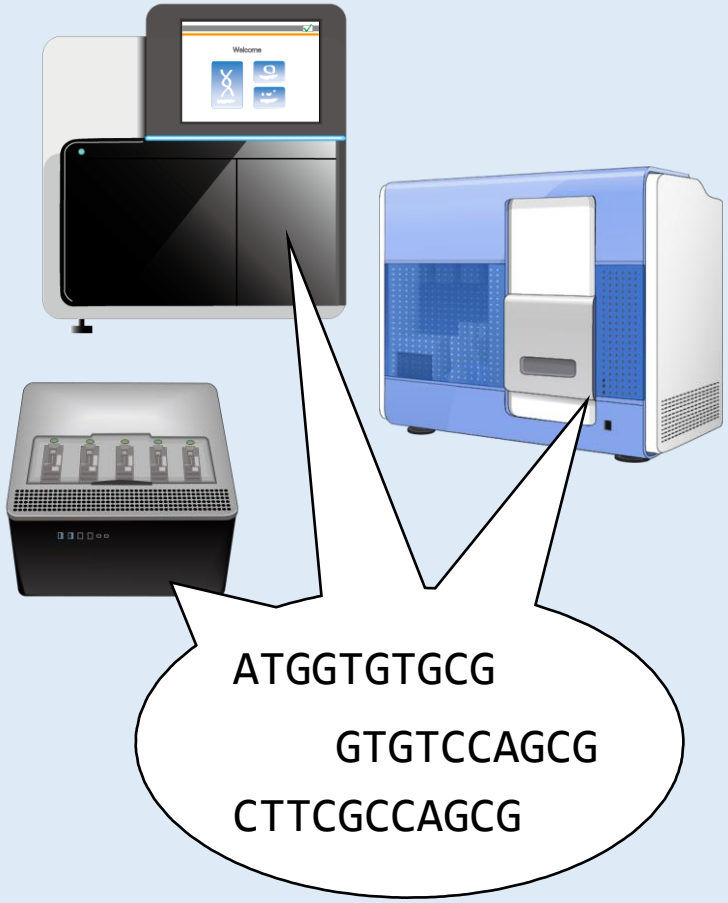
変異検出用ソフトウェア Sentieon を使用した Joint Genotyping

フィルジェン株式会社　バイオインフォマティクス部
(support@filgen.jp)

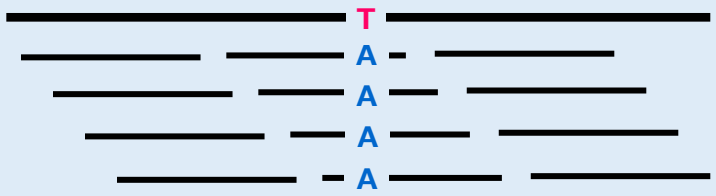
Sentieonについて

一次解析



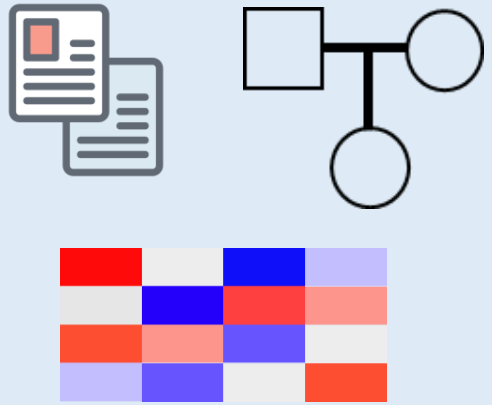
・ベースコール

二次解析



・リードマッピング
・バリアントコール

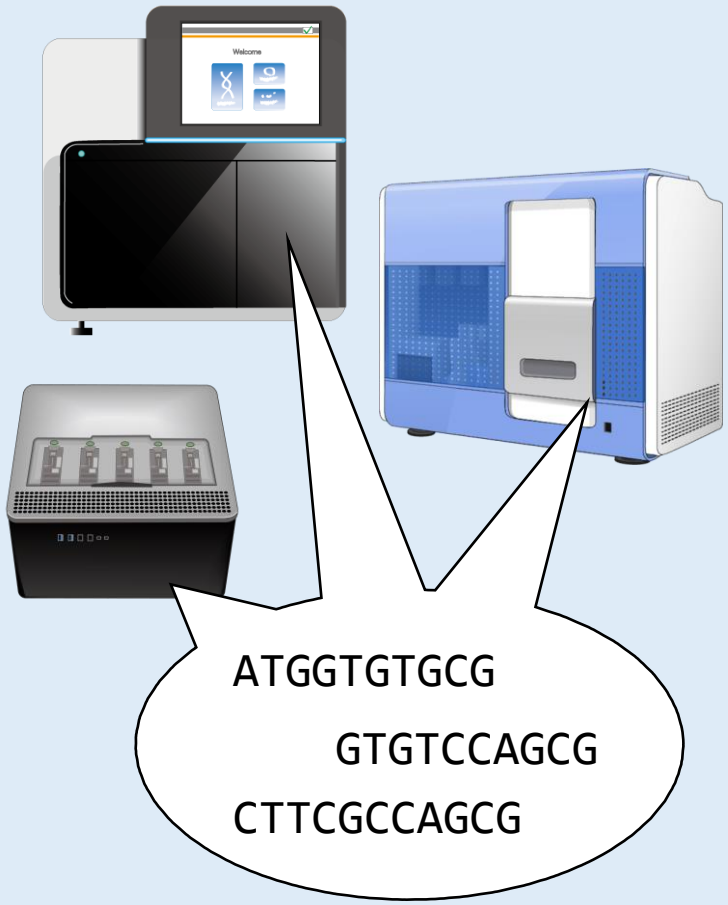
三次解析



・アノテーション付け
・フィルタリング
・トリオ解析
・ビジュアライゼーション
・レポート作成
など

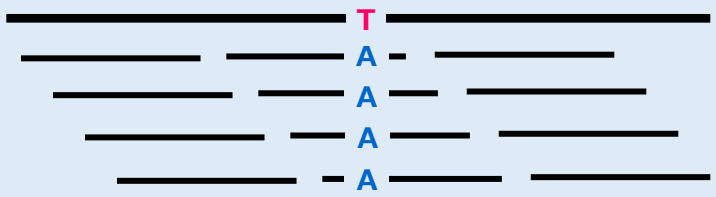
Sentieonについて

一次解析



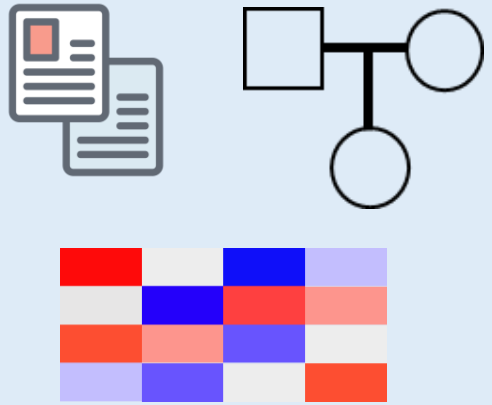
・ベースコール

二次解析



・リードマッピング
・バリアントコール

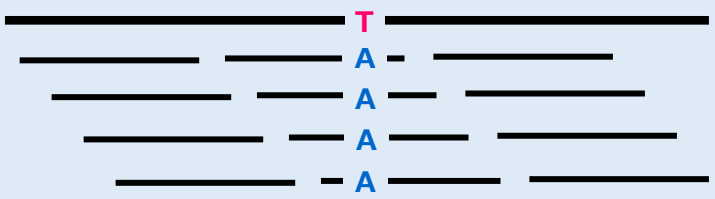
三次解析



・アノテーション付け
・フィルタリング
・トリオ解析
・ビジュアライゼーション
・レポート作成
など

二次解析における問題

二次解析



・リードマッピング
・バリアントコール

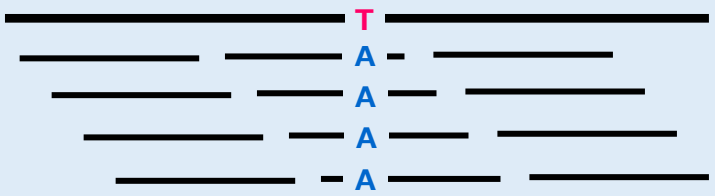
・ゲノム上に膨大な量のリードをマッピング

・シーケンスエラー等も考慮しながら、
変異サイトを検出

→大量の計算リソースを必要とするプロセス

二次解析用のソフトウェア

二次解析



・リードマッピング
・バリアントコール

二次解析用のソフトウェア

・Illumina DRAGEN
・GATK
・NVIDIA Parabricks
・Sentieon
など

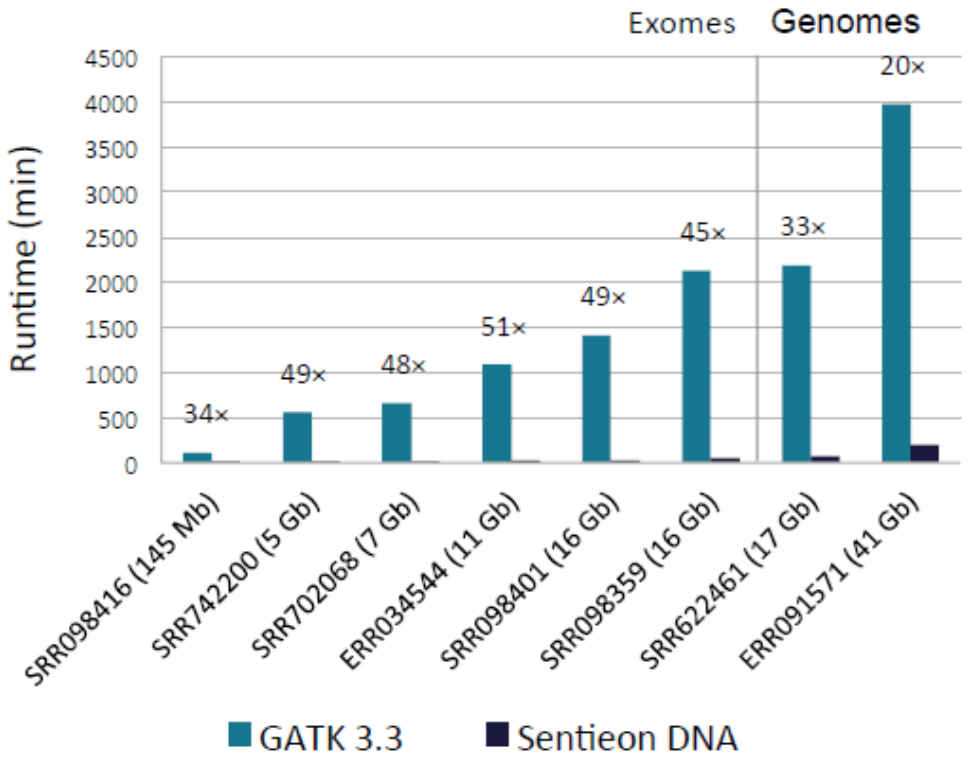
オープンソースのものから商用のものまで様々

Sentieonの特長

Sentieonを使うメリット

- 二次解析のゴールドスタンダードであるGATKと同じ数学モデルを使用しつつ、アルゴリズムを見直すことで高速化を実現
- 一般的なCPUベースのシステムで動作
- さまざまな解析用のサンプルスクリプトの提供
- ロングリードやハイブリッドアセンブリに対応したパイプラインも提供

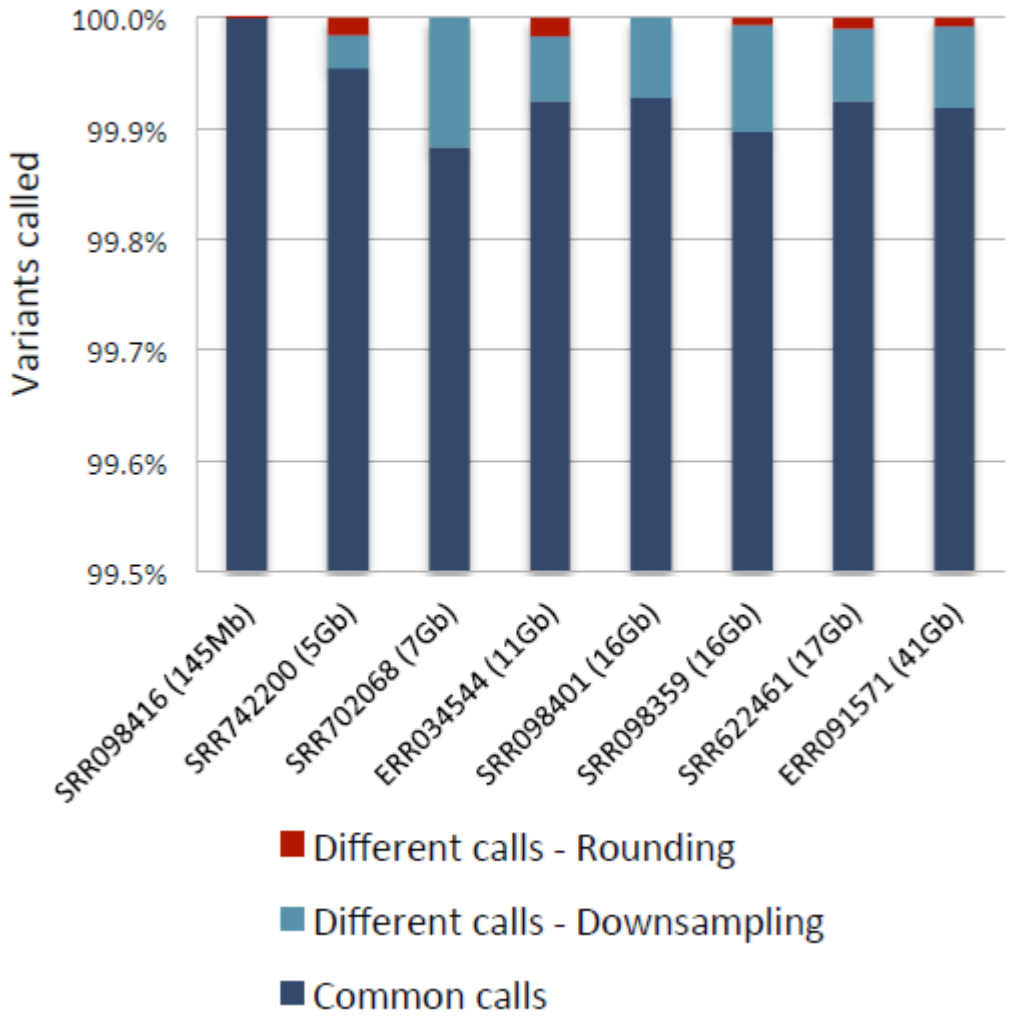
SentieonとGATKの比較



・BAMファイルからのVCFファイル作成において、SentieonはGATKに比べて30倍～50倍程度（エクソーム）、20倍～30倍程度（全ゲノム）の高速化を実現（Weber et al., 2016）

・FASTQファイルからVCFファイル作成の場合は、10倍程度の高速を実現

SentieonとGATKの比較



・SentieonとGATKでコールされたバリアントを比較した結果、99.8%のバリアントが共通してコールされていた。

誤差は、GATKのダウンサンプリング（※）や丸め誤差に起因するものであった（Weber et al., 2016）。

※GATKでは、高カバレッジ領域においてリードのダウンサンプリングを行うため、解析ごとに結果がばらつく

Sentieonのパイプライン

Sentieon DNAseq

- 生殖細胞系列変異の検出
- GATKと同じ結果だが、より高速

Sentieon DNAscope

- 生殖細胞系列変異の検出
- アセンブリアルゴリズムの改良と機械学習による精度向上

Sentieon TNseq

- 体細胞系列変異の検出
- Mutect/Mutect2と同じ結果だが、より高速

Sentieon TNscope

- 体細胞系列変異の検出
- SNV, INDEL, SVを包括的に検出

Sentieonのパイプライン

Sentieon DNAseq

- 生殖細胞系列変異の検出
- GATKと同じ結果だが、より高速

Sentieon DNAscope

- 生殖細胞系列変異の検出
- アセンブリアルゴリズムの改良と機械学習による精度向上

Illumina社、Complete Genomics社、PacBio社、Ultima Genomics社、Element Biosciences社、Oxford Nanopore社等、さまざまなプラットフォームに特有のモデルを使用し、精度を向上

DNAseq vs DNAscope

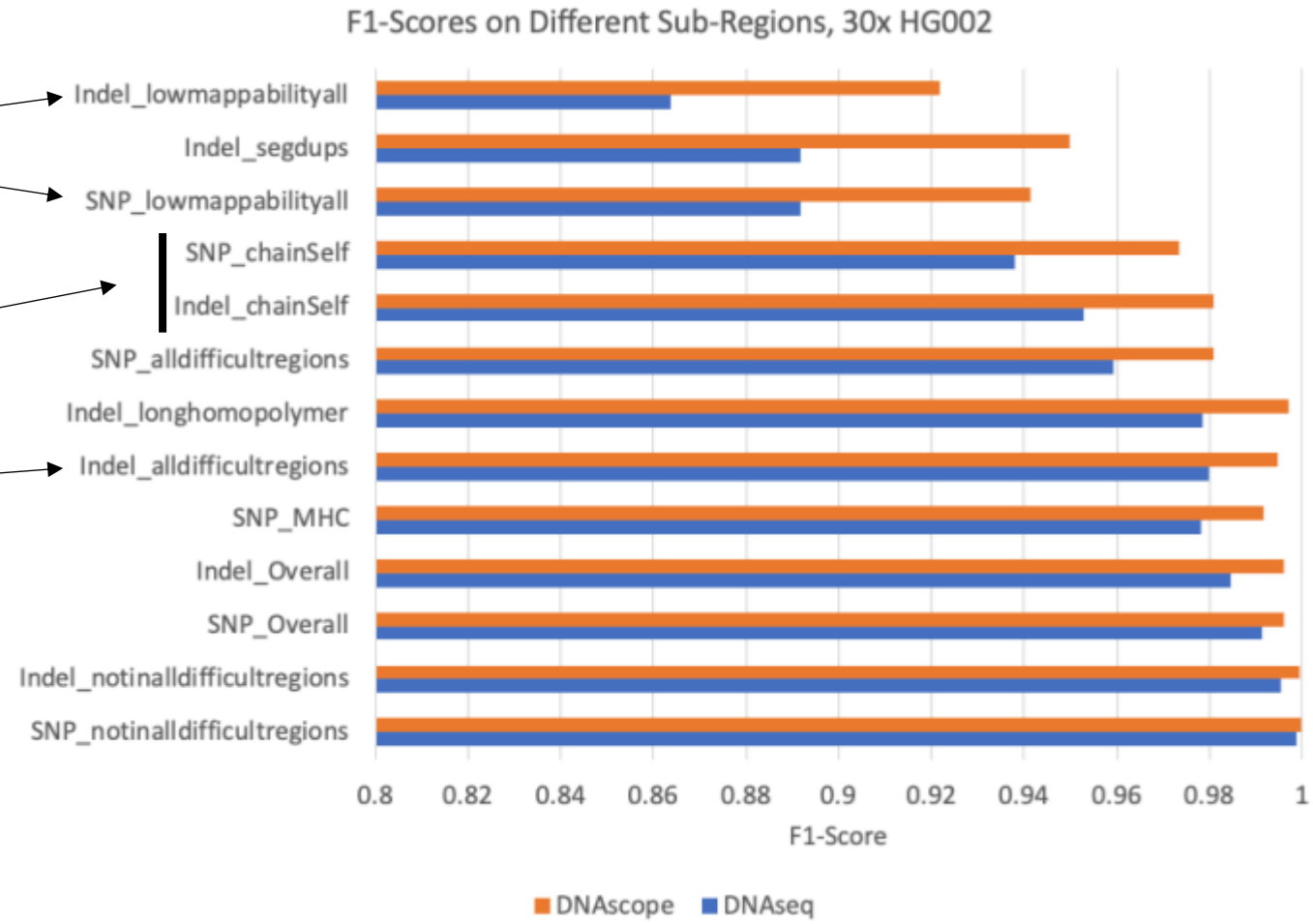
低”mappability”領域

似たような配列がゲノム中に散在し、
マッピングされにくい領域

Self-chain領域

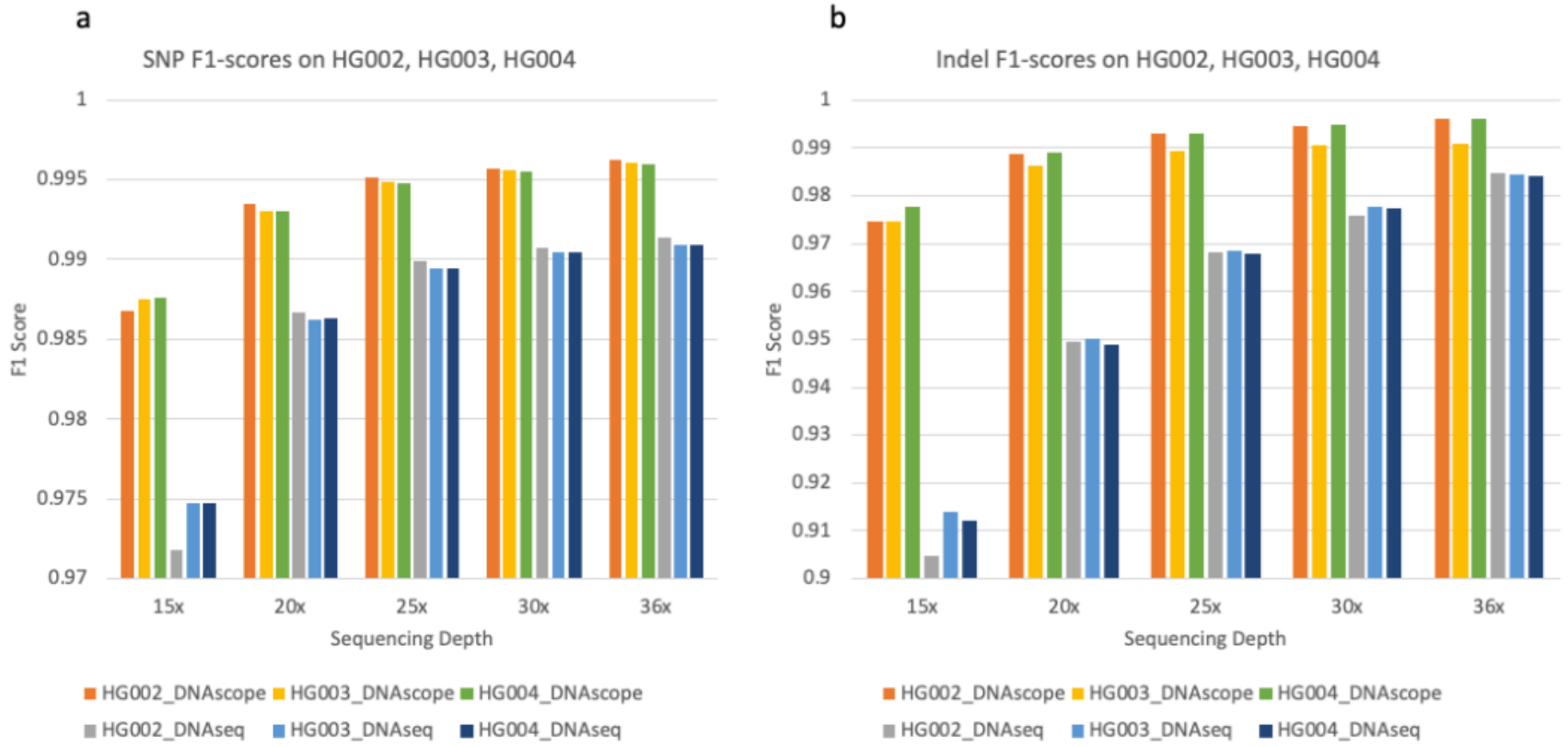
“alldifficult”領域

タンデム重複や長いホモポリマー等
複数の要因で正確なマッピングが難しい領域



GA4GHが既定するゲノム上のさまざまな領域（Krusche, P. et al., 2019）において、
DNAscopeはDNAseqの変異検出性能を上回る（Freed, P. et al., 2022）

DNAseq vs DNAscope



特に低カバレッジ領域において、DNAscopeはDNAseqの変異検出性能を上回る（Freed, P. et al., 2022）

その他のパイプライン

Sentieon DNAscope LongRead

- 生殖細胞系列変異の検出
- アセンブリアルゴリズムの改良と機械学習による精度向上
- ロングリードに対応

Sentieon DNAscope Hybrid

- 生殖細胞系列変異の検出
- アセンブリアルゴリズムの改良と機械学習による精度向上
- ハイブリッドアセンブリに対応

その他のパイプライン

Filgen® biosciences & nanosciences

Pipeline Functions

Pipelines	DNAscope (short reads)	Pangenome	DNAscope LongRead	DNAscope Hybird	DNAseq	TNscope (somatic)	TNseq (somatic)
Alignment	✓	✓	✓	✓	✓	✓	✓
SNP/Indel	✓	✓	✓	✓	✓	✓	✓
SV	✓	✓	✓	✓		✓	
CNV	✓	✓		✓			
Segdup Genes	✓	✓		✓			
UMI Process						✓	
Matching GATK					✓		✓

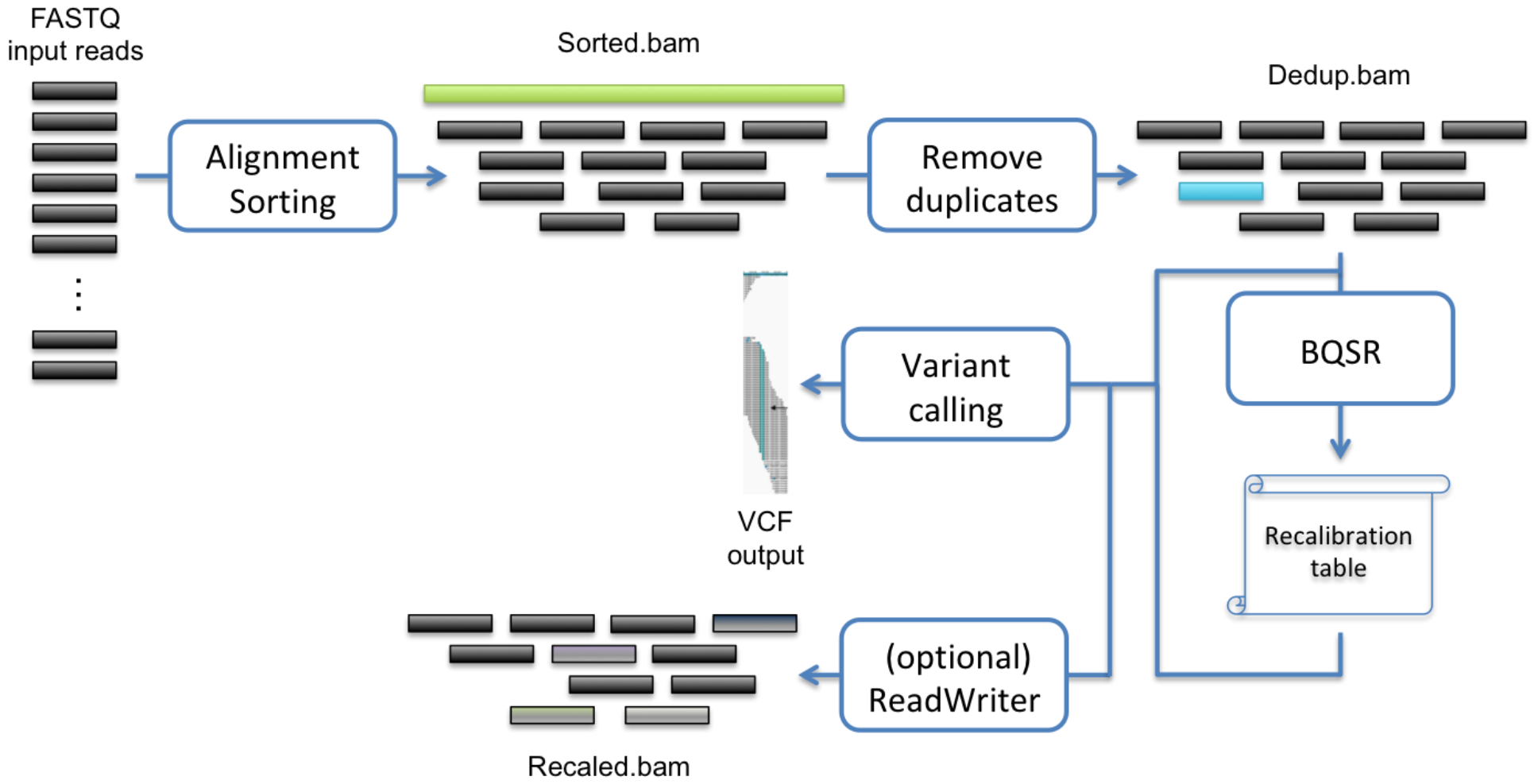
今回使用するパイプライン

Filgen® biosciences & nanosciences

Pipeline Functions

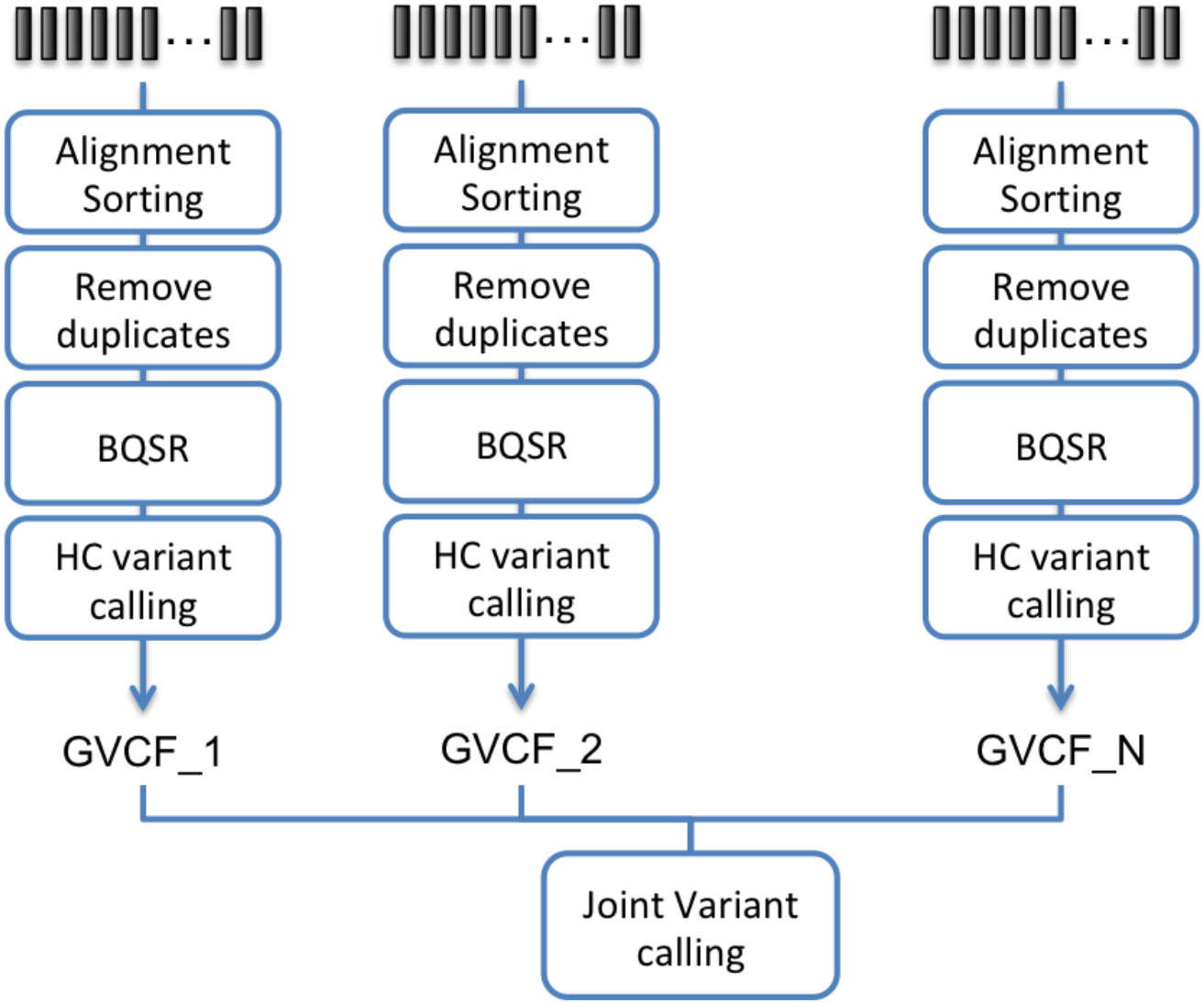
Pipelines	DNAscope (short reads)	Pangenome	DNAscope LongRead	DNAscope Hybird	DNAseq	TNscope (somatic)	TNseq (somatic)
Alignment	✓	✓	✓	✓	✓	✓	✓
SNP/Indel	✓	✓	✓	✓	✓	✓	✓
SV	✓	✓	✓	✓		✓	
CNV	✓	✓		✓			
Segdup Genes	✓	✓		✓			
UMI Process						✓	
Matching GATK					✓		✓

一般的なDNAseqパイプライン



Joint Genotyping用のパイプライン

DNAseqを使用する場合



DNAseqを使用したJoint Calling①

マッピングとソート

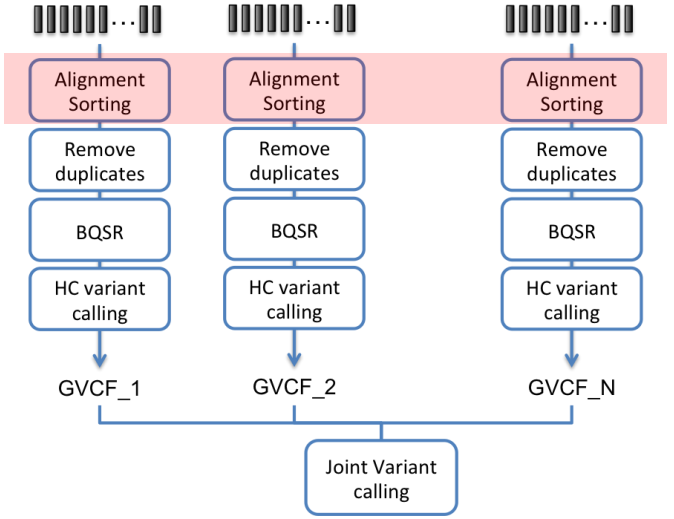
```
sentieon bwa mem -R '@RG¥tID:GROUP_NAME¥tSM:SAMPLE_NAME¥tPL:PLATFORM' -t
NUMBER_THREADS REFERENCE SAMPLE [SAMPLE2] || echo -n 'error' )
  | sentieon util sort -r REFERENCE -o SORTED_BAM -t NUMBER_THREADS --sam2bam -i -
```

sentieon bwa mem: マッピングを実行するコマンド

-R: BAMファイルのヘッダーを入力
  tID: リードに固有のID
  tSM: サンプルに固有のID
  tPL: プラットフォーム名
-t: スレッド数

Sentieon util sort: BAMファイルのソートを実行するコマンド

-r: リファレンス配列の場所／名前
-o: 出力ファイル名
-t: スレッド数



DNAseqを使用したJoint Calling②

重複リードの除去

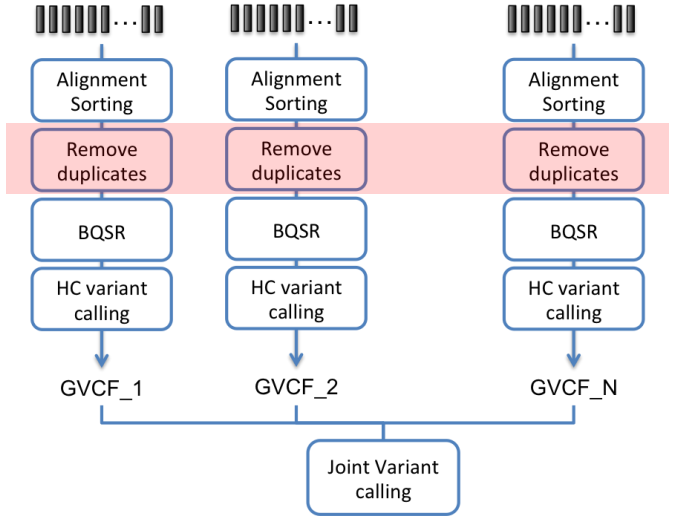
```
sentieon driver -t NUMBER_THREADS -i SORTED_BAM
  --algo LocusCollector --fun score_info SCORE.gz
sentieon driver -t NUMBER_THREADS -i SORTED_BAM
  --algo Dedup [--rmdup] --score_info SCORE.gz
  --metrics DEDUP_METRIC_TXT DEDUPED_BAM
```

sentieon driver: パイプラインを実行するための主なコマンド

-t: スレッド数
-i: 入力ファイル

--algo LocusCollector: 重複リード除去のために必要なリードの情報を集計
  --fun score_info: スコア関数（score_info）の指定

--algo Dedup: 重複リードにフラグを付ける
  --rmdup: フラグが付いたリードの除去
  --score_info: LocusCllectorの出力
  --metrics: 重複リード除去メトリクスの保存場所



DNAseqを使用したJoint Calling③

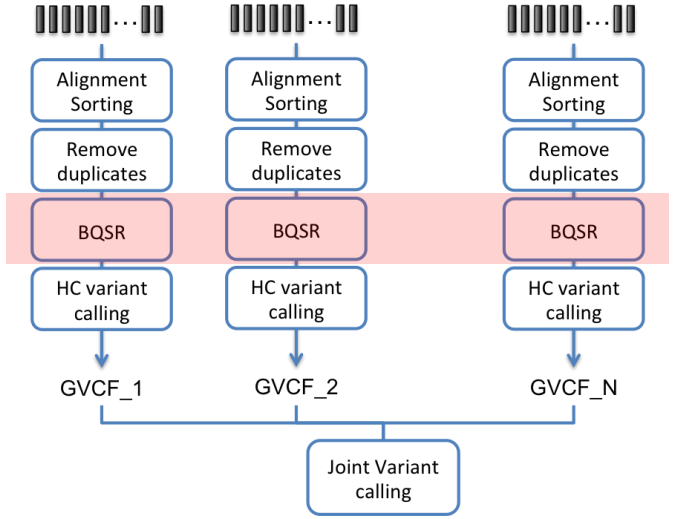
BQSR

```
sentieon driver -t NUMBER_THREADS -r REFERENCE
  -i DEDUPED_BAM --algo QualCal [-k KNOWN_SITES] RECAL_DATA.TABLE
```

sentieon driver: パイプラインを実行するための主なコマンド

-t: スレッド数
-i: 入力ファイル
-r: リファレンス配列の場所／名前

--algo QualCal: キャリブレーション用のテーブルを作成
  -k: 既知のバリアント情報（.vcf）



DNAseqを使用したJoint Calling④

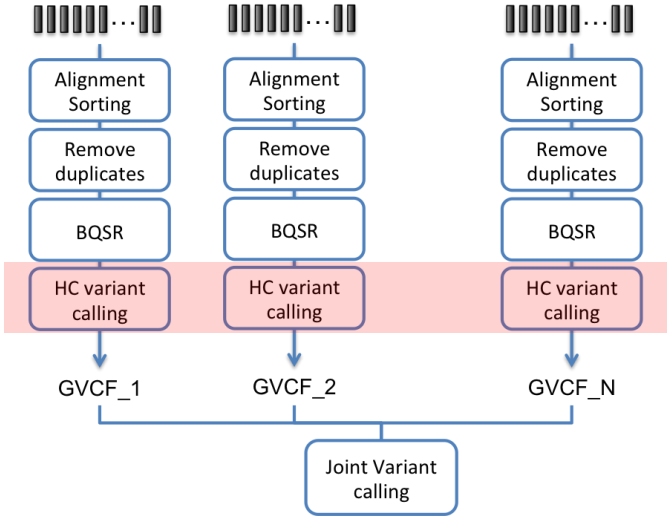
gVCFファイルの作成

```
sentieon driver -r REFERENCE -t NUMBER_THREADS -i DEDUPED_BAM -q RECAL_DATA_TABLE
--algo Haplotyper --emit_mode gvcf VARIANT_GVCF
```

sentieon driver: パイプラインを実行するための主なコマンド

-t: スレッド数
-r: リファレンス配列の場所／名前
-q: キャリブレーション用テーブル

--algo Haplotyper: バリアントコールのためのアルゴリズム
  --emit mode: 出力するバリアントの指定（variant, confident, all, gvcf）



DNAseqを使用したJoint Calling⑤

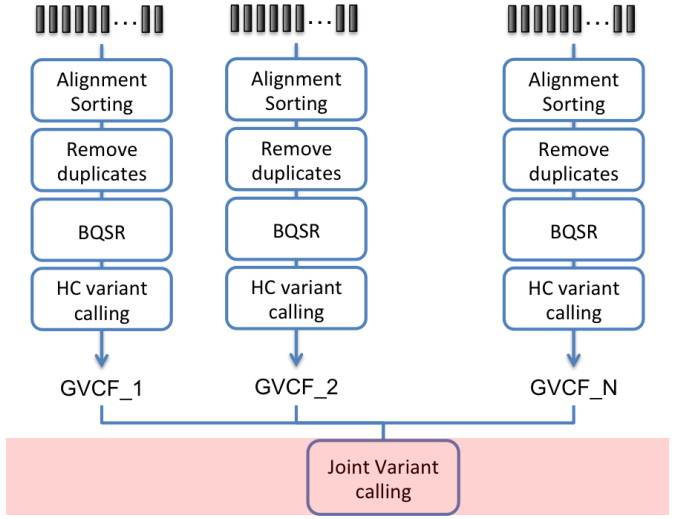
Joint Calling

```
sentieon driver -r REFERENCE --algo GVCFtyper -v s1_VARIANT_GVCF  -v s2_VARIANT_GVCF
-v s3_VARIANT_GVCF VARIANT_VCF
```

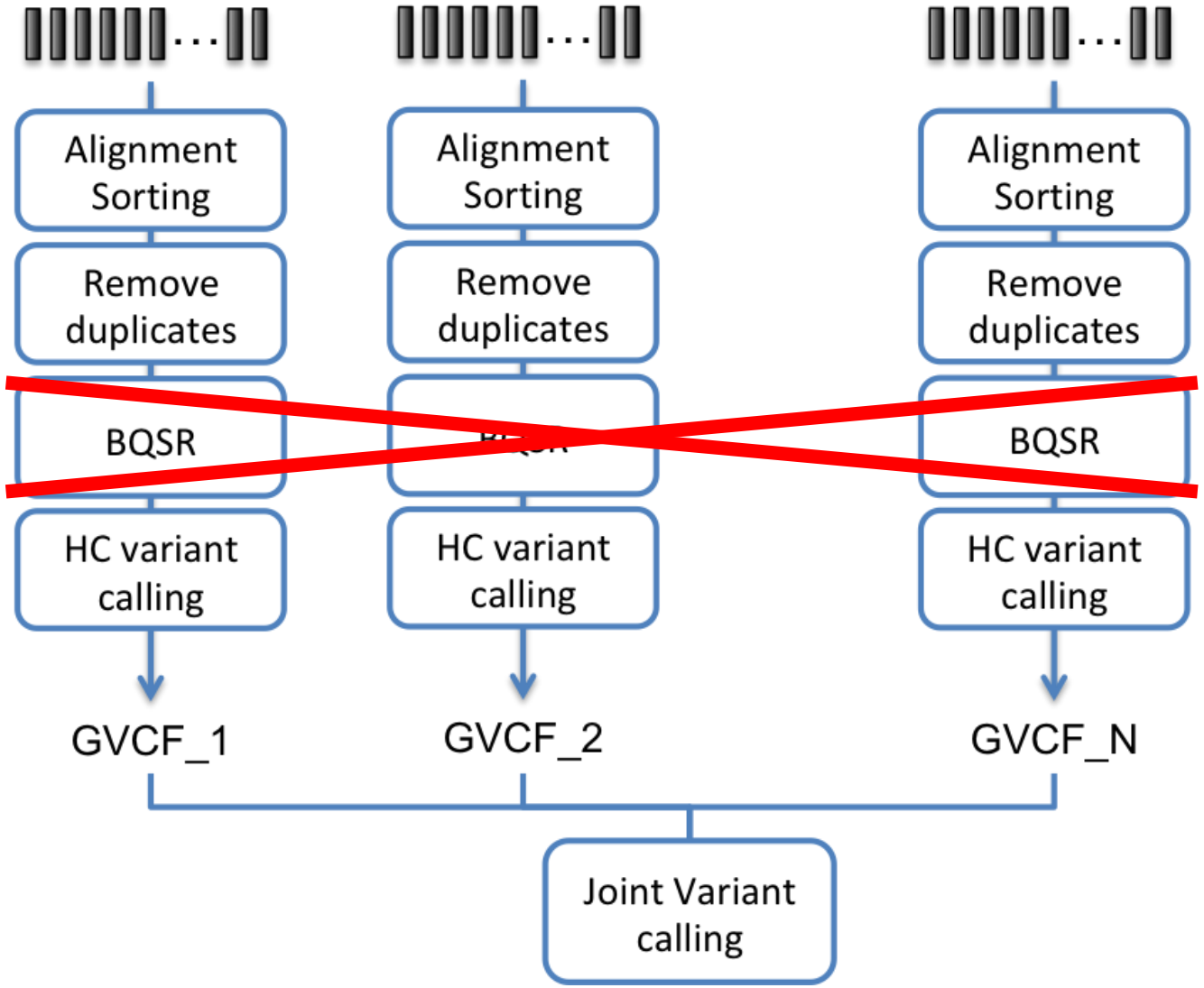
sentieon driver: パイプラインを実行するための主なコマンド

-r: リファレンス配列の場所／名前

--algo GVCFtyper: バリアントコールのためのアルゴリズム
  -v: 入力ファイル



DNAscopeを使用した場合のパイプライン



BQSRは不要
（未処理のデータで学習しているため）

DNAscopeを使用したJoint Calling①

マッピングとソート

```
sentieon bwa mem -R ‘@RG¥tID:GROUP_NAME¥tSM:SAMPLE_NAME¥tPL:PLATFORM’ -t NUMBER_THREADS
-x DNASCOPE_MODEL/bwa.model REFERENCE SAMPLE [SAMPLE2] || echo -n 'error’ )
  | sentieon util sort -r REFERENCE -o SORTED_BAM -t NUMBER_THREADS --sam2bam -i -
```

sentieon bwa mem: マッピングを実行するコマンド

-R: BAMファイルのヘッダーを入力
  tID: リードに固有のID
  tSM: サンプルに固有のID
  tPL: プラットフォーム名
-t: スレッド数
-x: モデルファイルの場所（Illumina以外のプラットフォームについては対応するモデルをgithubから入手）

Sentieon util sort: BAMファイルのソートを実行するコマンド

-r: リファレンス配列の場所／名前
-o: 出力ファイル名
-t: スレッド数

DNAscopeを使用したJoint Calling②

重複リードの除去

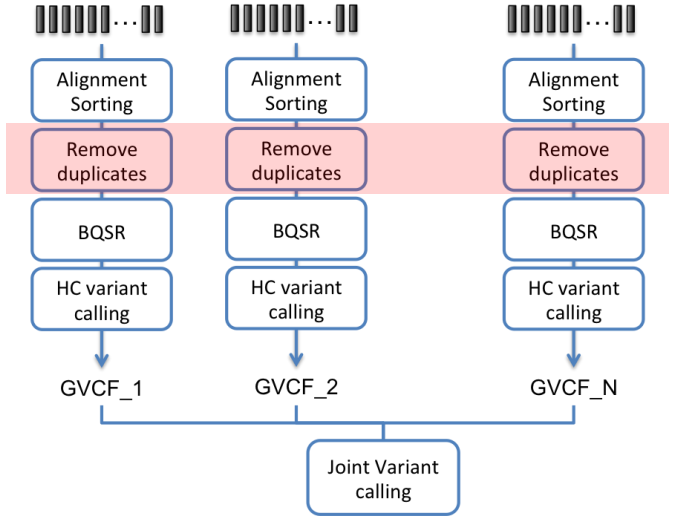
```
sentieon driver -t NUMBER_THREADS -i SORTED_BAM
  --algo LocusCollector --fun score_info SCORE.gz
sentieon driver -t NUMBER_THREADS -i SORTED_BAM
  --algo Dedup [--rmdup] --score_info SCORE.gz
  --metrics DEDUP_METRIC_TXT DEDUPED_BAM
```

sentieon driver: パイプラインを実行するための主なコマンド

-t: スレッド数
-i: 入力ファイル

--algo LocusCollector: 重複リード除去のために必要なリードの情報を集計
  --fun score_info: スコア関数（score_info）の指定

--algo Dedup: 重複リードにフラグを付ける
  --rmdup: フラグが付いたリードの除去
  --score_info: LocusCllectorの出力
  --metrics: 重複リード除去メトリクスの保存場所



DNAscopeを使用したJoint Calling③

gVCFファイルの作成（第１ステップ）

```
sentieon driver -t NUMBER_THREADS -r REFERENCE -i DEDUPED_BAM
[--interval INTERVAL_FILE] --algo DNAscope [-d dbSNP] [--pcr_indel_model none]
 --model DNASCOPE_MODEL/dnascope.model --emit_mode gvcf TMP_VARIANT_VCF
```

sentieon driver: パイプラインを実行するための主なコマンド

-t: スレッド数
-r:リファレンス配列の場所／名前
-i: 入力ファイル

--interval: BEDファイルの場所（オプション）

--algo DNAscope: DNAscopeでバリアントの検出
--pcr_indel_model: ライブラリ作成の際にPCRを使用していない場合はnone
--model:モデルファイルの場所
-d 既知バリアントデータ（オプション）
--emit mode: 出力するバリアントの指定（variant, confident, all, gvcf）

DNAscopeを使用したJoint Calling④

gVCFファイルの作成（第２ステップ）

```
sentieon driver -t NUMBER_THREADS -r REFERENCE --algo DNAModelApply --model
DNASCOPE_MODEL/dnascope.model -v TMP_VARIANT_GVCF VARIANT_GVCF
```

sentieon driver: パイプラインを実行するための主なコマンド

-t: スレッド数
-r:リファレンス配列の場所／名前

--algo DNAModelApply: DNAscopeによるバリアントの検出の第2ステップを実施
--model:モデルファイルの場所
-v: 第１ステップの出力データ

DNAscopeを使用したJoint Calling⑤

Joint Calling

```
sentieon driver -r REFERENCE --algo GVCFtyper -v s1_VARIANT_GVCF  -v s2_VARIANT_GVCF
-v s3_VARIANT_GVCF VARIANT_VCF
```

sentieon driver: パイプラインを実行するための主なコマンド

-r: リファレンス配列の場所／名前

--algo GVCFtyper: バリアントコールのためのアルゴリズム
  -v: 入力ファイル

その他の解析

Sentieonのドキュメントでは、さまざまな解析用のサンプルコードを提供

・生殖細胞変異検出（DNAseq, DNAscope）

・体細胞変異検出（TNseq, TNscope）

・RNAの変異検出

・ロングリードデータを使用した変異検出

・ハイブリッドアセンブルを利用した変異検出

など

お問い合わせ先：フィルジェン株式会社

TEL: 052-624-4388 (9:00～17：00)

FAX: 052-624-4389

E-mail: support@filgen.jp