本番(前処理)
res <- preprocessReads(file
              filenameMate=i
              outputFilename
              outputFilename
              Lpattern=param
res
```

R Console

```
 要求されたパッケージ Rbowtie をロード中です
>
> #本番(前処理)
> res <- preprocessReads(filename=in_f1, #前処理を実行
+                filenameMate=in_f2,     #前処理を実行
+                outputFilename=out_f1,  #前処理を実行
+                outputFilenameMate=out_f2,#前処理を実行
+                Lpattern=param_adapter)  #前処理を実行
Error in preprocessReads(filename = in_f1, filenameMate = in_f2, $
  Removing adapters from paired-end samples is not yet supported $
> res                                    #確認してるだけです
 エラー:  オブジェクト 'res' がありません
>
```

Tips: sessionInfo

①パッケージ名を指定してバージョン情報を得るやり方。②R本体のバージョンをちゃんと上げていかないと、インストールできるパッケージのバージョンも新しくできないので注意しましょう。



原著論文の引用

原著論文はきっちり引用しましょう。
①Bioconductorから提供されているQuasRパッケージのサイトへのリンク。②原著論文。③最新バージョン。

2. gzip圧縮FASTQ形式

乳酸菌RNA-seqデータS
は、約75MB、全リード1
して「TruSeq Adapter, I
"GATCGGAAGAGCAC
あったので、これを入力
ブター配列しかトリムし
す。残念ながら、QuasR
す。2015年6月23日に

```
in_f1 <- "SRR6162
in_f2 <- "SRR6162
out_f1 <- "hoge1_
out_f2 <- "hoge1_
param_adapter <-
param_nrec <- 500

#必要なパッケージを
library(QuasR)

#本番(前処理)
res <- preprocess
              fi
              ou
              outp
              Lpat

res
```

- SRP016842: Nie et a
- QuasR: Gaidatzis et al., Bioinformatics, 2015

①

QuasR

platforms some | downloads top 20% | posts
in Bioc 2 years | build ok | comm

Quantify and Annotate Short Reads

Bioconductor version: Release (3.1)

This package provides a framework for the quanti
complete workflow starting from raw sequence re
plots, to the quantification of genomic regions of

Author: Anita Lerch, Dimos Gaiditzis and Michael

Maintainer: Michael Stadler <michael.stadler at fr

Citation (from within R, enter citation("QuasR"

②

Gaidatzis D, Lerch A, Hahne F and Stadler MB (2015). "QuasR: Quantification and annotation of short reads in R." *Bioinformatics*, **31**(7), pp. 1130–1132. http://dx.doi.org/10.1093/bioinformatics/btu781, PMID:25417205.

Langmead B, Trapnell C, Pop M and Salzberg SL (2009). "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome." *Genome Biology*, **10**(3), pp. R25. http://dx.doi.org/10.1186/gb-2009-10-3-r25, PMID:19261174.

Au KF, Jiang H, Lin L, Xing Y and Wong WH (2010). "Detection of splice junctions from paired-end RNA-seq data by SpliceMap." *Nucleic Acids Research*, **38**(14), pp. 4570–4578. http://dx.doi.org/10.1093/nar/gkq211, PMID:20371516.

Details

biocViews	Alignment, ChIPSeq, Coverage, Genetics, MethylSeq, Preprocessing, Quality Control, RNASeq, Sequencing, Software
Version	1.8.3 ③
In Bioconductor since	BioC 2.13 (R-3.0) (2 years)
License	GPL-2
Depends	parallel, GenomicRanges(>= 1.13.3), Rbowtie
Imports	methods, zlibbioc, BiocGenerics, S4Vectors, IRanges, BiocInstaller, Biobase, Biostrings, BSgenome, Rsamtools(>= 1.19.38), GenomicFeatures(>= 1.17.13), ShortRead(>= 1.19.1), GenomicAlignments, BiocParallel, GenomeInfoDb
LinkingTo	Rsamtools
Suggests	rtracklayer, Gviz, RUnit, BiocStyle
SystemRequirements	
Enhances	
URL	
Depends On Me	
Imports Me	
Suggests Me	