

平成28年度NGSハンズオン講習会 Reseq解析

2016年7月26日

amelieff

本講義にあたって

- 代表的な解析の流れを紹介します。
 - 論文でよく使用されているツールを使用します。
- コマンドを沢山実行します。
 - スペルミスが心配な方は、コマンド例がありますのでコピーして実行してください。
 - 実行が遅れてもあせらずに、課題や休憩の間に追いついてください。

本講義の内容

前半パート (講義)

- Reseqとは
- 検出可能な遺伝子変異
- 解析パイプラインとは
- 公開データの取得と利用
- クオリティコントロールとは
- マッピングとは
- 変異検出とは
- アノテーションとは
- より高精度な分析のために
- 後半パート (実習)で行うこと

後半パート (実習)

- 公開データの確認
- クオリティコントロール
- マッピング
- 変異検出
- アノテーション
- 解析結果の可視化
- まとめ
- 最後に

講義パート

Reseqとは

whole genome sequence
whole exome sequence
amplicon sequence
target sequence
⋮

Reseq

DNAの変異検出を目的としたワークフローの総称

Reseq

- ① 公開データ取得
 - ② クオリティコントロール
 - ③ マッピング
 - ④ 変異検出
- SNVとIndelの検出を行います。

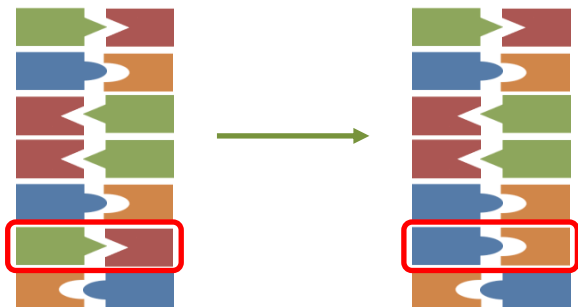
RNA-seq (明日実施)

- ① 公開データ取得
 - ② クオリティコントロール
 - ③ マッピング
 - ④ 発現定量
- FPKMを算出します。

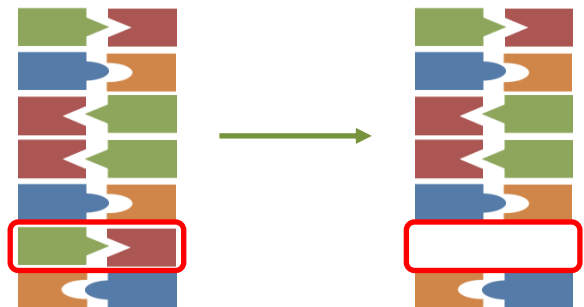
検出可能な遺伝子変異

ショートリードのシーケンスでも様々な変異を検出可能

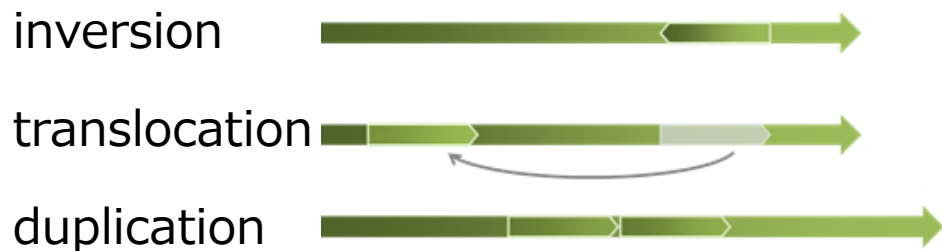
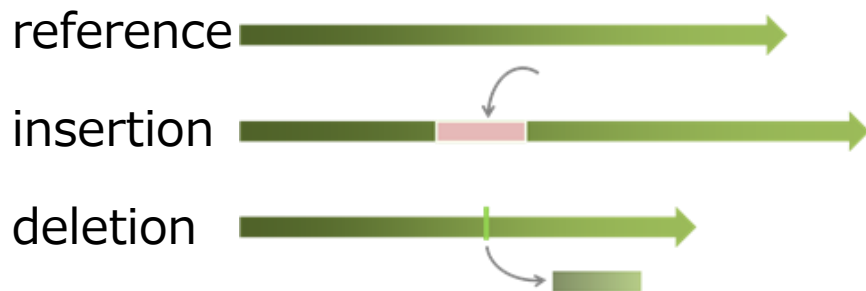
SNV
(Single Nucleotide Variant)



InDel
(Insertion & Deletion)



SV
(Structural Variation)

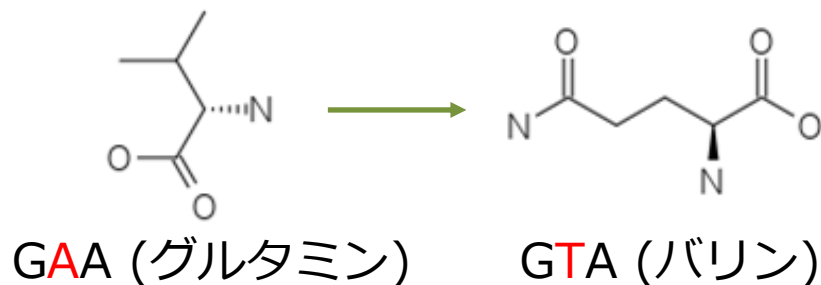


検出可能な遺伝子変異

各変異による影響例

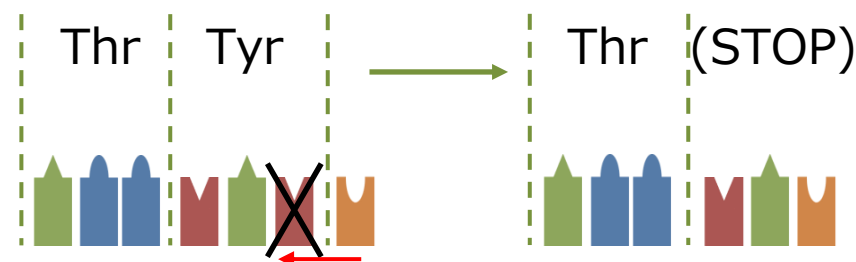
SNV

(Single Nucleotide Variant)



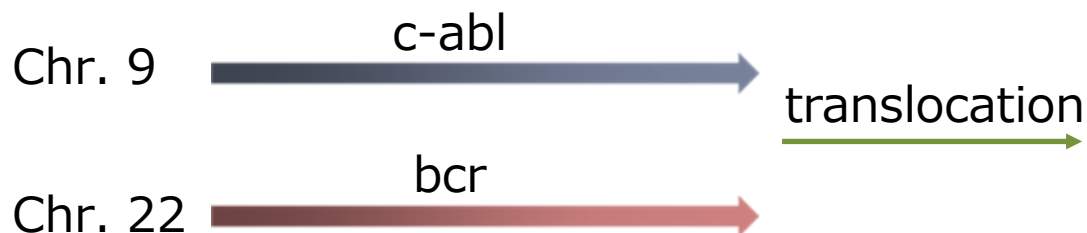
InDel

(Insertion & Deletion)



SV

(Structural Variation)

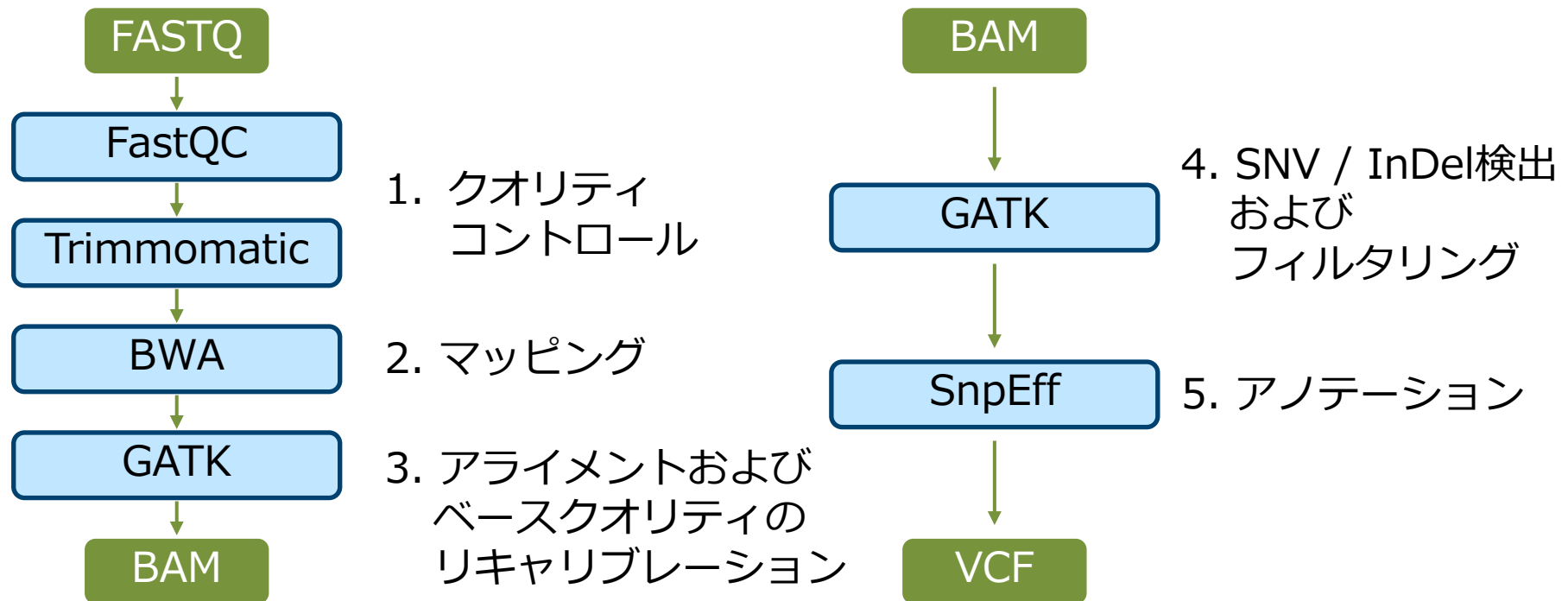


Philadelphia chromosome



解析パイプラインとは

あるソフトの出力結果が、次のソフトの入力ファイルとなる連続した解析処理の流れ。



公開データの取得と利用

「変異」 $\leftrightarrow$ 「疾患」の関連付け

コントロール群由来



疾患群由来



Deletionの発見



~~「変異」 $\leftrightarrow$ 「疾患」~~

疾患
人種
性別
普遍的な変異
⋮

解析結果



公開データ



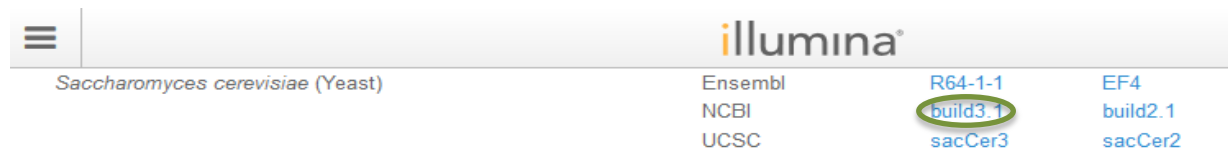
原因遺伝子変異特定

公開データの取得と利用

今回の解析に必要なデータ (ダウンロード済み)

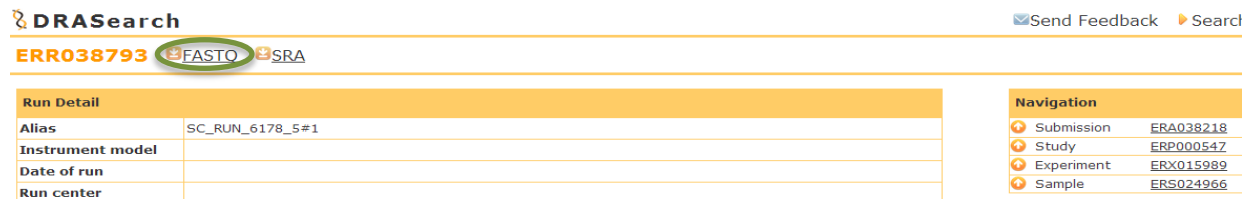
■ リファレンスゲノム

- http://support.illumina.com/sequencing/sequencing_software/igenome.html



■ 解析対象のシーケンスデータ

- ftp://ftp.ddbj.nig.ac.jp/ddbj_database/dra/fastq/ERA038/ERA038218/ERX015989/ERR038793_1.fastq.bz2
- ftp://ftp.ddbj.nig.ac.jp/ddbj_database/dra/fastq/ERA038/ERA038218/ERX015989/ERR038793_2.fastq.bz2



公開データの取得と利用

酵母のリファレンスゲノムデータの取得方法

```
$ wget ftp://igenome:G3nom3s4u@ussd-  
ftp.illumina.com/Saccharomyces_cerevisiae/NCBI/build3.1/Saccha  
romyces_cerevisiae_NCBI_build3.1.tar.gz  
$ tar zxvf Saccharomyces_cerevisiae_NCBI_build3.1.tar.gz
```

IlluminaのWebページからリファレンスゲノムを取得し解凍 (実行済み)。

```
$ ls -l /home/iu/genome/sacCer3  
:  
-rwxrwxr-x 1 iu iu 12400379 7月 4 19:50 genome.fa  
-rwxrwxr-x 1 iu iu 14 7月 4 19:50 genome.fa.amb  
-rwxrwxr-x 1 iu iu 562 7月 4 19:50 genome.fa.ann  
:
```

/home/iu/genome/sacCer3に配置してあります。

公開データの取得と利用

解析対象のシーケンスデータの取得方法 ① (実行済み)

<http://trace.ddbj.nig.ac.jp/dra/index.html>へアクセス。



[Login & Submit](#) | [Databases▼](#) | [Japanese](#) | [Contact](#)

Google™ Custom Search



Sequence Read Archive click!!

[Home](#)

[Handbook](#)

[FAQ](#)

[Search](#)

[Download▼](#)

[Pipeline](#)

[About DRA](#)

News

2014-05-13: **New DRA submission system is released.** [less...](#)

We have released the new DRA submission system. For major changes, please see the [slides](#) and [new handbook](#).

(6th, June, 2014)

For submissions with status "new" which had been created before 12th, May, 2014, addition or deletion of metadata objects could cause errors. It is recommended that download metadata as a tab-delimited text file and upload it into a newly created submission.

DDBJ Sequence Read Archive (DRA) is an archive database for output data generated by next-generation sequencing machines including Roche 454 GS System®, Illumina Genome Analyzer®, Applied Biosystems SOLiD® System, and others. DRA is a member of the International Nucleotide Sequence Database Collaboration (INSDC) and archiving the data in a close collaboration with NCBI Sequence Read Archive (SRA) and EBI Sequence Read Archive (ERA). Please submit the trace data from conventional capillary sequencers to DDBJ Trace Archive.

公開データの取得と利用

解析対象のシーケンスデータの取得方法 ②（実行済み）
ERR038793を検索。

 **DRASearch** type!!

Accession :

Organism :

StudyType :

CenterName :

Platform :

Keyword :

Show records Sort by

click!!

もちろんキーワード検索も可能

公開データの取得と利用

解析対象のシーケンスデータの取得方法 ③（実行済み）
実験詳細を確認。

DRASearch Send Feedback Search

ERR038793 FASTQ SRA ここからダウンロード可能

Run Detail	
Alias	SC_RUN_6178_5#1
Instrument model	
Date of run	
Run center	
Number of spots	739,873
Number of bases	147,974,600

Navigation	
Submission	ERA038218
Study	ERP000547
Experiment	ERX015989
Sample	ERS024966

click!!

READS (joined) quality show 10 rows << < 1 / 73988 Page > >>

```
>ERR038793.1
GGACAAGGTTACTTCCTAGATGCTATATGTCCCTACGGCCTTGTCTAACCCATCCAGCATGCAATAAGGTGACATAGAT
ATCCCAACACACCCACCCCTGTGGAGTTGGATATGGGTAAATTGGAGGGTAACGGTGGGTGAGTGGTAGTAAGTAGAGGGA
TGGATGGTGGTTCGGAGGGGTATGGTTGGATGGGACAGGG

>ERR038793.2
TGGTGGTATAAAGTGGTAGGGTAAGTATGTGTGTTATTTACGATCATTTGTTAGCGTTTCAATATGGTGGGTAAAAAC
GCAGGATAGTGAGTTACCGAACACACACCAACCCACACACCCACACACCCACACACCCACACACCCACACCCAC
ACCCACACCCACACCCACACACCCACACCCACTAACCCCTAA
```

公開データの取得と利用

解析対象のシーケンスデータの取得方法 ④（実行済み）

シーケンスデータの情報を確認。

Experiment Detail	
Title	
Design Description	Illumina sequencing of library 2414804, constructed from sample accession ERS024966 for study accession ERP000547. This is part of an Illumina multiplexed sequencing run (6178_5). This submission includes reads tagged with the sequence ATCACGTT.
Organism	Saccharomyces cerevisiae
Library Description	
Name	2414804
Strategy	WGS
Source	GENOMIC
Selection	RANDOM
Layout	PAIRED
Orientation	
Nominal Length	462
Nominal Sdev	41
Construction Protocol	Standard

※ 一部のみ記載

公開データの取得と利用

解析対象のシーケンスデータの取得方法 ⑤ (実行済み)

CUIでダウンロード。

```
$ cd /home/iu/reseq/data  
  
$ wget ¥  
ftp://ftp.ddbj.nig.ac.jp/ddbj_database/dra/fastq/ERA038/ERA038  
218/ERX015989/ERR038793_1.fastq.bz2  
  
$ wget ¥  
ftp://ftp.ddbj.nig.ac.jp/ddbj_database/dra/fastq/ERA038/ERA038  
218/ERX015989/ERR038793_2.fastq.bz2
```

/home/iu/reseq/dataにダウンロードしてあります。

公開データの取得と利用

解析対象のシーケンスデータの取得方法 ⑥ (実行済み)
圧縮ファイルの解凍。

```
$ ls
```

```
ERR038793_1.fastq.bz2  ERR038793_2.fastq.bz2
```

今回用いるデータはbz2形式で圧縮されていました。

```
$ bzip2 -d ERR038793_1.fastq.bz2
```

```
$ bzip2 -d ERR038793_2.fastq.bz2
```

/home/iu/reseq/dataには解凍済のファイルが配置してあります。
ソフトウェアによっては圧縮されたままのファイルを扱えるものもあります。

クオリティコントロールとは

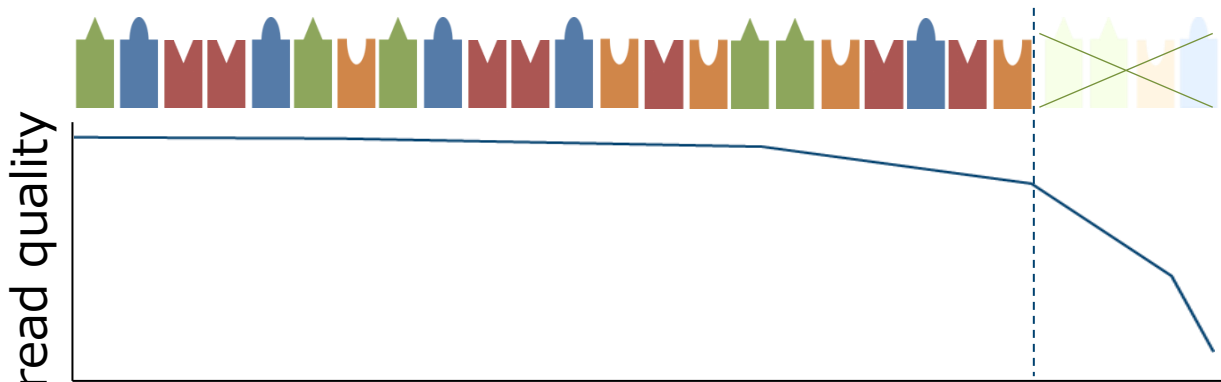
公開データをそのまま使うのは危険。



測定環境の違い
シーケンス結果のクオリティ
アダプター配列の有無
タグの有無
⋮



アダプター、タグの除去



シーケンスクオリティ
の悪い塩基をトリム
または
低クオリティのリードを
除去

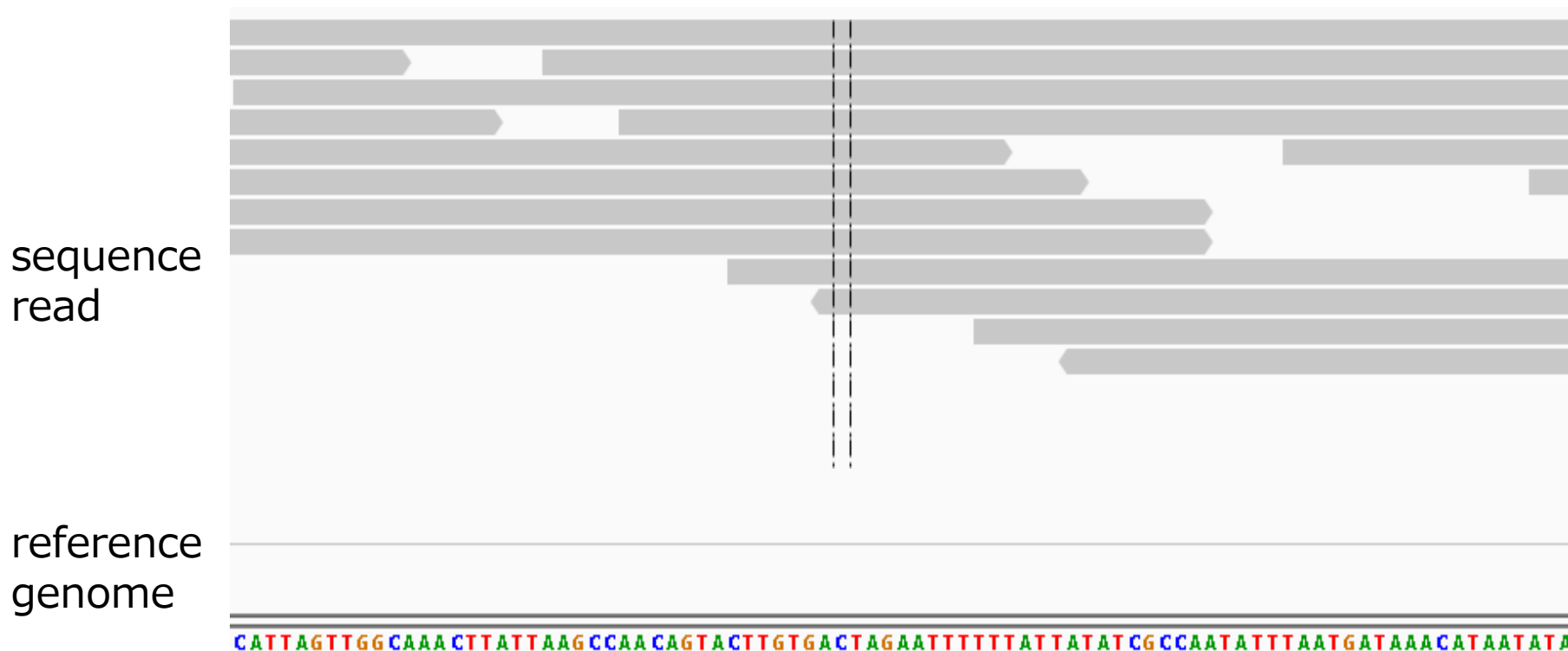
クオリティコントロールとは

ゲノム解析で用いられる主なクオリティコントロールツール。

- FastQC … クオリティチェック用ソフトウェア。
- FASTX-toolkit … Cで書かれた多機能クオリティコントロールツール。
- PRINSEQ … Perlで書かれた多機能クオリティコントロールツール。
- Trimmomatic … Javaで書かれたトリミングツール。
- etc...

マッピングとは

各リードはリファレンスゲノムのどこに位置するか調べる。



Reseq解析は、リファレンスに対して変異検出するので、リファレンス自体がどの程度確かなのかが非常に大切

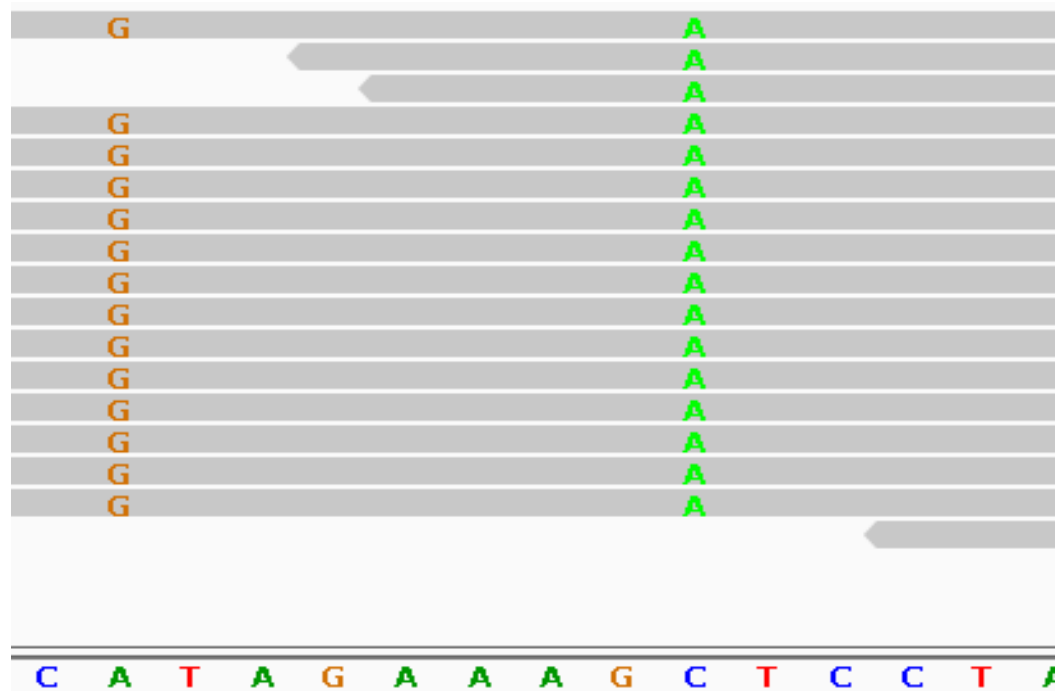
マッピングとは

ゲノム解析で用いられる主なマッピングツール。

- BWA … Indelに強いギャップ許容型のマッピングツール。
- Bowtie2 … ショートリード用のマッピングツール。
- SOAP2 … 大量ショートリード高速マッピングツール。Indel不可。
- Novoalign … NovoCraft社の製品。ギャップ許容型のマッピングツール。
- etc...

変異検出とは

マッピングされたリードを元にリファレンスゲノムとの比較を行う。



WGSではこういった変異が数万～数十万検出されるのでひとつずつ確認することは不可能です。

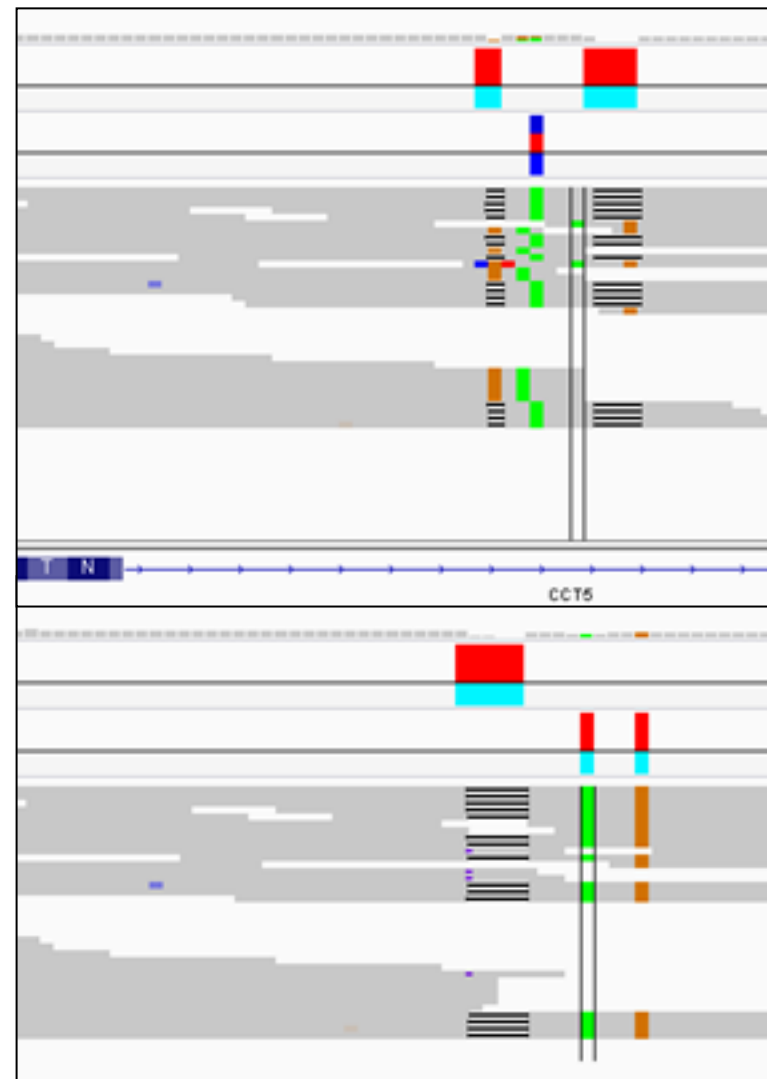
変異検出とは

変異検出の前に ① ~Realignment~

リアライメントとは？

1本のリードに複数の変異が含まれる場合に、アライメントスコアの計算上、SNVやIndelの正確な位置を決定できないことがあります。

このような領域を対象領域として抜き出して、改めて丁寧にアライメントを行うことで変異検出の信頼性を高めることができます。



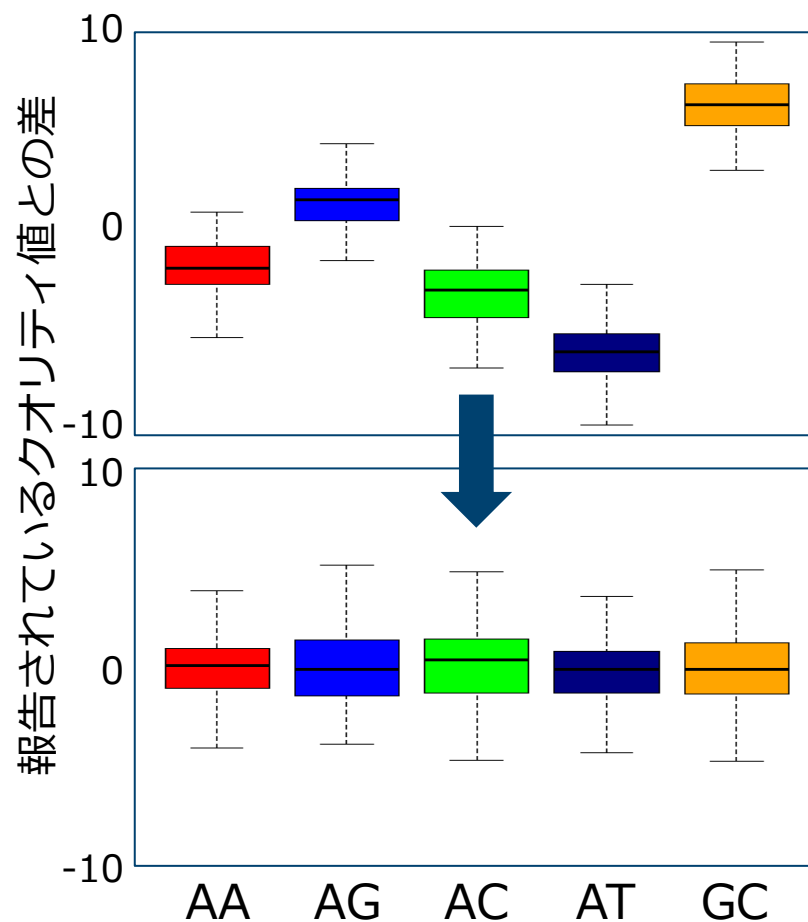
変異検出とは

変異検出の前に ② ~Base Recalibration~

ベースリキャリブレーションとは？

変異検出のアルゴリズムはクオリティスコアに大きく影響されます。

この行程では既知のSNP情報を用いて、測定環境により異なるクオリティスコアをノーマライズすることで、測定環境に依存しない変異検出が可能となります。



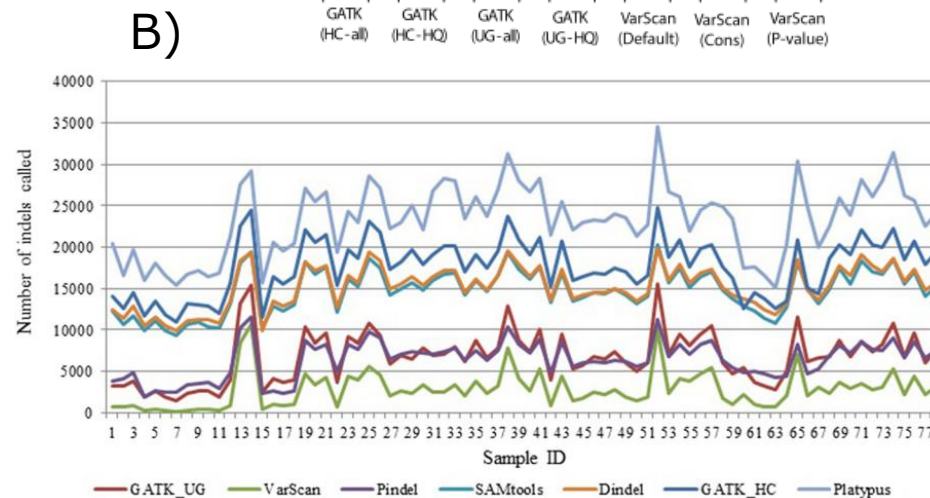
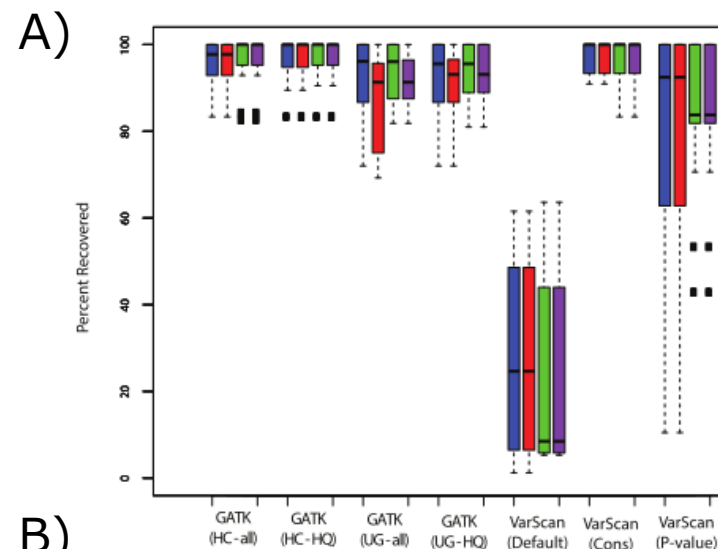
変異検出とは

ゲノム解析で用いられる変異検出ツール。

- GATK HaplotypeCaller
- GATK UnifiedGenotyper
- VarScan
- etc...

A)
Charles D. Warden *et al.*, *Peer J*, 2014

B)
Hasan *et al.*, *Human Genomics*, 2015



変異検出とは

最新版GATK

Realignmentは
GATK HaplotypeCallerや
MuTect2を使用すれば
必要ないということで、
GATK3.6から非推奨項目に・・・。

実習では
GATK HaplotypeCallerによる変異の一括検出を行います。

[Best-Practices](#) [Documentation](#) [Blog](#) [Forum](#) [Download](#)



Version highlights for GATK version 3.6

Posted by [Geraldine_VdAuwer](#) on 1 Jun 2016 (6)

What better way to start the summer than with a new GATK release?

Umm no don't answer that, there's loads of good options. You could have a barbecue, eat some ice cream, go on a hike if that's the sort of thing that floats your kayak... Or you might live somewhere where winter is just starting and everything I just said there was terribly insensitive. Sorry.

Ahem. As I mentioned in my recent [sneak preview](#) blog post, the bulk of our development effort (speed! copy number! unicorns!) is now going into the GATK4 project. Accordingly, development in the GATK3 framework is winding down, so this release consists mainly of bug fixes, added convenience functions, and relatively minor behavior tweaks.

That being said we do have a few new experimental features in the VQSR tools (which haven't yet been fully ported to GATK4, hence the ongoing development in GATK3) that are pushing the envelope of allele-specific filtering. So that's interesting, if not yet fully documented (someone should really get on that). And you'll probably care about some of those tweaks I casually mentioned above -- in fact I guarantee that at least one of these things will matter to you in some way. If you read through the whole thing and don't find *anything* relevant to you, tell me in the comments that I was wrong. That's what the Internet is for.

As usual, here I go over the changes that matter the most / to the most; consider going through the [release notes](#) as well for a full list of changes.

One version of Java to rule them all

Possibly the most sweeping change in this version is that it introduces support for Java 8. As noted recently, when we switched our test framework to Java 8 we encountered multiple failures in GATK 3.5 tests, which I discussed [here](#). We fixed the underlying issues, so from version 3.6 onward GATK now runs reliably on Java 8. As a nice bonus, this puts us back into sync with HTSJDK and the Picard toolkit, which had been running on Java 8 for a few months already. If you were doing it right, you had both versions of Java installed and ran each toolkit, GATK and Picard, on the appropriate version. How much hassle? Too much hassle! — Now you can run everything on Java 8.

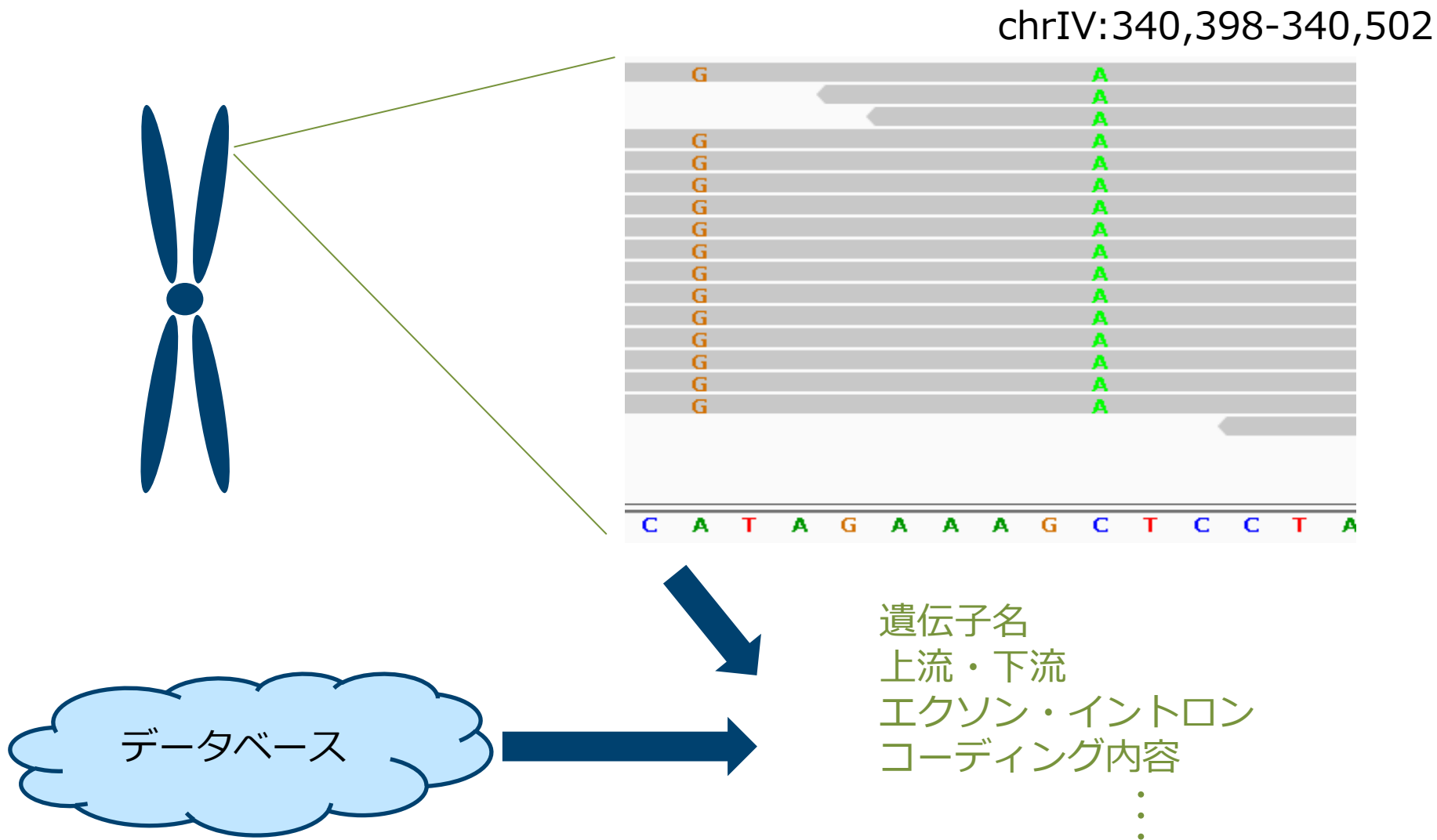
ET finally gets home; discovers phone bill, flees to Canada

Here's another change that will affect everyone regardless of use case, in a good way: we removed the [Phone Home](#) usage reporting system. It served its purpose for many years, but we've outgrown it. So we ripped it all out. If you previously used a key to deactivate it with the `-et NO_ET` and `-key <your_key>` arguments, you can stop and take those out. Or if you're just too busy, leave them in -- the Phone Home code is all gone but we left in the argument definitions, so the parser shouldn't fail if you leave them in your commands. This shouldn't break any scripts.

Indel realignment tools drop out, go open-source

In the next few days we'll be making some updates to our Best Practices documentation to reflect updates to our production pipelines, and one thing you'll notice is that local realignment around indels is no longer included in the pre-processing part of the pipelines. More on that in a follow-up post; in a nutshell, indel realignment is just not useful enough anymore when you're calling variants with haplotype-based tools like HaplotypeCaller and MuTect2. This does mean we will no longer support the indel realignment tools as actively; but since others may care more about preserving and possibly even expanding this functionality, we've decided to move the relevant tools, IndelRealigner and RealignerTargetCreator, and related classes to the part of the GATK that is open-sourced under the MIT license.

アノテーションとは



アノテーションとは

ゲノム解析で用いられるアノテーションツール。

- SnpEff … 高性能なアノテーションツール。ヒト以外にも対応。
- Annovar … 高性能なアノテーションツール。ヒトのみ。
- Seattle Seq … Webベースのアノテーションツール。
- etc...

後半パート (実習)で行うこと

本日実際に行う解析フロー。



実習パート

はじめに

reseqディレクトリに移動してください。

```
$ cd /home/iu/reseq  
$ ls  
data
```

講義に使用するテストデータが置いてあります。

使用する際には指示があります。

公開データの確認

fastaファイルの中身を見てみる。

```
$ less /home/iu/genome/sacCer3/genome.fa  
>chrI  
CCACACCACACCCACACACCCACACACCACACACCACACACCACACACC  
CACACACACACATCCTAACACTACCCTAACACAGCCCTAATCTAACCCCTG  
GCCAACCTGTCTCTCAACTTACCCTCCATTACCCTGCCTCCACTCGTTAC  
CCTGTCCCATTCAACCATACTACTCCGAACCACCATCCATCCCTCTACTT  
ACTACCACTCACCCACCGTTACCCTCCAATTACCCATATCCAACCCACTG  
:
```

1行目： コンティグ名

2行目以降： 実際の配列情報

※ 「q」 で閲覧を終了

公開データの確認

解析対象のfastqファイルを確認。

```
$ less data/ERR038793_1.fastq
```

```
@ERR038793.1 1 length=100
GGACAAGGTTACTTCCTAGATGCTATATGTCCCTACGGCCTTGTCTAACACCATCCAGCATGCA
ATAAGGTGACATAGATATACCCACACACCACACCCT
+ERR038793.1 1 length=100
D/DDBD@B>DFEEEEEEEEEF@FDEEEBEDBBDDD:AEEE<>CB?FCFF@F?FBFF@?:EEE:E
EBEEEB=EEE.>>?=AD=8CDFFFFEFEF@C?;DC
```

fastqファイルを見える。

- 1行目： @配列IDと付加情報
- 2行目： 塩基配列
- 3行目： +配列IDと付加情報
- 4行目： クオリティ

※ fastqファイルは1リードあたり4行で表記されます。

公開データの確認

解析対象データのリード数を確認。

```
$ wc -l data/ERR038793_1.fastq
2959488 data/ERR038793_1.fastq
```

2959488行あるので、リード数は
 $2959488 / 4 = 739872$ となる。

```
$ wc -l data/ERR038793_2.fastq
2959488 data/ERR038793_2.fastq
```

ペアエンドなのでERR038793_2.fastqは
もちろん同じリード数を持つ。

クオリティコントロール

シーケンスクオリティチェックソフトウェアFastQCの紹介

```
$ fastqc -v
```

```
FastQC v0.10.1
```

バージョンを確認 (2016年7月現在、最新版はv0.11.5)。

```
$ fastqc -h
```

```
FastQC - A high throughput sequence QC analysis tool
```

```
SYNOPSIS
```

```
fastqc seqfile1 seqfile2 .. seqfileN
```

```
fastqc [-o output dir] [--(no)extract] [-f fastq|bam|sam]  
      [-c contaminant file] seqfile1 .. seqfileN  
      :
```

.fastq以外に.samや.bamも可能、複数ファイルの指定も可能。

クオリティコントロール

シーケンスクオリティチェックソフトウェアFastQCの実行

```
$ mkdir fastqc_before  
$ fastqc -o fastqc_before -f fastq ¥  
  -c data/ERR038793_1.fastq data/ERR038793_2.fastq  
$ ls fastqc_before
```

```
ERR038793_1_fastqc      ERR038793_2_fastqc  
ERR038793_1_fastqc.zip ERR038793_2_fastqc.zip
```

解析結果のhtmlファイルができていますので、これをブラウザ (firefox)で確認してみます。

```
$ firefox ¥  
  fastqc_before/ERR038793_1_fastqc/fastqc_report.html ¥  
  fastqc_before/ERR038793_2_fastqc/fastqc_report.html
```

ブラウザでタブが2つ開かれ、
クオリティチェックの解析結果が確認できます。

クオリティコントロール

FastQCの結果確認 ①

Summary

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

✓ 問題なし

! 注意 (warning)

✗ 問題あり (failure)



Basic Statistics

Measure	Value
Filename	ERR038793_1.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	739872
Filtered Sequences	0
Sequence length	100
%GC	37

Basic Statistics

ファイルの基本的な情報。

ファイルタイプや、リード数、リード長などの情報が表示される。

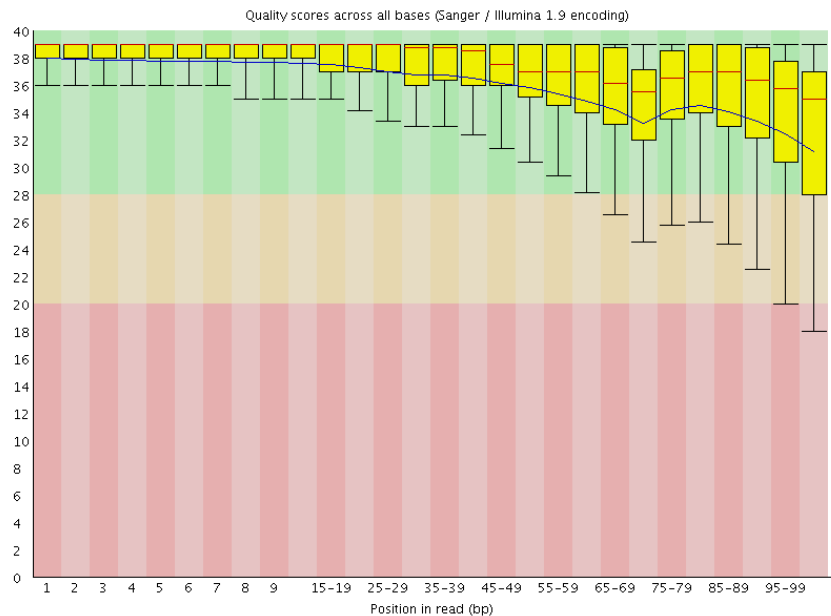
ここではwarning, failureは出ない。

クオリティコントロール

FastQCの結果確認 ②



Per base sequence quality



Per Base Sequence Quality

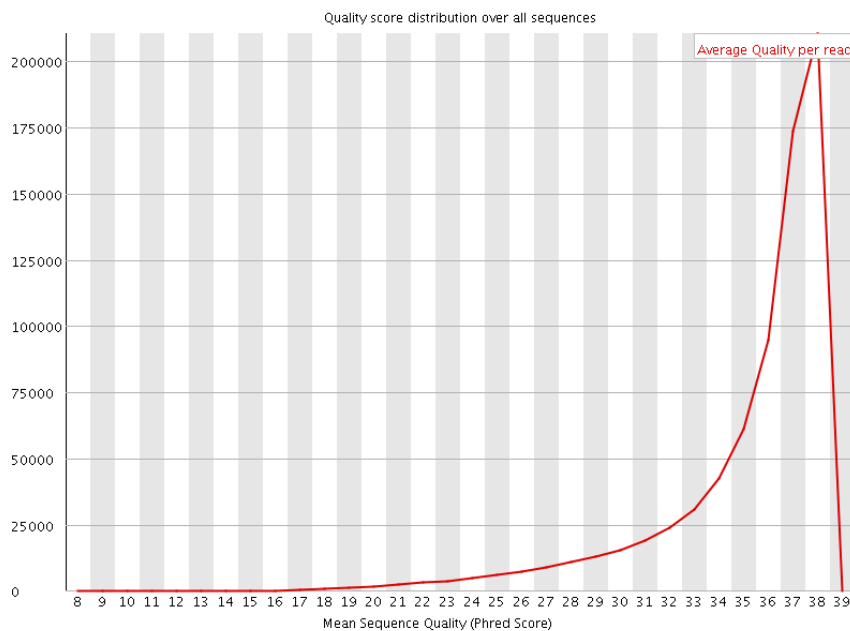
横軸はリード長、縦軸はquality valueを表す。

リードの位置における全体のクオリティの中央値や平均を確認できる。赤線は中央値、青線は平均値、黄色のボックスは25%～75%の領域を表す。上下に伸びた黒いバーが10%～90%の領域を意味する。

クオリティコントロール

FastQCの結果確認 ③

✔ Per sequence quality scores

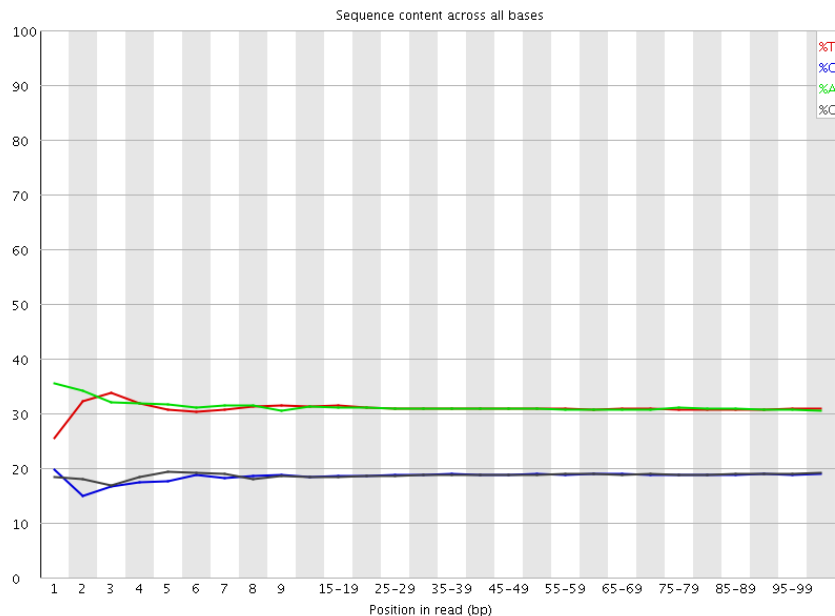


Per Sequence Quality Scores
縦軸がリード数、横軸がPhred
quality scoreの平均値。

クオリティコントロール

FastQCの結果確認 ④

🚩 Per base sequence content



Per Base Sequence Content

リードにおける位置での各塩基の割合を示す。

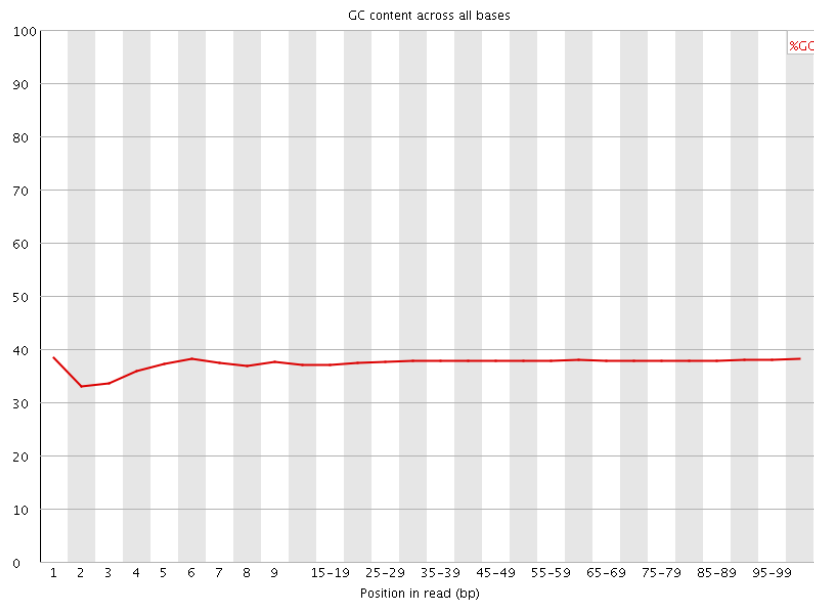
いずれかの位置で、AとTの割合の差、もしくはGとCの割合の差が10%以上だとwarning、20%以上でfailureとなる。

クオリティコントロール

FastQCの結果確認 ⑤



Per base GC content



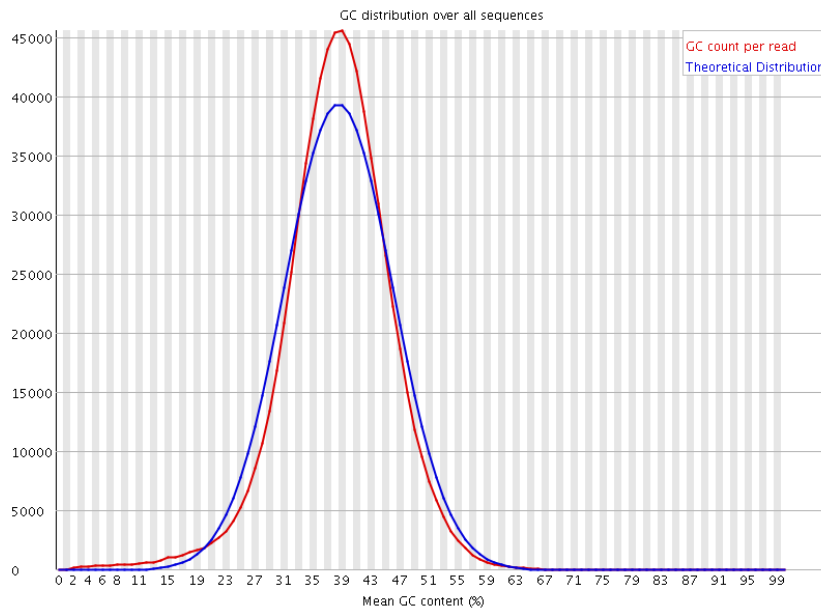
Per Base GC Content

リードにおける位置でのGC含量を表す。いずれかの位置で、全体でのGC含量の平均値より5%以上の差が開くと warning, 10%でfailureとなる。

クオリティコントロール

FastQCの結果確認 ⑥

✔ Per sequence GC content



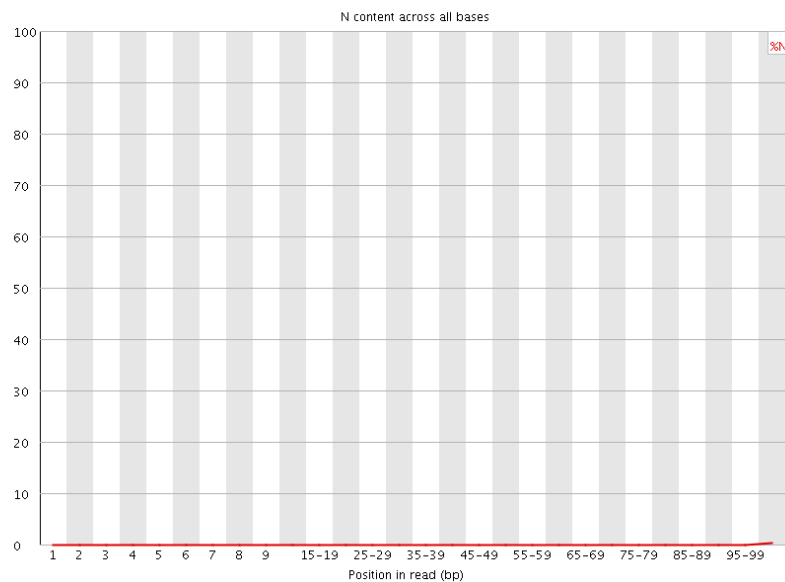
Per sequence GC content

リードの GC 含量の分布が示されている。リードに含まれる GC 含量は一般に正規分布に従うとされている。正規分布と比較し、その残差が 15% 以上ならば Warning とされる。また、30% 以上ならば Failure とされる。

クオリティコントロール

FastQCの結果確認 ⑦

✔ Per base N content



Per Base N Content

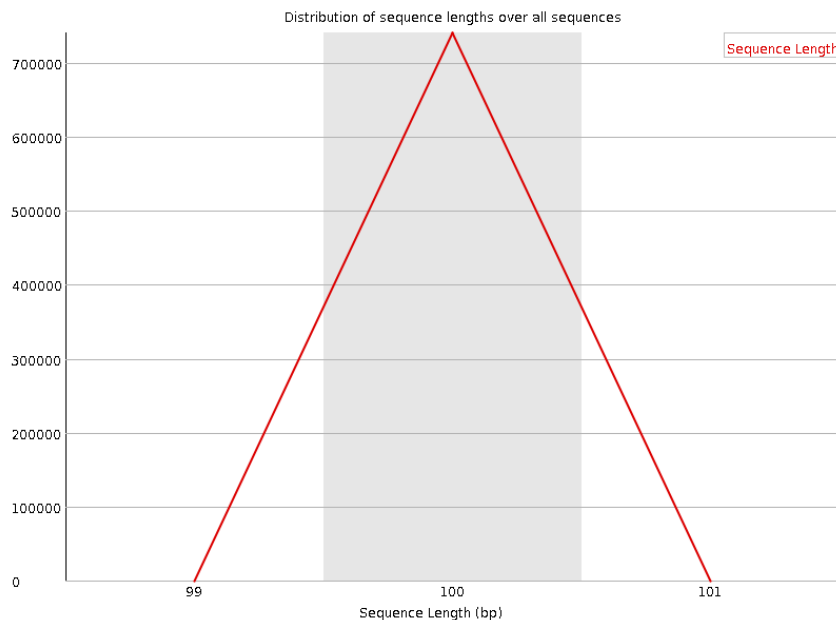
“N”はシーケンサーの問題でATGCいずれの塩基にも決定出来なかった場合に記述される。

リードのいずれかの位置で5%以上Nが存在するとwarning, 20%以上でfailureとなる。

クオリティコントロール

FastQCの結果確認 ⑧

✔ Sequence Length Distribution



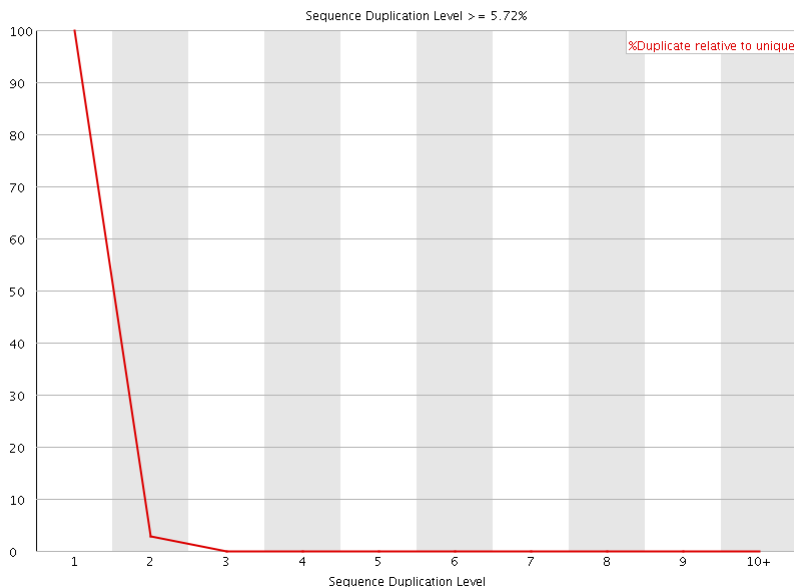
Sequence Length Distribution

リード長の全体の分布。
全てのリードの長さが同じであることを前提としており、一定でなければwarning、ゼロのものが含まれているとfailureになる。

クオリティコントロール

FastQCの結果確認 ⑨

✔ Sequence Duplication Levels



✔ Overrepresented sequences

No overrepresented sequences

Sequence Duplication Levels

リードの重複レベルを見ている。
1~10はそれぞれ重複のレベルで、
全体の20%以上がユニークでないものだとwarning, 50%以上がユニークでないとはfailureとなる。

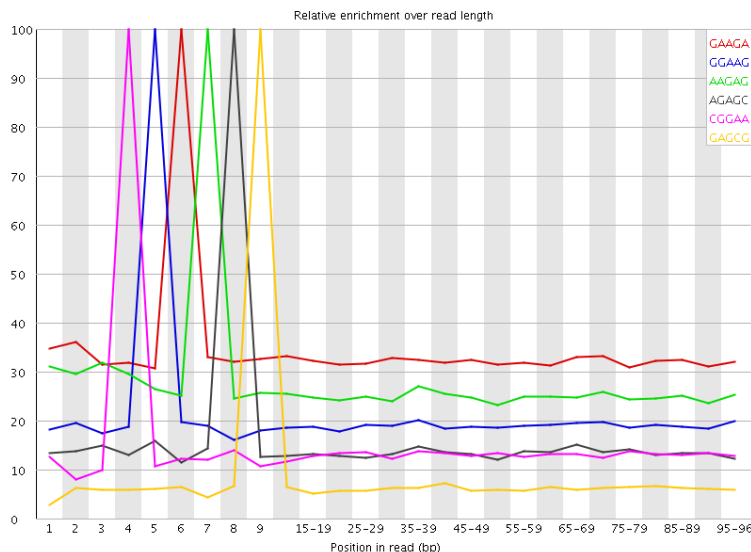
Overrepresented Sequences

重複している配列とその割合を表す。
特定の配列が全リードの0.1%を超えるとwarning、1%を超えるとfailureとなる。

クオリティコントロール

FastQCの結果確認 ⑩

⚠ Kmer Content



K-mer Content

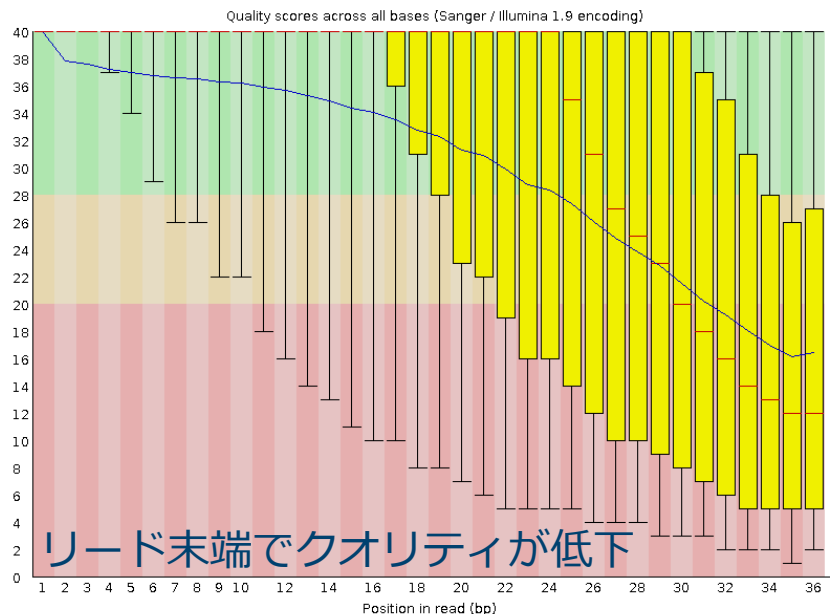
K bpの任意の配列(K-mer)を考えた時、ライブラリに含まれるATGCの割合を元に「実際に観測された値/理論的に観測される期待値」を計算している (デフォルトはK=7)。それぞれの任意の配列について、実測が期待値を大きく上回っている時、それはライブラリに配列的な偏りがあると解釈される。

「実測値/期待値」は、リード長全体における計算と、リードのある位置での計算を行い、全体における値が3倍、リードのある位置における値が5倍になるとwarning、リードのある位置における値が10倍になるとfailureとなる。

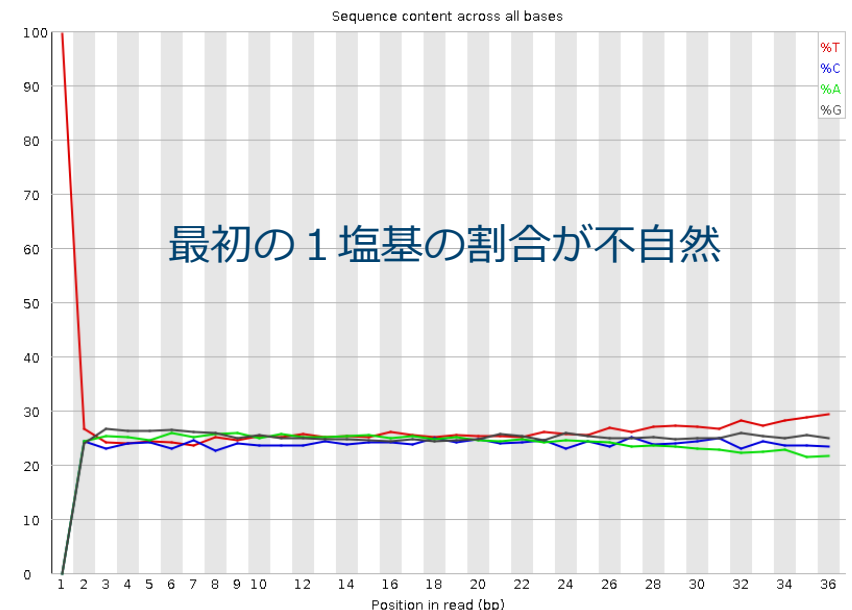
クオリティコントロール

補足) クオリティの悪いデータ

✖ Per base sequence quality



✖ Per base sequence content



マッピング率の低下や、変異の偽陽性が増加するなどの問題を引き起こす。

シーケンス技術が向上しクオリティの高いデータを目にする機会が増えましたが、試料・シーケンス・トリミングなどに、問題がないか確認することをおすすめします。

クオリティコントロール

クオリティを向上させるために (amelieffの場合)



様々な流儀が存在するが、大切なのは「処理の**内容**」と「処理の**順番**」。

例えば
ロングリードの場合、リードの大半が除外されてしまう可能性。

例えば
ペアエンドリードの場合、ペアが揃っていないとマッピングソフトが停止する可能性。

クオリティコントロール

今回のデータに対する処理 (Trimmomaticを用いた一括処理)

```
$ mkdir trimmed_data
$ java -jar /usr/local/bin/trimmomatic-0.36.jar ¥
PE -threads 2 -phred33 -trimlog trimmed_data/log.txt ¥
data/ERR038793_1.fastq ¥
data/ERR038793_2.fastq ¥
trimmed_data/paired_output_ERR038793_1.fastq ¥
trimmed_data/unpaired_output_ERR038793_1.fastq ¥
trimmed_data/paired_output_ERR038793_2.fastq ¥
trimmed_data/unpaired_output_ERR038793_2.fastq ¥
SLIDINGWINDOW:5:20 LEADING:20 TRAILING:20 MINLEN:80
```

Summary

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

今回解析するデータはFastQCによるクオリティチェックの結果、「問題あり」と判断された項目はありませんでした。そのため、今回はリード末端のクオリティが悪い部分をトリムすることでクオリティの底上げを図ります。

クオリティコントロール

Trimmomaticのオプション

- PE: ペアエンドの指定。
- -threds: 使用するスレッド数。
- -phred33: クオリティスコアの計算方法。
- -trimlog: logファイルの名前指定。
- SLIDINGWINDOW: ウィンドウサイズと平均クオリティの設定。
- LEADING: リードの先頭からトリム位置を探した時の下限クオリティ値。
- TRAILING: リードの末端からトリム位置を探した時の下限クオリティ値。
- MINLEN: リードそのものを除去する設定値。

クオリティコントロール

QC後の結果確認

```
$ mkdir fastqc_after  
  
$ fastqc -o fastqc_after -f fastq -c ¥  
  trimmed_data/paired_output_ERR038793_1.fastq ¥  
  trimmed_data/paired_output_ERR038793_2.fastq  
  
$ firefox ¥  
  fastqc_after/paired_output_ERR038793_1_fastqc/¥  
  fastqc_report.html ¥  
  fastqc_after/paired_output_ERR038793_2_fastqc/¥  
  fastqc_report.html
```

クオリティコントロール

Trimmomaticによるクオリティコントロールの結果

データクオリティチェック (FastQC)

クオリティ20未満のリード末端をトリム

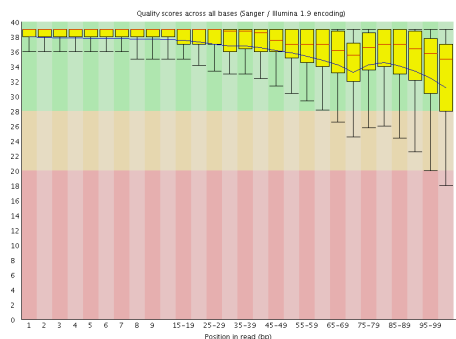
配列長が短いリード除去

片側のみのリードを除外

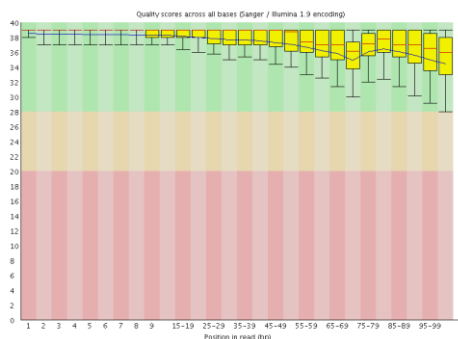
データクオリティチェック (FastQC)

解析するデータにアダプター配列が含まれている場合、Trimmomaticを用いてアダプター配列を除去することも出来る。

✓ Per base sequence quality



✓ Per base sequence quality



リード末端のクオリティが悪かった部分がトリムされました。

マッピング

BWA memによるマッピング準備

```
$ bwa mem -help
```

```
Usage: bwa mem [options] <idxbase> <in1.fq> [in2.fq]  
:
```

インデックスファイルと対象のfastqファイルが要求されている。

例) リファレンスゲノムのFASTAファイル
に対するインデックスファイル



マッピング

BWA memのインデックスファイル作成

```
$ mkdir -p Scerevisiae/BWAIndex  
$ cd Scerevisiae/BWAIndex  
$ ln -s /home/iu/genome/sacCer3/genome.fa  
$ bwa index genome.fa  
$ ls
```

genome.fa	genome.fa.ann	genome.fa.pac
genome.fa.amb	genome.fa.bwt	genome.fa.sa

インデックスファイルを格納するフォルダを作成し移動、
リファレンスゲノムのシンボリックリンクを作成し、
それを用いてインデックスファイルを作成する。

マッピング

BWA memによるマッピング (ここからは1ファイルで行います。)

```
$ cd /home/iu/reseq
$ mkdir mapping
$ bwa mem -M ¥
  -R '@RG¥tID:ERR038793_1¥tSM:ERR038793¥tPL:Illumina' ¥
  Scerevisiae/BWAIndex/genome.fa ¥
  trimmed_data/paired_output_ERR038793_1.fastq > ¥
  mapping/ERR038793_mapped.sam
```

-M: SAM/BAMファイルのFLAG列を他のソフトウェアに互換性のあるものに変更する。

-R: RG (read groups) の情報を付与する。複数のサンプル情報が混在している場合に有用。GATKでBAMファイルを扱うにはplatform (PL) および sample (SM)が必須。

PLの例 : 454, LS454, Illumina, Solid, ABI_Solid

マッピング

SAMファイルをBAMファイルに変換。

```
$ samtools view -b mapping/ERR038793_mapped.sam > ¥  
mapping/ERR038793_mapped.bam
```

```
$ ls -lh mapping
```

```
total 211M  
-rw-rw-r-- 1 iu iu 41M 7月 13 17:14 ERR038793_mapped.bam  
-rw-rw-r-- 1 iu iu 171M 7月 13 17:13 ERR038793_mapped.sam
```

171MのSAMファイルが41Mのバイナリファイルに変換された。

マッピング

BAMファイルをソート、インデックス作成。

```
$ samtools sort mapping/ERR038793_mapped.bam ¥  
  -o  mapping/ERR038793_sorted.bam  
$ samtools index mapping/ERR038793_sorted.bam  
$ ls mapping
```

```
ERR038793_mapped.bam  ERR038793_sorted.bam  
ERR038793_mapped.sam  ERR038793_sorted.bam.bai
```

BAMファイルを高速に扱うためのインデックスファイルを作成。

マッピング

マッピングの結果を確認する。

```
$ samtools idxstats mapping/ERR038793_sorted.bam
```

chrI	230218	8958	0
chrII	813184	34924	0
chrIII	316620	15039	0
chrIV	1531933	64370	0
chrIX	439888	19710	0
chrM	85779	22048	0

:

1列目： コンティグ名（fastaファイルのヘッダ）

2列目： コンティグの長さ

3列目： マッピングされたリード数

4列目： マッピングされなかったリード数

3列目の総和  マッピングされたリードの総数

4列目の総和  マッピングされなかったリードの総数

マッピング

マッピングの結果を計算、確認する。②

```
$ wc -l trimmed_data/paired_output_ERR038793_1.fastq | ¥  
  awk '{print $1/4;}'
```

```
597105
```

fastqファイルは4行 1 リードなので、fastqファイルの行数を4で割った数がリード数です。

```
$ samtools idxstats mapping/ERR038793_sorted.bam > multi.txt  
$ awk '{sum += $3} END {print sum}' multi.txt
```

```
576020
```

```
$ awk '{sum2 += $4} END {print sum2}' multi.txt
```

```
22086
```

マッピング

マッピングの結果を振り返る。

全リード数: 597105

マッピングされたリード数: 576020

マッピングされなかったリード数: 22086

計598106

全リード数

$\neq$

マッピングされたリード数

マッピングされなかったリード数

1001リード分はマルチヒットによるもの

マッピング

マルチヒットしたリードを除き、ユニークリードのみにする。

```
$ samtools view -b -F 256 mapping/ERR038793_sorted.bam > ¥  
mapping/ERR038793_unique.bam
```

- view : sam/bamを扱うサブコマンド
- -b : 出力をBAMファイルにする
- -F : 指定されたフラグが付与されたリードを除外する

マッピング状況を確認する。

```
$ samtools index mapping/ERR038793_unique.bam  
$ samtools idxstats mapping/ERR038793_unique.bam > ¥  
unique.txt
```

- index : BAMファイルのインデックスファイルを作成する
- idxstats : インデックスファイルのステータスを表示する

マッピング

マッピングの結果を再計算する。

```
$ awk '{sum += $3} END {print sum}' unique.txt
```

```
575019
```

```
$ awk '{sum2 += $4} END {print sum2}' unique.txt
```

```
22086
```

全リード数: 597105

マッピングされたリード数: 575019

マッピングされなかったリード数: 22086

} 計597105

変異検出

GATK HaplotypeCallerによる変異検出

```
$ mkdir variant_call
$ java -jar /usr/local/bin/GenomeAnalysisTK.jar ¥
  -R /home/iu/genome/sacCer3/genome.fa -T HaplotypeCaller ¥
  -I mapping/ERR038793_sorted.bam ¥
  -variant_index_type LINEAR -variant_index_parameter 128000 ¥
  -o variant_call/ERR038793.vcf
$ ls variant_call
```

```
ERR038793.vcf  ERR038793.vcf.idx
```

VCF (Variant Call Format)が作成されました。

変異検出

GATK HaplotypeCallerで設定したオプション

- -R: リファレンスゲノムの場所。
- -T: 使用するアルゴリズム。
- -I: 入力データ。
- -variant_index_type: LINEARで等間隔のインデックスを作成する。
- -variant_index_parameter: インデックスのbin幅。
- -o: 出力ファイル。

変異検出

VCFファイルの確認

```
$ less variant_call/ERR038793.vcf
```

```
                                :  
#CHROM      POS  ID   REF  ALT  QUAL    ...  ERR038793  
chrI        111  .    C    T    191.77  ...  0/1:3,6:9:90:220,0,90  
chrI        136  .    G    A    342.77  ...  1/1:0,9:9:29:371,29,0  
                                :
```

CHROM : 染色体番号

POS : 変異箇所の 1 塩基目の位置

ID : ID情報（情報がないので「.」と記載。）

REF : リファレンスゲノムの塩基配列

ALT : 変異のある塩基配列

QUAL : phred-scaleによるクオリティ値

FILTER : フィルタリング条件（情報がないので「.」と記載。）

INFO : 変異情報

変異検出

VCFファイルの確認 FORMAT

GT	AD	DP	GQ	PL
genotype	allelic depth	read depth	genotype quality	phred-scaled genotype likelihoods

ERR038793 (ファイル名)

C/Tのヘテロ ——— 0/1:3,6:9:90:220,0,90 ——— C/Tの最尤推定値が最も高い
10[^](-PL/10)

REFのカバレッジは3
ALTのカバレッジは6 リード数は計9つ 99.9999999%の信頼性

#CHROM	POS	ID	REF	ALT	...
chrI	111	.	C	T	...

※ ひとつ目のSNPを例に。

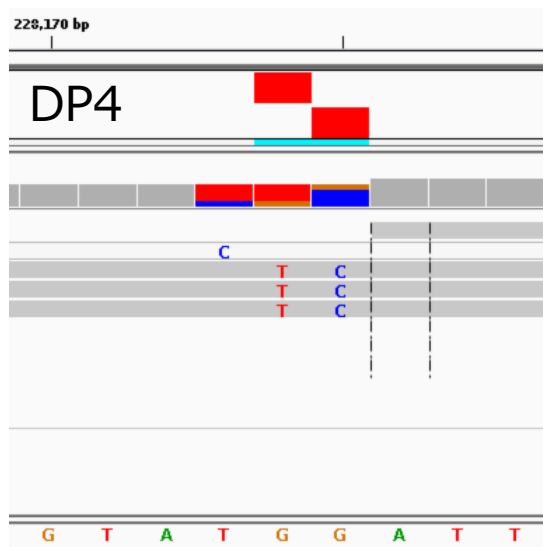
変異検出

検出したSNV、INDELの数を確認

```
$ grep -c -v '^#' variant_call/ERR038793.vcf
```

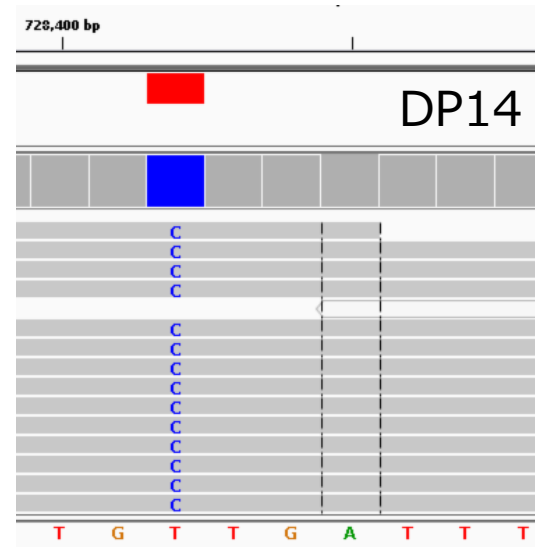
57869

57869個の変異が検出されましたが、この中にはカバレッジが低く、信頼性が十分に確保できない変異が存在してます。



信頼度

<



変異検出

Low coverageなSNVのカウント

```
$ awk '{print $10;}' variant_call/ERR038793.vcf | ¥ ...①  
  grep '0/1' | ¥ ...②  
  cut -d ':' -f 3 | ¥ ...③  
  awk '{if($1 < 10){print $1;}}' | ¥ ...④  
  wc -l ...⑤
```

667

カバレッジが10未満の信頼性の低い変異が667個存在しています。

- ①: SAMPLE列の抜きだし。
- ②: SNVのみにフォーカス。
- ③: SAMPLE列の「:」区切り3つめの要素のDP (coverage) を抜き出す。
- ④: DPが10未満のもののみ出力する。
- ⑤: 出力された行数を数える。

変異検出

変異のフィルタリング（FILTER列の活用）

```
$ java -jar /usr/local/bin/GenomeAnalysisTK.jar ¥  
-R /home/iu/genome/sacCer3/genome.fa -T VariantFiltration ¥  
-V variant_call/ERR038793.vcf ¥  
-o variant_call/ERR038793_fil.vcf ¥  
--clusterWindowSize 10 ¥  
--filterExpression 'DP < 10' ¥  
--filterName 'LowCoverage'
```

-R: リファレンスゲノムの場所

-V: 入力VCFファイル。

-o: 出力ファイル。

--filterExpression : フィルタリング条件。

--filterName : フィルター名。

変異検出

変異のフィルタリング（FILTER列の活用）

```
$ less variant_call/ERR038793_fil.vcf
```

```
          :  
#CHROM      POS  ID   REF  ALT  QUAL    FILTER      ...  
chrI        111  .    C    T    191.77  LowCoverage  ...  
chrI        136  .    G    A    342.77  LowCoverage  ...  
          :
```

カバレッジが10以下のSNPを消すわけではなく、“LowCoverage”というダグを付くことで、後ほどフィルタリングできるようになっています。

アノテーション

snpEffを用いたアノテーション方法

```
$ mkdir annotation
$ cd annotation
$ java -jar /usr/local/bin/snpEff.jar eff ¥
  -c /usr/local/bin/snpEff.config -i vcf -o vcf ¥
  R64-1-1.82 ../variant_call/ERR038793_fil.vcf 1> ¥
  ERR038793_eff.vcf
$ less ERR038793_eff.vcf
```

eff: 出力フォーマットの指定。

-c: コンフィグファイルの場所。様々なデータベースの情報が記載されている。

-i, -v: 入出力ファイルフォーマット。

R64-1-1.82: Scerevisiaeのデータベース。SacCer3 に対応。

アノテーション

snpEffを用いたアノテーション方法

						:		
#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	...
chrI	111	.	C	T	191.77	LowCoverage		...
chrI	136	.	G	A	342.77	LowCoverage		...
						:		

遺伝子名やコーディング情報、翻訳後のタンパク質に与えるインパクト等の情報が付与される。



アノテーション

snpEffを用いたアノテーション、エラーの回避

```
$ grep 'chrM' ERR038793_eff.vcf
```

```
          :  
          |||ERROR_CHROMOSOME_NOT_FOUND ...  
          :
```

今回作成したVCFファイルではミトコンドリアDNAを「chrM」と記述しています。しかしながら、今回用いたsnpEffのデータベース「R64-1-1.82」ではミトコンドリアのDNA情報が「chrMito」と記載されているために正しくマッチングが行われずエラーが起きています。

アノテーション

snpEffを用いたアノテーション、エラーの回避

```
$ sed -e 's/chrM/chrMito/g' ¥  
  ../variant_call/ERR038793_fil.vcf > ¥  
  ../variant_call/ERR038793_fil2.vcf  
$ java -jar /usr/local/bin/snpEff.jar eff ¥  
  -c /usr/local/bin/snpEff.config -i vcf R64-1-1.82 ¥  
  -o vcf ../variant_call/ERR038793_fil2.vcf 1> ¥  
  ERR038793_eff2.vcf  
$ grep 'chrM' ERR038793_eff2.vcf
```

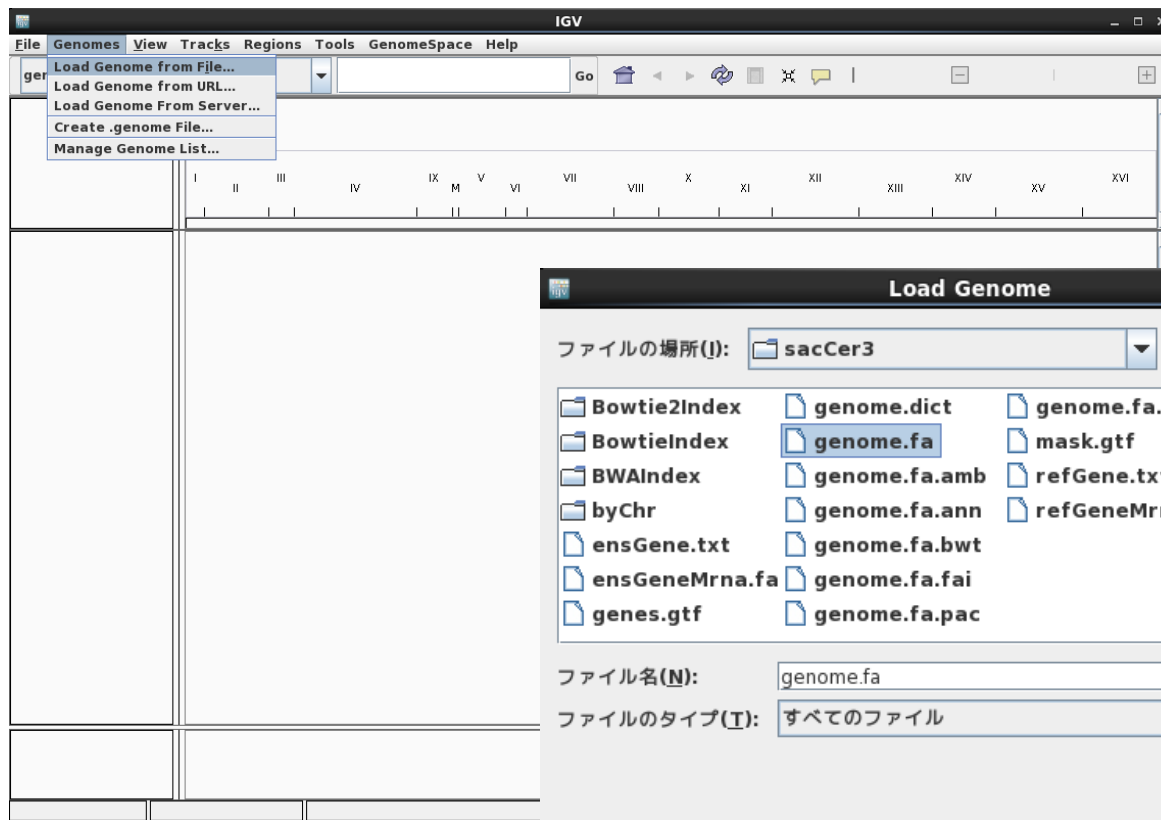
ミトコンドリアDNAもアノテーションされました。

sedコマンド: 文字列の全置換から行単位の抽出・削除まで行えるテキスト加工コマンド。

解析結果の可視化

Integrative Genomics Viewer (IGV)を用いた解析結果の確認 ①

```
$ igv.sh
```

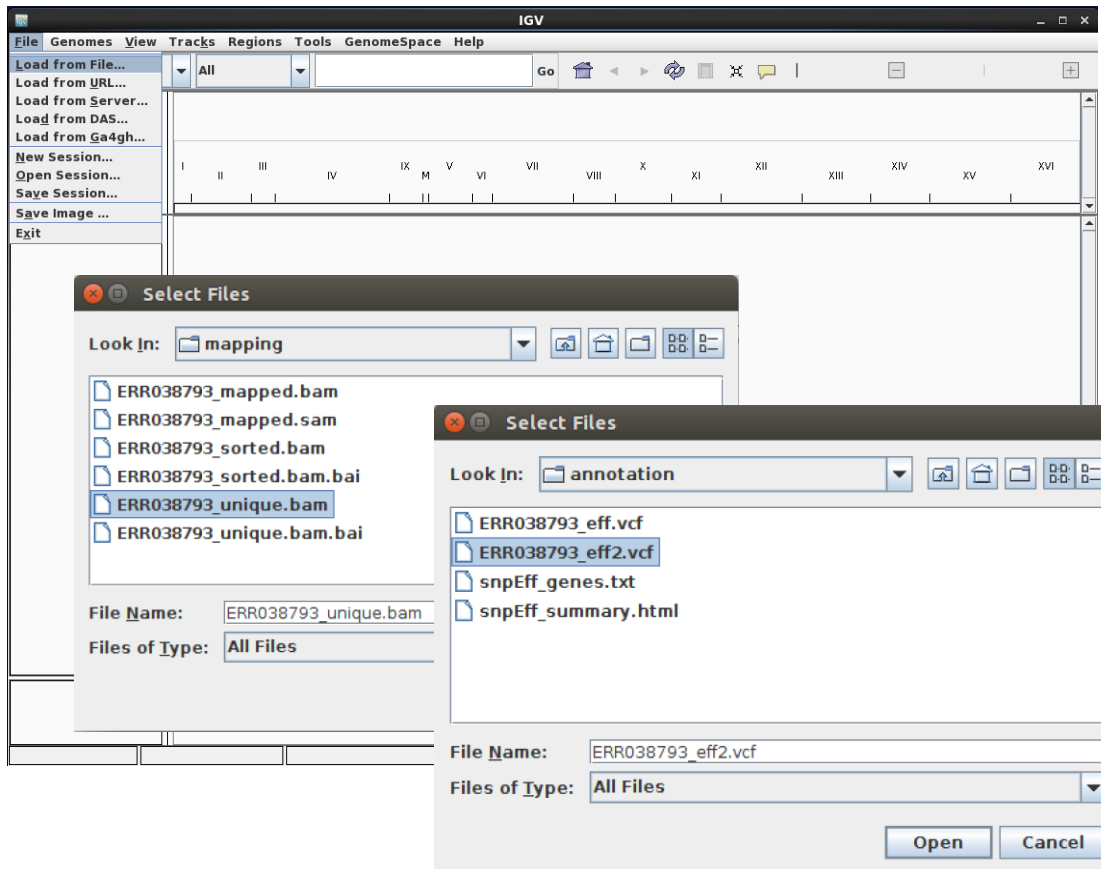


IGVを起動し、
Genomesタブから
Load Genomes from
File...を選択。

/home/iu/
genome/
sacCer3の下に
あるgenome.fa
を選択し開く。

解析結果の可視化

Integrative Genomics Viewer (IGV)を用いた解析結果の確認 ②

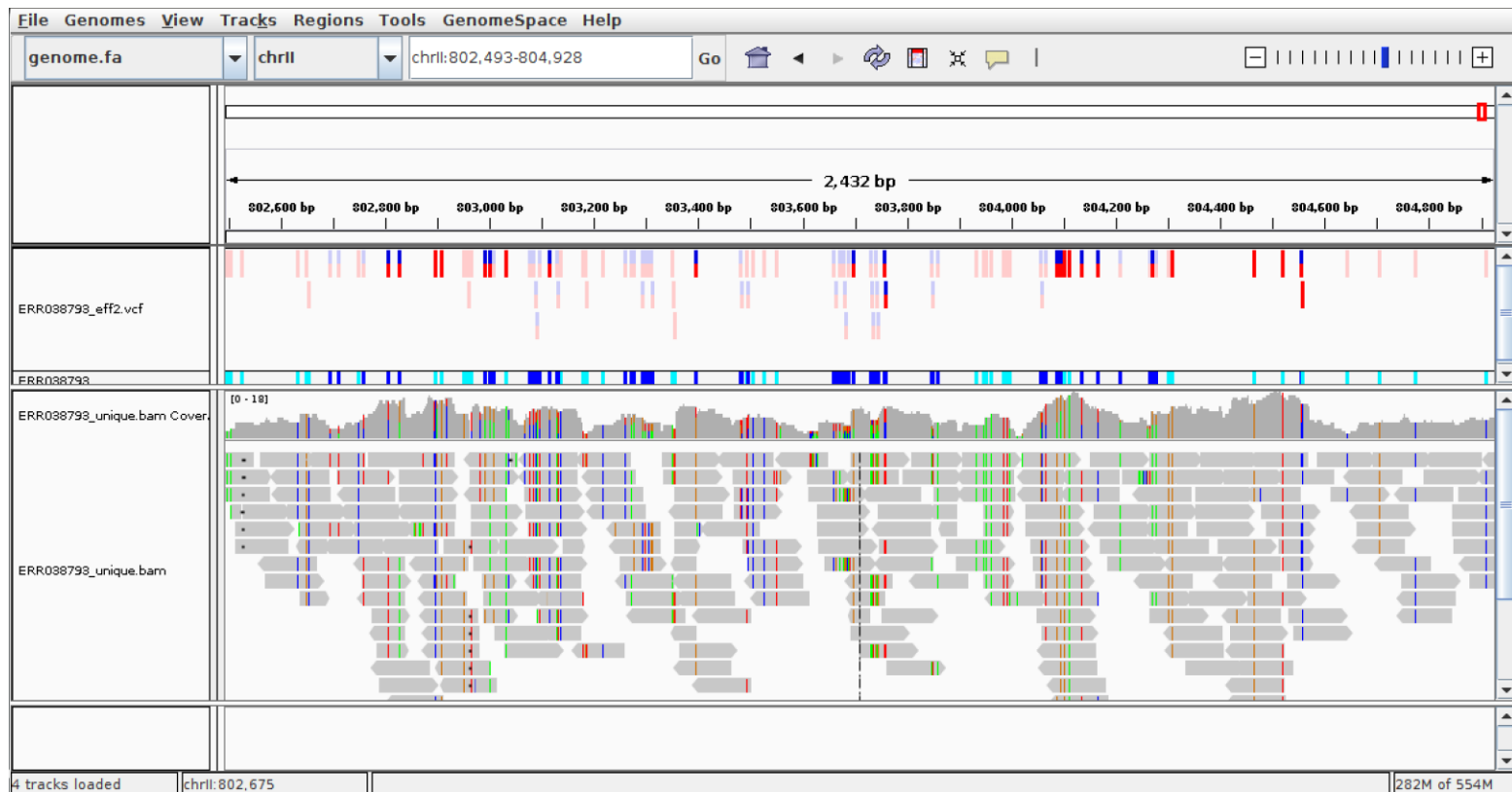


Fileタブから
Load from File...を選択。

ERR038793_unique.bam
ERR038793_eff2.vcf
の2ファイルを順次読み込む。

解析結果の可視化

Integrative Genomics Viewer (IGV)を用いた解析結果の確認 ③



サーチウィンドウにchrII:802,493-804,928と入力。
多くの変異が視認できる。

解析結果の可視化

Integrative Genomics Viewer (IGV)を用いた解析結果の確認 ④



chrII:804,080-804,116と入力し、拡大表示する。
この位置にカーソルを合わせるとgenotypeの概要を確認できる。

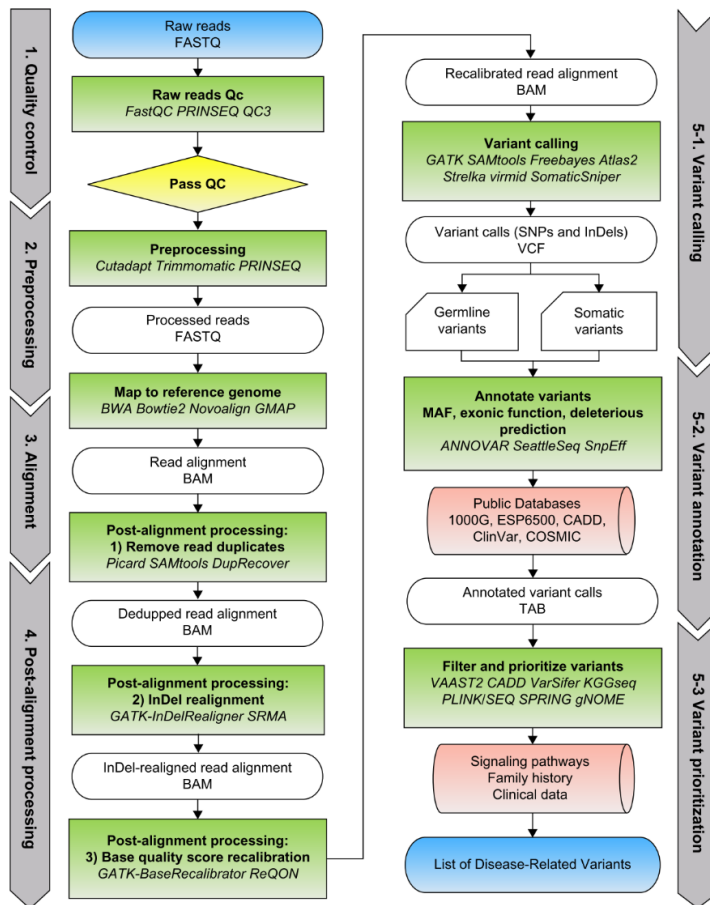
まとめ

本で行った解析のおさらい。



最後に

より高精度な解析を行うには。



本日行ってきたのはあくまで解析方法の一例です。ツールの選択に「正解」はありません。自身のデータに適したツールを選択し、より良い解析手順を確立していきましょう。

Riyue Bao et al., *CANCER INFORMATICS*, 2014, **13**(s2), 67–82