ALEA2

v.2.0.jra User Guide

7th December 2017

Contents

Introduction to ALEA2

Building on our previous work on the ALEA software package, a computational toolbox for allele-specific epigenomics analysis incorporating allelic variation data, we detail here ALEA's successor - ALEA2. This package dramatically increases the functionality and usability of ALEA, incorporating allelic variation with existing resources, allowing for the identification of significant associations of epigenetic modifications and specific allelic variants in human and mouse cells. Similar to ALEA, ALEA2 provides a customizable pipeline for allele-specific analysis of next-generation sequencing data which takes raw sequencing data for ChIP-seq, RNA-seq and DNA Methylation analysis, producing a UCSC track hub. ALEA2 takes advantage of the available genomic resources for human (The 1000 Genomes Project Consortium) and mouse (The Mouse Genome Project) to reconstruct diploid *in silico* genomes for human samples or hybrid mouse samples. Then, for each accompanying ChIP-seq, RNA-seq or DNA Methylation dataset, ALEA2 generates two wig files from short reads aligned differentially to each haplotype. This pipeline has been validated using human and hybrid mouse ChIP-seq, RNA-seq and DNA Methylation data (See Test Data section).

Quick Reference

Dependencies

To run ALEA2, the following dependencies are required: Java (1.7), Python (2.4 +), bwa (0.7.12) and/or bowtie (1.1.2 or 2.2.9), samtools (0.1.19 & 1.3.1), tabix (0.2.5 +), bgzip, bedGraphToBigWig, VCFtools (0.1.14), bedtools(2.17.0), Bismarck (0.16.3), tophat (2.1.1), STAR(2.5.1b). However, it is recommended to carry out installation and running of ALEA2 using Docker, the correct dependencies are built into the image.

Docker can be downloaded and installed using the OS-specific information at `https://www.docker.com/`.

Pipeline

As with ALEA version 1, ALEA2 uses the pipeline shown in Fig.1, taken from Younesy et al., 2014, for allele-specfic analysis. Further to this allele-specific alignment of NGS reads generated from sodium bisulphite-converted DNA and subsetquent allele-specific methylation analysis have been added to the pipeline, following the steps highlighted in Figs. 2 and 3 respectively.

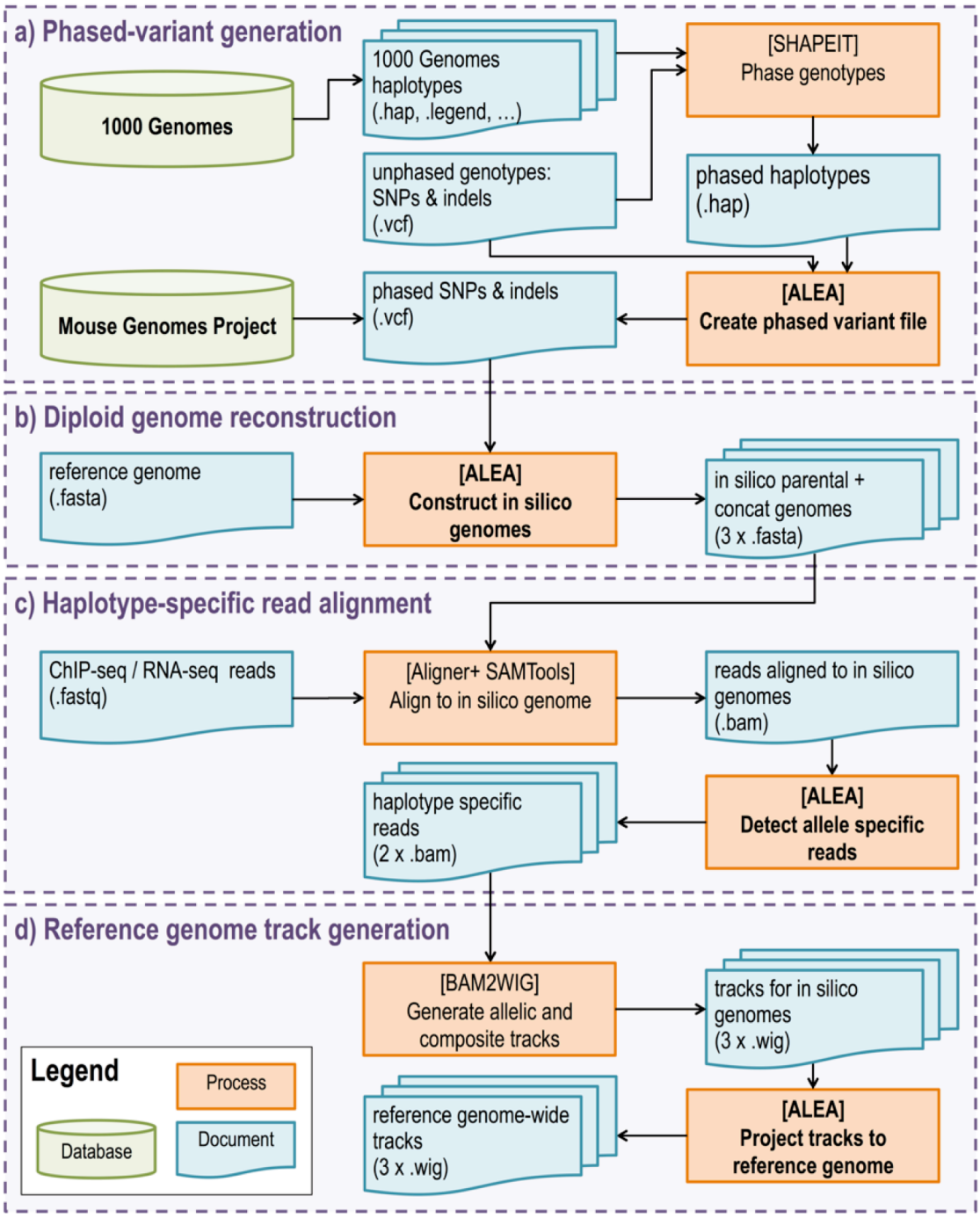

Figure 1: Allele-specific epigenomics analysis pipleine in ALEA2, taken from Younesy et al., 2014

Haplotype-specific bisulphite-seq aligns whole genome bisulphite sequencing (WGBS) reads to the *in silico* genomes constructed in step 2 of the original ALEA pipeline and detects reads that are uniquely aligned to only one of the two genomes. Reads aligned to multiple locations in either haplotype and reads mapping equally to both haplotypes at the same location are filtered out, hence only allele-specific reads mapping to the regions containing heterozygous SNPs are detected.

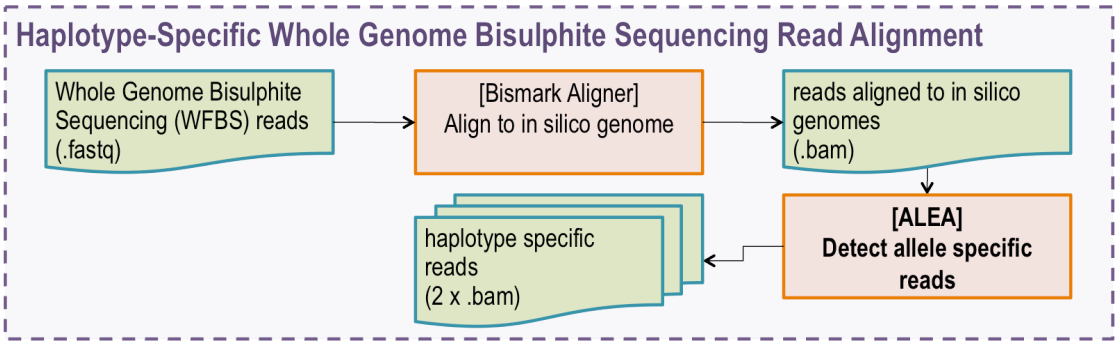

Figure 2: Allele-specfic epigenomics analysis pipleine in ALEA2

Methylation call and projection of the haplotype-specific methylation coordinates back to the reference genome. This module first calls CpG methylation from the allele-specific bam files generated in the previous module and then generates two haplotype-specific methylation track files by projecting the allele-specific coordinates back to the reference genome. Due to indels present in the construction of parental in silico genomes, the coordinates of the tracks are offset when compared with the reference genome. However, most visualization tools are based on alignment to reference genomes. Using a refmap created with the in silico genomes, ALEA2 maps the methylation tracks back to the reference genome.

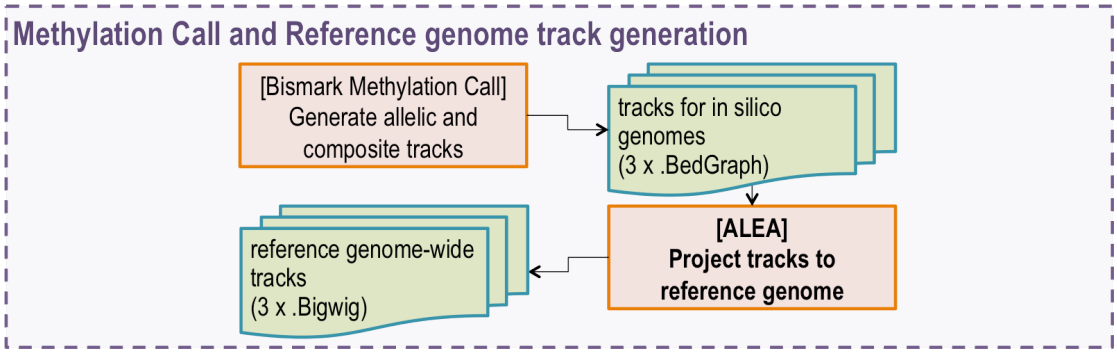

Figure 3: Allele-specfic epigenomics analysis pipleine in ALEA2

Synopsis

```
alea phaseVCF hapsDIR unphased.vcf outputPrefix

alea createGenome reference.fasta phased.vcf.gz strain1 strain2
     outputDir

alea createGenome -snps-indels-separately reference.fasta
     phased_snps.vcf.gz phased_indels.vcf.gz strain1 strain2
     outputDir

alea alignReads <-s/-p> input_reads_1 [input_reads_2] genome_concat
     genome_reference strain1 strain2 outputPrefix

alea createTracks <-s/-p> bamPrefix strain1 strain2 genome1.refmap
     genome2.refmap chrom.sizes outputDIR

alea createReport bamPrefix strain1 strain2 coordinates
```

Installation

ALEA2 has been developed to use a Docker container, isolating the execution from the operating system and ensuring all the dependencies are available

The latest version can be found at `https://github.com/hyounesy/ALEA/tree/master/docker`. Once the Dockerfile is downloaded, the container image can be built using:

```
docker build -t taskkoike/alea:2.0 /path/to/dockerfile-directory/
```

Hardware requirements

It is recommended to run ALEA on a 64-bit UNIX based machine with a minimum of 32GB available RAM and 60 GB available disk space. However, use of the Docker container renders ALEA2 OS-independent, it can be used under Windows or Mac OS.

Please note that ALEA uses STAR for RNA sequencing. Allele-specific STAR is memory intensive, the developer recommends 64GB of memory and 100GB of disk space.

Test Data

Our allele-specific pipeline was tested using Mouse RNA-seq, H3K27me3 ChIP-seq and Bisulphite Sequencing datasets. The original data is available for download under the GEO accession numbers GSM1625868, GSM2041978 and GSM1386021, respectively. Variant sequence data was obtained from UCSC (see Appendix A).

The test data can be downloaded as detailed in Appendices C, D and E, respectively.

UCSC Track Hubs

UCSC track hubs are a method of organising multiple genomic tracks to allow for unified data visualisation and interpretation. The ALEA2 createReport module creates a folder of UCSC-compatible bigWig files, which can be uploaded to UCSC genome browser to check the regions of interest, as shown in Figure 4 taken from Albert et al., 2017.

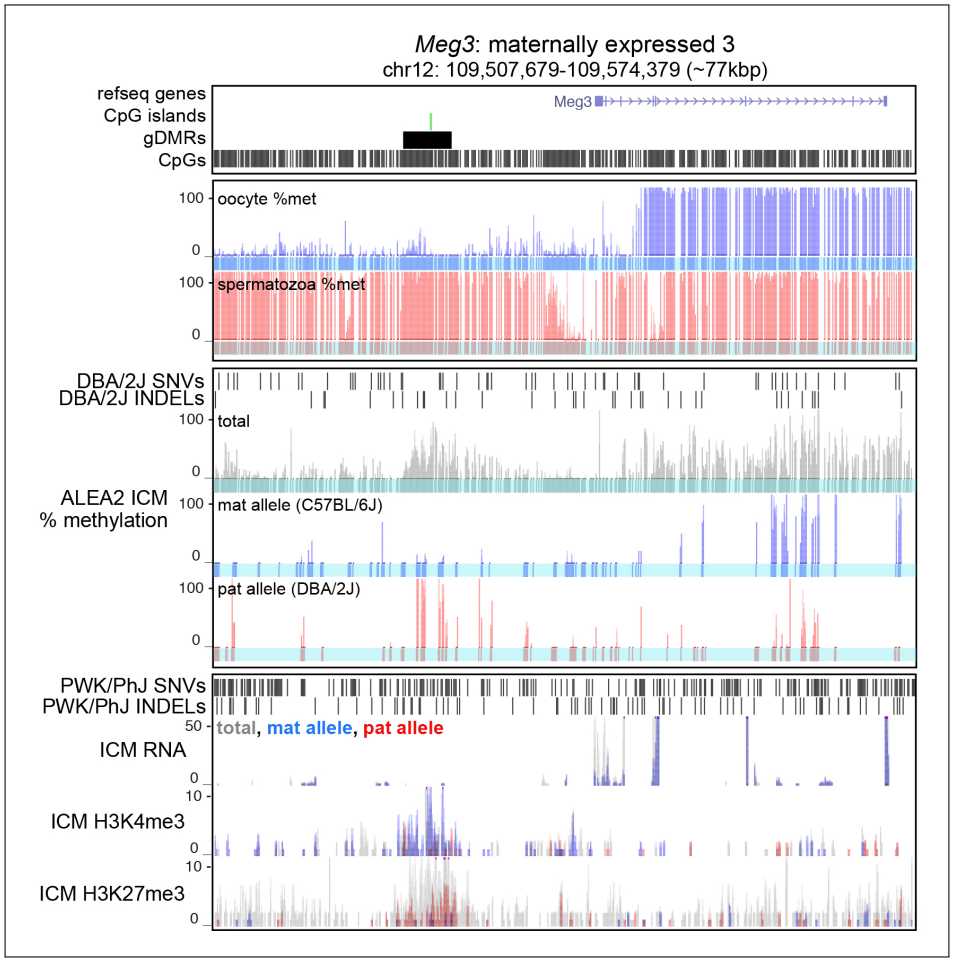

Figure 4: Genome browser screenshot, taken from Albert et al., 2017, of the Meg3 gDMR and downstream gene for visualization of parent-of-origin genetic imprinting (bisulphite-, RNA- and ChIP-seq) in ICM cells. Allelic RNA- and ChIP-datasets (bottom 3 tracks), ALEA2 automatically generates composite tracks containing total (grey), reference (blue) and non-reference (red) genomic tracks

.

Running ALEA2

Nota Bene *- The following environmental variable is set and used throughout this user guide to simplify both the running of commands in the terminal and the representation of those commands within this guide. Please ensure you set the variable using the correct syntax for the operating system upon which you are working. Also, the* `'-v'` *option links the host directory to the container mount point* `'/alea-data'`*, please use this mount point to ensure the configuration file is used correctly.*

```
export ALEA2='docker run -v <path/to/host/alea/data>:/alea-data -i
    taskkoike/alea:2.0'
```

Also, Docker containers will remain resident on the system until removal. This is easily achieved using

```
docker container prune
docker rm $(docker ps --filter status=exited --quiet) # remove
    stopped docker processes
docker rmi `docker images -aq`
```

Setting ALEA2 Options

```
$ALEA2 setting
```

In order to configure ALEA2, the `setting` function presents the user with the following interactive menu in preparation for the analysis. Simply enter the number corresponding to the desired setting for each:

```
Which type of analysis do you want to do?
```

1. `ChIPseq`
2. `RNAseq`
3. `BSseq`

Which type of alignment tool do you want to use?

1. BWA
2. Bowtie2
3. STAR
4. Tophat2
5. Bismarck

How would you like to specify the reference genome location?

1. Keep current settings
2. Automatically download genome, or change between pre-existing builds (recommended)
3. Manually set file paths to local files

Genotype phasing

For human samples, genotypes are typically called from whole genome-sequencing (WGS) short reads accompanying the epigenomic dataset under allele-specific study. SHAPEIT2 is then employed to phase genotypes using a publicly available reference panel of haplotypes provided by the 1000 Genomes project (Appendix F explains the details of running SHAPEIT2 using a reference panel of haplotypes). The output phased haplotype (.haps) is used together with the original unphased variant file (VCF) to create a phased variant file with two haplotypes containing homozygous and phased heterozygous SNPs and Indels (Figure 1a). For mouse datasets, ALEA2 accepts epigenomic marks from F1 hybrid offspring whose parents are among the 17 inbred strains available from the Mouse Genome Project. Therefore, the phased variant file is already available and the phasing step is not required. The phased SNPs and Indels files can be downloaded and placed under `<path/to/host/alea-data>` as described in appendix A. The following command are only required on human samples.

Usage:

```
$ALEA2 phaseVCF hapsDIR unphased.vcf outputPrefix
```

Options:

hapsDIR path to the directory containing the .haps files

unphased.vcf path to the vcf file containing unphased SNPs and Indels

outputPrefix output file prefix including the path but not the extension

Output:
creates the file outputPrefix.vcf.gz

In silico genome creation

The phased variant file is fed with the reference genome into the second module (Figure 1b) where haplotype regions are reconstructed from the individual haplotypes. Two *in silico* genomes are created for the use in identification of chromosomal or allele specific effects. For mouse, the two *in silico* genomes represent the parental haplotypes, however for human the two genomes will essentially be a mosaic of the parental haplotypes due to the very low likelihood of true long-range phasing. After creating the two *in silico* genomes, this module then concatenates these parental genomes into an artificial genome which is twice the size of the parental genomes. This new genome is used later for finding haplotype-specific epigenomic reads. The following commands can be used for both human and mouse samples.

The names of strains can be found from the phased.vcf.gz file. For human, it is always "hap1" and hap2" as used by SHAPEIT2 for phased haplotypes. For mouse, these are the names of the parental inbred mice crossing to create F1 hybrid mouse under study (e.g. CASTEiJ (Cast) and C57BL6J (B6) mice). Our tool accepts 17 mouse strains from the following list: C57BL6J, 129S1, AJ, AKR, BALBcJ, C3HHeJ, C57BL6NJ, CASTEiJ, CBAJ, DBA2J, FVB_NJ, LPJ, NODShiLtJ, NZO, PWKPhJ, Spretus, WSBEiJ.

Usage:

```
$ALEA2 createGenome phased.vcf.gz strain1 strain2 outputDir
```

```
$ALEA2 createGenome -snps-indels-separately phased_snps.vcf.gz
     phased_indels.vcf.gz strain1 strain2 outputDir
```

Options:

phased.vcf.gz	the phased variants vcf file (including SNPs and Indels)
strain1	name of strain1 exactly as specified in the vcf file (e.g. hap1)
strain2	name of strain2 exactly as specified in the vcf file (e.g. hap2)
outputDir	location of the output directory
-snps-indels-separately	use if SNPs and Indels are in two separate vcf files
phased-snps.vcf.gz	the phased SNPs (should be specified first)
phased-indels.vcf.gz	the phased Indels (should be specified second)

Output:

Creates two parental *in silico* genomes strain1.fasta and strain2.fasta as well as alignment indices, and a concatenated genome strain1_strain2.fasta.

Note:

It is possible to have SNPs and Indels in two separate vcf files. In that case use `-snps-indels-separately` option, and make sure you specify SNPs before Indels.

Allele-specific Alignment

We use BWA or Bowtie to align ChIP-seq, Tophat2 or STAR to align RNA-seq and Bismark to align bisulphite-seq reads in FASTQ format to the *in silico* genomes constructed by the previous module and detect the allelic reads that are uniquely aligned to each genome (Figure 1c). These reads map to the regions containing heterozygous SNPs and can be used to determine the allelic ratios for those regions. All reads aligned to multiple locations of either haplotypes or aligned to both haplotypes are filtered out. Thus, the module captures the haplotype-specific reads aligned only to one of the haplotypes. Samtools is employed to convert and sort the generated SAM files to BAM format. The following command can be applied on both human and mouse data. To find eligible names for human and mouse strains, please see the ***In silico* genome creation** subsection

Usage:

```
$ALEA2 alignReads <-s/-p> input_reads_1 [input_reads_2] genome_concat
     strain1 strain2 outputPrefix
```

Options:

-s	to align single-end reads (requires one input file)
-p	to align paired-end reads (requires two input files)
input_reads_1	the 1st input reads file in fastq. (fastq.gz or bam is supported when using BWA)
input_reads_2	(paired end) the 2nd input reads file in fastq. (fastq.gz or bam is supported when using BWA)
genome_concat	path to the indexed reference for concatenated *in silico* genome. for BWA, specifiy path to the fasta. for Bowtie2 and Tophat2, specify path and basename of index files for Bismark, specify genome folder, excluding <Bisulphite_Genome>
genome_reference	the reference fasta file previously used for *in silico* genome creation path to the indexed reference genome. for BWA, specifiy path to the fasta. for Bowtie, specify basename of index file for Bismark, specify genome folder
strain1	name of strain1 (e.g. hap1 or CASTEiJ)
strain2	name of strain2 (e.g. hap2 or C57BL6J)
outputPrefix	prefix for output files, including the full path, without an extension (e.g. ./TSC_H3K36me3)

Output:

outputPrefix_strain1_strain2.sam all reads aligned to the concatenated *in silico* genome

outputPrefix_strain1.bam allelic reads for strain1 genome (sorted bam)

outputPrefix_strain2.bam allelic reads for strain2 genome (sorted bam)

Examples:

```
$ALEA2 alignReads -s H3K36me3.fastq C57BL6J_CAST_EiJ.fasta mm10.fasta
     CAST_EiJ C57BL6J ./H3K36me3
```

```
$ALEA2 alignReads -p H3K36me3_1.fastq H3K36me3_2.fastq
     C57BL6J_CAST_EiJ.fasta CAST_EiJ C57BL6J ./H3K36me3
```

When using STAR for RNASeq

```
$ALEA2 alignReads -s H3K36me3.fastq
     createGenomeDir/STAR-index/hap1_hap2/ hap1 hap2
     ./hap1_hap2_H3K36me3
```

When using BOWTIE2 for DNA or TOPHAT for RNASeq

```
$ALEA2 alignReads -s H3K36me3.fastq
     createGenomeDir/bowtie2-index/hap1_hap2 hap1 hap2
     ./hap1_hap2_H3K36me3
```

When using BISMARK for WGBS

```
$ALEA2 alignreads -p WGBS_1.fastq WGBS_2.fastq
     createGenomeDir/hap1_hap2/ hap1 hap2 ./hap1_hap2_WGBS
```

Projecting to reference genome

The fourth module (Figure 1d) generates compressed Wig-format files which are then projected back onto the reference genome, producing strain-specific BedGraph and BigWig output files. The existence of Indels in the construction of parental *in silico* genomes causes the coordinates of the tracks to be offset relative to the reference genome. As most visualization tools require alignment to reference genomes, the refmap file created with the *in silico* genomes are used by ALEA2 to map the tracks back to the reference genome. The following command can be applied on both human and mouse data. To find eligible names for human and mouse strains, please see the ***In silico* genome creation** subsection.

Usage:

```
$ALEA createTracks <-s/-p> bamPrefix strain1 strain2 genome1.refmap
     genome2.refmap outputDIR
```

Options:

-s	to create tracks for single-end aligned reads
-p	to create tracks for paired-end aligned reads
bamPrefix	prefix used for the output of alignReads command
strain1	name of strain1 (e.g. hap1)
strain2	name of strain2 (e.g. hap2)
genome1.refmap	path to the refmap file created for *in silico* genome 1
genome2.refmap	path to the refmap file created for *in silico* genome 2
outputDIR	output directory (where to create track files)

Output:

outputDIR/outputPrefix_strain1.bedGraph

outputDIR/outputPrefix_strain1.bw read profiles for strain1 projected to reference genome

outputDIR/outputPrefix_strain2.bedGraph

outputDIR/outputPrefix_strain2.bw read profiles for strain2 projected to reference genome

outputDIR/outputPrefix_strain1.wig.gz

outputDIR/outputPrefix_strain2.wig.gz unprojected read profiles for strain1 and strain2

Reporting the tracks

createReport function provides the output from ALEA2 in an easily-accessible format that allows biologists with little computational experience to visualize the pipeline output. This is achieved by taking an allele-agnostic bam file and output files from `createTracks` and generating normalised BedGraph files for given intervals in all the input files. These BedGraphs are converted to BigWig and arranged in a UCSC trackhub-compatible hierarchical file structure. Allelic coverage and total RPKM are calculated over each interval, and output to a tab-delimited text file. These files can be uploaded to the UCSC genome browser for integrated visualization of allele-specific epigenomic tracks.

Usage:

```
$ALEA2 createReport alignReadsDir/bedGraphPrefix strain1 strain2
     interval_file
```

Options:

input_bam input bam generated by aligning to the reference genome (PREFIX_total.bam)

bedGraphPrefix bedGraphPrefix used in createTracks, can be path to file (e.g. alignReadsDir/bam_prefix). do not include _total.bedGraph in bedGraphPrefix

strain1 name of strain 1 (e.g. hap1)

strain2 name of strain 2 (e.g. hap2)

intervals General Feature Format (GFF) used for counting reads supplied in folder ./gff/ if using custom gff, please match supplied format

Output:

<interval file>_<date> a table with allelic coverage, total RPKM, parental ratios and other statistics for each interval in the GFF file.

Examples:

➢ using BWA (CHiP-Seq)

```
$ALEA2 createReport H3K4me3Liver C57BL6J CAST_EiJ
     mm10_transcription_start_sites.bed5
```

➢ Using STAR (RNA-Seq)

```
$ALEA2 createReport F1hybridLiver C57BL6J CAST_EiJ
     mm10_exons_RNA.bed5
```

Examples

Mouse RNA-Seq

Test data for GEO accession number GSM1625868 can be downloaded as described in Appendix C.

For mouse datasets, ALEA accepts epigenomic marks from F1 hybrid offspring whose parents are among the 17 inbred strains available from the Mouse Genome Project. Therefore, the phased variant file is already available and we do not need the phasing step.

➢ ALEA2 configuration

```
$ALEA2 setting
```

➢ *in silico* genome creation

```
$ALEA2 createGenome -snps-indels-separately \
        /alea-data/mm10/mgp.v5.merged.snps_all.dbSNP142.vcf.gz \
        /alea-data/mm10/mgp.v5.merged.indels.dbSNP142.normed.vcf.gz \
        PWK_PhJ C57BL6J /alea-data/mousernaseq
```

➢ Allele-specific alignment

```
$ALEA2 alignReads -s \
    /alea-data/H3K36me3.fastq \
    /alea-data/mousernaseq/bowtie2-index/PWK_PhJ_C57BL6J \
    PWK_PhJ C57BL6J /alea-data/mousernaseq/rs
```

➢ Projecting back to the reference genome

```
$ALEA2 createTracks -s /alea-data/mousernaseq/rs  PWK_PhJ C57BL6J \
    /alea-data/mousernaseq/PWK_PhJ.fasta.refmap \
    /alea-data/mousernaseq/C57BL6J.fasta.refmap \
    /alea-data/mousernaseq
```

➢ Create UCSC-compatible trackhub

```
$ALEA2 createReport /alea-data/mousernaseq/rs  PWK_PhJ C57BL6J \
    /alea-data/mm10/mm10_exons_RNA.bed6
```

Whilst creating the report for the RNAseq, ALEA2 uses a six column bed file format, a "transcript ID" column as column 6, in order to allow reporting the average value of all exons from a transcript.

Using TopHat2 and mm10 as the reference strain.

➢ *in silico* genome creation

```
$ALEA2 createGenome -snps-indels-separately \
        /alea-data/mm10/mgp.v5.merged.snps_all.dbSNP142.vcf.gz \
        /alea-data/mm10/mgp.v5.merged.indels.dbSNP142.normed.vcf.gz \
        PWK_PhJ C57BL6J /alea-data/mousernaseq
```

➢ Allele-specific alignment

```
$ALEA2 alignReads -s \
    /alea-data/H3K36me3.fastq \
    /alea-data/mousernaseq/bowtie2-index/PWK_PhJ_C57BL6J \
    PWK_PhJ C57BL6J /alea-data/mousernaseq/rs
```

➢ Projecting back to the reference genome

```
$ALEA2 createTracks -s /alea-data/mousernaseq/rs  PWK_PhJ C57BL6J \
    /alea-data/mousernaseq/PWK_PhJ.fasta.refmap \
    /alea-data/mousernaseq/C57BL6J.fasta.refmap \
    /alea-data/mousernaseq
```

➢ Create UCSC-compatible trackhub

```
$ALEA2 createReport /alea-data/mousernaseq/rs  PWK_PhJ C57BL6J \
    /alea-data/mm10/mm10_exons_RNA.bed6
```

Whilst creating the report for the RNAseq, ALEA2 uses a six column bed file format, a "transcript ID" column as column 6, in order to allow reporting the average value of all exons from a transcript.

Mouse ChIP-Seq

Test data for GEO accession number GSM2041078 can be downloaded as described in Appendix D.

For mouse datasets, ALEA2 accepts epigenomic marks from F1 hybrid offspring whose parents are among the 17 inbred strains available from the Mouse Genome Project. Therefore, the phased variant file is already available and we do not need the phasing step.

➢ Alea2 configuration

```
$ALEA2 setting
```

This examplar uses Bowtie2 for alignment and mm10 as the reference strain.

➢ *in silico* genome creation

```
$ALEA2 createGenome -snps-indels-separately \
        /alea-data/mm10/mgp.v5.merged.snps_all.dbSNP142.vcf.gz \
        /alea-data/mm10/mgp.v5.merged.indels.dbSNP142.normed.vcf.gz \
        PWK_PhJ C57BL6J /alea-data/mousechipseq
```

➢ Allele-specific alignment

```
$ALEA2 alignReads -s /alea-data/mousechipseq/SRR4022245.fastq \
    /alea-data/mousechipseq/bowtie2-index/PWK_PhJ_C57BL6J \
    PWK_PhJ C57BL6J /alea-data/mousechipseq/msc_
```

➢ Projecting back to the reference genome

```
$ALEA2 createTracks -s /alea-data/mousechipseq/msc_ \
    PWK_PhJ C57BL6J \
    /alea-data/mousechipseq/PWK_PhJ.fasta.refmap \
    /alea-data/mousechipseq/C57BL6J.fasta.refmap \
    /alea-data/mousechipseq
```

➢ Create UCSC-compatible trackhub

```
$ALEA2 createReport  /alea-data/mousechipseq/msc_PWK_PHJ_C57BL6J  \
    PWK_PhJ C57BL6J
         /alea-data/mm10/mm10_transcription_start_sites.bed5
```

Mouse Whole Genome Bisulphite Sequencing

Test data, from the GEO sample GSM1386021, should be downloaded as described in Appendix E.

For mouse datasets, ALEA2 accepts epigenomic marks from F1 hybrid offspring whose parents are among the 17 inbred strains available from the Mouse Genome Project. Therefore, the phased variant file is already available and we do not need the phasing step.

➢ Alea2 configuration

```
$ALEA2 setting
```

➢ *in silico* genome creation

```
$ALEA2 createGenome -snps-indels-separately \
    /alea-data/mm10/mgp.v5.merged.snps_all.dbSNP142.vcf.gz \
    /alea-data/mm10/mgp.v5.merged.indels.dbSNP142.normed.vcf.gz \
    DBA_2J C57BL6J /alea-data/mousebs
```

➢ Allele-specific alignment

```
$ALEA2 alignReads -p \
    /alea-data/mousebs/SRR1286778_1.fastq \
    /alea-data/mousebs/SRR1286778_2.fastq \
    /alea-data/mousebs/DBA_2J_C57BL6J \
    DBA_2J C57BL6J /alea-data/mousebs/bs
```

➢ Projecting back to the reference genome

```
$ALEA2 createTracks -p /alea-data/mousebs/bs DBA_2J C57BL6J \
    /alea-data/mousebs/DBA_2J.fasta.refmap \
    /alea-data/mousebs/C57BL6J.fasta.refmap \
    /alea-data/mousebs/bs_out
```

➢ Create UCSC-compatible trackhub

```
$ALEA2 createReport  /alea-data/mousebs/bs_out/bs  \
    DBA_2J C57BL6J
         /alea-data/mm10/mm10_transcription_start_sites.bed5
```

Appendix

A. Downloading Test Data - Mouse mm10 references

Whilst ALEA2 setting will download the mm10 reference information for the analysis, it is still necessary to separately down load the phased SNP and Indel information.

```
mkdir ../mm10
cd !$
wget --timestamping 'ftp://ftp-mouse.sanger.ac.uk/current_snps/mgp*'
md5sum -c *.md5
```

B. Downloading Test Data - Mouse mm10 Interval Data

In order to create UCSC gene browser-compatible tracks, ALEA2 requires bed format interval files. These files are currently provided within the source `.tar.gz` file.

C. Downloading Test Data - Mouse RNA-Seq Example

The samples used in the RNA sequencing are publicly available from the NCBI (`https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM1625868`) and can be downloaded using the NCBI SRA tools (`https://www.ncbi.nlm.nih.gov/sra`).

```
mkdir ../mouserna
cd !$
prefetch SRR1840522
fastq-dump --split-3 SRR1840522
wget -c \
ftp://ftp.bcgsc.ca/supplementary/ALEA/files/test-data/mouse/H3K36me3.fastq.gz
gunzip H3K36me3.fastq.gz
```

D. Downloading Test Data - Mouse ChIP-Seq Example

The samples used in the CHiP sequencing are publicly available from the NCBI (`https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM2041078`) and can be downloaded using the NCBI SRA tools (`https://www.ncbi.nlm.nih.gov/sra`).

```
mkdir ../mousechipseq
cd !$
prefetch SRR4022245
fastq-dump --split-3 SRR4022245
```

E. Downloading Test Data - Mouse Bisulphite Example

The samples used in the Bisuphite sequencing are publicly available from the NCBI (`https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM1386021`) and can be downloaded using the NCBI SRA tools (`https://www.ncbi.nlm.nih.gov/sra`).

```
mkdir ../mousebs
cd !$
prefetch SRR1286778
fastq-dump --split-3 SRR1286778
```

F. Using SHAPEIT2 for genotype phasing

We employed SHAPEIT2 for genotype phasing. Since we generated the genotype of our human data in VCF format, we needed a few pre-processing steps to convert our VCF file to the most appropriate format used by SHAPEIT2, i.e. Plink BED/BIM/FAM format. This can be done by a series of commands running VCFtools and Plink before we run SHAPEIT2:

➢ Converting VCF format to Plink PED/MAP format

```
vcftools --vcf skin01_a20921.vcf --force-index-write --plink --out
     skin01_a20921
```

➢ Splitting the genotypes in PED/MAP format by chromosomes and converting them to BED/BIM/FAM format

```
for chr in $(seq 1 22) ;
do
plink --file skin01_a20921 --chr $chr --recode --out chr$chr --noweb
     ;
done

plink --file skin01_a20921 --chr X --recode --out chrX --noweb

for chr in $(seq 1 22) ;
do
plink --file chr$chr --make-bed --out chr$chr --noweb;
done

plink --file chrX --make-bed --out chrX --noweb
```

To run SHAPEIT2 on the reference panel of haplotypes provided by the 1000 Genomes project, one should first download the 1,000 Genomes phase1 haplotype set in the correct format. The following two commands run SHAPEIT2 on split genotype for each chromosome (in BED/BIM/FAM format) using reference panel of haplotypes:

➢ Alignment of the SNPs between the genotype and the reference panel

```
/bin/SHAPEIT/shapeit.v2.r644.linux.x86_64 -check
-B ../skin01_a20921_break/chr1
-M /1000_Genomes_phase1_haplotype/⌟
     ALL_1000G_phase1integrated_v3_impute/⌟
     genetic_map_chr1_combined_b37.txt
-R /1000_Genomes_phase1_haplotype/⌟
     ALL_1000G_phase1integrated_v3_impute/⌟
     ALL_1000G_phase1integrated_v3_chr1_impute.hap.gz
/1000_Genomes_phase1_haplotype/ALL_1000G_phase1integrated_v3_impute/⌟
     ALL_1000G_phase1integrated_v3_chr1_impute.legend.gz
/1000_Genomes_phase1_haplotype/ALL_1000G_phase1integrated_v3_impute/⌟
     ALL_1000G_phase1integrated_v3.sample --output-log
     chr1.alignments
```

➢ Phasing the genotype using the reference panel of haplotypes

```
/bin/SHAPEIT/shapeit.v2.r644.linux.x86_64
-B ../skin01_a20921_break/chr1
-M /1000_Genomes_phase1_haplotype/
     ALL_1000G_phase1integrated_v3_impute/
     genetic_map_chr1_combined_b37.txt
-R /1000_Genomes_phase1_haplotype/
     ALL_1000G_phase1integrated_v3_impute/
     ALL_1000G_phase1integrated_v3_chr1_impute.hap.gz
/1000_Genomes_phase1_haplotype/ALL_1000G_phase1integrated_v3_impute/
     ALL_1000G_phase1integrated_v3_chr1_impute.legend.gz
/1000_Genomes_phase1_haplotype/ALL_1000G_phase1integrated_v3_impute/
     ALL_1000G_phase1integrated_v3.sample -O chr1.phased
     --exclude-snp
chr1.alignments.snp.strand.exclude --no-mcmc --thread 8
```

References

Albert, Julien Richard et al. (2017). "ALEA2.0: an expanded computational toolbox for allele-specific epigenomic analysis". In: *Submitted*.

Younesy, Hamid et al. (Apr. 2014). "ALEA: a toolbox for allele-specific epigenomics analysis". In: *Bioinformatics* 30.8, pp. 1172–1174. URL: http://dx.doi.org/10.1093/bioinformatics/btt744.