Sequencing Solution: Technical Note March 2014

NimbleGen SeqCap EZ 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 EZ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。一般的な解析ワークフローは、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、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 EZのデータ解析方法の一例を示します。BAMファイル作成とバリアントコールのために生データのFASTQファイルを扱う方法の紹介に加え、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をする際には、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)。

ソフトウェアとモジュールについての概要

ソフトウェア (バージョン)	モジュール	本資料中で使用する機能
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延長 (パディング)。
BWA (0.7.5a-r405)	index	FASTA シークエンスからインデックスが付与されたゲノムを作成。
	mem	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。
FastQC (0.10.1)	fastqc	シークエンスリードのクオリティを評価 (塩基毎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 (この文書では使用されていません)。*3. リファレンスWebサイト*をご参照ください。
Picard (1.98)	CreateSequenceDictionary	リファレンスゲノムのシークエンスディクショナリファイル (.dict)を生成。
	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リード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ファイルの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ファイルにインデックスを付与。
	faidx	リファレンスゲノムのFASTAインデックスを作成。
	flagstat	BAMファイルのマッピング結果概要レポートを出力。
	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	クオリティによる生リードのトリミング。

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。*3. リファレンス Web サイト*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。

リファレンスゲノムへのインデックス付与

クロモソーム順にソートされている (chr1, chr2, ..., chr10, chr11, … chrX, chrY, chrM, chr1_random など) FASTA ファイルフォーマットのゲノム配列にインデックスを付与します。下記の例では、リファレンスゲノムファイル名が‘ref.fa’であるとして記述していますが、実際のファイル名(*例.* ‘hg19.fa’)に置換して記述してください。

ソフトウェア／モジュール	BWA ／ index SAMtools ／ faidx Picard ／ CreateSequenceDictionary
インプット	ref.fa
アウトプット	ref.fa {indexed} ref.fa =加工されていないリファレンスゲノム ref.fa.amb, ref.fa.ann, ref.fa.bwt, ref.fa.pac, ref.fa.sa = インデックス化リファレンスファイル ref.fa.fai =インデックス化FASTA ref.dict =リファレンスシークエンスディクショナリ
リファレンスゲノムインデックスの作成 /path/to/bwa index -a bwtsw /path/to/ref.fa	
FASTAインデックスの作成 /path/to/samtools faidx /path/to/ref.fa	
シークエンスディクショナリの作成 java -Xmx4g -Xms4g -jar /path/to/Picard/CreateSequenceDictionary.jar REFERENCE=/path/to/ref.fa OUTPUT=ref.dict	

上記の例で Xmx、Xms でオプションに指定された値はそれぞれメモリの最大値、メモリの初期値を示しています。

ゲノムへインデックスを付与する操作は 1 つのゲノムバージョンに対して 1 度だけ必要です。ここで作成されたインデックスが付与されたゲノムは、その他のマッピング解析にも使用できます。本紙での以降のステップでインデックスが付与されたリファレンスゲノムが必要な場合には、そのプロセスのインプットセクションに ‘ref.fa {indexed}’と記載しています。インデックスが付与されたリファレンスゲノムは、ゲノム FASTA ファイルと本章で作成されたファイルで構成され、全て同じディレクトリに存在しているものとします。

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。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
bad_sequence_fastqc/fastqc_report.html

並べ替え、インデックスを付与した BAM ファイルの作成

このプロセスを実施する前に、ここまでのページで記述した*リファレンスゲノムへのインデックス付与* (2 ページ)、*FASTQ ファイルの解凍* (3 ページ)、*シークエンスリードクオリティの評価* (3 ページ) の各項の処理をしておく必要がある場合があります。このプロセスでのインプットファイルである 'adapters.fa' ファイルは、シークエンス用アダプター配列を含む FASTA ファイルです (例えば Trimmomatic のダウンロードサブディレクトリ 'adapters' をご参照ください)。

ソフトウェア／モジュール	Trimmomatic ／ illuminaclip BWA ／ mem SAMtools ／ view SAMtools ／ sort Picard ／ MarkDuplicates SAMtools ／ index
インプット	adapters.fa ref.fa {indexed} SAMPLE_R1.fastq SAMPLE_R2.fastq
アウトプット	SAMPLE.bam {indexed} SAMPLE.bam = sorted and duplicate-marked BAM file SAMPLE.bam.bai = BAM index file SAMPLE_picard_markduplicates_metrics.txt
リードのトリミング (必須では無いが推奨) 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	
リードのマッピング /path/to/bwa mem –R "@RG¥tID:SAMPLE¥tPL:illumina¥tLB:SAMPLE¥tSM:SAMPLE" /path/to/ref.fa –t NumProcessors –M SAMPLE_R1_trimmed.fq SAMPLE_R2_trimmed.fq | /path/to/samtools view –Sb – > SAMPLE_unsorted.bam	
BAMファイル内の情報の並べ替え /path/to/samtools sort SAMPLE_unsorted.bam SAMPLE_sorted	
重複リードのチェック java –Xmx4g –Xms4g –jar /path/to/Picard/MarkDuplicates.jar VALIDATION_STRINGENCY=LENIENT INPUT=SAMPLE_sorted.bam OUTPUT=SAMPLE.bam METRICS_FILE=SAMPLE_picard_markduplicates_metrics.txt REMOVE_DUPLICATES=false ASSUME_SORTED=true	
BAMファイルへインデックスを付与 /path/to/samtools index SAMPLE.bam	

「リードのトリミング」ステップで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」ステップでは、-R オプションでリードグループ（'@RG'）を定義することで、アウトプットファイルである BAM ファイルのヘッダの記述を変更することができます。今回の例では、サンプル ID（'ID'）、シークエンシングプラットフォーム（'PL'）、ライブラリ名（'LB'）、検体名（'SM'）を指定できます。ライブラリ名、ID、検体名が一致する場合には上記の例のように SAMPLE という1つの記述を使用することもできます。

「BAMファイル内の情報の並べ替え」ステップでは自動的に拡張子が追加されるため、コマンドラインのアウトプットファイル名として'.bam'というファイル拡張子を追加する必要は有りません。

「重複リードのチェック」ステップでは、続けて使用するツールが重複フラグを適切に認識するかどうかが分からない場合にはREMOVE_DUPLICATES=trueを使うことが望ましいかもしれません。ペア、非ペア、重複リード数はSAMPLE_picard_markduplicates_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、これらのリードは重複リードの割合（PERCENT_DUPLICATION）の計算には含まれておりません。ペアおよび非ペアの重複リードとして数えられています。

並べ替えられ、重複リードがマークされたSAMPLE.bamファイルは以下の様々な例でのインプットとして使用されます。後続ステップでインデックス化されたBAMファイルを使用する場合、インプットセクションに'SAMPLE.bam {indexed}'と記載します。SAMPLE.bamと同じディレクトリにSAMPLE.bam.baiインデックスファイルが存在している必要があります。

マッピング結果概要(SAMtools)

マッピング結果概要は SAMtools を使って SAMPLE.bam ファイルから作成することができます。このプロセスを実施する前に*並べ替え、インデックスを付与した BAM ファイルの作成*（3 ページ）の各項の処理をしておく必要がある場合があります。

ソフトウェア／モジュール	SAMtools ／ flagstat
インプット	SAMPLE.bam
アウトプット	SAMPLE_samtools_flagstat_metrics.txt
/path/to/samtools flagstat SAMPLE.bam > SAMPLE_samtools_flagstat_metrics.txt	

アウトプットファイルには、BAMファイル中の全リード数、マップされたリード数、ペア・非ペア（singletons）のマップされたリード数、（除去する代わりにマークするオプションを使用した場合には）重複リード数が記載されます。

マッピング結果概要(Picard)

マッピング結果概要は Picard を使って SAMPLE.bam と ref.fa {indexed}ファイルから作成することができます。このプロセスを実施する前に*リファレンスゲノムへのインデックス付与*（2 ページ）、*並べ替え、インデックスを付与した BAM ファイルの作成*（3 ページ）の各項の処理をしておく必要がある場合があります。

ソフトウェア／モジュール	Picard ／ CollectAlignmentSummaryMetrics
インプット	ref.fa {indexed} 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	

アウトプットファイルについては *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。このプロセスを実施する前に*並べ替え、インデックスを付与した BAM ファイルの作成*（3 ページ）の各項の処理をしておく必要がある場合があります。

ソフトウェア／モジュール	Picard ／ CollectInsertSizeMetrics
インプット	SAMPLE.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.bam OUTPUT=SAMPLE_picard_insert_size_metrics.txt	

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作成することもできます。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。R がインストールされている場合には、SAMPLE_picard_insert_size_plot.pdf というプロットも作成され、PDF ファイルに保存されます。

BED ファイルからの位置情報の抽出

BEDファイルは列数を自由に使うことのできるタブ区切り形式のファイルで、最初の3列 (染色体、開始位置、終了位置)のみが必須の情報となります(詳細は *http://genome.ucsc.edu/FAQ/FAQformat.html#format1* をご参照ください)。SeqCap EZには、(a)1つのトラックで構成された2つのファイルセット、(b)2つのトラックで構成された1つのファイル、のどちらかの状態でBEDファイルが添付されます。①primary target(プローブ設計対象の領域でtarget regionとも呼ぶ)、または、②capture target(プローブでカバーされる領域でtiled regionとも呼ぶ) 、の2つのトラック情報はそれぞれのヘッダ行以下に位置情報が記載されています。

解析ツールを使用するにあたって、ヘッダ行の削除、位置情報の並べ替え、領域の統合 (2つ領域の位置関係が、一方の開始位置と他方の終了位置が隣り合っている場合や重なり合っている場合) が必要です。2つのトラックで構成された1つのBEDファイル(b)が添付されている場合は、これらの処理の前にトラックごとに分割する必要があります。アウトプットファイルの最初と最後の数行をチェックすることで、抽出が成功しているかどうかを確認すると良いでしょう。

下記の例では、最初の行にヘッダ行が存在し、それ以下の行に242,305行の位置情報の行が続くというprimary targetのトラック(target region)と、同様に最初の行にヘッダ行があり、それ以下の行に381,429行の位置情報の行が続くというcapture targetのトラック(tiled region)の情報を記述したBEDファイルを使用した場合で例記しています。これら2つのトラックが1つのBEDファイルに保存されている場合、BEDファイルの総行数は2つのヘッダ行を含めて623,736行となります。このとき、primary targetのヘッダが1行目に記述されており、capture targetのヘッダは242,307行目に記述されていることになります。

(a) primary target と capture target が別々の BED ファイルに保存されている場合

Linux の wc と tail コマンドを使用してヘッダ行を削除し、BEDtools コマンドを使用して並べ替えと統合を実施します。

ソフトウェア／モジュール	wc tail BEDtools ／ sort BEDtools ／ merge
インプット	SAMPLE_RNG_primary.bed (下記ではこのファイルを使用した場合について例記) SAMPLE_RNG_capture.bed
アウトプット	SAMPLE_primary_coord.bed SAMPLE_capture_coord.bed
インプットファイルの行数の計測 wc -l SAMPLE_RNG_primary.bed 242306 {ヘッダを含む総行数が*242306*であることが示されました}	
領域位置情報の抽出 (ヘッダ行の削除) tail -n 242305 SAMPLE_RNG_primary.bed > SAMPLE_primary_coord_unmerged_unsorted.bed {位置情報の行は「ヘッダを含む総行数」から1を引いた数になります}	
領域位置情報の並べ替え /path/to/bedtools sort -i SAMPLE_primary_coord_unmerged_unsorted.bed > SAMPLE_primary_coord_unmerged_sorted.bed	
重なり合った、あるいは隣り合った領域の位置情報の統合 /path/to/bedtools merge -i SAMPLE_primary_coord_unmerged_sorted.bed > SAMPLE_primary_coord.bed	

この例では、primary target のトラックのみが記述された BED ファイルでの操作例を示しています。capture target のトラックのみが記述された BED ファイルについても、ファイル名を適切に置き換え同じ操作で処理できます。このアウトプットファイルである SAMPLE_primary_coord.bed ファイルと SAMPLE_capture_coord.bed ファイルは様々なコマンドで使用されますが、Picard interval lists を作成するのに必須なファイルです。

(b) 2 つのトラック情報を持つ BED ファイルからの primary target 領域の抽出

トラック 1 最終行の次から存在するトラック 2 のヘッダの場所を確認する為に、Linux の grep コマンドを使用します。次に head コマンドでヘッダ行を含む primary target トラック全体を抽出し(track 1, target region)、最後に tail コマンドによりヘッダ行を削除します。その後、BEDtools コマンドを使用して並べ替えと統合を実施します。

ソフトウェア／モジュール	grep head tail BEDtools ／ sort BEDtools ／ merge
インプット	SAMPLE_RNG.bed
アウトプット	SAMPLE_primary_coord.bed
トラックのヘッダ行数の検索 grep -n track SAMPLE_RNG.bed 1:track name=target_region description="Target Regions" 242307:track name=tiled_region description="NimbleGen Tiled Regions" {トラック2のヘッダは*242307*行目にあることが示されました}	
primary target領域の位置情報の抽出 head -242306 SAMPLE_RNG.bed | tail -242305 > SAMPLE_primary_coord_unmerged_unsorted.bed {トラック1の最終行は、「トラック2のヘッダ行数」から1を引いた数になります} {トラック1の位置情報行数は、トラック2のヘッダ行数から2を引いた数になります}	
領域位置情報の並べ替え /path/to/bedtools sort -i SAMPLE_primary_coord_unmerged_unsorted.bed > SAMPLE_primary_coord_unmerged_sorted.bed	
重なり合った、あるいは隣り合った領域の位置情報の統合 /path/to/bedtools merge -i SAMPLE_primary_coord_unmerged_sorted.bed > SAMPLE_primary_coord.bed	

このアウトプットファイルである SAMPLE_primary_coord.bed ファイルは様々なコマンドで使用され、Picard interval lists を作成するのに必須なファイルです。

(b) 2 つのトラック情報を持つ BED ファイルからの capture target 領域の抽出

トラック 2 のヘッダの場所を確認する為に、Linux の grep コマンドを使用し、次に wc コマンドにより capture target の位置情報の行数を計算するために BED ファイルの総行数をカウントし(track 2, tiled region)、最後に tail コマンドによりヘッダ行を削除します。その後、BEDtools コマンドを使用して並べ替えと統合を実施します。

ソフトウェア／モジュール	grep wc tail BEDtools ／ sort BEDtools ／ merge
インプット	SAMPLE_RNG.bed
アウトプット	SAMPLE_capture_coord.bed
トラックのヘッダ行数の検索 grep -n track SAMPLE_RNG.bed 1:track name=target_region description="Target Regions" 242307:track name=tiled_region description="NimbleGen Tiled Regions" {トラック2のヘッダは*242307*行目にあることが示されました}	
行数の計測 wc -l SAMPLE_RNG.bed 623736 {総行数は*623736*であることが示されました}	
capture target領域の位置情報の抽出 tail -381429 SAMPLE_RNG.bed > SAMPLE_capture_coord_unmerged_unsorted.bed {トラック2の位置情報行数は「総行数」から「トラック2のヘッダ行数」を引いた数になります}	
領域位置情報の並べ替え /path/to/bedtools sort -i SAMPLE_capture_coord_unmerged_unsorted.bed > SAMPLE_capture_coord_unmerged_sorted.bed	
重なり合った、あるいは隣り合った領域の位置情報の統合 /path/to/bedtools merge -i SAMPLE_capture_coord_unmerged_sorted.bed > SAMPLE_capture_coord.bed	

このアウトプットファイルである SAMPLE_capture_coord.bed ファイルは様々なコマンドで使用されますが、Picard interval lists を作成するのに必須なファイルです。

capture target 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 リード数の確認*(6 ページ) で記載されている 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。このプロセスを実施する前に *BED ファイルからの位置情報の抽出*(5 ページ) の各項の処理をしておく必要がある場合があります。

ソフトウェア／モジュール	BEDtools ／ slop BEDtools ／ sort BEDtools ／ merge
インプット	SAMPLE_capture_coord.bed chromosome_sizes.txt
アウトプット	SAMPLE_padded_capture_coord.bed
Capture Target領域へのパディング /path/to/bedtools slop -i SAMPLE_capture_coord.bed -b 100 -g chromosome_sizes.txt > SAMPLE_capture_coord_padded_unmerged_unsorted.bed	
パディング済み領域の並べ替え /path/to/bedtools sort -i SAMPLE_padded_capture_coord_unmerged_unsorted.bed > SAMPLE_padded_capture_coord_unmerged_sorted.bed	
重なり合った、あるいは隣り合った領域の位置情報の統合 /path/to/bedtools merge -i SAMPLE_padded_capture_coord_unmerged_sorted.bed > SAMPLE_padded_capture_coord.bed	

上記の例では抽出処理後のcapture region位置情報 (capture region BED file) にパディングをしていますが、BEDtools はアウトプットファイルからトラック行を除去するため、1つのファイルに1つのトラック情報が記載されたBEDファイルであれば *(a) primary targetとcapture targetが別々のBEDファイルに保存されている場合*(5ページ) のプロセスをしていない場合でも処理を行うことができます。

「capture target領域へのパディング」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のようなフォーマットである必要があり (*例* 'chr1 249250621') 、インプットのBEDファイルに存在するそれぞれ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。

一つのトラックの BED ファイルでの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 ファイルの条件については、前項 (*capture target 領域へのパディング*) をご参照ください。このプロセスを実施する前に *BED ファイルからの位置情報の抽出*(5 ページ) の各項の処理をしておく必要がある場合があります。

ソフトウェア／モジュール	BEDtools ／ genomecov grep cut
インプット	chromosome_sizes.txt SAMPLE_capture_coord.bed (下記ではこのファイルを使用した場合について例記) or SAMPLE_primary_coord.bed or SAMPLE_RNG_capture.bed or SAMPLE_RNG_primary.bed
アウトプット	{サイズが表示されます}
/path/to/bedtools genomecov -i SAMPLE_capture_coord.bed -g chromosome_sizes.txt -max 1 | grep -P "genome¥t1" | cut -f 3	

上記の例では capture target の位置情報ファイルから総サイズを確認していますが、同じコマンド操作で 1 つのトラックで構成された他の BED ファイルの総サイズを確認することもできます。

On-Target リード数の確認

このプロセスを実施する前に *並べ替え、インデックスを付与した BAM ファイルの作成*(3 ページ) 、*BED ファイルからの位置情報の抽出*(5 ページ) 、*capture target 領域へのパディング*(6 ページ) の各項の処理をしておく必要がある場合があります。

ソフトウェア／モジュール	BEDtools ／ intersect wc
インプット	SAMPLE.bam SAMPLE_primary_coord.bed or SAMPLE_capture_coord.bed or SAMPLE_padded_capture_coord.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 SAMPLE_primary_coord.bed | wc -l or /path/to/bedtools intersect -bed -abam SAMPLE.bam -b SAMPLE_capture_coord.bed | wc -l or /path/to/bedtools intersect -bed -abam SAMPLE.bam -b SAMPLE_padded_capture_coord.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。on-target リードの割合 (on-target 率) を算出するには、マップされ重複リードを除いたリードの総数でこの数を割ります。マップされ重複リードを除いたリードの総数を調べるには、*マッピング結果概要(SAMtools)*(4 ページ) あるいは *マッピング結果概要(Picard)*(4 ページ) をご参照ください。

カバレージの確認

このプロセスを実施する前に*リファレンスゲノムへのインデックス付与*(2ページ)、*並べ替え、インデックスを付与した BAM ファイルの作成*(3ページ)、*BED ファイルからの位置情報の抽出*(5ページ)の各項の処理をしておく必要がある場合があります。

ソフトウェア／モジュール	GATK ／ DepthOfCoverage
インプット	ref.fa {indexed} SAMPLE.bam {indexed} SAMPLE_primary_coord.bed or SAMPLE_capture_coord.bed
アウトプット	SAMPLE_gatk_primary_target_coverage.sample_summary or SAMPLE_gatk_capture_target_coverage.sample_summary {この他のアウトプットファイルが作成される場合があります}
java -Xmx4g -Xms4g -jar /path/to/GATKFramework/GenomeAnalysisTK.jar -T DepthOfCoverage -R /path/to/ref.fa -I SAMPLE.bam -o SAMPLE_gatk_primary_target_coverage -L SAMPLE_primary_coord.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.bam -o SAMPLE_gatk_capture_target_coverage -L SAMPLE_capture_coord.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塩基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* でマニュアルを参照してください。

バリアントの検出とフィルタリング

このプロセスを実施する前に*リファレンスゲノムへのインデックス付与*(2ページ)、*並べ替え、インデックスを付与した BAM ファイルの作成*(3ページ)、*BED ファイルからの位置情報の抽出*(5ページ)の各項の処理をしておく必要がある場合があります。SAMtools の mpileup コマンドを使用してバリアント(SNVs と小さな indel)を検出し、varFilter コマンドを使用してそのバリアントをカバレージによりフィルタリングします。

ソフトウェア／モジュール	SAMtools ／ mpileup SAMtools ／ BCFtools>view SAMtools ／ vcfutils>varFilter
インプット	ref.fa {indexed} SAMPLE.bam {indexed}
アウトプット	SAMPLE_vcfutils_var.flt.vcf
ゲノムバリアントの検出 /path/to/samtools mpileup -uf /path/to/ref.fa SAMPLE.bam | /path/to/bcftools view -bvcg - > SAMPLE_samtools_var.raw.bcf	
検出したバリアントのフィルタリング /path/to/bcftools view SAMPLE_samtools_var.raw.bcf | /path/to/vcfutils.pl varFilter -d 12 -D 250 > SAMPLE_vcfutils_var.flt.vcf	

varFilterコマンドにはカバレージをコントロールする最少(-d)、最大(-D)という2つのフィルタリングオプションがあり、上記の例ではそれぞれ'12'と'250'を使用しています。最少カバレージ値を増加させると、カバレージが低い真のバリアントを見逃す可能性がありますが、カバレージが5よりも低いバリアントはシークエンシングエラーとの区別が難しいために一般的に信頼性は低くなります。

この資料では記述しておりませんが、追加の下流のバリアント解析には特定のサンプルにおける既知バリアントとの比較、dbSNP と検出した SNP の比較、バリアントの分類、バリアントの効果の解析などがあります。

Picard のインターバルリストの作成

Picardのインターバルリストとは、PicardのCalculateHsMetricsコマンドに必要とされるゲノム区間を記述したファイルです。このファイルは、各区間で、リファレンスゲノムと位置情報セットが鎖と名前の情報とともに記載されており、SAMのようなヘッダを持っています。Picardの'target interval'は'primary target'に、Picardの'bait interval'は'capture target'に相当しています。このプロセスを実施する前に*並べ替え、インデックスを付与したBAMファイルの作成*(3ページ)、*BEDファイルからの位置情報の抽出*(5ページ)の各項の処理をしておく必要がある場合があります。

■ Picard のターゲットインターバルリストの作成

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、Linux の gawk コマンドで SAMPLE_primary_coord.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、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 SAMPLE_primary_coord.bed
アウトプット	SAMPLE_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 SAMPLE_primary_coord.bed | gawk '{print $1 "¥t" $2+1 "¥t" $3 "¥t+¥tinterval_" NR}' > SAMPLE_target_body.txt	
ヘッダと本文の連結 cat SAMPLE_bam_header.txt SAMPLE_target_body.txt > SAMPLE_target_intervals.txt	

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

■ Picard のベイトインターバルリストの作成

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、Linux の gawk コマンドで SAMPLE_capture_coord.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、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 SAMPLE_capture_coord.bed
アウトプット	SAMPLE_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 SAMPLE_capture_coord.bed | gawk '{print $1 "¥t" $2+1 "¥t" $3 "¥t+¥tinterval_" NR}' > SAMPLE_bait_body.txt	
ヘッダと本文の連結 cat SAMPLE_bam_header.txt SAMPLE_bait_body.txt > SAMPLE_bait_intervals.txt	

SAMPLE_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*リファレンスゲノムへのインデックス付与*(2 ページ)、*並べ替え、インデックスを付与した BAM ファイルの作成*(3 ページ)、*Picard のインターバルリストの作成*(7 ページ) の各項の処理をしておく必要がある場合があります。

ソフトウェア／モジュール	Picard ／ CalculateHsMetrics
インプット	ref.fa {indexed} SAMPLE.bam {indexed} SAMPLE_target_intervals.txt SAMPLE_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=SAMPLE_bait_intervals.txt
TARGET_INTERVALS=SAMPLE_target_intervals.txt INPUT=SAMPLE.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“TARGET_INTERVALS” (primary target) と“BAIT_INTERVALS” (capture target) は、両方のインターバルファイルの違いが大きいほど結果に違いが生じます。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。

3. リファレンスWebサイト

BEDtools・・・・・・・・・・・・・ *https://code.google.com/p/bedtools/*
BWA・・・・・・・・・・・・・・・・ *http://bio-bwa.sourceforge.net/*
FastQC・・・・・・・・・・・・・・ *http://www.bioinformatics.babraham.ac.uk/projects/fastqc/*
GATK・・・・・・・・・・・・・・・ *http://www.broadinstitute.org/gatk/*
IGV・・・・・・・・・・・・・・・・ *http://www.broadinstitute.org/igv/*
Picard・・・・・・・・・・・・・・ *http://picard.sourceforge.net/*
SAMtools (BCFtoolsとVCFutilsを含む) ・・・ *http://samtools.sourceforge.net/*
seqtk・・・・・・・・・・・・・・・ *https://github.com/lh3/seqtk*
Trimmomatic・・・・・・・・・ *http://www.usadellab.org/cms/?page=trimmomatic*

※　弊社ではこれらのウェブサイトの内容について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のこと。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 interval (Picard)・ Capture targetの項を参照。
BAM file・・・・・・・・・・・・ SAMファイル形式を圧縮した形式。
BCF file・・・・・・・・・・・・ VCF ファイル形式を圧縮した形式。
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・・・・・・・ アライメント時により迅速な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。Bait interval (Picard)とTarget interval (Picard)の項を参照。
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を記述するためによく使用される標準形式。
Target interval (Picard) ・Primary targetの項を参照。
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

1405R