Roche

Sequencing Solution: Technical Note April 2014

NimbleGen SeqCap Epi ターゲットエンリッチメントデータの評価方法

オープンソースツールのコマンド例文

目次

1. はじめに

Illumina社のシークエンシングシステムを用いたRoche NimbleGenのSeqCap Epiターゲットエンリッチメントのシークエンスデータは、しばしばオープンソース解析ツールを用いて解析されます。一般的なメチレーションおよびバリアント検出解析のワークフローは、シークエンスリードのクオリティ評価、リードフィルタリング、リファレンスゲノムへのマッピング、重複リードの除去、カバレージ統計評価、メチレーション解析、バリアントコール、バリアントフィルタリング、というステップで構成されます。本紙では、公的に利用可能かつ一般的なツールを利用したSeqCap Epi のデータ解析方法の一例を示します(図 1)。BAMファイル作成とメチレーション解析を実施するための生データのFASTQファイルを扱う方法に加え、NimbleGenより提供されるBEDファイルの取り扱い方法やPicardによる解析に必要なインターバルリストファイルをどのように作成するかというサポートワークフローについても記述しています。一方で、本紙で紹介されるツール以外にも同様の処理が可能なツールが存在しますので、本紙とは別の解析ワークフローで解析することも可能です。

本紙は、バイオインフォマティクス解析の経験のある研究者が、Roche NimbleGenで使用されている基本的な解析ワークフローをご理解いただける内容となっております。このため、実際にご自身のデータを解析する際には、それぞれの研究に最適なワークフローとなるように、慎重に追加オプションを検討する必要があります。また、本紙で例示した方法は効果的に動作することを確認していますが、*公的に利用可能なオープンソースソフトウェアツールは改変される可能性があり、そうした改変やその内容は弊社の管理下にはございません。このため、本紙で記述したツールを用いたときに得られる解析結果に対して、弊社では保証・責任を負いません。サポートやドキュメンテーションについては、各ツールの作成者からの情報をご参照ください。*

2. 方法

シークエンシング生データは、第三者により開発された無償のオープンソースツールにより様々な加工や解析を行うことが出来ます。本紙では最小限のデータ加工ステップとワークフローを説明しています。

ご自身の実験データに最適なワークフローを開発するためには、Coriell Instituteの細胞バンクから取得することのできるHapMapサンプルなどの標準/コントロールサンプルを用いた解析を実施することが理想です。HapMapサンプルの既知バリアント情報は、HapMap Project (*http://hapmap.ncbi.nlm.nih.gov*)、1000 Genomes Project (*http://www.1000genomes.org*)、GATKのリソースバンドルのような特別な情報集積ページ (*http://www.broadinstitute.org/gatk/download*) からダウンロードすることができます。

注) 本紙の例文で'SAMPLE'と記述されている箇所は、解析したい実ファイル名に置換してください。同様に'/path/to/…'という記述例についても、有効なパスに置換してください。カレントディレクトリはインプットファイルの場所であるとし、このディレクトリにアウトプットファイルやレポートファイルが作成されます。

本紙で複数行にわたって記載されている場合でも、各ステップのコマンドは一綴りで入力してください。このとき、ファイルパス中にはスペースは入力しませんが、それぞれのオプションの前後にはスペースが必要です (OS システムにもよりますが、Tab キーを使用したパスとファイル名の自動補完をご利用ください)。

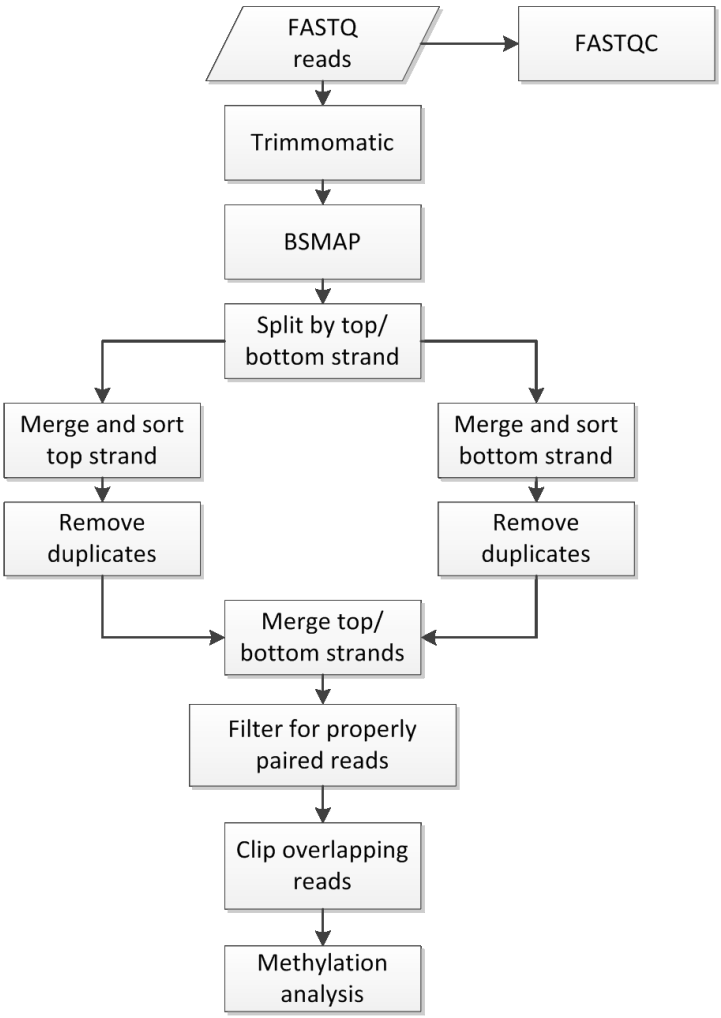

図 1: メチレーション解析ワークフローの基本スキーム

ツールについての概要

パッケージ (バージョン)	ツール	本資料中で使用する機能
bamUtil (1.0.10)	clipOverlap	BAM ファイル内のペアリードがオーバーラップしていた場合、クオリティがより高い配列を残すように切り取り。
BEDtools (2.17.0)	intersect	マップされたリード (BAM フォーマット) をターゲット領域のリスト (BED フォーマット) に参照させ、on-target リード率を算出。
	sort	ターゲット領域を並べ替え (BEDフォーマット)。
	merge	BEDファイル内のオーバーラップ領域を統合。
	genomecov	BEDファイル内のターゲット領域の総サイズを算出。
	slop	ターゲット領域の長さを拡張 (パディング)。
BisSNP (0.82.2)	BisulfiteCountCovariates	再キャリブレーションのためのデータファイルを作成。
	BisulfiteTableRecalibration	SNP コール前に再キャリブレートしたベースクオリティを算出。
	BisulfiteGenotyper	各ゲノム位置でのメチル化/非メチル化数の算出と、SNP コール。
	sortByRefAndCor.pl	VCF ファイルの名前とゲノム位置による並べ替え。
	VCFpostprocess	CpGs と SNPs をフィルタリング。
	vcf2bed6plus2.pl	VCF ファイルを、BED ファイルに変換。
BSMAP (2.74)	bsmap	インデックスが付与されたゲノムへシークエンスリードをマッピング。
	methratio.py	個々の塩基のメチル化率を算出。
FastQC (0.10.1)	fastqc	シークエンスリードのクオリティを評価 (1塩基単位でのクオリティプロット)。
GATK Framework (2.7-2)	DepthOfCoverage	シークエンシングのカバレッジを算出 (平均値、中央値、詳細情報)。
IGV	igv	BAMとBEDファイルのためのゲノムビューア (本紙では使用されていません)。
methylKit (0.9.2)	read	BSMAPメチレーション結果のインポート
	filterByCoverage	カバレージに従ったメチレーションデータのフィルタリング。
	unite	全てのサンプルによりカバーされているゲノム位置を統合。
	calculateDiffMeth	各ゲノム位置でのメチル化率とサンプル間での差を算出。
	get.methylDiff	メチレーション変化の絶対値、q値、領域タイプ (hypo, hyper, 全て) によりサンプル間のメチル化率の差をフィルタリング。
	tileMethylCounts	ゲノム上のタイリング領域内またはスライディングウィンドウ内のメチル化/非メチル化塩基数を集計。
Picard (1.98)	AddOrReplaceReadGroups	SAMまたはBAMファイルへリードグループ情報を追加。
	CollectInsertSizeMetrics	インサートサイズの平均および標準偏差を推定。インサートサイズ分布をプロット。
	CalculateHsMetrics	ターゲットエンリッチメントのパフォーマンスを評価。
	MarkDuplicates	重複リードを除去またはチェック。ペア、非ペア、Optical duplicateのリード数をレポート。
	CollectAlignmentSummaryMetrics	BAMファイルからマッピング結果概要レポートを出力。
	SamToFastq	SAMやBAMファイルからFASTQファイルを作成。
SAMtools (0.1.18)	sort	BAMファイル内の情報を並べ替え。
	index	並べ替えられたBAMファイルからインデックスファイルを作成。
	view	ヘッダやリードデータを視覚化または抽出。
	mpileup	BAMファイル内のバリアントをコール。
	BCFtools ⇒ view	VCFとBCF間でフォーマットを変換。
	vcfutils ⇒ varFilter	検出されたバリアントのフィルタリング。
seqtk (1.0-r31)	sample	FASTQファイル (s) からリードをランダムに抽出。
Trimmomatic (0.30)	illuminaclip	クオリティによるリードのトリミング。
Bamtools (2.3.0)	split	BAMファイルを分割。
	merge	BAMファイルを統合。
	filter	BAMファイルから、paired readsとしてマッピングされなかったreadsを除外。

表 1: 本テクニカルノートで使用した解析ツール一覧。 本紙では各括弧書きのバージョンのツールを使用して動作を確認しています。*Reference* に記載のリンクからインストールの方法と各コマンドオプションの説明をご確認ください。これらのツールは Linux システムで動作確認をしていますが、MacOS でも使用することができます。

FASTQ ファイルの解凍

FASTQ ファイルが圧縮されている場合 (拡張子.gz) には、それらを解凍する必要があります。

ソフトウェア／モジュール	gunzip
インプット	SAMPLE_R1.fastq.gz SAMPLE_R2.fastq.gz
アウトプット	SAMPLE_R1.fastq SAMPLE_R2.fastq

```
gunzip -c SAMPLE_R1.fastq.gz > SAMPLE_R1.fastq
gunzip -c SAMPLE_R2.fastq.gz > SAMPLE_R2.fastq
```

BAM ファイルからの FASTQ ファイルの作成

異なるパイプラインを用いてリードを再マップしたいが元の FASTQ ファイルを入手できない場合、BAM ファイルから FASTQ ファイルを作成することができます。

ソフトウェア／モジュール	Picard ／ SamToFastq
インプット	SAMPLE_file.bam
アウトプット	SAMPLE_R1.fastq SAMPLE_R2.fastq

```
java -Xmx4g -Xms4g -jar /path/to/Picard/SamToFastq.jar
VALIDATION_STRINGENCY=LENIENT INPUT=SAMPLE_file.bam FASTQ=SAMPLE_R1.fastq
SECOND_END_FASTQ=SAMPLE_R2.fastq
```

重複リードやクオリティの低いリード除去などの操作や、ベースクオリティの再キャリブレーションを実施していない場合に限り、作成されたFASTQファイルは元のファイルを復元しています。

FASTQ ファイルからの一部のリードデータの抽出

ランダムサンプリングはデータセットごとのリード数が異なるデータの比較を行う場合の正規化方法として有用な方法です。アダプタートリミングやクオリティフィルタリング処理の前にリードを抽出するか、それらの処理の後に高品質で最短リード長以上のリードのみのデータをサンプリングするかについては、実験目的に合わせてご自身でご判断ください。ペアエンドのリードデータの場合、その 2 つのファイルを同じシード値 (-s)、リード数に設定することが重要です。この seqtk アプリケーションは圧縮 (.gz) された FASTQ ファイルにも適用することができますが、アウトプットファイルは圧縮されない FASTQ ファイルとなります。

ソフトウェア／モジュール	seqtk ／ sample
インプット	SAMPLE_R1.fastq SAMPLE_R2.fastq
アウトプット	SAMPLE_subset_R1.fastq SAMPLE_subset_R2.fastq
/path/to/seqtk sample -s 10000 SAMPLE_R1.fastq 10000000 > SAMPLE_subset_R1.fastq /path/to/seqtk sample -s 10000 SAMPLE_R2.fastq 10000000 > SAMPLE_subset_R2.fastq	

上記の例ではペアエンドの FASTQ ファイルから、ランダムな 10M (1000 万) リードを抽出しています。同じシード値 (-s) を適用することで FASTQ レコードの順序が維持され、マッピング等にペアエンドの情報を残したまま使用できるようになります。

注) seqtk は抽出するリード数に比例した RAM を必要とします。

シークエンスリードクオリティの評価

マッピング結果の解析には時間が掛かるため、その操作を実施する前に fastqc ツールにより生データの塩基あたりのシークエンスクオリティプロットとレポートを作成し、マッピング処理に進めるかどうかの判断をすると効率的です。fastqc ツールは圧縮された FASTQ ファイルにも圧縮されていない FASTQ ファイルにも使用することができます。

ソフトウェア／モジュール	FastQC ／ fastqc
インプット	SAMPLE_R1.fastq SAMPLE_R2.fastq
アウトプット	SAMPLE_R1_fastqc.zip SAMPLE_R2_fastqc.zip
/path/to/fastqc --nogroup SAMPLE_R1.fastq SAMPLE_R2.fastq	

.zip ファイルと圧縮されていないディレクトリがそれぞれの SAMPLE インプットファイルに対して作成されます。これらのフォルダには fastqc_report.html という名前の HTML 形式のレポートが含まれており、インターネットブラウザで見ることができます。様々なシークエンス結果における QC レポートの例が下記の URL に掲載されています。
http://www.bioinformatics.babraham.ac.uk/projects/fastqc/good_sequence_short_fastqc/fastqc_report.html
http://www.bioinformatics.babraham.ac.uk/projects/fastqc/bad_sequence_fastqc/fastqc_report.html

アダプター配列のトリミングとクオリティフィルタリング

リードをマッピングする前に、リード内のシークエンシング用アダプター配列をリードから除去 (トリミング) し、シークエンシングクオリティによるトリミングまたはフィルタリングを実施してください。Trimmomatic アプリケーションはこの両方の処理を実施することができます。

ソフトウェア／モジュール	Trimmomatic ／ illuminaclip
インプット	SAMPLE_R1.fastq SAMPLE_R2.fastq
アウトプット	SAMPLE_R1_trimmed.fq SAMPLE_R1_unpaired.fq SAMPLE_R2_trimmed.fq SAMPLE_R2_unpaired.fq
java -Xms4g -Xmx4g -jar /path/to/trimmomatic.jar PE -threads NumProcessors -phred33 SAMPLE_R1.fastq SAMPLE_R2.fastq SAMPLE_R1_trimmed.fq SAMPLE_R1_unpaired.fq SAMPLE_R2_trimmed.fq SAMPLE_R2_unpaired.fq ILLUMINACLIP:/path/to/adapters.fa:2:30:10 LEADING:20 TRAILING:20 SLIDINGWINDOW:5:20 MINLEN:75	

Trimmomatic アプリケーションでは 4 つのファイルが作成されます。SAMPLE_R1_trimmed.fq と SAMPLE_R2_trimmed.fq は、アダプタートリミングとクオリティフィルタリング後もペアであるリード (ペアリード) で構成されています。他方、SAMPLE_R1_unpaired.fq と SAMPLE_R2_unpaired.fq には、クオリティが悪いか、リード長が 75bp 以下となってしまったためにペアが成立しなかったリード (シングルトンリード) が記録されています。

シングルトンリードを必ずしも解析から排除する必要はありませんが、大抵のアライメントアプリケーションはシングルトンリードをペアリードとは別個に解析する必要があり、その後の解析を複雑化する可能性があります。また、もし解析ワークフローにマッピングされたペアリードのみが対象のフィルタリングステップが含まれている場合には、シングルトンリードをマッピングに用いても解析データには反映されません。

上記例に示したパラメーターは、2x100bpのシークエンスリードの解析時に最適な数値となっています。この条件を一般的なデータに適用した場合、リードの約90%がこれらのクオリティフィルターをパスすると予想できます。このパーセンテージを上げたい場合にはTrimmomaticのパラメーター (特にトリミング後の最短リード長を決定するMINLENパラメーター) を調節してください。

リードのマッピング

バイサルファイト変換済みシークエンスリードをリファレンスゲノムにマッピングするアライメントソフトウェアは、ここでご紹介するものに限らず様々なものが利用されています。リード長とリードのタイプ (100bp ペアエンドリード、76bp シングルエンドリードなど)、ゲノムサイズや GC 含量、利用可能な計算リソースによりアライメントソフトウェアを選択してください。

BSMAP は 100bp のペアエンドリードの場合にアライメントに掛かる時間と全体的なアライメントの質とのバランスの良さが認められています。BSMAP では、リファレンスゲノムへのインデックスは BSMAP の実行時に付与されますので、他のアライメントソフトウェアのように予めインデックスを付与しておく必要はありません。BSMAP により SAM ファイルが作成されますので、ワークフローの次の工程に進める前にこれを BAM ファイルに変換する必要があります。

ソフトウェア／モジュール	bsmap Picard ／ AddOrReplaceReadGroups
インプット	ref.fa SAMPLE_R1.fastq SAMPLE_R2.fastq
アウトプット	SAMPLE.bam
リードのマッピング /path/to/bsmap -r 0 -s 16 -n 1 -a SAMPLE_R1.fastq -b SAMPLE_R2.fastq -d path/to/ref.fa -p NumProcessors -o SAMPLE.sam	
リードグループの追加、BAMファイルへの変換 java -Xmx4g -Xms4g -jar /path/to/Picard/AddOrReplaceReadGroups.jar VALIDATION_STRINGENCY=LENIENT INPUT=SAMPLE.sam OUTPUT=SAMPLE.bam CREATE_INDEX=TRUE RGID=SAMPLE RGLB=SAMPLE RGPL=illumina RGSM=SAMPLE RGPU=platform_unit	

上記例は、両鎖にユニークにリードをマッピングするためのパラメーターです。デフォルトの設定ではマップされなかったリードについてはレポートされませんので、もし、こうしたリードについても解析する場合は、 -u オプションを使用してください。

BSMAPのアウトプットファイル (SAMファイル) 内のリードには、リードグループ情報 (キャプチャーIDやリードグループライブラリID、リードグループサンプルIDなど) が割り当てられていませんので、上記の例文では、こうした情報をSAMファイルからBAMファイルに変換すると同時に追加しています。リードグループライブラリIDやリードグループサンプルIDは、複数のデータを統合する場合の識別のために使用します。platform unitとはフローセルとレーンを特定するための情報で、一般的にはFASTQのIDのコロン (:) で区切られた初めの4つの領域 (太字箇所) で把握することができます。
(例:**@DJDPWKN1 : 239 : C3DL6ACXX : 2**: 1101:1181:2050 1:N:0:CGATGT)

リファレンスゲノムファイルについては、注意すべき点がいくつかあります。まず、FASTAフォーマットのゲノム配列は、chr1, chr2, ..., chr10, chr11, … chrX, chrY, chrM, chr1_randomなどのように染色体の核型順にソートされている必要があります。本紙では、'hg19.fa' のような実際のリファレンスゲノムファイル名の代わりに 'ref.fa' と記載しています。また、SeqCap Epi製品はラムダDNAをスパイクインコントロールとして使用していることから、GenBank accession NC_001416の配列もリファレンスゲノムファイルに追加しておく必要があります。このラムダDNA配列にマッピングされたリードを解析することで、バイサルファイト処理の効率を調べることができます。さらに、ヒトリファレンスゲノムにユニークにマップされるリードを解析する場合、Y染色体上の擬似常染色体領域 (PAR, pseudo-autosomal region) をNでマスクしたリファレンスゲノムファイルを用い、リードをPARと同様の配列を持つX染色体にマップさせることをお勧めいたします。

ヒトを検体とした実験で100bpペアリードのデータの場合、一般的に90%以上のリードがヒトリファレンスゲノムにマップされると期待できます。ヒト以外の生物種の場合においては、ゲノムサイズやGC含量、リピート配列、リファレンスゲノムのクオリティなどのような様々なファクターにより、リードのマッピング率は変動すると考えられます。

ソーティングと重複リードの除去

バイサルファイト変換したシークエンスデータを取り扱う際には、変換処理を行わない標準的なゲノムキャプチャーの場合と比較して、さらなる課題が生じます。これはバイサルファイト処理により非メチル化 C が T に変換されることで、top 鎖と bottom 鎖 (ワトソン鎖とクリック鎖) が元のゲノム配列に対する相補配列とならなくなるためです (図 2)。

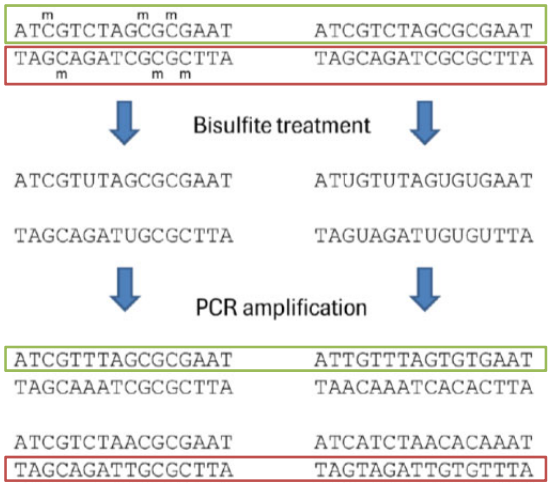

図 2: バイサルファイト処理により生じる非相補的な配列。 メチル化 C はバイサルファイト処理による影響を受けませんが、非メチル化 C は U に変換され、その後の PCR により T に置換されます。元のゲノム配列は相補的ですが、この分子それぞれから変換された分子は互いに相補的ではありません (四角枠内)。また、図中の左右の配列のように、メチル化状態が異なると、変換後の配列も異なります。

重複リードを除去するための標準的なツールである Picard の MarkDuplicates には、バイサルファイトシークエンスや非相補的な top 鎖と bottom 鎖を扱うオプションがありません。このため、重複リードを除去するには、マップされたリードを top 鎖と bottom 鎖に分ける必要があり、重複リードを除去した後にもう一度マージします。BSMAP のアライメントでは、マップされた箇所が top 鎖か bottom 鎖か、フォワードリードかリバースリードか、を示すタグが BAM ファイルに含まれていますので、このタグを利用してデータ処理を行います。

ソフトウェア／モジュール	bamtools ⇒ split bamtools ⇒ merge SAMtools ⇒ sort Picard ⇒ MarkDuplicates
インプット	SAMPLE.bam
アウトプット	SAMPLE.TAG_ZS_++.bam SAMPLE.TAG_ZS_+-.bam SAMPLE.top.bam SAMPLE.top.sorted.bam SAMPLE.top.rmdups.bam SAMPLE.top.rmdups_metrics.txt SAMPLE.TAG_ZS_-+.bam SAMPLE.TAG_ZS_--.bam SAMPLE.bottom.bam SAMPLE.bottom.sorted.bam SAMPLE.bottom.rmdups.bam SAMPLE.bottom.rmdups_metrics.txt SAMPLE.rmdups.bam

BAMファイルの分離

```
/path/to/bamtools split -tag ZS -in SAMPLE.bam
```

ストランドBAMファイルの統合

```
/path/to/bamtools merge -in SAMPLE.TAG_ZS_++.bam -in SAMPLE.TAG_ZS_+-.bam -out SAMPLE.top.bam
/path/to/bamtools merge -in SAMPLE.TAG_ZS_-+.bam -in SAMPLE.TAG_ZS_--.bam -out SAMPLE.bottom.bam
```

BAMファイルの並べ替え

```
/path/to/samtools sort SAMPLE.top.bam SAMPLE.top.sorted
/path/to/samtools sort SAMPLE.bottom.bam SAMPLE.bottom.sorted
```

重複の除去

```
java -Xmx4g -Xms4g -jar /path/to/Picard/MarkDuplicates.jar
VALIDATION_STRINGENCY=LENIENT INPUT=SAMPLE.top.sorted.bam
OUTPUT=SAMPLE.top.rmdups.bam METRICS_FILE=SAMPLE.top.rmdups_metrics.txt
REMOVE_DUPLICATES=true ASSUME_SORTED=true CREATE_INDEX=true

java -Xmx4g -Xms4g -jar /path/to/Picard/MarkDuplicates.jar
VALIDATION_STRINGENCY=LENIENT INPUT=SAMPLE.bottom.sorted.bam
OUTPUT=SAMPLE.bottom.rmdups.bam METRICS_FILE=SAMPLE.bottom.rmdups_metrics.txt
REMOVE_DUPLICATES=true ASSUME_SORTED=true CREATE_INDEX=true
```

重複リードを除去したBAMファイルの統合

```
/path/to/bamtools merge -in SAMPLE.top.rmdups.bam -in SAMPLE.bottom.rmdups.bam -out SAMPLE.rmdups.bam
```

「重複の除去」のステップでは、REMOVE_DUPLICATES=falseを使うこともできます。このように変更すると、重複リードはマークされますが除去はされません。重複リードを含んだデータを更に解析すると、この後の処理で作成されるファイルサイズが大きくなりますが、さらに下流の解析時に重複リードを含める/含めないを切り替えて結果を確認することができます。

ペア、非ペア、重複リード数はSAMPLE.strand.rmdups_metrics. txtファイルに記載されます。このアウトプット結果概要の詳細については *http://picard.sourceforge.net/picard-metric-definitions.shtml* をご参照ください。このファイルでは、シークエンスの類似性とシークエンスクラスター距離に従って‘optical duplicates’がレポートされますが、このファイル中の重複リードの割合 (PERCENT_DUPLICATION) の計算では、これら‘optical duplicates’のリード数は計算に使用されておらず、ペアおよび非ペアの重複リード数からの計算結果となります。

適切なペアリードのフィルタリング

バイサルファイトリードはマッピングが難しいことから、ペアリードの両方がマップされ、ペアの方向が正しく、ライブラリのインサートサイズと位置関係が矛盾しないというような適切なリードペアのみを用いて解析することにより、全体のデータクオリティを向上させることができます。一方で、こうしたフィルタリングを適用すると構造的なバリアントは検出されなくなりますので、このようなバリアントの検出を目的とする場合には、本紙では記載していない別の解析ワークフローをセットアップする必要があります。

ソフトウェア／モジュール	bamtools ⇒ filter
インプット	SAMPLE.rmdups.bam
アウトプット	SAMPLE.filtered.bam

```
/path/to/bamtools filter -isMapped true -isPaired true -isProperPair true -forceCompression -in SAMPLE.rmdups.bam -out SAMPLE.filtered.bam
```

ヒトを検体とした SeqCap Epi の実験では、BSMAP によりマッピングされたリードのうち約5%がこのフィルタリングにより除かれます。

オーバーラップリードの切り取り

180～220bp の典型的なインサートサイズをもつ次世代シークエンサー用のライブラリを 100bp のペアエンドでシークエンスすると、大部分のペアエンドリードは互いに一部がオーバーラップします。こうしたオーバーラップ領域の塩基を二重に数えてしまうと、特にカバレージの低い塩基のメチル化率を正確に評価できなくなります。この問題を回避するために、bamUtil の clipOverlap モジュールのように、片方あるいは両方のリードを‘soft-clip’してペアリードがオーバーラップしなくなるよう処理する必要があります。

ソフトウェア／モジュール	bamUtil ⇒ clipOverlap
インプット	SAMPLE.filtered.bam
アウトプット	SAMPLE.clipped.bam

```
/path/to/bam clipOverlap --stats --in SAMPLE.filtered.bam --out SAMPLE.clipped.bam
```

clipOverlap の‘soft-clip’は、ペアリードをチェックしオーバーラップが確認された場合には、オーバーラップ領域での平均クオリティが低い配列を切り取り、必要に応じてアライメント位置を上書きします。詳細は *http://genome.sph.umich.edu/wiki/BamUtil:_clipOverlap* をご参照ください。

ライブラリの平均インサートサイズやライブラリのサイズ分布に依存してこの処理が適用されるリードの数は変動します。*インサートサイズ分布の評価*の項をご参照ください。

BAM ファイルへのインデックス付与

BAM ファイルを処理する多くのツールでは、処理速度を向上させ計算を効率化させるために BAM ファイルにインデックスを付与しておく必要があります。Picard ツールでは、コマンドラインに CREATE_INDEX=true というパラメーターを入れておくと自動的にインデックスが付与されます。その他の BAM ファイルを作成するツールでは、下記のように SAMtools により各自で処理する必要があります。

ソフトウェア／モジュール	SAMtools ⇒ index
インプット	SAMPLE.bam
アウトプット	SAMPLE.bam.bai

```
/path/to/samtools index SAMPLE.bam
```

このコマンドでは元のファイル名に.bai という拡張子を付けたファイルが作成されます。

マッピング結果概要 (Picard)

マッピング結果概要は SAMPLE.bam と ref.fa ファイルから Picard を用いて作成することができます。

ソフトウェア／モジュール	Picard ⇒ CollectAlignmentSummaryMetrics
インプット	ref.fa SAMPLE.bam
アウトプット	SAMPLE_picard_alignment_metrics.txt
java -Xmx4g -Xms4g -jar /path/to/Picard/CollectAlignmentSummaryMetrics.jar METRIC_ACCUMULATION_LEVEL=ALL_READS INPUT=SAMPLE.bam OUTPUT=SAMPLE_picard_alignment_metrics.txt REFERENCE_SEQUENCE=/path/to/ref.fa VALIDATION_STRINGENCY=LENIENT	

ここでは VALIDATION_STRINGENCY=LENIENT と設定することにご注意ください。これは、BAM ファイルを扱う全てのツールが、BAM の仕様に完全に準拠してアウトプットファイルを作成しているわけではないためで、このパラメーターを用いなければ、Picard ツールは非準拠の記述を 1 つでも検出すると停止してしまいます。アウトプットファイルの詳細については *http://picard.sourceforge.net/picard-metric-definitions.shtml* をご参照ください。

Picard インターバルリストの作成

Picardインターバルリストとは、ゲノムを区間に分けて詳細に記述したファイルで、リファレンスゲノムと位置情報のセット (鎖情報と名前情報) が各区間で記載された、SAMファイルのヘッダに似た情報を含んだファイルです。Picardの‘target interval’はSeqCap Epi Enrichment Kit付属のデザインファイル中の‘primary target’BEDファイルに、Picardの‘bait interval’は同‘capture target’BEDファイルに相当しています。このPicardインターバルリストは、PicardのCalculateHsMetricsコマンドの実行に必要です。

注) SeqCap Epi 製品に添付されるデザインファイルですが、‘capture target’に相当するカバレージターゲット BED ファイルが 1 ファイルしか提供されない場合があります。このような場合には、以降のコマンドでの‘primary target’と‘capture target’のどちらにもそのファイルを指定してください。

■ Picard ターゲットインターバルリストの作成

先ず、SAMPLE.bam ファイルからヘッダを抽出するために SAMtools の view コマンドを使用します。次に、Linux の gawk コマンドで SAMPLE_primary_target.bed ファイルからインターバルリストの本体を抽出し書式を整えます。最後に、cat コマンドで二つの要素を適切な書式のインターバルリストとして結合させます。

ソフトウェア／モジュール	SAMtools ⇒ view cat gawk
インプット	SAMPLE.bam DESIGN_primary_target.bed
アウトプット	DESIGN_target_intervals.txt
Picardのインターバルリストのヘッダの作成 /path/to/samtools view -H SAMPLE.bam > SAMPLE_bam_header.txt	
Picardのターゲットインターバルリストの本文の作成 cat DESIGN_primary_target.bed | gawk '{print $1 "¥t" $2+1 "¥t" $3 "¥t+¥tinterval_" NR}' > DESIGN_target_body.txt	
ヘッダと本文の連結 cat SAMPLE_bam_header.txt DESIGN_target_body.txt > DESIGN_target_intervals.txt	

DESIGN_target_intervals.txtファイルはPicardのCalculateHsMetricsコマンドに使用します。

■ Picard ベイトインターバルリストの作成

先ず、SAMPLE.bam ファイルからヘッダを抽出するために SAMtools の view コマンドを使用します。次に、Linux の gawk コマンドで SAMPLE_capture_target.bed ファイルからインターバルリストの本体を抽出し書式を整えます。最後に、cat コマンドで二つの要素を適切な書式のインターバルリストとして結合させます。

ソフトウェア／モジュール	SAMtools ⇒ view cat gawk
インプット	SAMPLE.bam DESIGN_capture_target.bed
アウトプット	DESIGN_bait_intervals.txt
Picardのインターバルリストのヘッダの作成 /path/to/samtools view -H SAMPLE.bam > SAMPLE_bam_header.txt	
Picardのベイトインターバルリスト本文の作成 cat DESIGN_capture_target.bed | gawk '{print $1 "¥t" $2+1 "¥t" $3 "¥t+¥tinterval_" NR}' > DESIGN_bait_body.txt	
ヘッダと本文の連結 cat SAMPLE_bam_header.txt DESIGN_bait_body.txt > DESIGN_bait_intervals.txt	

DESIGN_bait_intervals.txt ファイルは Picard の CalculateHsMetrics コマンドに使用します。

Hybrid Selection (HS) 解析結果概要

CalculateHsMetrics コマンドはターゲットエンリッチメントリードのクオリティを評価する結果概要を出力します。このプロセスを実施する前に *BisSNP を用いた SNP/メチレーションの検出*の各項の処理が必要な場合があります。

ソフトウェア／モジュール	Picard ⇒ CalculateHsMetrics
インプット	ref.fa {indexed} SAMPLE.clipped.bam {indexed} DESIGN_target_intervals.txt DESIGN_bait_intervals.txt
アウトプット	SAMPLE_picard_hs_metrics.txt
java -Xmx4g -Xms4g -jar /path/to/Picard/CalculateHsMetrics.jar BAIT_INTERVALS=DESIGN_bait_intervals.txt TARGET_INTERVALS=DESIGN_target_intervals.txt INPUT=SAMPLE.clipped.bam OUTPUT=SAMPLE_picard_hs_metrics.txt METRIC_ACCUMULATION_LEVEL=ALL_READS REFERENCE_SEQUENCE=/path/to/ref.fa VALIDATION_STRINGENCY=LENIENT TMP_DIR=.	

PicardのCalculateHsMetricsのBAIT_INTERVALSとTARGET_INTERVALSに、同一のインターバルリストのみをインプットする場合と異なるファイルをインプットする場合とでは、結果に違いが生じます。この違いはprimary target とcapture targetの差の大きさに依存します。SeqCap Epi Enrichment KitにBEDファイルが1つしか提供されていない場合には、そのファイルから作成したPicardインターバルリストをBAIT_INTERVALSとTARGET_INTERVALSの両方に使用してください。アウトプットファイルの詳細については *http://picard.sourceforge.net/picard-metric-definitions.shtml* をご参照ください。

インサートサイズ分布の評価

ランダムな物理的断片化とその後のサイズセレクションによりシークエンスサンプルが調製されていることから、通常は各リードのインサートは一定の範囲内に様々なサイズで存在します。しかし、その分布が大きくゆがんでいる場合、on-target 率や、少なくとも 1 本のリードでカバーされる塩基の割合に悪影響を与える可能性があります。このレポートを作成するには、*適切なペアリードのフィルタリング*で作成した SAMPLE.filtered.bam ファイルを適用してください。

ソフトウェア／モジュール	Picard ⇒ CollectInsertSizeMetrics
インプット	SAMPLE.filtered.bam
アウトプット	SAMPLE_picard_insert_size_metrics.txt SAMPLE_picard_insert_size_plot.pdf
java -Xmx4g -jar /path/to/Picard/CollectInsertSizeMetrics.jar VALIDATION_STRINGENCY=LENIENT HISTOGRAM_FILE=SAMPLE_picard_insert_size_plot.pdf INPUT=SAMPLE.filtered.bam OUTPUT=SAMPLE_picard_insert_size_metrics.txt	

R がインストールされいてる場合には、SAMPLE_picard_insert_size_metrics.txt からサンプル間のインサートサイズ分布プロットを PDF ファイルとして作成することもできます (SAMPLE_picard_insert_size_plot.pdf)。このファイルの説明については *http://picard.sourceforge.net/picard-metric-definitions.shtml* をご参照ください。

ターゲット領域へのパディング

ハイブリダイゼーションを原理としたターゲットエンリッチメントでは、ターゲット領域の一部を含むライブラリフラグメントがキャプチャーされるために、ターゲット領域に隣接した領域にもシークエンスカバレージが得られます。ターゲットに隣接している off-target リードの量を評価する必要がある場合には、on-target 率の評価の前に、この項で説明するパディング処理をターゲット領域にしておく必要があります。パディング、並べ替え、統合の 3 つの処理は、下記のコマンドにより実行することができます。

ソフトウェア／モジュール	BEDtools ⇒ slop BEDtools ⇒ sort BEDtools ⇒ merge
インプット	DESIGN_capture_target.bed chromosome_sizes.txt
アウトプット	DESIGN_padded_capture_target.bed
パディング、ソーティング、重なり合うあるいは隣り合った領域の位置情報の統合 /path/to/bedtools slop -i DESIGN_capture_target.bed -b 100 -g chromosome_sizes.txt | /path/to/bedtools sort -i - | /path/to/bedtools merge -i - > DESIGN_padded_capture_target.bed	

最初のステップの -b オプションに指定された値は、ターゲット領域の両側に追加する塩基数を示しています。上記の例では全てのターゲット領域の両側に100塩基が付加されます。インプットファイルである chromosome_sizes.txtファイルは、ChrName<tab>ChrSizeのフォーマットである必要があり (*例* 'chr1 249250621')、インプットであるBEDファイルに存在する全ての染色体に対する情報が記載されている必要があります。'-i-' は、前のコマンドからの標準出力を次のコマンドが引き継ぐことを指示するためのものです。

ターゲット領域の総サイズの確認

ここでインプットファイルとして使用する chromosome_sizes.txt ファイルについては、*ターゲット領域へのパディング*をご参照ください。

ソフトウェア／モジュール	BEDtools ⇒ genomecov grep cut
インプット	chromosome_sizes.txt DESIGN.bed
アウトプット	{サイズが表示されます}
/path/to/bedtools genomecov -i DESIGN.bed -g chromosome_sizes.txt -max 1 | grep -P "genome¥t1" | cut -f 3	

または、gawk のみで実施することも可能です:

```
cat DESIGN.bed | gawk -F'¥t' 'BEGIN{SUM=0}{SUM+=$3-$2}END{print SUM}'
```

On-Target リード数の確認

On-Target リード (1 塩基以上ターゲット領域とオーバーラップしているマップされたリードの数) は、BEDtools の intersect を用いて、ここまでのプロセスで作成された様々な BAM ファイルから算出することができます。

ソフトウェア／モジュール	BEDtools ⇒ intersect wc
インプット	SAMPLE.bam DESIGN_primary_target.bed or DESIGN_capture_target.bed
アウトプット	{on-target リード数が表示されます}
/path/to/bedtools intersect -bed -abam SAMPLE.bam -b DESIGN_primary_target.bed | wc -l or /path/to/bedtools intersect -bed -abam SAMPLE.bam -b DESIGN_capture_target.bed | wc -l	

このコマンドでは、ターゲット領域と 1 塩基以上オーバーラップする全てのリード ("on-target リード") の数を出力します。この後の解析のために、アウトプットをファイルとして保存することもできます。on-target リードの割合 (on-target 率) を算出するには、重複リードを除いたマップされたリードの総数でこの数を割ります。重複リードを除いたマップされたリードの総数を調べるには、*マッピング結果概要 (Picard)* をご参照ください。

カバレージの確認

Primary target または capture target 領域のカバレージは、GATK の DepthOfCoverage コマンドで算出することができます。

ソフトウェア／モジュール	GATK ⇒ DepthOfCoverage
インプット	ref.fa {indexed} SAMPLE.clipped.bam {indexed} DESIGN_primary_target.bed or DESIGN_capture_target.bed
アウトプット	SAMPLE_gatk_primary_target_coverage.sample_summary or SAMPLE_gatk_capture_target_coverage.sample_summary {この他にもアウトプットファイルが作成されます}
java -Xmx4g -Xms4g -jar /path/to/GATKFramework/GenomeAnalysisTK.jar -T DepthOfCoverage -R /path/to/ref.fa -I SAMPLE.clipped.bam -o SAMPLE_gatk_primary_target_coverage -L DESIGN_primary_target.bed -ct 1 -ct 10 -ct 20 or java -Xmx4g -Xms4g -jar /path/to/GATKFramework/GenomeAnalysisTK.jar -T DepthOfCoverage -R /path/to/ref.fa -I SAMPLE.clipped.bam -o SAMPLE_gatk_capture_target_coverage -L DESIGN_capture_target.bed -ct 1 -ct 10 -ct 20	

アウトプットの sample_summary ファイルには、-L で設定したターゲット領域 (興味対象領域) 上の平均およびメディアンカバレージの他、-ct で設定したカバレージ以上でシークエンスされた領域の割合についても記載されています。より詳細な情報は、*http://www.broadinstitute.org/gatk/gatkdocs/org_broadinstitute_sting_gatk_walkers_coverage_DepthOfCoverage.html* をご参照ください。

BSMAP を用いたメチル化率の算出

検体ゲノムの C 塩基のメチル化率は、*オーバーラップリードの切り取り*で作成された BAM ファイルから methratio.py スクリプトで算出することができます。

ソフトウェア／モジュール	BSMAP ⇒ methratio.py
インプット	ref.fa SAMPLE.clipped.bam
アウトプット	SAMPLE.methylation_results.txt
メチル化率の算出 python /path/to/methratio.py -d hg19.fa -s /path/to/samtools -m 1 -z -i skip -o SAMPLE.methylation_results.txt SAMPLE.clipped.bam	

-m オプションを利用してカバレージによりフィルタリングして結果を出すことができます。この値を高く設定するには (例えば、'-m 5')、より多くのリードがターゲット領域をカバーしている必要があります。 -z オプションはメチル化率が'0' (ゼロ) のゲノム位置についてもレポートするために必要です。'-i skip' により、CからTへの塩基置換 (SNP) の可能性があるゲノム位置を無視するように指示しています。このようなSNPの可能性があるゲノム位置の解析には*BisSNPを用いたSNP/メチレーションの検出*の方法を使用してください。全てのCに関してメチル化率が算出されることから、このステップで作成されるアウトプットファイルはサイズが大きく、数GBになる場合があります。追加オプションとして -c を用いれば、解析範囲を限定させることができます (例えば '-c chr1' のように記述すると、chr1についてのみ処理します)。このオプションと -m オプションとを組み合わせることで、より小さく合理的なサイズの結果ファイルを作成することができます。

BSMAP を使用したバイサルファイト変換効率の算出

算出されるメチル化率は、バイサルファイト変換効率が高い場合には信頼できると考えることができますが、そうでない場合にはメチル化率を過大評価してしまう可能性があります。バイサルファイト変換効率を評価する最も良い方法は、メチル化されていないインプット DNA を用いて非変換の C (バイサルファイト変換されなかったとみなされる塩基) の数を算出することです。

ヒトおよび動物の場合はミトコンドリアゲノムが、植物の解析の場合はミトコンドリアとクロロプラストゲノムの両方がこのアプローチの対象として利用できる可能性があります。しかしながら、これらのオルガネラのゲノムは常に十分に解析されているとは限りませんので、Roche NimbleGen では、ラムダファージ DNA を供給し、スパイクインコントロールとしての使用をプロトコールに組み込んでいます。全ての SeqCap Epi Enrichment Kit 製品のキャプチャープローブプール溶液にはラムダファージの 4500 から 6500bp (GenBank Accession NC_001416) のゲノム領域をキャプチャーするためのプローブが含まれています。マッピングの際には、リードをラムダゲノム配列にもマッピングさせるために、実験生物種のリファレンス配列にラムダゲノム配列を追加しておく必要があります。ラムダゲノムにマップされたリード上の C の数を実験全体のバイサルファイト変換効率の算出のために利用します。

ソフトウェア／モジュール	BSMAP ⇒ methratio.py
インプット	ref.fa SAMPLE.clipped.bam
アウトプット	SAMPLE.NC_001416.methylation_results.txt
メチル化率の算出 python /path/to/methratio.py -d hg19.fa -s /path/to/samtools -m 1 -z -i skip -c NC_001416 -o SAMPLE.NC_001416.methylation_results.txt SAMPLE.clipped.bam	

'-c NC_001416' はメチレーション解析をラムダゲノムにのみ限定させるためのものです。このスパイクインコントロールを添加していない実験の場合には、このスイッチをミトコンドリアゲノム名 (例. chrM) かクロロプラストゲノム名 (例. chrC) に置換してください。解析完了後、アウトプットファイルであるSAMPLE.NC_001416.methylation_results.txtをMicrosoft Excel などのスプレッドシートアプリケーションで開き、(必要であればキャプチャー領域以外の行を削除し)、下記の計算式で変換効率を算出します:

変換効率 = 1 - (sum (C_count) / sum (CT_count))

一般的には、バイサルファイト変換効率が99.5%以上であれば実験は成功していると考えることができます。

BisSNP を用いた SNP/メチレーションの検出

検体ゲノムの C 塩基位置のメチル化率および SNP の可能性については、*オーバーラップリードの切り取り*ステップで作成した BAM ファイルを BisSNP のパッケージで処理することで算出することができます。通常 BisSNP は最初のステップで Indel Realignment を検出しますが (詳細は BisSNP のユーザーガイドをご参照ください)、BSMAP ソフトウェアはギャップのあるアライメントに対応していませんので、このステップはスキップしても構いません。

ソフトウェア／モジュール	BisSNP ⇒ BisulfiteCountCovariates BisSNP ⇒ BisulfiteTableRecalibration BisSNP ⇒ BisulfiteGenotyper BisSNP ⇒ sortByRefAndCor.pl BisSNP ⇒ VCFpostprocess BisSNP ⇒ vcf2bed6plus2.strand.pl
インプット	ref.fa {indexed} genome.snps.vcf DESIGNS_capture_target.bed SAMPLE.clipped.bam
アウトプット	SAMPLE.cpg.raw.vcf SAMPLE.cpg.raw.sorted.vcf SAMPLE.cpg.filtered.vcf SAMPLE.cpg.filter.summary.txt SAMPLE. cpg.filtered.strand.6plus2.bed SAMPLE.snp.raw.vcf SAMPLE.snp.raw.sorted.vcf SAMPLE.snp.filtered.vcf SAMPLE.snp.filter.summary.txt SAMPLE. cpg.filtered.strand.6plus2.bed
ベースクオリティの再キャリブレーション java -Xmx10g -jar /path/to/BisSNP-0.82.2.jar -R ref.fa -I SAMPLE.clipped.bam -T BisulfiteCountCovariates -cov ReadGroupCovariate -cov QualityScoreCovariate -cov CycleCovariate -recalFile SAMPLE.recalFile_before.csv -nt NumProcessors -knownSites genome.snps.vcf java -Xmx10g -jar /path/to/BisSNP-0.82.2.jar -R ref.fa -I SAMPLE.clipped.bam -o SAMPLE.recal.bam -T BisulfiteTableRecalibration -recalFile SAMPLE.recalFile_before.csv -maxQ 40	
SNPとメチレーションの検出 java -Xmx10g -jar /path/to/BisSNP-0.82.2.jar -R ref.fa -I SAMPLE.recal.bam -T BisulfiteGenotyper -D genome.snps.vcf -vfn1 SAMPLE.cpg.raw.vcf -vfn2 SAMPLE.snp.raw.vcf -L DESIGNS_capture_target.bed -stand_call_conf 20 -stand_emit_conf 0 -mmq 30 -mbq 0 -nt NumProcessors	
Sort VCFファイル perl /path/to/sortByRefAndCor.pl --k 1 --c 2 SAMPLE.snp.raw.vcf ref.fa.fai > SAMPLE.snp.raw.sorted.vcf perl /path/to/sortByRefAndCor.pl --k 1 --c 2 SAMPLE.cpg.raw.vcf ref.fa.fai > SAMPLE.cpg.raw.sorted.vcf	
Filter SNP/methylation calls java -Xmx10g -jar /path/to/BisSNP-0.82.2.jar -R ref.fa -T VCFpostprocess -oldVcf SAMPLE.snp.raw.sorted.vcf -newVcf SAMPLE.snp.filtered.vcf -snpVcf SAMPLE.snp.raw.sorted.vcf -o SAMPLE.snp.filter.summary.txt java -Xmx10g -jar /path/to/BisSNP-0.82.2.jar -R ref.fa -T VCFpostprocess -oldVcf SAMPLE.cpg.raw.sorted.vcf -newVcf SAMPLE.cpg.filtered.vcf -snpVcf SAMPLE.snp.raw.sorted.vcf -o SAMPLE.cpg.filter.summary.txt	
Convert VCF to BED file perl /path/to/vcf2bed6plus2.strand.pl SAMPLE.snp.filtered.vcf perl /path/to/vcf2bed6plus2.strand.pl SAMPLE.cpg.filtered.vcf	

BisSNPではSNP検出プロセスの一部で既知のSNP情報を利用します。こうしたSNP情報セットが無い場合でも動作しますが、指定することにより特に低カバレージの領域 (例：10x以下) でのSNP検出の正確性を向上させることができます。SNP VCFファイル内の順番 (クロモソーム・ポジション) はリファレンスゲノムファイルやインプットBAMファイル内の順番と一致している必要があります。BisSNPのウェブサイト (*http://epigenome.usc.edu/publicationdata/ bissnp2011/utilies.html*) には、ヒトゲノムHG18およびHG19、マウスゲノムMM9のSNPファイルがVCF形式で保存されています。.

BisSNPのデフォルト設定ではSNPを下記のように フィルタリングします:

- クオリティスコアが 20未満
- カバレージが 120xより大きい
- ストランドバイアスが -0.02より大きい
- カバレージに対するクオリティスコアが 1.0未満
- Heterozygous SNP検出のmapping_quality_zeroフィルター が 0.1 より大きい
- ひとつの20bpウィンドウ内に2個のSNPsがある

作成されるBEDファイル (SAMPLE. cpg.filtered.strand.6plus2.bed) の7列目にはメチル化率が、8列目にはC/Tカバレージが記載されます。このファイルをRoche NimbleGenの提供するゲノムビューワーであるSignalMapソフトウェアで開くと、メチル化率は0から1000のスコアとして縦軸にプロットされます。スコア0とはメチル化率が0%であることを示し、スコア1000とはメチル化率が100%であることを示しています。この時、ストランドを示すカラムは削除されることから、両鎖のメチル化率は正の値として表示されます。C/Tカバレージは画面上に表示されません。SignalMapソフトウェアv2.0 は *http://www.nimblegen.com/products/software/signalmap/*から無償で入手することができます。

DMR の検出

2 検体または 2 条件を比較する場合、DMR (differentially methylated region) を検出するためのソフトウェアパッケージには様々なものがあります。このうちの 1 つが DNA メチレーション解析用の R パッケージである methylKit です。R のインストールについては *http://cran.r-project.org/doc/manuals/R-admin.html*、methylKit やその関連ツールのインストール方法については *https://code.google.com/p/methylkit/* をご参照ください。

ソフトウェア／モジュール	R/methylKit ⇒ read R/methylKit ⇒ filterByCoverage R/methylKit ⇒ unite R/methylKit ⇒ calculateDiffMeth R/methylKit ⇒ get.methylDiff R/methylKit ⇒ tileMethylCounts
インプット	BSMAP methylation results files
アウトプット	{様々です}

```
# DMRを検出するためのRコード
# ファイルをロードする
library(methylKit)
file.list = list("SAMPLE.methylation_results.txt","CONTROL.methylation_results.txt")
sample.list = list("TEST","CONTROL")

methData <-read(
    file.list,
    sample.id=sample.list,
    treatment=c(0,1),
    assembly="hg19",
    header=TRUE,
    context="CpG",
    resolution="base",
    pipeline=list(
        fraction=TRUE,
        chr.col=1,
        start.col=2,
        end.col=2,
        coverage.col=6,
        strand.col=3,
        freqC.col=5
    )
)
```

```
#基本的な統計情報を作成する
getMethylationStats(methData,plot=T,both.strands=T)
getCoverageStats(methData,plot=T,both.strands=T)
```

```
#カバレージ (>10X)またはメチル化率(<99.9 パーセンタイル)によるフィルタリング
methData.filtered = filterByCoverage(methData, lo.count = 10, lo.perc=NULL,
hi.count=NULL,hi.per = 99.9)
```

```
#Merge samples to locations covered by all samples
meth = unite(methData.filtered,destrand=FALSE)
```

```
#Calculate per-base differential methylation p-values and q-values
methDiff = calculateDiffMeth(meth)
```

```
#Find differentially methylated bases with 25% difference and qvalue<0.01
Diff25p=get.methylDiff(methDiff,difference=25,qvalue=0.01)
```

```
#Find differentially hypo methylated bases with 25% difference and qvalue<0.01
Diff25pHypo =get.methylDiff(methDiff,difference=25,qvalue=0.01,type="hypo")
```

```
#Find differentially hyper methylated bases with 25% difference and qvalue<0.01
Diff25pHyper=get.methylDiff(methDiff,difference=25,qvalue=0.01,type="hyper")
```

```
#Find differentially methylated regions with 25% difference and qvalue<0.01
tiles = tileMethylCounts(methData.filtered,win.size=500,step.size=100)
tileMeth = unite(tiles,destrand=FALSE)
tileDiff = calculateDiffMeth(tileMeth,num.cores=8)
tile25pct <- get.methylDiff(tileDiff,difference=25,qvalue=0.01)
```

```
#Write data to file
write.table(getData(Diff25p),file="diff25pct.txt",row.names=F,col.names=T,sep="¥t",quote=F)
write.table(getData(tile25pct),file="tile25pct.txt",row.names=F,col.names=T,sep="¥t",quote=F)
```

レプリケートデータセットはインポート (read)時に指定した処理値 (treatment=c(・・・)) によって自動的に統合することができます。Q値と percent differencesを調節することで、厳密性を調節できます。また、methylKitのサイト (*https://code.google.com/p/methylkit/*) では、相関・クラスタリング・PCAやアノテーションの追加などの追加的な解析についての例とチュートリアルが提供されています。

3. リファレンスWebサイト

BamUtil :
http://genome.sph.umich.edu/wiki/BamUtil

BEDtools :
https://code.google.com/p/bedtools/

BisSNP :
http://sourceforge.net/projects/bissnp/

BSMAP :
https://code.google.com/p/bsmap/

FastQC :
http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

GATK:
http://www.broadinstitute.org/gatk/

IGV :
http://www.broadinstitute.org/igv/

methylKit :
https://code.google.com/p/methylkit/

Picard :
http://picard.sourceforge.net/

R :
http://www.r-project.org/

SAMtools (including BCFtools and VCFutils) **:**
http://samtools.sourceforge.net/

seqtk :
https://github.com/lh3/seqtk

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

※　これらのウェブサイトの内容について Roche NimbleGen では責任を負いかねます。

4. 用語

BAI file・・・・・・・・・・・・ BAMインデックスファイル。インデックスが付与されたBAMファイルが必要なツールのために、BAIファイルはBAMファイルと同じ場所に保存されている必要があります。

Bait・・・・・・・・・・・・・・・ Capture targetの項を参照。

BAM file・・・・・・・・・・・・ SAMファイルを圧縮したバイナリ形式のファイル。

BCF file・・・・・・・・・・・・ VCF ファイルを圧縮したバイナリ形式のファイル。

BED file・・・・・・・・・・・・ ゲノム領域/間隔を記述するためのファイル形式。BEDファイルのスタート位置情報 (左端) は0と示されます。

Capture target・・・・・・・ Roche NimbleGenにより定義された単語で、一本以上のプローブでカバーされる領域を示します。NimbleGenのBEDファイルではこの領域をTiled regionsと呼んでいます。PicardでのBait intervalsに相当します。

FASTA file・・・・・・・・・・ 核酸配列を記述するための標準ファイル形式。

FASTQ file・・・・・・・・・・ ベースクオリティ情報も含むシークエンスリードを記述するための標準ファイル形式。

Genomic index・・・・・・・ より迅速なアライメント時の比較を可能とするリファレンスゲノム配列の形式。

Picard interval file・・・ リファレンスゲノム情報が記載されたヘッダを含む、ゲノム領域/間隔を記述するためのファイル形式。Picard intervalファイルのスタート位置情報は1と示されます。

Primary target・・・・・・・ Roche NimbleGenにより定義された単語で、プローブ設計対象領域を示します。NimbleGenのBEDではこの領域を単にTarget regionsと記載しています。ほとんどの領域は元々の研究対象領域 (regions of interest) と同一ですが、100bp以下の領域についてはプローブ選択を容易にするために100bpに拡張しています。PicardでのTarget intervalsに相当します。

SAM file・・・・・・・・・・・・ Sequence Alignment / Mapファイル。リファレンスゲノムへのシークエンスリードのアライメント結果を記述するためによく使用される標準形式。

Target region・・・・・・・・・ Primary targetの項を参照。

Tiled region・・・・・・・・・・ Capture targetの項を参照。

VCF file・・・・・・・・・・・・ Variant call format ファイル。リファレンスゲノムに対するバリアント検出結果を記述するためによく使用される標準形式。

ロシュ・ダイアグノスティックス株式会社
シークエンスソリューション
〒105-0014 東京都港区芝2 丁目6 番1 号
TEL. 03-5443-5287

 1408R