平成28年度NGSハンズオン講習会
NGS解析基礎

2016年7月25日

amelieff

最近のシーケンサ

illumina®

MiniSeq
MiSeq
NextSeq
HiSeq
HiSeqX

IonPGM
IonProton

ThermoFisher
SCIENTIFIC

PACIFIC
BIOSCIENCES™

MinION

Oxford
NANOPORE
Technologies

Sequel
PacBio

10x
GENOMICS™

目次

- NGSデータ解析で主に使用するファイル形式
- データの可視化
- データのクオリティチェックとクリーニング
- NGSデータのマッピング
- 【実践！】新しいソフトウェアの導入

■ **資料の見方**

```
$ pwd
```

```
/home/user/analysis/NGShandson
```

実際に入力する**コマンド**を、**紺枠の四角**の中に示します。
コマンドの**結果**を、紺枠・**グレー地の四角**の中に示します。

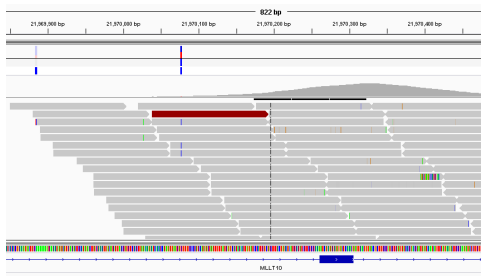
基本的なNGS解析の流れ

シーケンシング

リード配列情報

ゲノム参照配列

マッピング



配列のアライメント情報

Reseq
変異検出

RNA-seq
発現量定量

ChIP-seq
ピーク検出

変異情報

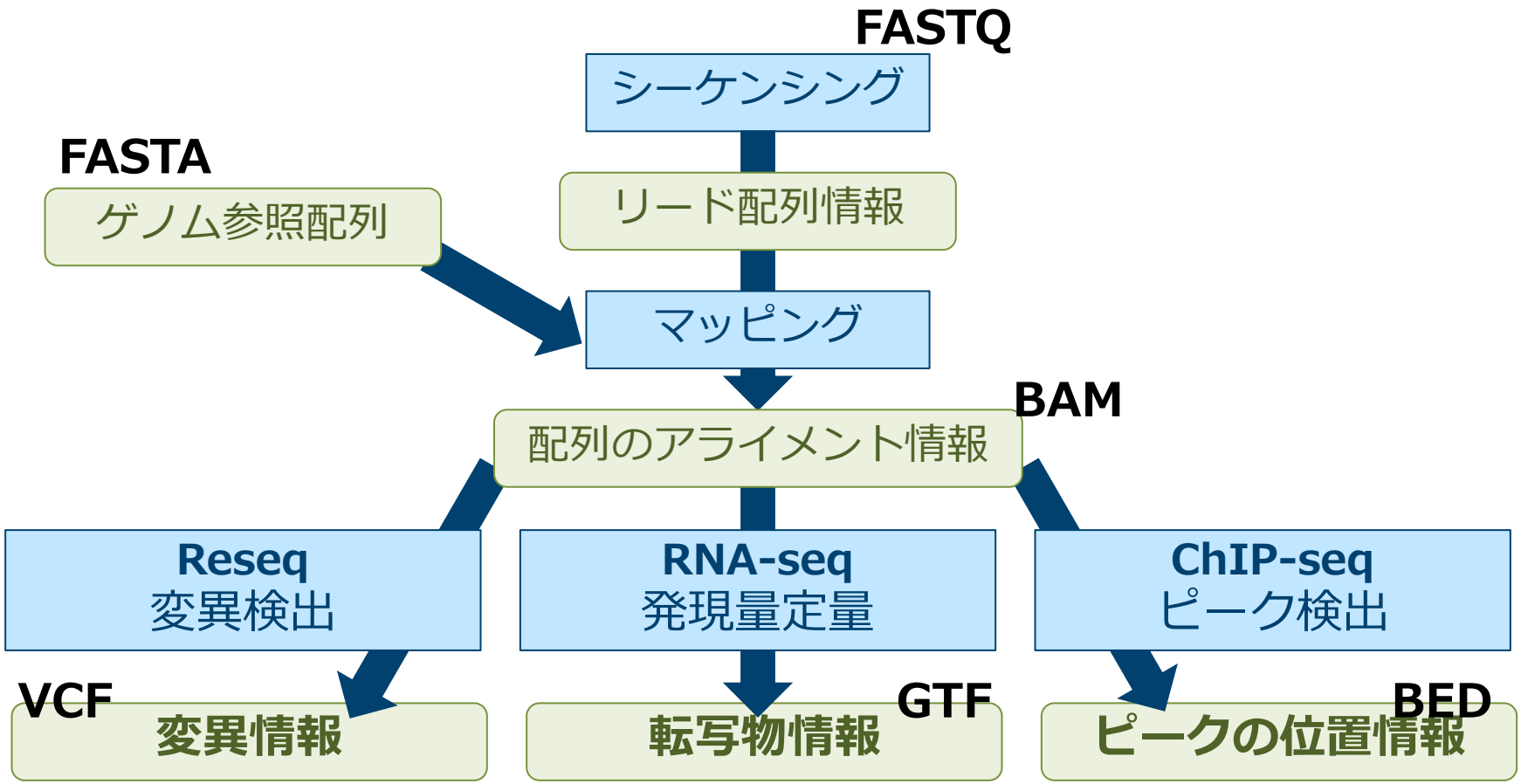
転写物情報

ピークの位置情報

NGSデータ解析で主に使用するファイル形式

拡張子	記載されている情報
FASTA	塩基配列やアミノ酸配列の情報
FASTQ	シーケンサが出力するリード情報
BAM / SAM	リードをゲノムにマッピングしたアライメント情報
VCF	変異情報
BED	ゲノム上の領域の情報
GFF/GTF	ゲノム上のfeature (遺伝子、転写産物等) の情報

NGSデータ解析で主に使用するファイル形式



NGSデータ解析で主に使用するファイル形式

- **FASTAファイル**
  - 塩基やアミノ酸などの**配列の情報**。ここではリファレンスゲノムの塩基配列のfastaについて説明する。
  - **ヘッダ**：「>」から始まる。
  - **データ**：塩基配列。60～80文字で折り返す。
  - 拡張子が統一されておらず、.fa、.fasta、.fna、.fasなどが使われていることがある。

【例】

```
$ less sacCer_chrI.fa
```

```
>I
CCACACCACACCCACACACCCACACACCACACCACACACCACACCACACCCACACACACACATCCTAACA
CTACCCTAACACAGCCCTAATCTAACCCTGGCCAACCTGTCTCTCAACTTACCCTCCATTACCCTGCCTC
CACTCGTTACCCTGTCCCATTCAACCATACCACTCCGAACCACCATCCATCCCTCTACTTACTACCACTC
                                        :
```

NGSデータ解析で主に使用するファイル形式

- **FASTQ**
  - シーケンサーが読んだ**シーケンスの情報**
  - 1リードの情報を4行で表したファイル
  - 拡張子は fastq または fq

	必須の情報	**オプション**
1行	@から始まる配列ID	付加情報
2行	リードの塩基配列	
3行	+	配列ID、または1行目と同じ
4行	各塩基のクオリティ	

NGSデータ解析で主に使用するファイル形式

- **FASTQ**
  - ファイルサイズが大きいため、圧縮されていることが多い。
  - GZ　…よく使われる圧縮方法。シーケンサから出力されることが多い。
  - BZ2　…圧縮・展開に時間がかかるが、高効率な圧縮方法。
  - SRA　…配列ファイルに特化した圧縮方法。SRA-toolkitで扱う。
  - ZIP　…一般的によく使われる圧縮方法。

Tips
ファイルの圧縮・展開コマンドを覚えておくと便利（→P.60、P.70）。

NGSデータ解析で主に使用するファイル形式

■ **FASTQ**

【例】

```
$ less SRR504515_R1.fastq
1 @SRR504515.1 HWI-ST423_0087:2:1:1183:2098 length=101
2 AAANGACGGTTGGTCCTTAAAATTCCATGGATGTAGATCTTATCCCCACACCCAGACTCTAG
3 +SRR504515.1 HWI-ST423_0087:2:1:1183:2098 length=101
4 @>?#>ABAA>FFHEHHEHDHHGHAHFGFDGGFGEFGE=F<D@BCA5DCB=A:@BB#######
1 @SRR504515.2 HWI-ST423_0087:2:1:1192:2129 length=101
2 TGGNTAGCTGAGCTTGGTGCTGTAGACTAAAGCACATTCCTTCATGGCAAATCACTTACAGT
3 +SRR504515.2 HWI-ST423_0087:2:1:1192:2129 length=101
4 >>=#7<<88>?CDCDBC6ADDCBBDC9DD4C@+@0:7=97*@@?####################
:                                   :
```

NGSデータ解析で主に使用するファイル形式

- **FASTQ**
  - FASTQのクオリティは「記号のASCIIコード - 33」と対応する

【例】クオリティ値：? → 実際のクオリティ：63 - 33 = 30

ASCIIコード表

33:!	34:"	35:#	36:$	37:%	38:&	39:'	40:(
41:)	42:*	43:+	44:,	45:-	46:.	47:/	48:0
49:1	50:2	51:3	52:4	53:5	54:6	55:7	56:8
57:9	58::	59:;	60:<	61:=	62:>	63:?	64:@
65:A	66:B	67:C	68:D	69:E	70:F	71:G	72:H
73:I	74:J	75:K	76:L	77:M	78:N	79:O	80:P
81:Q	82:R	83:S	84:T	85:U	86:V	87:W	88:X
89:Y	90:Z	91:[	92:\	93:]	94:^	95:_	96:`
97:a	98:b	99:c	100:d	101:e	102:f	103:g	104:h
105:i	106:j	107:k	108:l	109:m	110:n	111:o	112:p
113:q	114:r	115:s	116:t	117:u	118:v	119:w	120:x
121:y	122:z	123:{	124:|	125:}	126:~		

NGSデータ解析で主に使用するファイル形式

- **FASTQ**
  - P = 10-Q/10
  - Q = -10 log10(P)

Q score =**30** のとき
エラー率 = **0.00100**

ASCII_BASE=33 Illumina, Ion Torrent, PacBio and Sanger

Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII
0	1.00000	33 !	11	0.07943	44 ,	22	0.00631	55 7	33	0.00050	66 B
1	0.79433	34 "	12	0.06310	45 -	23	0.00501	56 8	34	0.00040	67 C
2	0.63096	35 #	13	0.05012	46 .	24	0.00398	57 9	35	0.00032	68 D
3	0.50119	36 $	14	0.03981	47 /	25	0.00316	58 :	36	0.00025	69 E
4	0.39811	37 %	15	0.03162	48 0	26	0.00251	59 ;	37	0.00020	70 F
5	0.31623	38 &	16	0.02512	49 1	27	0.00200	60 <	38	0.00016	71 G
6	0.25119	39 '	17	0.01995	50 2	28	0.00158	61 =	39	0.00013	72 H
7	0.19953	40 (	18	0.01585	51 3	29	0.00126	62 >	40	0.00010	73 I
8	0.15849	41)	19	0.01259	52 4	30	0.00100	63 ?	41	0.00008	74 J
9	0.12589	42 *	20	0.01000	53 5	31	0.00079	64 @	42	0.00006	75 K
10	0.10000	43 +	21	0.00794	54 6	32	0.00063	65 A			

NGSデータ解析で主に使用するファイル形式

- **SAM / BAM**
  - リードをゲノムにマッピングした**アライメント情報**。

SAM	テキストデータ
BAM	SAMを圧縮したバイナリデータ

  - 相互変換には主に **SAMtools** というソフトを使用する。

- samからbam　（-b: bamとして出力）

```
$ samtools view -b sam > bam
```

- bamからsam　（-h: ヘッダ付きで出力）

```
$ samtools view -h bam > sam
```

NGSデータ解析で主に使用するファイル形式

- **SAMファイルの中身**
  - ヘッダ行：@から始まる。
  - データ行：タブ区切りで、1行に1リードの情報が記載されている。

【例】

```
@HD     VN:1.0  GO:none SO:coordinate
@SQ     SN:chr1 LN:249250621    UR:file:/home/genome/hg19/genome.fa
@SQ     SN:chr2 LN:243199373    UR:file:/home/genome/hg19/genome.fa
@SQ     SN:chr3 LN:198022430    UR:file:/home/genome/hg19/genome.fa
@SQ     SN:chr4 LN:191154276    UR:file:/home/genome/hg19/genome.fa
@SQ     SN:chr5 LN:180915260    UR:file:/home/genome/hg19/genome.fa
@SQ     SN:chr6 LN:171115067    UR:file:/home/genome/hg19/genome.fa
@SQ     SN:chr7 LN:159138663    UR:file:/home/genome/hg19/genome.fa
@SQ     SN:chr8 LN:146364022    UR:file:/home/genome/hg19/genome.fa
                                :
SRR504515.1962973       129     chr1    10146   10      7S56M5D38M      chr11   134946
SRR504515.36684253      129     chr1    10149   0       27M1I36M        chr15   917952
SRR504515.12321503      163     chr1    11585   0       101M    =       11810   292
SRR504515.48945773      163     chr1    11714   0       101M    =       11940   276
SRR504515.45196577      99      chr1    11806   0       50M     =       11991   286
SRR504515.12321503      83      chr1    11810   0       67M     =       11585   -292
```

ヘッダ行

データ行

NGSデータ解析で主に使用するファイル形式

- **SAMファイルの中身**
  - データ行：最初の11列は必須。

列	項目	意味	例
1	QNAME	リード名	ERR038793.1
2	FLAG	フラグ	113
3	RNAME	染色体名	XII
4	POS	リードのスタートポジション	1065143
5	MAPQ	マッピングクオリティ	4
6	CIGAR	CIGAR (アライメントステータス)	12M4I84M
:	:	:	:

NGSデータ解析で主に使用するファイル形式

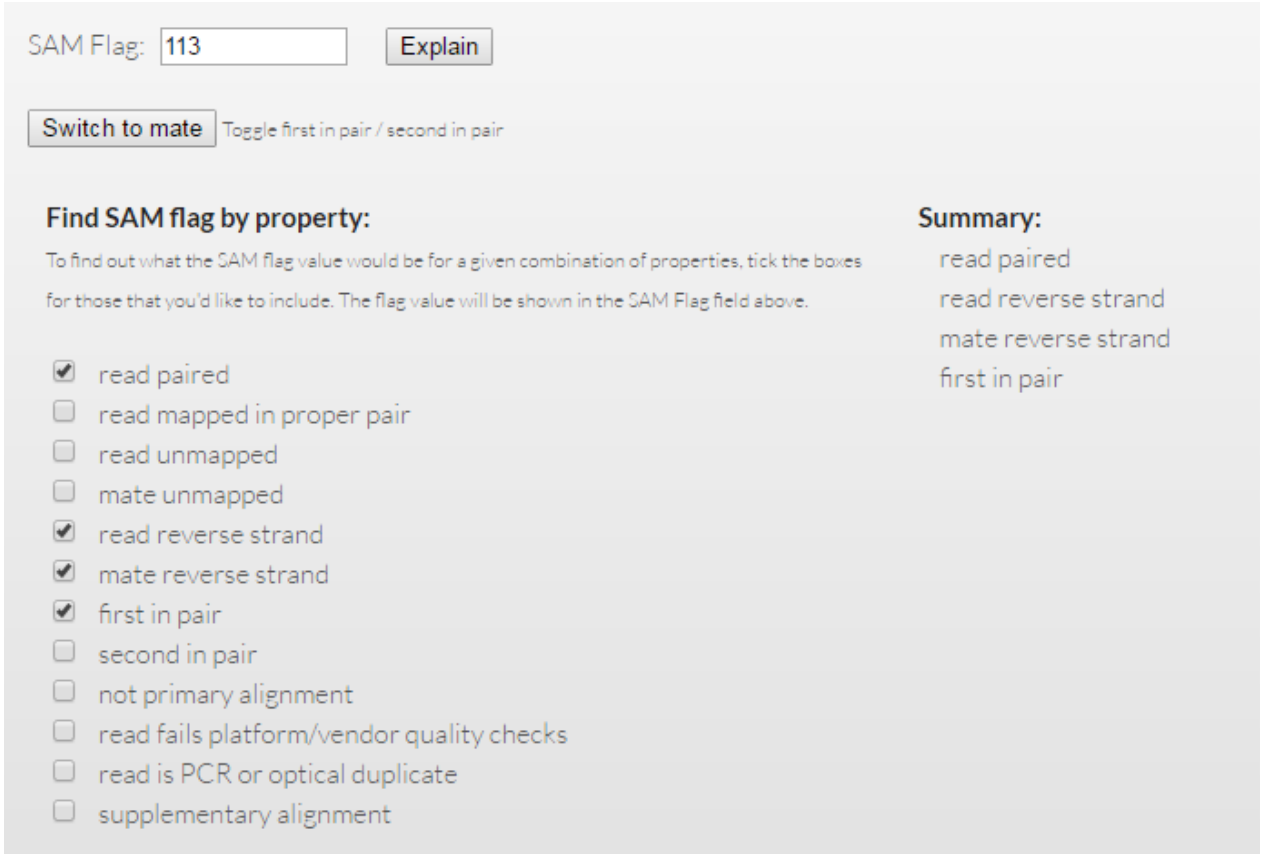
- **SAMファイルの中身**
  - データ行：最初の11列は必須。

列	項目	意味	例
:	:	:	:
7	RNEXT	ペアリードがある染色体名	I
8	PNEXT	ペアリードのスタート位置	150
9	TLEN	ペア間の距離＋各リード長	0
10	SEQ	リード配列	AGGGTGTGGTGTGTGGGTATATCTATGTCACCTTATTGCATGCTGGATGGTGTTAGACAAGGCCGTAGGGACATATAGCATCTAGGAAGTAACCTTGTCC
11	QUAL	リードクオリティ	CD;?C@FEFEFFFFFDC8=DA=?>>.EEE=BEEEBEE:EEE:?@FFBF?F@FFCF?BC><EEEA:DDDBBDEBEEEDF@FEEEEEEEEFFD>B@DBDD/D
:	:	:	:

NGSデータ解析で主に使用するファイル形式

■ **SAMファイルの中身**

フラグ自動計算：https://broadinstitute.github.io/picard/explain-flags.html



NGSデータ解析で主に使用するファイル形式

- **VCFファイル**
  - ゲノム上の**変異の情報**。
  - **ヘッダ行**：「#」で始まる。

【例】

```
##fileformat=VCFv4.1
##FILTER=<ID=HARD_TO_VALIDATE,Description="MQ0 >= 4 && ((MQ0 / (1.0 * DP))
##FILTER=<ID=HRUN,Description="HRun > 5">
##FILTER=<ID=LowCoverage,Description="DP < 10">
##FILTER=<ID=LowQD,Description="QD < 1.5">
##FILTER=<ID=LowQual,Description="QUAL >= 30.0 && QUAL < 50.0">
##FILTER=<ID=SnpCluster,Description="SNPs found in clusters">
##FILTER=<ID=StrandBias,Description="SB > -0.1">
##FILTER=<ID=VeryLowQual,Description="QUAL < 30.0">
##FORMAT=<ID=AD,Number=.,Type=Integer,Description="Allelic depths for the ref and alt all
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Approximate read depth (reads with MQ=
##FORMAT=<ID=GQ,Number=1,Type=Float,Description="Genotype Quality">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="Normalized, Phred-scaled likelihoods f
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes, for each ALT
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency, for each ALT allele, in
                                        :
```

ヘッダ行

NGSデータ解析で主に使用するファイル形式

- **VCFファイル**
  - ゲノム上の**変異の情報**。
  - **データ行**：1行に1変異の情報が、タブ区切りで記載されている。

【例】

```
                                                    :
chr1    14522   .       G       A       69.94   HARD_TO_VALIDATE;Str
chr1    14542   .       A       G       82.09   HARD_TO_VALIDATE;StrandBias
chr1    63516   .       A       G       51.82   LowCoverage;StrandBias
chr1    753269  .       C       G       31.86   HARD_TO_VALIDATE;LowQual;StrandB
ias
chr1    753405  .       C       A       93.71   LowCoverage;StrandBias
chr1    808922  .       G       A       689.48  HARD_TO_VALIDATE
chr1    808928  .       C       T       731.06  HARD_TO_VALIDATE
chr1    887801  .       A       G       37.30   LowCoverage;LowQual;StrandBias
```

データ行

NGSデータ解析で主に使用するファイル形式

- **VCFファイル**
  - ゲノム上の**変異の情報**。
  - **データ行**：1行に1変異の情報が、タブ区切りで記載されている。

列	項目	説明	例
1	#CHROM	変異がある染色体名	I
2	POS	変異のポジション(最初のポジションは**1**)	111
3	ID	rsID、COSMIC IDなど	rs987324
4	REF	リファレンスゲノムのアリル	C
5	ALT	変異のアリル	T
6	QUAL	変異のクオリティ	105.93
7	FILTER	変異検出ソフトが変異につける変異のクオリティ	LowCoverage
:	:	:	:

NGSデータ解析で主に使用するファイル形式

- **VCFファイル**
  - ゲノム上の**変異の情報**。
  - **データ行**：1行に1変異の情報が、タブ区切りで記載されている。

列	項目	説明	例
:	:	:	:
8	INFO	検出ソフトやアノテーションソフトが、「;」区切りで変異につける変異の情報やアノテーション。記述は自由	AC=1;AF=0.50;AN=2
9	FORMAT	以降の列に「:」区切りで記載される、サンプルごとの変異情報の書式説明	GT:AD:DP:GQ:PL
:	サンプル列	変異の情報。 書式はFORMATに従う	0/1:5,4:9:99:136,0,173

NGSデータ解析で主に使用するファイル形式

- **BEDファイル**
  - ゲノム上の**領域の情報**。
  - ChIP-seqで検出されたピークを表したり、exome-seq、target-seqなどで解析範囲を指定するために用いられる

列	項目	説明	例
1	chrom	染色体	XII
2	chromStart	開始ポジション (最初のポジションは**0**)	1065142
3	chromEnd	終了ポジション	1065238

※最初の3列はすべてのBEDに共通して必須だが、以降の列は必要ではなく、内容も自由度が高い

【例】

```
chr4    103790204    103790390
chr4    103997665    103997765
chr4    106394637    106394799
chr4    110354820    110355016
chr4    111806132    111806324
chr4    113152875    113153158
chr4    114682396    114682607
chr4    114683705    114683825
chr4    120133405    120133592
chr4    120133809    120133943
chr4    120375605    120375791
chr4    120988065    120988280
chr4    121843277    121843492
chr4    122722370    122722670
chr4    123073343    123073530
chr4    123747867    123748087
```

NGSデータ解析で主に使用するファイル形式

- **GFF/GTFファイル**
  - ゲノム上の **feature の情報**。
  - 遺伝子や転写産物などの情報を記載するために使用する。RNA-seqでは、既知転写産物情報がマッピング精度向上のため使用されたり、発現している転写産物情報をGTF形式にすることがある。

【例】

```
chrI    sacCer3_ensGene start_codon     130799  130801  0.000000        +       .       gene_id "YAL012W"; transcript_id "YAL012W";
chrI    sacCer3_ensGene CDS     130799  131980  0.000000        +       0       gene_id "YAL012W"; transcript_id "YAL012W";
chrI    sacCer3_ensGene stop_codon      131981  131983  0.000000        +       .       gene_id "YAL012W"; transcript_id "YAL012W";
chrI    sacCer3_ensGene exon    130799  131983  0.000000        +       .       gene_id "YAL012W"; transcript_id "YAL012W";
chrI    sacCer3_ensGene start_codon     335     337     0.000000        +       .       gene_id "YAL069W"; transcript_id "YAL069W";
chrI    sacCer3_ensGene CDS     335     646     0.000000        +       0       gene_id "YAL069W"; transcript_id "YAL069W";
chrI    sacCer3_ensGene stop_codon      647     649     0.000000        +       .       gene_id "YAL069W"; transcript_id "YAL069W";
chrI    sacCer3_ensGene exon    335     649     0.000000        +       .       gene_id "YAL069W"; transcript_id "YAL069W";
chrI    sacCer3_ensGene start_codon     538     540     0.000000        +       .       gene_id "YAL068W-A"; transcript_id "YAL068W-A";
chrI    sacCer3_ensGene CDS     538     789     0.000000        +       0       gene_id "YAL068W-A"; transcript_id "YAL068W-A";
chrI    sacCer3_ensGene stop_codon      790     792     0.000000        +       .       gene_id "YAL068W-A"; transcript_id "YAL068W-A";
chrI    sacCer3_ensGene exon    538     792     0.000000        +       .       gene_id "YAL068W-A"; transcript_id "YAL068W-A";
chrI    sacCer3_ensGene stop_codon      1807    1809    0.000000        -       .       gene_id "YAL068C"; transcript_id "YAL068C";
chrI    sacCer3_ensGene CDS     1810    2169    0.000000        -       0       gene_id "YAL068C"; transcript_id "YAL068C";
chrI    sacCer3_ensGene start_codon     2167    2169    0.000000        -       .       gene_id "YAL068C"; transcript_id "YAL068C";
chrI    sacCer3_ensGene exon    1807    2169    0.000000        -       .       gene_id "YAL068C"; transcript_id "YAL068C";
chrI    sacCer3_ensGene start_codon     2480    2482    0.000000        +       .       gene_id "YAL067W-A"; transcript_id "YAL067W-A";
chrI    sacCer3_ensGene CDS     2480    2704    0.000000        +       0       gene_id "YAL067W-A"; transcript_id "YAL067W-A";
chrI    sacCer3_ensGene stop_codon      2705    2707    0.000000        +       .       gene_id "YAL067W-A"; transcript_id "YAL067W-A";
```

NGSデータ解析で主に使用するファイル形式

- **GFF/GTFファイル**
  - ゲノム上の **feature の情報**。
  - 遺伝子や転写産物などの情報を記載するために使用する。RNA-seqでは、既知転写産物情報がマッピング精度向上のため使用されたり、発現している転写産物情報をGTF形式にすることがある。

列	項目	説明	例
1	seqname	染色体名またはsccaffold名	I
2	source	Featureを検出したプログラム・プロジェクト名	sacCer3_ensGene, unknown
3	feature	Featureの種類	CDS, start_codon, exon
4	start	Featureの開始ポジション。(最初のポジションは**1**)	335
5	end	Featureの終了ポジション	646
:	:	:	:

あるfeatureについて、start codon、exon、CDSなど、複数行にわたって記載されることもある

NGSデータ解析で主に使用するファイル形式

- **GFF/GTFファイル**
  - ゲノム上の **feature の情報**。
  - 遺伝子や転写産物などの情報を記載するために使用する。RNA-seqでは、既知転写産物情報がマッピング精度向上のため使用されたり、発現している転写産物情報をGTF形式にすることがある。

列	項目	説明	例
:	:	:	:
6	score	0-1000まで、または「.」	105.93
7	strand	ストランド	+または-、 不明な場合は「.」
8	frame	Featureがexonのとき、最初の塩基のreading frameを表す0-2までの数字。Exon以外の場合は「.」	2
:	:	:	:

NGSデータ解析で主に使用するファイル形式

- **GFF/GTFファイル**
  - GTFとGFFの違い

➢ GFF

列	項目	説明	例
:	:	:	:
9	Group	Group名。同じグループに属する行は、すべて同じGroup名を持つ	Transcript YAL069W

➢ GTF

列	項目	説明	例
:	:	:	:
9	attribute	各featureに関する詳細を「;」区切りで記述	gene_id "YAL067W-A"; transcript_id "YAL067W-A";

データの可視化

はじめに

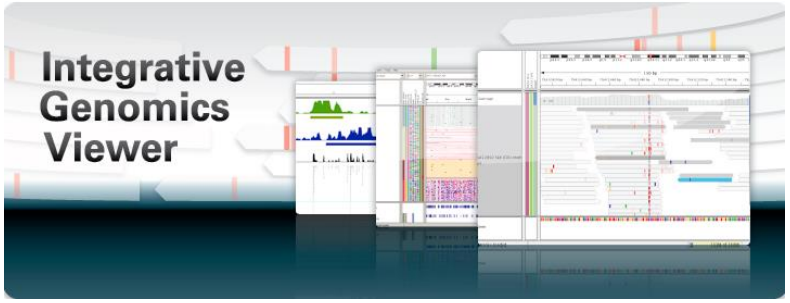
- NGS基礎解析ディレクトリに移動してください。

```
$ cd /home/iu/ngsbasics
$ ls
```

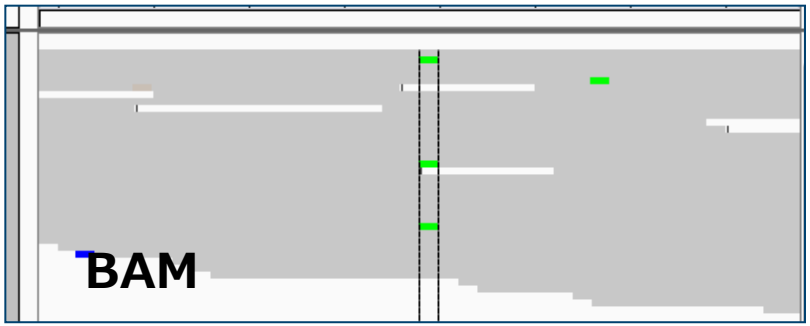
```
sacCer_chrI.fa
sacCer_chrI.gtf
SRR504515.bam
SRR504515.bed
SRR504515_R1.fastq
SRR504515_R2.fastq
SRR504515.vcf
Trimmomatic-0.36.zip
```

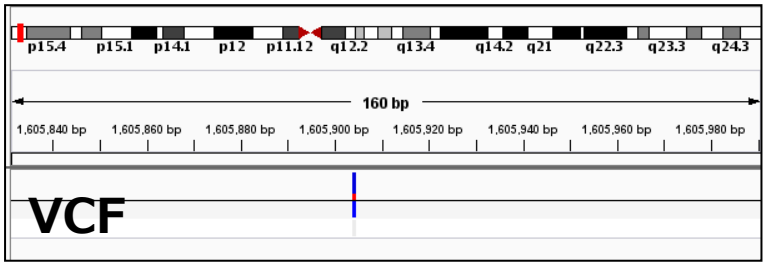
講義に使用するテストデータが置いてあります。

データの可視化



- Integrative Genomics Viewer（IGV）
  - 米 Broad Instituteが開発した
    ゲノムブラウザ
  - GUIで直感的な操作が行える
  - BAM、BED、VCFなどのファイル形式に対応
    （可視化できる形式一覧は
    http://www.broadinstitute.org/software/igv/FileFormats）
  - **Windows、MacOS、LinuxのいずれのOSでも動作する**
  - クローズドな環境で使用でき、セキュリティ上安全

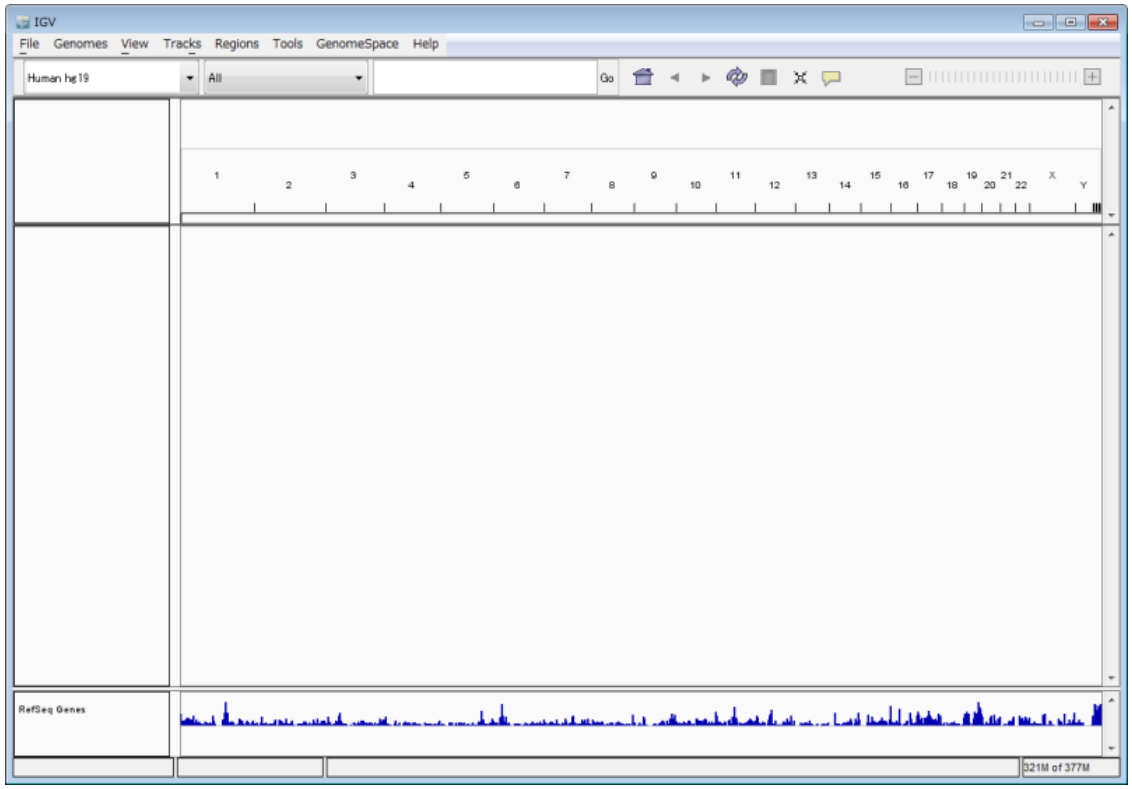




データの可視化

- IGVの起動

```
$ igv.sh
```



データの可視化

- インデックスの作成
  - サイズが大きなファイルを高速に扱うため、サイズの大きなインデックス（目次）ファイルが必要なことが多い

  - BAMファイルのインデックス
    - ファイル名は「***.bai、***.bam.bai」。
    - SAMtoolsで作成する。

  - VCFファイルのインデックス
    - ファイル名は「***.vcf.idx」
    - IGV（igvtools）で作成する。

データの可視化

■ BAMファイルのインデックス作成

1. BAMファイルを確認する。

```
$ ls
```

```
1K_ERR038793.bam
```

2. BAMファイルをソートする。（ソート済みの場合は不要）

```
$ samtools sort 1k_ERR038793.bam 1k_ERR038793_sort
$ls
```

```
1k_ERR038793.bam     1k_ERR038793_sort.bam
```

3. インデックスを作成する。

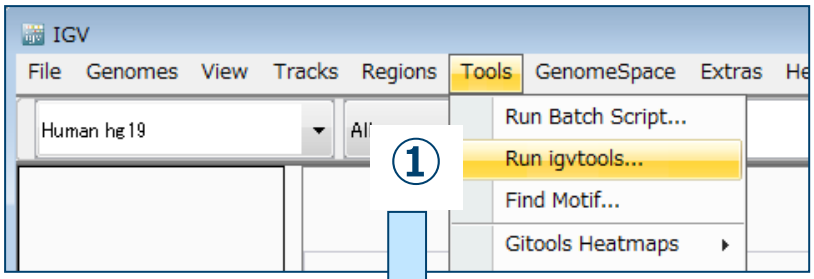
```
$ samtools index 1k_ERR038793_sort.bam
$ ls
```

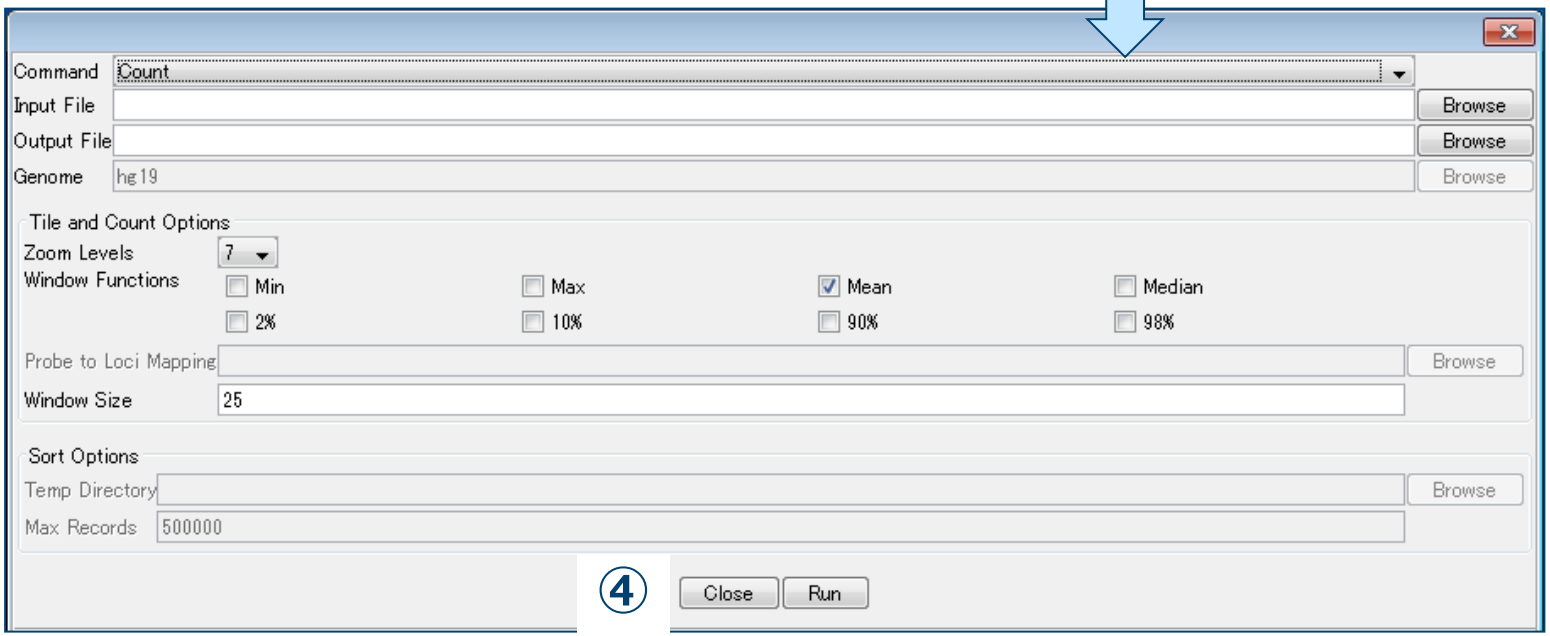
```
1k_ERR038793.bam     1k_ERR038793_sort.bam
```

データの可視化

■ VCFファイルのインデックス作成

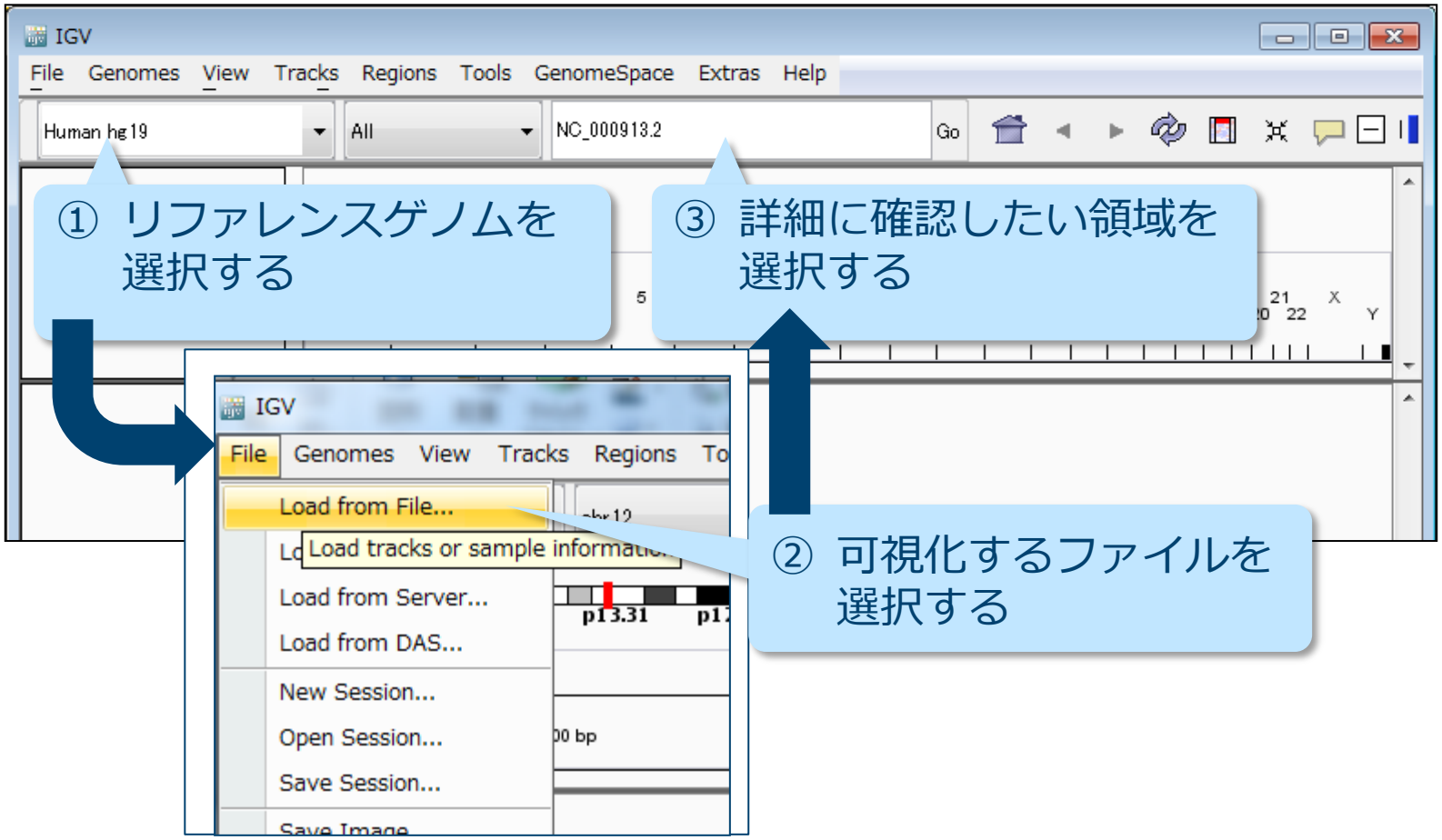
1. IGVからigvtoolsを起動する。
2. Commandを「index」に設定する。
3. Input Fileを選択する。
4. 「Run」ボタンを押して実行する。





データの可視化

- BAM/BED/VCF/GTFをIGVで可視化する



データのクオリティチェック

データのクオリティチェックとクリーニング

- NGSデータ解析において1番重要なことは

解析データのクオリティ

“Garbage in, garbage out”

データのクオリティが悪いと、どんなすばらしいインフォマティシャンが解析しても、**いい結果は出ない**。

データのクオリティチェックとクリーニング

- クオリティチェック
  - 低クオリティなデータは、多くの偽陽性やエラーの元となる。
    - アダプター配列の混入
    - 低クオリティ塩基・リードの混在
    - Poly-A/T tail
    - 他生物のDNAのコンタミ

- シーケンスリードのQC
- マッピング率の確認

- クオリティクリーニング
  - アダプター配列の除去
  - 低クオリティ塩基・リードの除去
  - Poly-A/T tailの除去

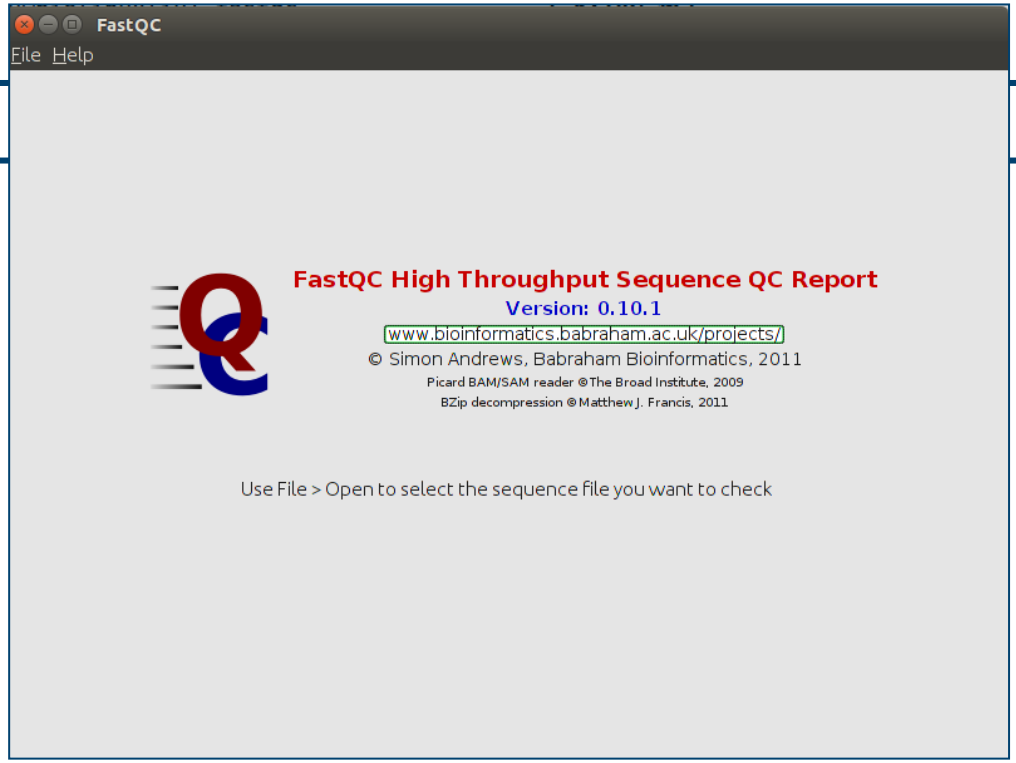
クリーニングのいずれか、または複数を実行できるソフトウェアを用途に応じて使用する

Fastx-toolkit	Cutadapt
tagcleaner	Prinseq
Trimmomatic	seqtk

データのクオリティチェックとクリーニング

- FastQC
  シーケンスリードのクオリティを確認するソフトウェア。FASTQまたはBAMを用いる。
  - GUIで操作する場合

```
$ fastqc
```



データのクオリティチェックとクリーニング

- FastQC
  FASTQまたはBAMのクオリティを確認するソフトウェア。
  - CUIで操作する場合

1. Usageの確認

```
$ 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
                                 :
```

データのクオリティチェックとクリーニング

- FastQC
  FASTQまたはBAMのクオリティを確認するソフトウェア。

1. FASTQファイルの確認

```
$ ls
```

```
1K_ERR038793.fastq
```

2. 実行

```
$ fastqc -f 1K_ERR038793.fastq
```

```
Started analysis of 1K_ERR038793_1.fastq
Approx 5% complete for 1K_ERR038793_1.fastq
Approx 10% complete for 1K_ERR038793_1.fastq
:
:
Approx 100% complete for 1K_ERR038793_1.fastq
Analysis complete for 1K_ERR038793_1.fastq
```

データのクオリティチェックとクリーニング

- FastQC
  FASTQまたはBAMのクオリティを確認するソフトウェア。

3. 結果：レポートがあるディレクトリと、ディレクトリの圧縮ファイル

```
$ ls
```

```
1K_ERR038793_1.fastq        1K_ERR038793_1_fastqc
1K_ERR038793_1_fastqc.zip
```

4. 解析レポート

```
$ cd 1K_ERR038793_1_fastqc
$ ls
```

```
Icons   fastqc_data.txt     summary.txt
Images  fastqc_report.html
```

データのクオリティチェックとクリーニング

- FastQC
  FASTQまたはBAMのクオリティを確認するソフトウェア。

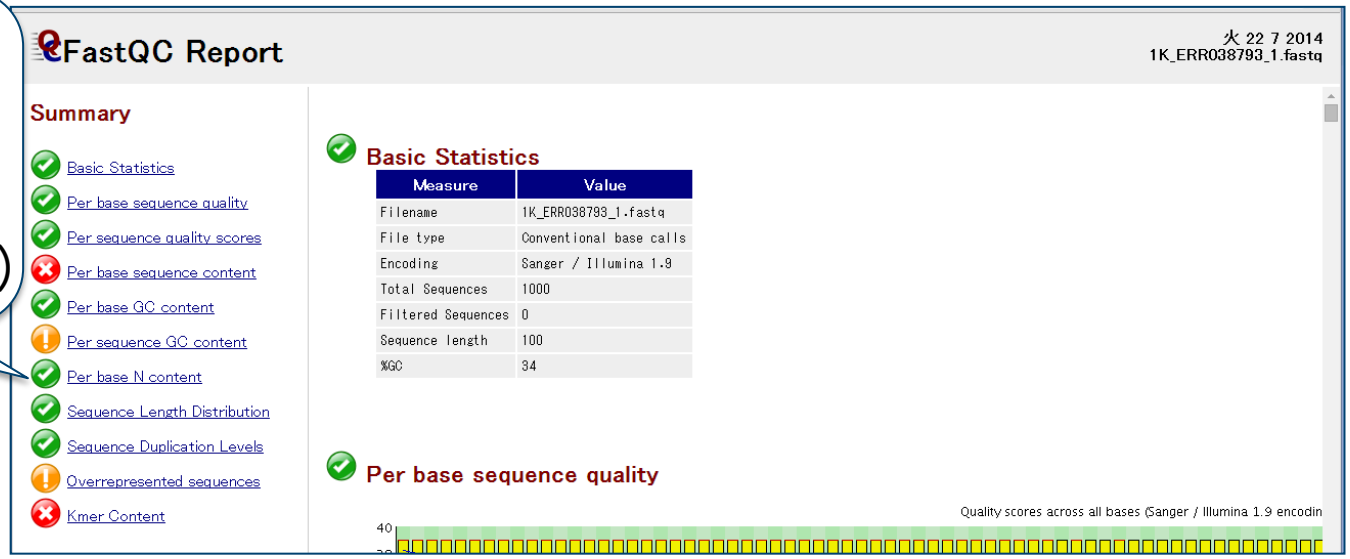
5. ウェブブラウザでレポートを開く

```
$ firefox fastqc_report.html
```

問題なし

注意 (warning)

問題あり (failure)



データのクオリティチェックとクリーニング

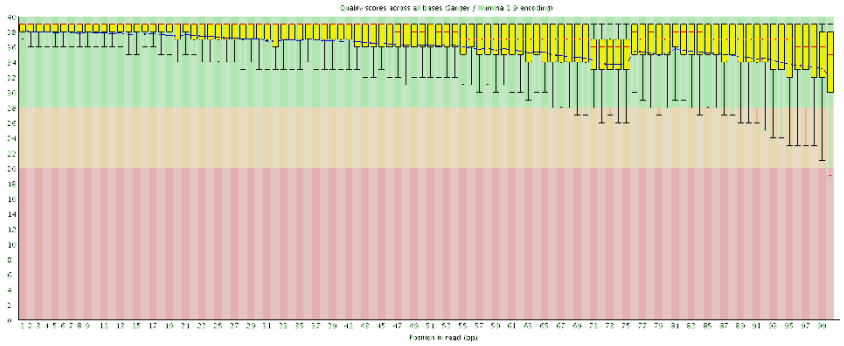
- FastQCのレポート

Basic Statistics

Measure	Value
Filename	1K_ERR038793_1.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	1000
Filtered Sequences	0
Sequence length	100
%GC	34

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。

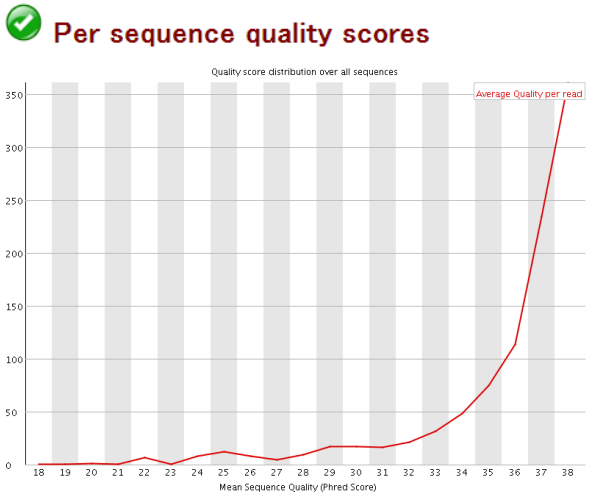
Per base sequence quality



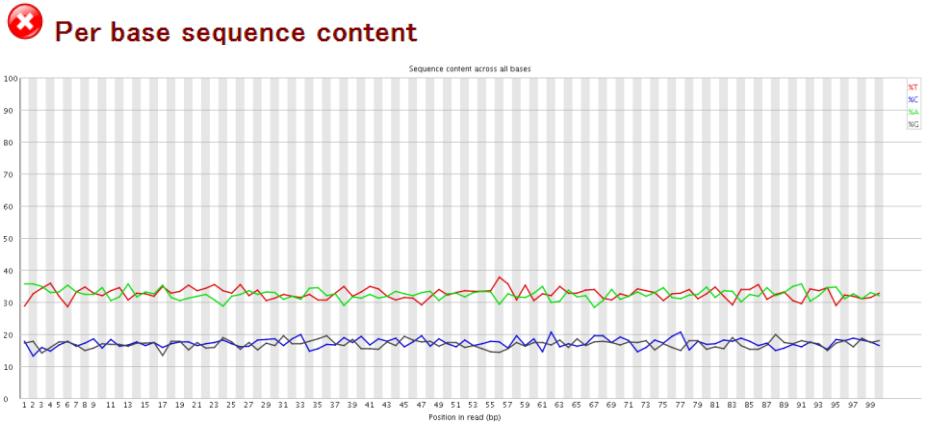
Per Base Sequence Quality
横軸はリード長、縦軸はquality valueを表す。
リードの位置における全体のクオリティの中央値や平均を確認できる。赤線は中央値、青線は平均値、黄色のボックスは25%～75%の領域を表す。上下に伸びた黒いバーが10%～90%の領域を意味する。

データのクオリティチェックとクリーニング

- FastQCのレポート



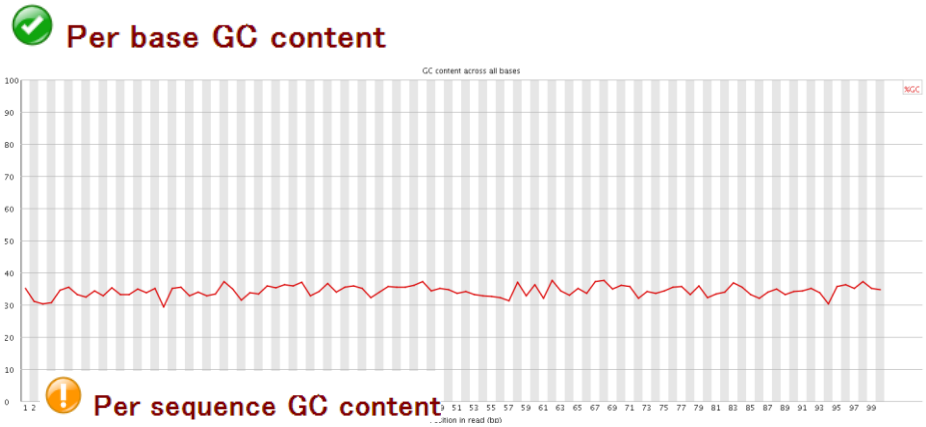
Per Sequence Quality Scores
縦軸がリード数、横軸がPhred quality scoreの平均値。

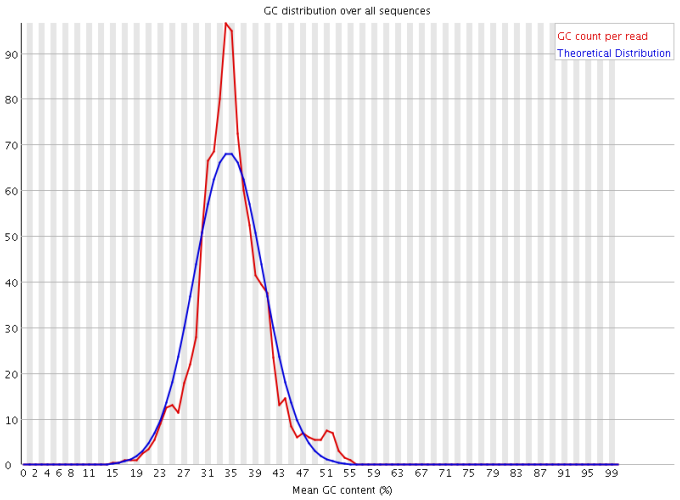


Per Base Sequence Content
リードにおける位置での各塩基の割合を示す。
いずれかの位置で、AとTの割合の差、もしくはGとCの割合の差が10%以上だとwarning,20%以上でfailureとなる。

データのクオリティチェックとクリーニング

- FastQCのレポート



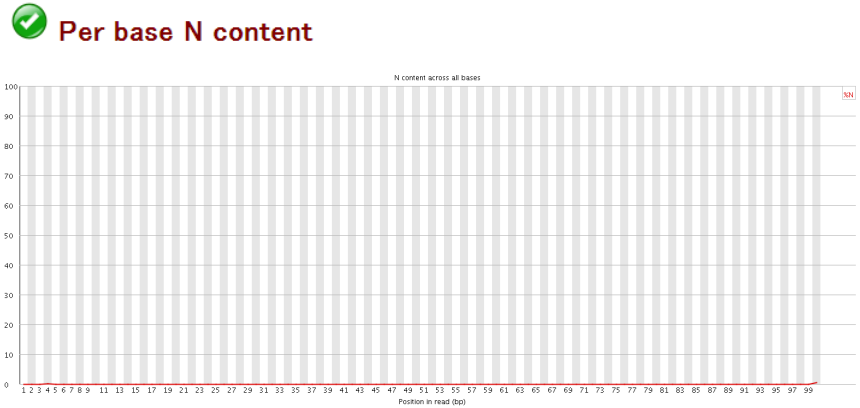


Per Base GC Content
リードにおける位置でのGC含量を表す。
いずれかの位置で、全体でのGC含量の平均値より5%以上の差が開くとwarning, 10%でfailureとなる。

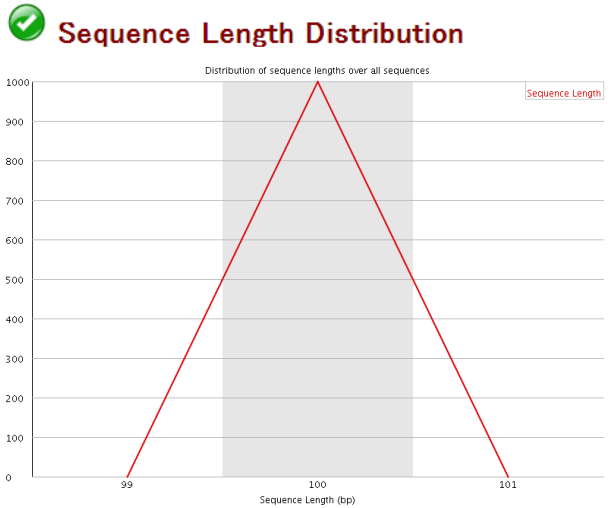
Per Sequence GC Content
各リードにおけるGC含量の平均の分布(赤線)と、理論分布(青線)。
理論分布との偏差の合計が、総リードの15%以上でwarning, 30%以上でfailureとなる。

データのクオリティチェックとクリーニング

■ FastQCのレポート



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。

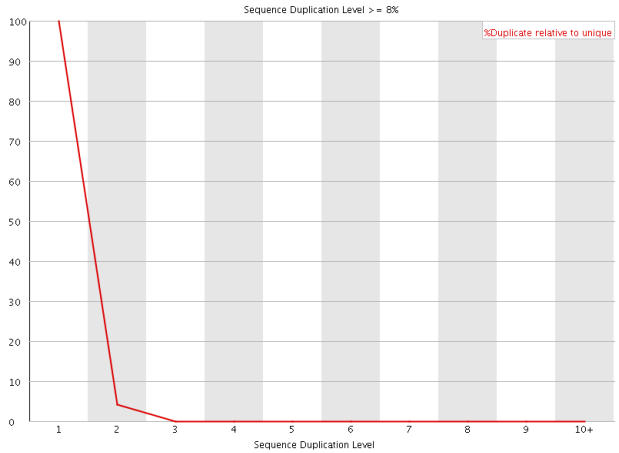


Sequence Length Distribution
リード長の全体の分布。
全てのリードの長さが同じであることを前提としており、一定でなければwarning、ゼロのものが含まれているとfailureになる。

データのクオリティチェックとクリーニング

- FastQCのレポート

Sequence Duplication Levels



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failureとなる。

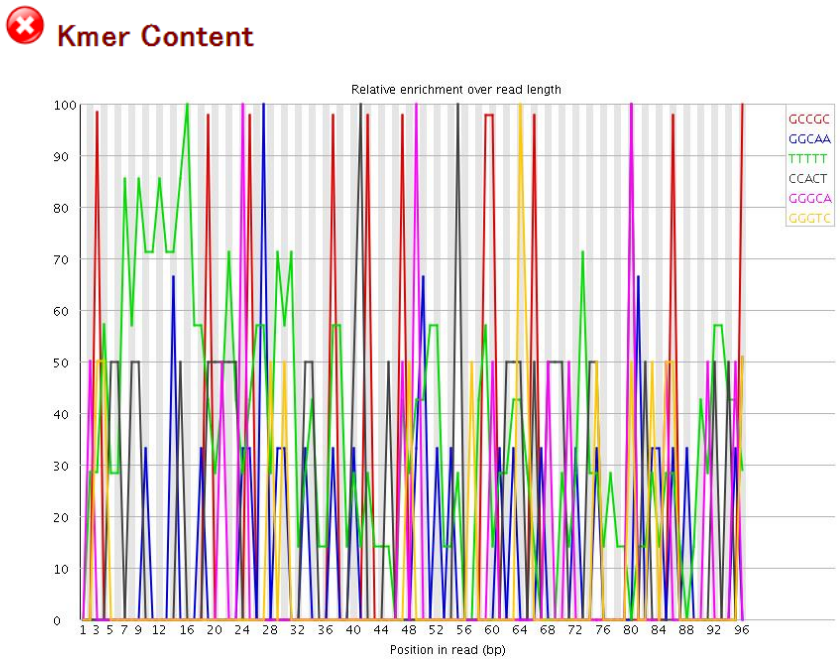
Overrepresented sequences

Sequence	Count	Percentage	Possible Source
AGTATTAATATTTCACTGTCTTGATATCGTTATCCCCATCGTAACGTGAA	2	0.2	No Hit
GCTTTAAACGGCTTCCGCGGAAGAAATATTTCCATCTCTTGAATTCGTAC	2	0.2	No Hit
CTTTTACACCATATACTAACCACTCAATTTATATACACTTATGCCAATAT	2	0.2	No Hit
CCTGTCCCATTCAACCATACCACTCCGAACCACCATCCATCCCTCTACTT	2	0.2	No Hit
AACCCGCTACGTTGACTACAAGCTCAAACCCGAATACCACATCTGCACGT	2	0.2	No Hit
GTCAATTTCTACTTGCCTCATTAGGGAAAAATTTAATAGCAGTTGTTATA	2	0.2	No Hit
CCATTATGACAAAGTTAAGGAGTTACGCGTGCTACATCACCGTAAAAATT	2	0.2	No Hit

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。

データのクオリティチェックとクリーニング

- FastQCのレポート



K-mer Content

5 bpの任意の配列(5mer)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。
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。

データのクオリティチェックとクリーニング

- マッピング率の確認
  - リファレンスゲノムへのマッピング率が一般的な割合より著しく低い場合、他生物ゲノムのコンタミなどが疑われる。

Mapped reads / Total reads

解析	一般的なマッピング率
Reseq	90~99%
RNA-seq	約80%
ChIP-seq	約70%

- あくまで一般的な割合。実験手法や解析手法が特殊な場合は、これらの数値から離れることがある。

データのクオリティチェックとクリーニング

- マッピング率の確認
  - マルチマップされたリードを除き、ユニークリードのみにする

```
$ samtools view -b -F 256 SRR504515.bam > SRR504515_uniq.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 SRR504515_uniq.bam
$ samtools idxstats SRR504515_uniq.bam > SRR504515_idxstats.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

データのクオリティチェックとクリーニング

- マッピング率の確認
  - idxstatsの見方

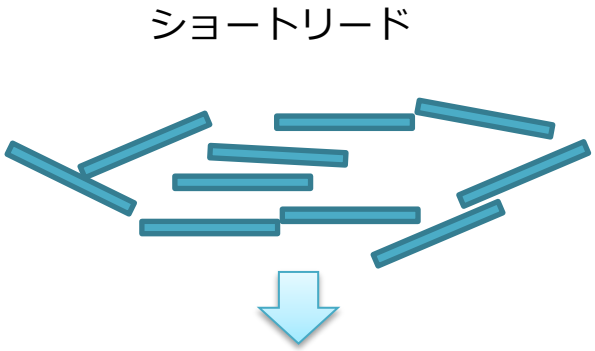
Seq name	Sequence length	Mapped reads	Unmapped reads
chr1	249250621	63735	0
chr2	243199373	0	0
:	:	:	:
chrM	16571	0	0
*	0	0	0

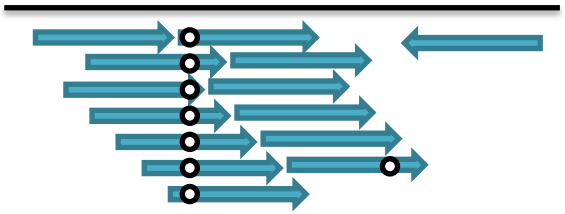
マッピング率＝

マップされたリード / (マップされたリード+マップされなかったリード)

NGSデータのマッピング

- シーケンサから得られたリード（DNA配列）を、リファレンスゲノムや転写産物上の類似した配列に対して並べること。
- BLASTのような従来のマッピングソフトは正確だが時間がかかり、NGS解析に向かないため、NGS解析用の高速なマッピングソフトが使われる。





NGSデータのマッピング

解析の種類	マッピングソフトの特徴	主なマッピングソフト
Reseq	大きなゲノムファイルに対して数カ所のミスマッチを許容しながら高速にマッピングする	BWA、Bowtie
RNA-seq	既知の転写産物やスプライシングにより生じるギャップを考慮しながらマッピングする	STAR、HISAT
Methyl-seq	メチル化を考慮してマッピングする	BSMAP、Bisulfighter

【実践！】
新しいソフトウェアの導入

【実践！】新しいソフトウェアの導入

「○○ってソフトがいいよ！」
と勧められた

この論文で使っているソフト、
使ってみたい

でも、使い方がわからないからあきらめよう…

新しいソフトを
使えるようになりましょう！

【実践！】新しいソフトウェアの導入

■ 導入の手順

1. 検索サイトで検索をして、ソフトウェアの配布サイトを探す。
2. ソフトウェアをダウンロードする。
3. 解凍する。
4. インストール方法を調べる。

5-1. コンパイルして実行ファイルを作成する。

5-2. コンパイルは必要ない。実行ファイルが配布されている。

【実践！】新しいソフトウェアの導入

- **Trimmomatic**：アダプターの除去、低クオリティリードの除去など、多様なシーケンスリードクリーニング機能をもつソフトウェア
  - Bolger, A. M., Lohse, M., & Usadel, B. (2014). Trimmomatic: A flexible trimmer for Illumina Sequence Data. Bioinformatics, btu170.

以下の順番でクリーニングが実行される

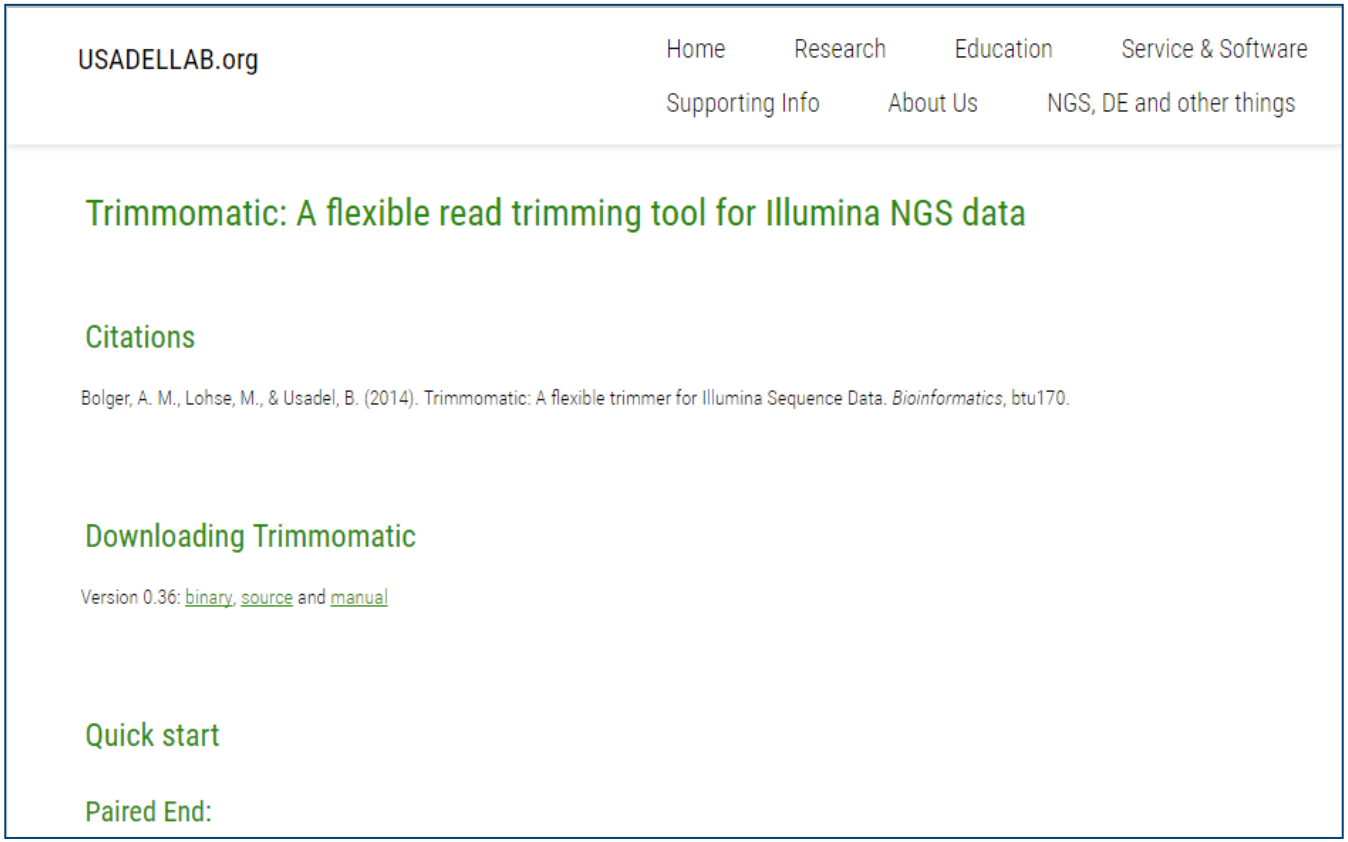
The current trimming steps are:

- ILLUMINACLIP: Cut adapter and other illumina-specific sequences from the read.
- SLIDINGWINDOW: Perform a sliding window trimming, cutting once the average quality within the window falls below a threshold.
- LEADING: Cut bases off the start of a read, if below a threshold quality
- TRAILING: Cut bases off the end of a read, if below a threshold quality
- CROP: Cut the read to a specified length
- HEADCROP: Cut the specified number of bases from the start of the read
- MINLEN: Drop the read if it is below a specified length
- TOPHRED33: Convert quality scores to Phred-33
- TOPHRED64: Convert quality scores to Phred-64

【実践！】新しいソフトウェアの導入

1. ソフトウェアの配布サイトを探す。

http://www.usadellab.org/cms/?page=trimmomatic



【実践！】新しいソフトウェアの導入

2. ソフトウェアの配布サイトを探すソフトウェアをダウンロードする。

リンクをクリックしてダウンロード、
またはソフトウェアのURLから
wgetコマンドでダウンロード


```
$ wget ¥
  http://www.usadellab.org/cms/uploads/supplementary/Trimmomati
  c/Trimmomatic-0.36.zip
```

その他にHP上で適切なダウンロード方法が指示されている場合は、その手順に従う。

【実践！】新しいソフトウェアの導入

3. 解凍する。
   - ダウンロードしたファイルの拡張子に適した解凍方法を用いる。

拡張子	圧縮形式	コマンド
.tar.gz	gzip	$ tar zxvf [ファイル名]
.tar.bz2	gzip2	$ tar jxvf [ファイル名]
.gz	gzip	$ gunzip [ファイル名]
		$ gzip -d [ファイル名]
.bz2	bzip2	$ bunzip2 [ファイル名]
		$ bzip2 -d [ファイル名]
.zip	zip	$ unzip [ファイル名]
.tar	tar	$ tar xvf [ファイル名]

【実践！】新しいソフトウェアの導入

3. 解凍する。
   – ダウンロードしたファイルの拡張子に適した解凍方法を用いる。

```
$ ls Trimmomatic-0.36.zip
$ unzip Trimmomatic-0.36.zip
```

```
Archive:  Trimmomatic-0.36.zip
   creating: Trimmomatic-0.36/
  inflating: Trimmomatic-0.36/LICENSE
  inflating: Trimmomatic-0.36/trimmomatic-0.36.jar
   creating: Trimmomatic-0.36/adapters/
  inflating: Trimmomatic-0.36/adapters/NexteraPE-PE.fa
  inflating: Trimmomatic-0.36/adapters/TruSeq2-PE.fa
  inflating: Trimmomatic-0.36/adapters/TruSeq2-SE.fa
  inflating: Trimmomatic-0.36/adapters/TruSeq3-PE-2.fa
  inflating: Trimmomatic-0.36/adapters/TruSeq3-PE.fa
  inflating: Trimmomatic-0.36/adapters/TruSeq3-SE.fa
```

【実践！】新しいソフトウェアの導入

4. インストール方法を調べる。
   - 「README」や「INSTALL」というファイル内にインストール方法が記載されていることが多い。

```
$ cd Trimmomatic-0.36
$ ls -ls
```

```
-rw-r--r-- 1 iu iu  35147  4月 27 10:45 2011 LICENSE
drwxr-xr-x 2 iu iu   4096  3月 21 16:27 2016 adapters
-rw-r--r-- 1 iu iu 126230  3月 21 16:27 2016 trimmomatic-0.36.jar
```

```
$ cd ../
```

【実践！】新しいソフトウェアの導入

5. 実行する

「.jar」ファイルはプログラミング言語Javaで書かれたコンパイル済みのプログラム。下記のコマンドで、すぐ実行できる。

```
$ java -jar Trimmomatic-0.36/trimmomatic-0.36.jar
```

```
Usage:
       PE [-version] [-threads <threads>] [-phred33|-phred64] [-trimlog
  <trimLogFile>] [-quiet] [-validatePairs] [-basein <inputBase> |
  <inputFile1> <inputFile2>] [-baseout <outputBase> | <outputFile1P>
  <outputFile1U> <outputFile2P> <outputFile2U>] <trimmer1>...
   or:
       SE [-version] [-threads <threads>] [-phred33|-phred64] [-trimlog
  <trimLogFile>] [-quiet] <inputFile> <outputFile> <trimmer1>...
   or:
       -version
```

※ 使用方法は後日の講義で説明します

【実践！】新しいソフトウェアの導入

5. 実行する

「.jar」ファイルはプログラミング言語Javaで書かれたコンパイル済みのプログラム。下記のコマンドで、すぐ実行できる。

```
$ java -jar Trimmomatic-0.36/trimmomatic-0.36.jar
```

```
Usage:
       PE [-version] [-threads <threads>] [-phred33|-phred64] [-trimlog
  <trimLogFile>] [-quiet] [-validatePairs] [-basein <inputBase> |
  <inputFile1> <inputFile2>] [-baseout <outputBase> | <outputFile1P>
  <outputFile1U> <outputFile2P> <outputFile2U>] <trimmer1>...
   or:
       SE [-version] [-threads <threads>] [-phred33|-phred64] [-trimlog
  <trimLogFile>] [-quiet] <inputFile> <outputFile> <trimmer1>...
   or:
       -version
```

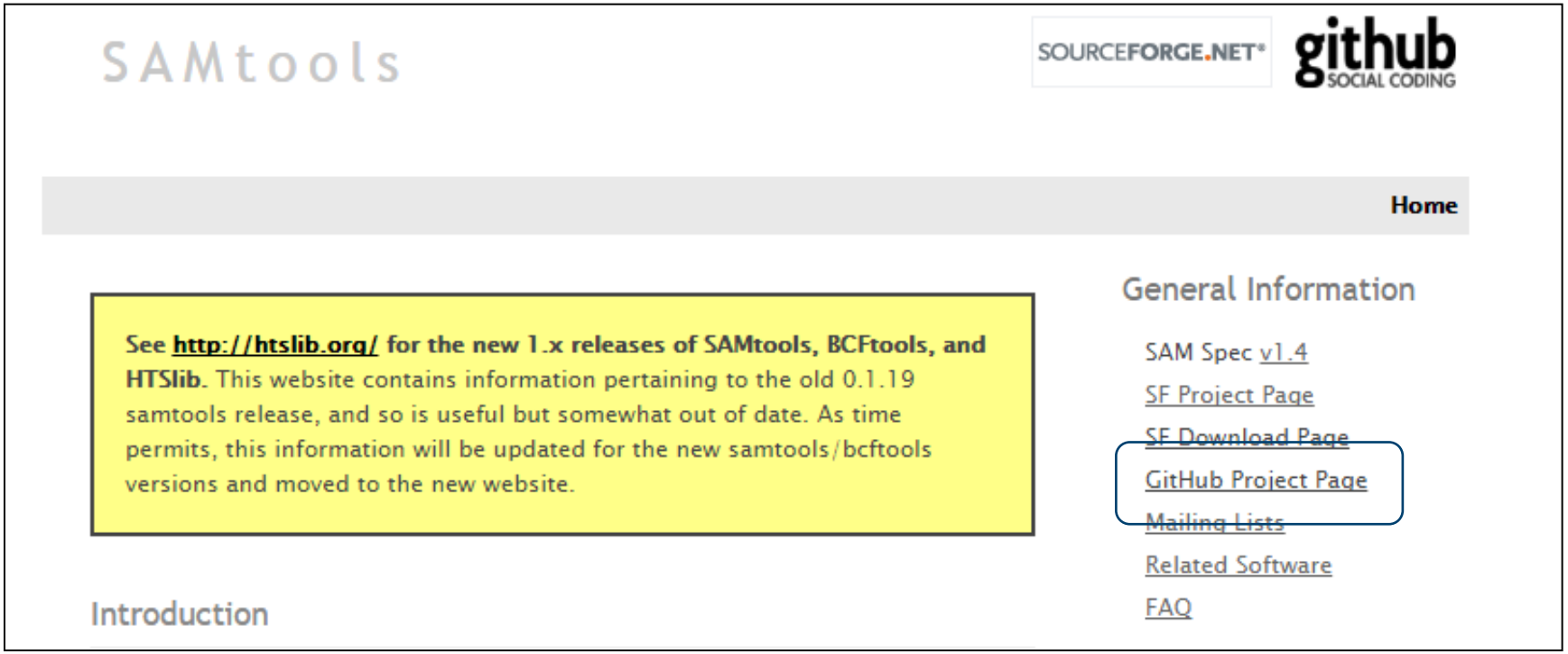
※ 使用方法は後日の講義で説明します

【実践！】新しいソフトウェアの導入

疑問解決①　GitHubとは？

頻繁に更新されるソフトウェアは、GitHub（ソフトウェア開発のための共有サービス）で配布されていることも多い。

【例】SAMtools

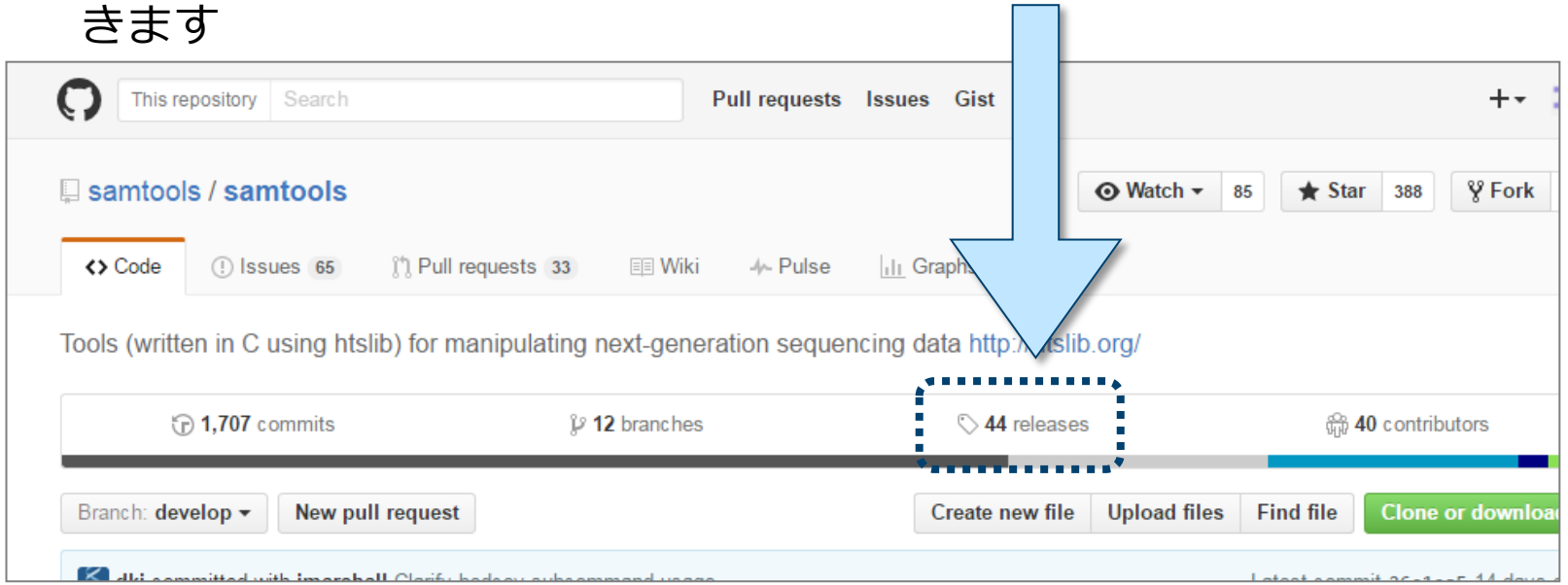


【実践！】新しいソフトウェアの導入

疑問解決①　GitHubとは？

頻繁に更新されるソフトウェアは、 GitHub（ソフトウェア開発のための共有サービス）で配布されていることも多い。

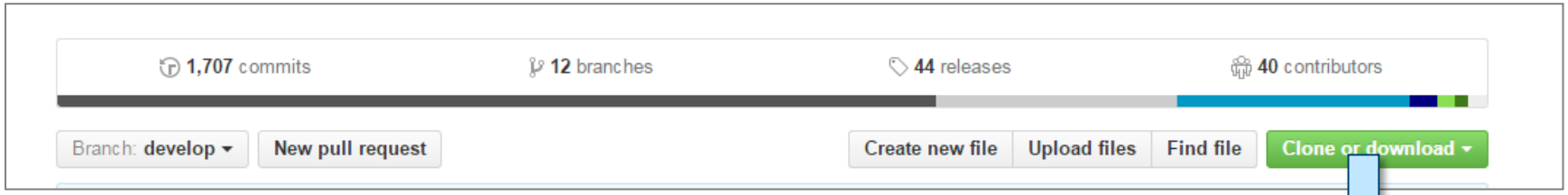
- GitHubからのダウンロード方法①　GitHubのRelease機能を使って配布用バイナリやソースコードを配布している場合は、ここからダウンロードできます



【実践！】新しいソフトウェアの導入

疑問解決①　GitHubとは？

- GitHubからのダウンロード方法②　GitHubのレポジトリ（ファイルなどの管理を行う場所）をClone（コピー）する



リモート（=オンライン上の）レポジトリのURLをコピーしてローカルにクローン（＝複製）します

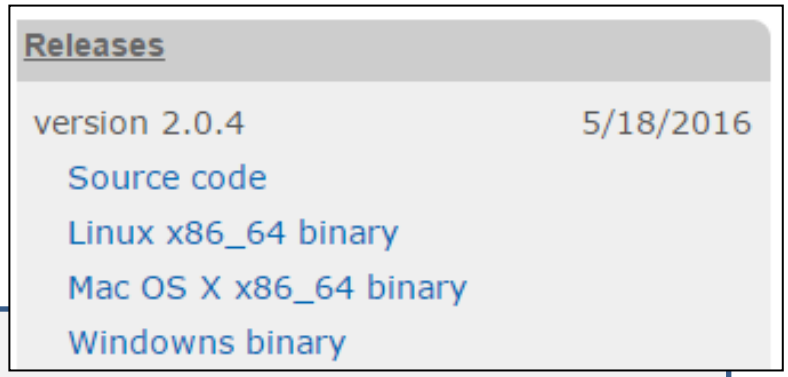

```
$ git clone https://github.com/samtools/samtools.git
```

【実践！】新しいソフトウェアの導入

疑問解決②たくさんの種類が配布されている場合、どれを選べばいい？

■ 使用する**OS**にあった**バイナリ**ファイルを選ぶ

【例】RNA-seqマッピングソフトHISAT2→



■ **Tips**

Source: プログラミング言語で書いたソフトウェア

Binary: プログラミング言語で書いたソフトウェアを**コンパイル**した、すぐ実行できる状態のソフトウェア

Source codeをダウンロードしてコンパイルして使用することもできるが、コンパイル時にエラーが起きたりしてうまくいかないこともあるため、source codeしか配布されていない場合や、binaryを使ってみてうまくいかなかった場合を除き、binaryを使用したほうがいい。

ご聴講
ありがとうございました

おまけ・gz圧縮ファイルを扱うコマンド

- 圧縮

```
$ gzip SRR504515_R1.fastq
$ ls
```

```
SRR504515_R1.fastq.gz
```

- 解凍

```
$ gunzip SRR504515_R1.fastq.gz
$ ls
```

```
SRR504515_R1.fastq
```

おまけ・gz圧縮ファイルを扱うコマンド

■ 圧縮したままファイルの中を見る

```
$ zless SRR504515_R1.fastq.gz
```

```
@SRR504515.1 HWI-ST423_0087:2:1:1183:2098 length=101
AAANGACGGTTGGTCCTTAAAATTCCATGGATGTAGATCTTATCCCCACACCCAGACTCTAGTG
```

類似のコマンドに zmore がある。

■ 複数の圧縮ファイルをまとめて1つのgzファイルにする

```
$ gunzip -c SRR504515_L001_R1.fastq.gz ¥
  SRR504515_L002_R1.fastq.gz | gzip -c > ¥
  SRR504515_R1.fastq.gz
$ ls
```

```
SRR504515_R1.fastq
```

-c ：結果をファイルではなく標準出力に出力するオプション