Table of Contents

1. Introduction to MELT

The Mobile Element Locator Tool (MELT) is a software package, written in Java, that discovers, annotates, and genotypes non-reference Mobile Element Insertions (MEIs) in Illumina DNA paired-end whole genome sequencing (WGS) data. MELT was first conceived as a tool to identify non-reference MEIs in large genome sequencing projects, specifically as part of the 1000 Genomes Project, and has been further optimized to run on a wide range of data types. MELT is optimized for performing discovery in a large number of individual sequencing samples using the Sun Grid Engine (SGE). MELT also has two additional workflows: analysis without SGE (for adaptability to other parallel computing platforms) and single genome analysis. MELT is highly scalable for many different types of data, and the different workflows are outlined and detailed in this documentation. MELT was also programmed to be user friendly, with only a small number of common and easily installed bioinformatic software dependencies (see Required software dependencies). Input is standard BWA (MEM or ALN) WGS alignment[1], while output is in the Variant Call File (VCF) 4.1 format[2]. This allows users to integrate MELT into pre-existing pipelines that utilize BWA alignments and accept VCF as input. Additionally, In keeping with the user-friendly design ideology, MELT operates with several safeguards to ensure uninterrupted runtime from beginning to end, with minimal user interaction.

MELT uses an approach that is fairly standard for many forms of Structural Variation (SV) discovery, but has been highly specialized for MEI discovery. In short, MELT first collects all discordant pairs from a WGS alignment, and aligns them to provided MEI references using the Bowtie2[3] sequencing aligner. It then 'walks' across the reference genome and creates a list of putative MEIs based on total read support at each putative site. MELT will then merge the initial MEI calls across the datasets provided, and determine specific information and exact breakpoints for each of the putative MEIs. Finally, all sites are genotyped according to a modified (for SV genotyping) version of Heng Li's genotype likelihood equation[4], and subsequently filtered based on true positive calls. Output is then generated in VCF format.

1.1. Readme Conventions

This readme uses the following conventions:

- A table with the following format contains options for the MELT **Runtime** being covered in the current section:

Option	Required?	Description
a	YES	A Required option
b	NO	An optional option

- Commands you should input into your terminal are given as:

```
$ cmd options for command
```

- Specific MELT Runtimes (Programs within the MELT.jar) are highlighted in **bold red** when discussed in the text (They will not be highlighted in command line examples).
- Important information is marked by '***Note:***'.

2. MELT System Requirements

2.1. Development

MELT is coded entirely in Java, specifically Java SE Runtime Environment build 1.7.0_65-b14-462 (Java 1.7). MELT was developed using a machine with the Red Hat Enterprise Linux Server release 5.9 OS with a 64-bit (x86_64) processor architecture with the following system specs:

- 8-Core Intel Xeon CPU with 3.00GHz/core
- 16GB RAM

MELT has also been tested using Java 1.8, with no differences between JREs during tests.

2.2. Software Testing

All three MELT workflows have been tested on the following linux setup:

- Local Machine: Red Hat Enterprise Linux Server release 5.9
- Java VM: Java HotSpot 64-Bit Server VM (build 20.6-b01, mixed mode)
- Grid Architecture: Sun Grid Engine (GE vers. 6.2u5)

MELT-SPLIT and MELT-SINGLE workflows have been tested on the following Macintosh setup:

- Local Machine: Macintosh OSX Yosemite version 10.10.2
- Java VM: Java HotSpot 64-Bit Server VM (build 25.0-b70, mixed mode)

MELT was coded in Java for its high level of portability. It is expected that as long as Java 1.7 is provided as the JRE, MELT should function on any standard *nix machine.

Note: Please do not attempt to run MELT on Windows PCs, as the file structure and environment assumptions are much different then those on *nix based systems.

2.3. Hardware Requirements

If running locally, MELT minimally requires 6GB of system memory, and at least a single x64 3.00 GHz CPU (This is simply for MELT, and do not include system overhead). In addition, MELT requires ~20GB of TEMPORARY storage space per 60X genome in order to run. This does not represent final storage size for MELT data.

When running MELT using **SGE**, additional pieces of information should be taken into account:

1. Number of nodes.
2. Number of memory on each node.

As an example, each MELT job submitted to the grid uses a maximum of 6GB of memory. If the grid is made up of 4 nodes with 12GB of memory each, then it is advised to limit the total number of jobs to 7 jobs at a time (12 x 4 = 48, 32 GB of total memory, 7 jobs = 42 GB used by MELT with room for overhead). MELT will communicate to the grid the amount of memory needed for each job and throttle jobs based on provided user parameters. SGE additionally accounts for this in job submission, but it is advised to run MELT with an eye towards grid submission stability. And as always, please take into account your fellow grid users!

3. Installing MELT

3.1. MELT Installation

MELT installation is fairly straightforward. First, extract the MELT tarball:

```
$ tar xf MELTvX.X.tar
```

This will create a MELTvX.X directory in your current directory (./MELTvX.X/). This folder will contain the MELT.jar jar file. MELT runs from a single .jar file, with the first argument representing the specific runtime that you want to execute. For example:

```
$ java –jar MELT.jar SGE <SGE Options>
```

Would run the MELT-SGE runtime. Possible MELT Runtimes are described in short below (or by providing --help/-help/-h to MELT.jar), and in more detail further on in the README:

Table 1: MELT Runtimes

Runtime Name	Purpose
Preprocess	Processes BAM Alignments for input into MELT
SGE	SGE Executable
Single	Run MELT analysis on a Single .bam file
IndivAnalysis	Step 1 of Non-SGE MELT Pipeline
GroupAnalysis	Step 2 of Non-SGE MELT Pipeline
Genotype	Step 3 of Non-SGE MELT Pipeline
MakeVCF	Step 4 of Non-SGE MELT Pipeline
Deletion-Genotype	MELT Reference MEI Deletion Genotyper
Deletion-Merge	MELT Reference MEI Deletion VCF writer
CALU	CAlu Alu classification software
LINEU	LineU L1 classification software
BuildTransposonZIP	Build custom TE discovery repositories

You can also get help for individual runtimes by doing:

```
$ java –jar MELT.jar <Runtime> --help/-help/-h
```

There are also several other folders included in the ./MELTvX.X/ directory that will be explained at length later in this tutorial.

Next, decide whether or not you will be running MELT using SGE or not. If running with SGE, please read the section Installing MELT Using SGE, otherwise, you can skip ahead to Required Software dependencies.

3.2. Installing MELT Using SGE

To run MELT using SGE, the environment variable $MELT_EXEC_SITE must be set to the full location of MELT.jar. Please ensure that this is the FULL PATH, as if run on SGE, MELT will not run without it! This can be set at runtime, or can be added to your path. An example of both follows:

- At runtime:

```
$ export MELT_EXEC_SITE=/full/path/to/install/dir/MELTvX.X/MELT.jar
```

- In system path (add the following to your .bashrc file [should be located in your home folder):

```
export MELT_EXEC_SITE=/path/to/install/dir/MELTvX.X/MELT.jar
```

Additionally, to run on SGE, MELT requires the SGE library dir for your grid system. You should be able to find this using the following command:

```
$ echo $SGE_ROOT
```

This will list a directory, and the lib directory will be within this. As an example, $echo SGE_ROOT could return /usr/local/packages/sge-root/ and the required lib dir for MELT will be /usr/local/packages/sge-root/lib/lx24-amd64/. There are typically several folders in the lib directory – please check with your System Admin to determine which one is correct (lx24-amd64 is for the 64-bit AMD architecture present at University of Maryland, Baltimore).

3.3. Required Software Dependencies

The only external dependency of MELT is Bowtie2 (MELT was tested with version 2.0.5 & 2.2.4. No differences were seen between versions during tests). Bowtie must be included on your system path (and Grid path if using SGE). To test this, you can simply try typing:

```
$ bowtie2 --help
```

This should show the version number and options for bowtie2 in your terminal window.

3.4. Required External File Dependencies

Several files are required for MELT to run:

1. Reference in FASTA format with .fai index file (can make .fai from fasta using samtools faidx). This can be recovered from the BAM file header by using:

   ```
   $ samtools view –H bam_to_be_analyzed.sorted.bam
   ```

   It is listed, typically, on each @SQ line in the SAM header under the flag AS (sequence name) or UR (URL download location)

   Note: Must be the same reference as used to align reads! **This is extremely important!** Not following this rule can result in any number of unexpected errors! These may include increased number of false positive/false negative sites, as well as unrecoverable runtime errors.

2. Gene Annotation in BED format. The human and chimpanzee versions of this file are included in the MELTvX.X.tar (in /add_bed_files/). Must be sorted according to coordinate. This file can be downloaded from the UCSC Genome Browser website[5,6]. Simple download the GENE track from the TABLES section of the website. The following table outlines the format of this file:

   Table 2: Gene File Format

Column	**Description**
1	chromosome (in either chrX or X format, where X is the chromosome ID)
2	Gene start position
3	Gene stop position
4	Gene ID (Any format)
5	Quality Score (default 0, no need to set, MELT does not use)
6	Strand
7	Coding start position
8	Coding stop position
9	Quality Score (default 0, no need to set, MELT does not use)
10	Number of exons
11	Exon lengths (comma delimited)
12	Exon starts (comma delimited)

   Note: Please ensure the chromosome names match those in the reference .fasta file you plan to use for BWA alignment and MELT analysis. A common example is the 'chr' prefix in front of chromosome names.

3. MEI .zip files which include the following files (A complete description is below in the section Transposon Reference Files:
   a. FASTA sequence (and .fai index)
   b. Bowtie2 index (of the same FASTA file as above)
   c. Reference insertions in BED format (for filtering known hits)

3.5. Optional External Files

MELT can also use a collection of MEI sites, or priors, from other studies. This is supplied to MELT in the form of a standard site VCF file. All columns past 9 are not used, and can be excluded when providing priors. It is recommended to use MELT discovered sites for prior input, as several pieces of information in the MELT VCF output (see MELT Output for more information) can help with improved genotyping. MELT uses priors in two ways:

1. Allow for genotyping of sites not discovered in the current data set.
2. Provide more accurate breakpoints for use during genotyping. This is only possible if MELT formatted MEIs are included as priors.

As is outlined above, priors are used mainly to increase total number of genotyped sites, and increase accuracy of that genotyping. Input of priors depends on the MELT workflow used:

- If using MELT-SGE or MELT-SINGLE, prior files are provided in the –t input file as a second column. Please see the respective workflow for a working example.
- If using MELT-SPLIT, use the –v option while running **GroupAnalysis**.

Note: Adding priors will likely increase the runtime of MELT, as all new sites must be genotyped in new samples.

After initial discovery, discovered sites are checked against the list of priors to see if any of the prior sites have been rediscovered. Matches are determined based on a +/- 100 bp window around the prior site. If a prior is matched as a hit, MELT then checks whether or not the provided prior site is a MELT formatted VCF. If using a MELT formatted VCF as prior input, the ASSESS score of the discovered site is checked against the ASSESS score in the prior file. If the ASSESS score of the prior is greater than the ASSESS score of the discovered site, the position of the prior is used for further analysis. Otherwise the discovered site position is used. If not using a MELT formatted VCF, the name (Column 3) of the prior is simply cataloged for record keeping.

A collection of VCF priors for L1 (### sites), *Alu* (sites), and SVA (sites) are provided in the folder ./MELTvX.X/prior_files/. These sites were discovered as part of the 1000 Genomes Project, Phase III. Please see the How to Cite MELT section for more information on these sites, and how to cite when using them.

Please see the section MELT Output for information on how priors are reported.

4. Transposon Reference Files

4.1. General Description

MELT requires transposon .zip files to direct MEI discovery. Six MEI.zip files (LINE1-Human-Hg19, Alu-Human-Hg19, LINE1-Chimp-PanTro4, Alu-Chimp-PanTro4, SVA-Human-Hg19, & HERVK-Human-Hg19) are included within the MELT tar-ball, and are located in the ./MELTvX.X/me_refs/ folder after extraction. The provided .zip files listed above do not require further modification in order to be used for MEI discovery. So, if you only want to discover human or chimp transposons in the Hg19 or panTro4 reference sequences, you can stop reading here and jump to the next section. If you want to discover non-reference transposons in another species or reference, please read on.

4.2. Creating Custom MEI.zip files

Advanced users can also create their own MEI reference files for use with MELT (Tested MEI refs that will run out of the box are included in the MELT tarball in the ./MELTvX.X/me_refs/ folder). A rundown on how to make your own .zip using the **BuildTransposonZIP** runtime is given step by step below:

4.2.1. The MELT .zip file structure

Each MEI_MELT.zip repository is simply a collection of 5 different files/indices. Please note – file suffixes are crucial to creating the .zip file, do not change them (i.e. .fa versus .fasta):

Table 3: mei.zip file structure

File	Short Description
Fasta sequence (.fa)	Sequence of transposon of interest in fasta format
Fasta index (.fa.fai)	.fai index of the Fasta sequence created by samtools faidx
Reference Transposon Mask (.bed)	Bed file of all reference insertions in the genome of interest.
Bowtie2 index (.bt2)	Bowtie2 index of .fa file. Created using bowtie2-build
Info file (.info)	Allows MELT to direct specificity and naming of transposon discovery

4.2.2. Deciding on an MEI Consensus

Creating a consensus sequence for the transposon you want to discover is very important to accurate MEI discovery. The simplest way to get an accurate consensus sequence is to simply look at the literature for the element of interest. If information on the element is scarce on non-existent, then the following steps may help to generate an accurate consensus for MEI discovery.

Simply using the repbase[7] provided sequence may not be enough to discover that particular transposon if it is highly variable from insertion to insertion. This is typically true of the smaller SINE elements from genome to genome, but typically not true for the larger LINE elements. Using a combination of the MELT-Deletion caller and RepeatMasker, one can evaluate the genome of interest for transposons that are polymorphic in the reference genome (i.e. when sequencing another individual, these transposons are 'deleted').

1. Run the MELT-Deletion caller on the pre-existing RepeatMasker masked sites in the genome.
2. Extract the fasta sequence of around 100 polymorphic sites
3. Perform a Multiple Sequence Alignment (MSA) using one of several availbile software packages[8,9]
4. Generate a consensus from this MSA using a tool such as cons from the EMBOSS package[10].
5. Remove any Ns from the consensus (these typically represent InDels in some, but not all, extracted elements), as they will hamper MELT MEI discovery.
6. Use this generated consensus when running BuildTransposonZIP.

4.2.3. MELT-BuildTransposonZIP

General Usage:

```
java –Xmx1G –jar MELT.jar BuildTransposonZIP \
   Transposon_Sequence.fa|Transposon_Sequence.fasta \
   Transposon_Mask.bed \
   NAME[A-Z,a-z,0-9] \
   ERROR[0-9+]
```

File suffixes are required by **MakeTransposonZIP** to match the requirements outlined in the general usage above. Naming requirements and a more detailed explanation of each element of the command line is explained below:

Note: **BuildTransposonZIP** will rename the fasta and bed file to *.fa.old or *.bed.old if they share the same name as **BuildTransposonZIP** intermediary files.

Fasta File

Create a fasta file of the transposon reference sequence. The first line of the file can be anything as long as it matches '>[Non-Whitespace]'. No spaces allowed. This name will NOT be used at all, and will be replaced by the NAME input. Following this, either multiple or single lines of sequence can be placed below. For example, Alu is 283 bp long; you can either place these full 283 bp on one line, or split them so there are 60 (fairly standard fasta convention) characters on each line. MELT does not accept 'N' for alignment (ATCG ONLY). 'N' will confuse the alignment algorithms during discovery. Characters can be either upper OR lower case. Only include one sequence in the fasta file you want to analyze, otherwise **MakeTransposonZIP** will fail.

BED File

Create a repeat masked bed file for your element of interest. Download the repeat masker[11] track (in BED format) from the UCSC Genome Browser webpage[5,6]:

1. From the homepage go to Tables section (top of webpage)
2. Choose your organism
3. Select Repeats from the 'groups' dropdown box
4. Select Repeatmasker from the 'track' dropdown box
5. Choose BED (browser extensible data) from the 'output format' dropdown box
6. Name the file and click 'get output'

Note: MELT currently does not support discovery of Transposons that have not been masked in their respective organism.

Note: Please ensure the chromosome names match those in the Reference .fasta file you plan to use for BWA alignment and MELT analysis. A common example is the 'chr' prefix in front of chromosome names.

Next, extract all of the reference coordinates of the type of element that you will be looking for (an Alu Human example):

```
$ grep Alu hg19.repeatmasker.bed > ALU.bed
```

This will grab ALL reference coordinates for your MEI of interest. You can also create a custom mask (i.e. masking coordinates you do not wish to discover in) if you like, just ensure that the first three columns of the file are chr, start, end in that order. This will prevent discovery in and around the bed coordinates provided.

Name

Decide on a name for your element. This name will be a standard feature of each of the files that **BuildTransposonZIP** will create, and will be kept across all file types.

Note: This will affect the ability of MELT to classify Family and Subfamily for Alu and L1 (see MEI Species Classification), and twin priming for L1 if not named properly. If you want to discover primate lineage L1 or Alu and get all the features of MELT, pleasure ensure ALU or LINE1 is used for name.

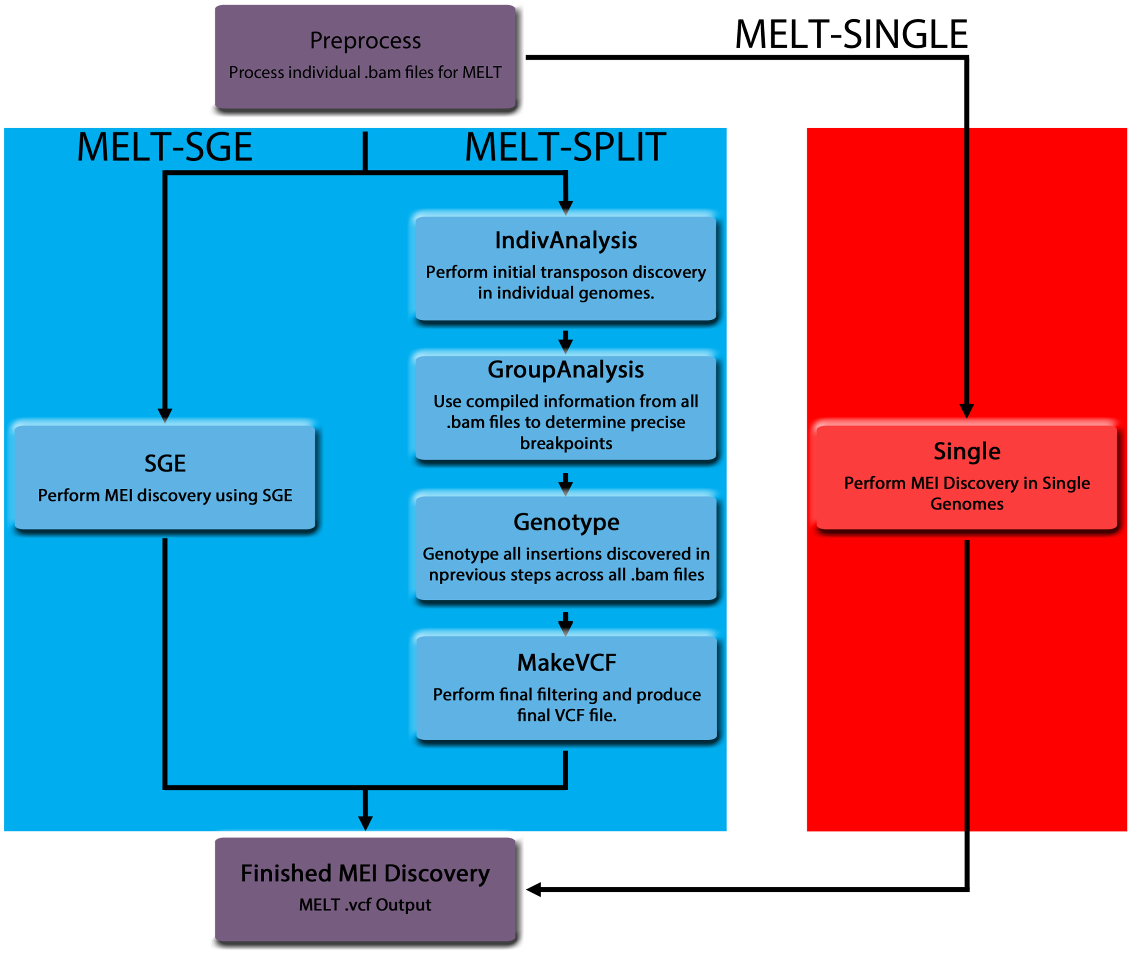
Error

ERROR tells MELT how many mismatches to allow per 100 bases of the MEI reference during alignment. This is a specificity parameter, and can only bet set here (not during MELT MEI Discovery). This number should reflect the biology of the transposon. If we were performing this for L1, we would set this number at 3, since L1 elements change at a much slower rate. *Alu* elements, on the other hand, mutate at a high rate, will be more divergent from element to element and we set ERROR to 10. Higher numbers are less specific, lower are more. The ERROR number in the .zip files included with MELT was empirically tested using the 1000 Genomes Phase III data.

5. Running MELT

As stated before, MELT can be run in one of three ways depending on the number of .bam files you wish to analyze. If running MELT on multiple .bam files (Figure 1 – in Blue), it is possible to run with SGE (MELT-SGE) or without SGE (MELT-SPLIT). If discovery is to be performed on a single .bam file (Figure 1 – In Red), MELT-SINGLE is the preferred method. A schematic describing possible workflows is shown below:

Figure 1: MELT MEI Discovery Workflows


Preprocess
Process individual .bam files for MELT
MELT-SINGLE
MELT-SGE
MELT-SPLIT
IndivAnalysis
Perform initial transposon discovery in individual genomes.
GroupAnalysis
Use compiled information from all .bam files to determine precise breakpoints
SGE
Perform MEI discovery using SGE
Genotype
Genotype all insertions discovered in nprevious steps across all .bam files
MakeVCF
Perform final filtering and produce final VCF file.
Single
Perform MEI Discovery in Single Genomes
Finished MEI Discovery
MELT .vcf Output

Each Runtime works with a slightly different set of options, but all option letters should roughly carry over between .jar files (i.e. –h is always for the reference fasta). All options discussed below are also available, albeit less verbosely, via command line:

```
$ ./MELT.jar <Runtime> --help/-help/-h
```

It is **HIGHLY** recommended to run multiple .bam files through MELT at one time, as multiple .bam files provide a large amount of power to MEI discovery.

5.1. Preprocessing .bam files for MELT

Beyond the .jar files discussed below, you can additionally 'pre-process' bam files to speed up MELT's runtime. This is a required step for MELT-SPLIT, but is optional for MELT-SGE or MELT-SINGLE (you can use the flag '-k' to tell MELT that .bam files have already been processed). To process a .bam file using **Preprocess**, simply use the following syntax:

```
$ java –Xmx2G –jar MELT.jar Preprocess <name.sorted.bam>
```

.bam files are expected to be both sorted and indexed, with at least the .bam suffix (.sorted.bam and .bam will both work). The .bai suffix is expected to be AFTER the .bam suffix, not in place of it (i.e. .sorted.bam.bai, NOT .sorted.bai). MELT will not run if this convention is not used. **Preprocess** generates three files, all as suffixes to the current bam file:

1. sorted.bam.disc – discordant pairs from the current BAM file
2. sorted.bam.disc.bai – index of these discordant pairs
3. sorted.bam.disc.fq – fastq version of all discordant pairs for MEI alignment

These files will be in the same directory as the provided .bam file, and can be used as input for further MEI discovery using MELT.

Note: If you run preprocessing on a .bam file, but forget to flag MELT-SGE with –k, processing will have to be performed again!

5.2. Which version of MELT should I use?

This depends largely on whether SGE is installed at your home institution and the number of .bam files you wish to analyze. It also depends on how much interactivity/input you would like to have running MELT. MELT-SGE is simply a SGE scheduling wrapper, and MELT-SGE, MELT-SPLIT, and MELT-SINGLE are identical programmatically. The only major differences between them being:

1. MELT-SGE handles running each of the individual steps
2. MELT-SGE & MELT-SINGLE can run multiple MEIs at once
3. MELT-SGE throttles the number of concurrent jobs on a computational grid using input from the user
4. MELT-SGE is less interactive, and the only input from the user is at the beginning of a run
5. MELT-SINGLE can only run on a single .bam file.

You can also use MELT-SPLIT and MELT-SINGLE if you have SGE, you will just have to submit and monitor the jobs manually. That means ensuring that each step completes properly, and that you are not overloading whatever system architecture/computer you are currently using. To use MELT-SGE, please read the section Running MELT using MELT-SGE. To use MELT-SPLIT, please read the section Running MELT using MELT-SPLIT. To use MELT-SINGLE, please read the section Running MELT using MELT-SINGLE.

5.3. Running MELT using MELT-SGE

5.3.1. Description

Perform MEI discovery using Sun Grid Engine (SGE). This will look in multiple genomes (-l) for multiple transposons (-t).

General usage:

```
$ java –Xmx1g -jar MELT.jar SGE <options>
```

5.3.2. Options

Note: Due to the nature of SGE, it is recommended to provide FULL PATH for all options.

Note: While –b is not a required option, it is crucial to make sure the genome you are using does not have any 'decoy' or 'fake' chromosomes. An example is the hs37d5 'decoy' chromosome attached to the 1000 Genomes human genome fasta file. This 'chromosome' works to help BWA to align reads properly to the reference[12], and if included in MELT analysis will result in increased runtime and discovery of MEIs on this fake chromosome.

An Example file_list.txt file for –l:

```
/full/path/to/alignment1.bam
/full/path/to/alignment2.bam
.
.
.
/full/path/to/alignmentN.bam
```

OR if providing coverage information with –l:

```
/full/path/to/alignment1.bam<TAB>15
/full/path/to/alignment2.bam<TAB>32
.
.
.
/full/path/to/alignmentN.bam<TAB>17
```

Note: Coverage only affects MELT analysis under 6x coverage, and the default [30x] is fine for analysis at greater coverage.

An example transposon_file_list.txt file for –t:

```
/full/path/to/LINE_MELT.zip
/full/path/to/ALU_MELT.zip
/full/path/to/SVA_MELT.zip
```

OR if utilizing priors with –t:

```
/full/path/to/LINE_MELT.zip<TAB>/full/path/to/LINE1_priors.VCF
/full/path/to/ALU_MELT.zip<TAB>/full/path/to/ALU_priors.VCF
/full/path/to/SVA_MELT.zip<TAB>/full/path/to/SVA_priors.VCF
```

Note: Priors do not have to be provided for each MEI class, but please ensure that the priors provided are only specific to the MEI type being discovered (i.e. only provide L1 priors on the LINE_MELT.zip line).

Table 4: MELT-SGE Options

Option	Required?	Description
h	YES	FULL PATH to the reference fasta. Has to be the same as the reference used to align samples being used for MEI detection, and must have a .fai index at the same location.
i	YES	The sge directory, see above section on Installing MELT Using SGE for further explanation on what this is.
l	YES	List of .bam alignments with one file per line. This is the .sorted.bam or .bam file, NOT the .disc, .disc.bai, or .fq file from MELT-Preprocess.jar, regardless of –k flag. Can also provide an additional column for coverage in the format "file<TAB>coverage". Coverage is not required, and will default to 30.
n	YES	The gene annotation for the FASTA reference (-h). Must be sorted by coordinate. An annotation for Hg19 is included in MELTvX.X.tar (hg19.genes.bed).
t	YES	List of MEI.zip files with one file per line. See above section on Transposon Reference Files for more information.
u	YES	Additional options required by the SGE runtime at your institution. I.e. if to run an SGE job you need to put 'qsub –q <que name> -P <project-name>', run MELT using –u '–q <que name> -P <project-name>'.
w	YES	The working dir for this project.
a	NO	Boolean flag to tell MELT that alignments were performed with BWA-ALN rather than BWA-MEM [**false**]
b	NO	Exclusion list for chromosomes. By default MELT performs MEI discovery on all chromosomes greater than 1Mb. To exclude chromosomes greater than 1Mb, simply list them, delimited by a forward slash. For example, to exclude chromosomes 1,2, and 4, use –b 1/2/4. [**null**]
d	NO	Minimum length of chromosome to analyze. [**1000000**]
e	NO	Fragment length of the samples. [**500**]

g	NO	Log file. [**STDOUT**]
j	NO	No call filter cutoff, see above for more exact description. [**25**]
k	NO	.bam files have already ben preprocessed. See above for description of what this means. [**false**]
m	NO	Number of jobs to be submitted to the grid at a time. [**100**]
q	NO	Alignments are pre Illumina 1.3 quality encoding [**false**]
r	NO	Read length of the supplied bam file. [**100**]
s	NO	STDEV cutoff [**2.0**]
x	NO	Number of time for any job to run on SGE, in minutes. Set higher for bigger bam files. [**120**]
z	NO	Maximum reads to load into memory at any given time. Best to leave alone, but if looking at a .bam file larger than ~200Gb, set moderately higher. [**5000**]

5.4. Running MELT Using MELT-SPLIT

5.4.1. Description

Perform MEI discovery without using Sun Grid Engine. Rather than one runtime, MELT-SPLIT uses 4 different runtimes to perform analysis. This is to ensure that the user gets the same population level MEI discovery as MELT-SGE, but without having to use SGE as a parallel computing platform. The pipeline follows the following 4 general steps:

I. **IndivAnalysis** – MEI discovery in individual samples
II. **GroupAnalysis** – Merge discovery information across all genomes in project to determine accurate breakpoint information. Also determines several MEI specific pieces of information such as length, strand, subfamily, target site duplication (TSD), among others.
III. **Genotype** – Genotype all samples using merged MEI discovery information.
IV. **MakeVCF** – Perform final filtering and merging of individual samples into final VCF.

For more information on how each individual step is performed, please see our publication (see Citing MELT below). For actual information on running each individual step, please continue following the README.

5.4.2. IndivAnalysis

General Usage:

```
java –Xmx6G –jar MELT.jar IndivAnalysis <options>
```

If performing an analysis across multiple .bam files, it is recommended to use the same working directory for each step. For example:

```
$ java –Xmx2G –jar MELT.jar IndivAnalysis –w /path/to/MEI_NAME/ -l
/path/to/file1.bam <other_options>
$ java –Xmx2G –jar MELT.jar IndivAnalysis –w /path/to/MEI_NAME/ -l
/path/to/file2.bam <other_options>
$ java –Xmx2G –jar MELT.jar IndivAnalysis –w /path/to/MEI_NAME/ -l
/path/to/file3.bam <other_options>
```

Where –w is the working directory, and –l is the .bam file of interest (described further below in Table 5). This is because **IndivAnalysis** creates several files as output, and **GroupAnalysis** looks for them all in one folder in initialization.

Note: **IndivAnalysis** assumes MELT-PreProcess.jar has been run on the supplied .bam file (-l).

Note: **IndivAnalysis** can only perform detection on one MEI at a time. Run multiple times to perform discovery for other transposons.

Note: Do not use the same working directory for multiple transposons! i.e. for LINE1 use LINE1, for Alu use ALU. This is to ensure GroupAnalysis can retrieve the proper files.

Note: While –b is not a required option, it is crucial to make sure the genome you are using does not have any 'decoy' or 'fake' chromosomes. An example is the hs37d5 'decoy' chromosome attached to the 1000 Genomes human genome fasta file. This 'chromosome' works to help BWA to align reads properly to the reference[12], and if included in MELT analysis will result in increased runtime and discovery of MEIs on this fake chromosome.

Table 5: IndivAnalysis Options

Option	Required?	Description
h	YES	FULL PATH to the reference fasta. Has to be the same as the reference used to align samples being used for MEI detection, and must have a .fai index at the same location.
l	YES	.bam alignment to perform MEI discovery on. This is the .sorted.bam or .bam file, NOT the .disc, .disc.bai, or .fq file from MELT-Preprocess.jar. PreProcessing is assumed to have been run already.
t	YES	Transposon .zip file to perform discovery. See above section on Transposon Reference Files for

		more information.
w	YES	The working dir for this project.
a	NO	Boolean flag to tell MELT that alignments were performed with BWA-ALN rather than BWA-MEM [**false**]
b	NO	Exclusion list for chromosomes. By default MELT performs MEI discovery on all chromosomes greater than 1Mb. To exclude chromosomes greater than 1Mb, simply list them, delimited by a forward slash. For example, to exclude chromosomes 1,2, and 4, use –b 1/2/4. [**null**]
c	NO	Coverage level of supplied bam file. [**30**] ***Note***: Coverage only affects MELT analysis under 6x coverage, and the default [30x] is fine for analysis at greater coverage.
d	NO	Minimum length of chromosome to analyze. [**1000000**]
q	NO	Alignment is pre Illumina 1.3 quality encoding [**false**]
r	NO	Read length of the supplied bam files [**100**]
z	NO	Maximum reads to load into memory at any given time. Best to leave alone, but if looking at a .bam file larger than ~200Gb, set moderately higher. [**5000**]

5.4.3. GroupAnalysis

General Usage:

```
$ java –Xmx4g –jar MELT.jar GroupAnalysis <options>
```

The key output from this process is an <MEI_EXAMINED>.pre_geno.tsv file. This file is used in the input of the final two steps.

The same rules used for **IndivAnalysis** apply here. i.e. Ensure you only analyze one transposon (-t) at a time, and use the same working directory (-w) for all samples.

Table 6: GroupAnalysis Options

Option	Required?	Description
h	YES	FULL PATH to the reference fasta. Has to be the same as the reference used to align samples being used for MEI detection, and must have a .fai index at the same location.

l	YES	STEP 1 Directory. This is the directory you provided with –w for **IndivAnalysis**.
n	YES	The gene annotation for the FASTA reference (-h). Must be sorted by coordinate. An annotation for Hg19 is included in MELTvX.X.tar (hg19.genes.bed).
t	YES	Transposon .zip file to perform discovery. See above section on Transposon Reference Files for more information.
w	YES	The working dir for this project.
a	NO	Boolean flag to tell MELT that alignments were performed with BWA-ALN rather than BWA-MEM [**false**]
q	NO	Alignment is pre Illumina 1.3 quality encoding [**false**]
r	NO	Read length of the supplied bam file. [**100**]
v	NO	Priors VCF for current MEI [**null**].
z	NO	Maximum reads to load into memory at any given time. Best to leave alone, but if looking at a .bam file larger than ~200Gb, set moderately higher. [**5000**]

5.4.4. Genotype

General Usage:

```
java –Xmx2G –jar MELT.jar Genotype <options>
```

Table 7: Genotype Options

Option	**Required?**	**Description**
h	YES	FULL PATH to the reference fasta. Has to be the same as the reference used to align samples being used for MEI detection, and must have a .fai index at the same location.
l	YES	.bam alignment to genotype. This is the .sorted.bam or .bam file, NOT the .disc, .disc.bai, or .fq file from MELT-Preprocess.jar. PreProcessing is assumed to have been run already.
p	YES	<MEI_EXAMINED>.pre_geno.tsv file created by MELT-GroupAnalysis.jar. Will be located in the top level of the directory provided by –w to MELT-GroupAnalysis.jar

t	YES	Transposon .zip file to perform discovery. See above section on Transposon Reference Files for more information.
w	YES	The working dir for this project.
e	NO	Fragment length of the samples. [**500**]
q	NO	Alignments are pre Illumina 1.3 quality encoding [**false**]
z	NO	Maximum reads to load into memory at any given time. Best to leave alone, but if looking at a .bam file larger than ~200Gb, set moderately higher. [**5000**]

5.4.5. MakeVCF

General Usage:

```
Java –Xmx2G –jar MELT.jar MakeVCF <options>
```

This step, beyond merging all MEI calls, also performs final filtering using three parameters:

1. (-j) No-call filter. If sites are no-call (./.) for greater than, default, 25% of individuals, then the call will be flagged with the r25 flag in the FILTER column.
2. (-s) 5' and 3' evidence filter. If sites have too much evidence on one side (only evidence on 3' or 5' of predicted insertion site), default 2.0 standard deviations outside of the mean for this discovery project, then the call will be flagged with the rSD flag in the FILTER column. Will not filter a site with fewer than 10 reads of combined evidence. For a more detailed explanation of this, please see read out paper (see Citing MELT below).

If the site passes both of these filters PASS will be listed in column 7. Please see MELT Output for a further explanation of the VCF output provided by MELT.

Table 8: MakeVCF Options

Option	**Required?**	**Description**
f	YES	List of files to merge. This is one file per line, of genotyped MEI calls. These files are located in the top level of the dir provided by –w to MELT-Genotype-jar, and are suffixed with <MEI_EXAMINED>.tsv.
h	YES	FULL PATH to the reference fasta. Has to be the same as the reference used to align samples being used for MEI detection, and must have a .fai index at the same location.

p	YES	<MEI_EXAMINED>.pre_geno.tsv file created by MELT-GroupAnalysis.jar. Will be located in the top level of the directory provided by –w to MELT-GroupAnalysis.jar
t	YES	Transposon .zip file to perform discovery. See above section on Transposon Reference Files for more information.
w	YES	The working dir for this project.
j	NO	No call filter cutoff, see above for more exact description. [**25**]
o	NO	Name of output VCF. [**./out.vcf**]
s	NO	STDEV cutoff [**2.0**]

5.4.6. Example MELT-SPLIT Workflow

As a simulated example to demonstrate MELT-SPLIT based MEI discovery, suppose we have sequenced four human individuals (Person1.sorted.bam, Person2.sorted.bam, Person3.sorted.bam, Person4.sorted.bam). We then used bwa-mem to align them to the Hg19 human reference sequence. We no want to know if they have any LINE1 non-reference Mobile Element Insertions. We first want to run Preprocess on all four samples to ensure MELT has the proper information to discover MEIs:

```
$ java –Xmx2G –jar MELT.jar Preprocess /path/to/Person1.sorted.bam
$ java –Xmx2G –jar MELT.jar Preprocess /path/to/Person2.sorted.bam
$ java –Xmx2G –jar MELT.jar Preprocess /path/to/Person3.sorted.bam
$ java –Xmx2G –jar MELT.jar Preprocess /path/to/Person4.sorted.bam
```

Next, we want to do initial MEI site discovery in all 4 genomes (only required options shown):

```
$ java –Xmx6G –jar MELT.jar IndivAnalysis \
–l /path/to/Person1.sorted.bam \
–w ./LINE1DISCOVERY/ \
-t /path/to/LINE_MELT.zip \
–h /path/to/human_reference.fa;

$ java –Xmx6G –jar MELT.jar IndivAnalysis \
–l /path/to/Person2.sorted.bam \
–w ./LINE1DISCOVERY/ \
-t /path/to/LINE_MELT.zip \
–h /path/to/human_reference.fa;

$ java –Xmx6G –jar MELT.jar IndivAnalysis \
–l /path/to/Person3.sorted.bam \
–w ./LINE1DISCOVERY/ \
-t /path/to/LINE_MELT.zip \
–h /path/to/human_reference.fa;

$ java –Xmx6G –jar MELT.jar IndivAnalysis \
–l /path/to/Person4.sorted.bam \
–w ./LINE1DISCOVERY/ \
-t /path/to/LINE_MELT.zip \
–h /path/to/human_reference.fa;
```

Next, we want to combine the initial discovery across these four genomes into a single piece of information to aid genotyping and filtering of false positive hits:

```
$ java –Xmx4G –jar MELT.jar GroupAnalysis \
-l ./LINE1DISCOVERY/ \
-w ./LINE1DISCOVERY/ \
-t /path/to/LINE_MELT.zip \
-h /path/to/human_reference.fa \
-n /path/to/human_annotation.bed;
```

Next, we will genotype each of the samples using the merged MEI information file:

```
$ java –Xmx2G –jar MELT.jar Genotype \
-l /path/to/Person1.sorted.bam \
-t /path/to/LINE_MELT.zip \
-h /path/to/human_reference.fa \
-w ./LINE1DISCOVERY/ \
-p ./LINE1DISCOVERY/LINE1.pre_geno.tsv;

$ java –Xmx2G –jar MELT.jar Genotype \
-l /path/to/Person2.sorted.bam \
-t /path/to/LINE_MELT.zip \
-h /path/to/human_reference.fa \
-w ./LINE1DISCOVERY/ \
-p ./LINE1DISCOVERY/LINE1.pre_geno.tsv;

$ java –Xmx2G –jar MELT.jar Genotype \
-l /path/to/Person3.sorted.bam \
-t /path/to/LINE_MELT.zip \
-h /path/to/human_reference.fa \
-w ./LINE1DISCOVERY/ \
-p ./LINE1DISCOVERY/LINE1.pre_geno.tsv;

$ java –Xmx2G –jar MELT.jar Genotype \
-l /path/to/Person4.sorted.bam \
-t /path/to/LINE_MELT.zip \
-h /path/to/human_reference.fa \
-w ./LINE1DISCOVERY/ \
-p ./LINE1DISCOVERY/LINE1.pre_geno.tsv;
```

Finally, we will create a list of genotyped MEIs for each individual, and provide this is input to MakeVCF to generate the final VCF file:

```
$ ls ./LINE1DISCOVERY/*.LINE1.tsv > ./list.txt;
$ java –Xmx2G –jar MELT.jar MakeVCF \
-f ./list.txt \
-t /path/to/LINE_MELT.zip \
-w ./LINE1DISCOVERY/ \
-p ./LINE1DISCOVERY/LINE1.pre_geno.tsv;
```

This will ultimately result in a final VCF file, which you can name with –o, or by default in your current directory: ./out.vcf.

5.5. Running MELT Using MELT-SINGLE

5.5.1. Description

MELT-SINGLE is a method to call MEIs on a single genome without having to manually perform each step individually using MELT-SPLIT. This method is advantageous for users who only have to analyze one high coverage genome for MEIs. It is **not recommended** to use MELT-SINGLE on low coverage genomes, as there could potentially be a higher number of false negative sites. Unlike MELT-SPLIT, however, MELT-SINGLE will look for multiple transposons at once in the provided genome.

General Usage:

```
$ java –Xmx6G –jar MELT.jar Single <options>
```

5.5.2. Options

Note: MELT-SINGLE makes use of the –k flag. If you have already run Preprocess on your .bam alignment, ensure you use –k or MELT-SINGLE will re-perform this step.

Note: While –b is not a required option, it is crucial to make sure the genome you are using does not have any 'decoy' or 'fake' chromosomes. An example is the hs37d5 'decoy' chromosome attached to the 1000 Genomes human genome fasta file. This 'chromosome' works to help BWA to align reads properly to the reference[12], and if included in MELT analysis will result in increased runtime and discovery of MEIs on this fake chromosome.

An example transposon_file_list.txt file for –t:

```
/full/path/to/LINE_MELT.zip
/full/path/to/ALU_MELT.zip
/full/path/to/SVA_MELT.zip
```

OR if utilizing priors with –t:

```
/full/path/to/LINE_MELT.zip<TAB>/full/path/to/LINE1_priors.VCF
/full/path/to/ALU_MELT.zip<TAB>/full/path/to/ALU_priors.VCF
/full/path/to/SVA_MELT.zip<TAB>/full/path/to/SVA_priors.VCF
```

Note: Priors do not have to be provided for each MEI class, but please ensure that the priors provided are only specific to the MEI type being discovered (i.e. only provide L1 priors on the LINE_MELT.zip line).

Table 9: Single Options

Option	Required?	Description
h	YES	FULL PATH to the reference fasta. Has to be the same as the reference used to align samples being used for MEI detection, and must have a .fai index at the same location.
l	YES	Single .bam alignment. This is the .sorted.bam or .bam file, NOT the .disc, .disc.bai, or .fq file from Preprocess, regardless of –k flag.
n	YES	The gene annotation for the FASTA reference (-h). Must be sorted by coordinate. An annotation for Hg19 is included in MELTvX.X.tar (hg19.genes.bed).
t	YES	List of MEI.zip files with one file per line. See above section on Transposon Reference Files for more information.
w	YES	The working dir for this project.
a	NO	Boolean flag to tell MELT that alignments were performed with BWA-ALN rather than BWA-MEM [**false**]
b	NO	Exclusion list for chromosomes. By default MELT performs MEI discovery on all chromosomes greater than 1Mb. To exclude chromosomes greater than 1Mb, simply list them, delimited by a forward slash. For example, to exclude chromosomes 1,2, and 4, use –b 1/2/4. [**null**]
c	NO	Coverage level of supplied bam file. [**30**]
d	NO	Minimum length of chromosome to analyze. [**1000000**]
e	NO	Fragment length of the samples. [**500**]
j	NO	No call filter cutoff, see above for more exact description. [**25]**
k	NO	.bam files have already ben preprocessed. See above for description of what this means. [**false**]
s	NO	STDEV cutoff [**2.0**]
q	NO	Alignments are pre Illumina 1.3 quality encoding [**false**]

r	NO	Read length of the supplied bam file. [**100**]
z	NO	Maximum reads to load into memory at any given time. Best to leave alone, but if looking at a .bam file larger than ~200Gb, set moderately higher. [**5000**]

6. MELT Output

MELT provides one output VCF file for each MEI analyzed, and when run with MELT-SGE, a BED file for each MEI for each sample. See VCF format specifications for further details on default columns (columns 1-6, 9+). MELT, by default, does not provide any information for the NAME (Column 3) column. However, if a prior file is provided, the NAME column will be the prior name.

In column 7, MELT provides one of three FILTERS – PASS, rSD, or s25. Please see section MELT-MakeVCF for more information on these filters.

MELT also provides several custom fields in the VCF INFO column (column 8) to provide additional information to the user.

Table 10: MELT Info Columns

INFO	**Description**
TSD	Provides the sequence of the Target Site Duplication (TSD) for each insertion. Will be 'null' if was not able to be determined.
ASSESS	Provides a accuracy assessment, and the information used, to determine the breakpoint of each insertion. - 0 = No overlapping reads at site - 1 = Imprecise breakpoint due to greater than expected distance between evidence - 2 = discordant pair evidence only – No split read information - 3 = left side TSD evidence only - 4 = right side TSD evidence only - 5 = TSD decided with split reads, highest possible quality. Please read our paper to learn more above individual scores (see citing MELT).
INTERNAL	Provides information on if the insertion is internal to a gene in the provided reference annotation file. The first piece of information is the gene name; the second is the location within that gene (either exon, intron, 5_UTR, or 3_UTR).
SVTYPE	Gives basic information about what type of MEI is represented in the record.
SVLEN	Length of this MEI.

MEINFO	Provides estimate information on the start, stop, and length of the MEI against its own reference. If polarity is unknown, will be 'null'. If either start or stop is not know, will be -1.
DIFF	Provides differences for this element compared to the MEI reference provided. Please see MEI Species Classification below for more information.
LP	Number of read pairs supporting the left, 5', side of this insertion.
RP	Number of read pairs supporting the right, 3', side of this insertion.
RA	Ratio of left and right read pairs.
ISTP	If the SVTYPE info field is L1 or LINE1, the site of twin priming in BP. If insertion is not twin-primed, will be 0. If no measurement was made, will be -1.
PRIOR	If a site has was found in priors file, will be true. If site was not found in priors file, or no priors file provided, will be false.

7. Other MELT Tools

7.1. MEI Species Classification

Built into MELT is a sequence assembly algorithm that rebuilds the sequence of the MEI from discordant read pairs discovered by MELT, and then classifies the insertion according to known family relationships. This algorithm currently contains three parts:

1. Sequence assembler – Assembles discordant pairs from WGS alignments into a complete sequence for an MEI.
2. Difference Discovery – Looks for differences in assembled sequence according to the provided MEI reference.
3. Classifier – Determines family and sub-family information about specific transposon types.

Part one only runs during MELT analysis, and currently is not stand-alone.

For Parts two and three, analysis depends on the transposon type. MELT can currently only classify Alu and LINE-1 insertions into family and subfamily. If, during a MELT Run, the name provided in the .info file matches either 'Alu', 'LINE1', or 'L1' (with any combination of lower and uppercase) family classification will be performed. If not, MELT will stop at part two as outline above and provide a list of differences from the MEI .fa file.

Part two and three have a stand-alone .jar for both *Alu* and LINE-1 difference discovery and classification, MELT-CALLU.jar and MELT-LINEU.jar, respectively. These two programs are discussed at length below.

7.1.1 CAlu

CAlu is a standalone tool that, given an unknown Alu sequence, will classify it according to family and mutations. CALU.jar is provided in the MELTvX.X.tar repository, and can be run standalone. It will be located with the other .jar files in the ./MELTvX.X/execs/ folder. Please see melt.igs.umaryland.edu/CALU/ for more information on classification.

General Usage (When using multi-fasta):

```
java –Xmx2g –jar MELT.jar CALU –f /path/to/sequences.fasta
```

General Usage (When providing sequence from the command line):

```
java –Xmx2g –jar MELT.jar CALU –s <Input sequence [ATCGN]>
```

-o can additionally be specified to direct output to a specific file. Otherwise, will print to standard out.

CalU output is a tab-delimited and contains the following information:

Table 11: Calu/LinU Output

Column	**Description**
1	Name of input sequence
2	Family
3	Subfamily
4	Start
5	Stop
6	Number of diagnostic changes (used to determine classification in columns 2 & 3)
7	Number of overall changes
8	Changes, delimited by '|'
9	Original input sequence

7.1.2 LineU

LineU is a standalone tool for LINE-1 subfamily analysis that will perform a similar analysis to CalU. I/O is identical for both.

7.2. MELT-Deletion

MELT-Deletion is a tool to determine presence/absence of reference mobile elements (those included in the reference release for a given species). It is included in the ./MELTvX.X/execs/ folder.

MELT-Deletion uses the repeat masker track (the same file used as a mask for MELT MEI discovery analysis) to determine if a reference MEI is present or absent. Please refer to the Detailed Description of .zip Creation for specifics on how to generate this file. MELT-Deletion uses two methods depending on the length of the reference

MEI. If the insertion is longer than the fragment read length, MELT-Deletion uses a joint discordant and split read pair analysis to estimate the genotype. If the insertion is shorter than the fragment read length, MELT-Deletion only uses a split read pair genotyping approach.

MELT-Deletion is run in two steps:

1. **Genotype** – Analyze an individual .bam file for reference MEI deletions.
2. **Merge** – Merge and filter individual deletion calls from multiple .bam files.

Each of these two processes is included in the MELT.jar file, and is specified with either Deletion-Genotype or Deletion-Merge. MELT-Deletion is first run on multiple .bam files, and then the resulting output files are merged. For example:

```
$ java –Xmx2g –jar MELT.jar Deletion-Genotype –l file1.sorted.bam –w
/path/to/working/dir/ <other options>
$ java –Xmx2g –jar MELT.jar Deletion-Genotype –l file2.sorted.bam –w
/path/to/working/dir/ <other options>
$ java –Xmx2g –jar MELT.jar Deletion-Genotype –l file3.sorted.bam –w
/path/to/working/dir/ <other options>

$ ls *.del.tsv > list.of.outputs.txt

$ MELT-Deletion –v Merge –l list.of.outputs.txt <other options>
```

Deletion-Genotype will create a single output tsv file, using the name of the bam file. The input of file1.sorted.bam would result in the output file1.del.tsv being placed in the working directory specified by -w.

Note: Do not change the suffix (.del.tsv), as this is required by Deletion-Merge.

Note: The default minimum chromosome size to analyze is 1Mb (1000000) but can be set lower if necessary. However, it is **not** recommended to set it below about 100Kb (100000), as this may cause errors during merging.

7.2.1. MELT-Deletion Genotype

General Usage:

```
java –Xmx2G –jar MELT.jar Deletion-Genotype <options>
```

Table 12: MELT-Deletion Genotype Options

Option	Required?	Description
b	YES	.bed coordinates to examine for reference MEI deletions
h	YES	FULL PATH to the reference fasta. Has to be the same as the reference used to align samples being used for MEI detection, and must have a

		.fai index at the same location.
l	YES	.bam alignment to genotype.
w	YES	The working dir for this project.
e	NO	Fragment length of the samples. [**500**]
z	NO	Maximum reads to load into memory at any given time. Best to leave alone, but if looking at a .bam file larger than ~200Gb, set moderately higher. [**5000**]

7.2.2. MELT-Deletion Merge

General Usage:

```
java –Xmx1G –jar MELT.jar Deletion-Merge <options>
```

Table 13: MELT-Deletion Merge Options

Option	Required?	Description
b	YES	.bed coordinates to examine for reference MEI deletions
h	YES	FULL PATH to the reference fasta. Has to be the same as the reference used to align samples being used for MEI detection, and must have a .fai index at the same location.
l	YES	List of genotype files to merge, one per line.
o	YES	output VCF file.
d	NO	Minimum length of chromosome to merge. [**1000000**]

7.2.3. MELT-Deletion Output

MELT-Deletion outputs in VCF 4.1 format using similar conventions to regular MELT MEI discovery analysis. The primary difference lies in the INFO column.

Table 14: MELT-Deletion Info Columns

INFO	**Description**
SVTYPE	Transposon subfamily according to Repeat Masker.
SVLEN	Length of this MEI.
ADJLEFT	If breakpoint determined by MELT-Deletion is different from Repeat Masker on the 5’ end, list adjusted breakpoint here.
ADJRIGHT	If breakpoint determined by MELT-Deletion is different from Repeat Masker on the 3’ end, list adjusted breakpoint here.

8. What MELT is for

MELT is a highly accurate caller with FDR established by a blind study performed by a very subjective group separate from our own. FDR estimates for the MEI types already discovered and studied by the 1000 Genomes project are <5%, with MELT designed to be just as sensitive at discovering low allele frequency insertions, as high allele frequency insertions. The sequencing depth of the samples that are provided are the only limits to MELT. MELT is also highly optimized for ease of use, and due to the portability of Java, users should not experience issues when moving to other platforms.

9. What MELT is not for

1.) MELT is NOT repeat masker! It will not discover MEIs in the reference sequence from your recently sequenced genome. It either discovers polymorphic MEIs in large populations or genotypes reference MEIs as compared to the reference.
2.) MELT does very well on most transposons. However, MELT has only been tested for non-LTR transposons (*Alu*, L1, and SVA in humans) and not LTR transposons such as ERVs or DNA transposons such as Mariner. Due to the nature of HERV-K's LTRs having high sequence homology with SVA, MELT can sometimes miscall an SVA insertion as a HERV-K insertion. Software development is underway to ensure that these transposons reach the high standard that we have already set for non-LTR transposons. Feel free to try these types of MEIs, but results may vary. A HERV-K.zip file has been included in the MELT release, but a FDR for discovery has not been established.
3.) MELT has currently been tested only in human (hg19), chimpanzee (panTro4), and dog (canFam3.1). MELT should do an excellent job at discovering transposons in ANY sequenced organism, but tests to determine this have yet to be performed.
4.) MELT has only been tested on BWA-MEM and BWA-ALN alignments. MELT does not discriminate between these types of alignments, though performing BWA-MEM alignments is typically easier and less time-intensive. It is recommended that you use these types of alignments, as MELT has not been tested on other types of alignments. More information on performing BWA alignments is provided at the BWA website (http://bio-bwa.sourceforge.net/), and in the BWA publication.

10. How to Cite MELT

If you use MELT to discovery MEIs please cite:

To be determined Gardner et al. Paper

If you use the included MEI priors' list, please additionally cite:

To be determined SV AND 1000Gs paper

11. References

1 Li, H. & Durbin, R. Fast and accurate long-read alignment with Burrows-Wheeler transform. *Bioinformatics (Oxford, England)* **26**, 589-595, doi:10.1093/bioinformatics/btp698 (2010).

2 Danecek, P. *et al.* The variant call format and VCFtools. *Bioinformatics (Oxford, England)* **27**, 2156-2158, doi:10.1093/bioinformatics/btr330 (2011).

3 Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with Bowtie 2. *Nature methods* **9**, 357-359, doi:10.1038/nmeth.1923 (2012).

4 Li, H. A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics (Oxford, England)* **27**, 2987-2993, doi:10.1093/bioinformatics/btr509 (2011).

5 Kent, W. J. *et al.* The human genome browser at UCSC. *Genome research* **12**, 996-1006, doi:10.1101/gr.229102. Article published online before print in May 2002 (2002).

6 Karolchik, D. *et al.* The UCSC Table Browser data retrieval tool. *Nucleic acids research* **32**, D493-496, doi:10.1093/nar/gkh103 (2004).

7 Jurka, J. *et al.* Repbase Update, a database of eukaryotic repetitive elements. *Cytogenetic and genome research* **110**, 462-467, doi:10.1159/000084979 (2005).

8 Larkin, M. A. *et al.* Clustal W and Clustal X version 2.0. *Bioinformatics (Oxford, England)* **23**, 2947-2948, doi:10.1093/bioinformatics/btm404 (2007).

9 Katoh, K., Kuma, K., Toh, H. & Miyata, T. MAFFT version 5: improvement in accuracy of multiple sequence alignment. *Nucleic acids research* **33**, 511-518, doi:10.1093/nar/gki198 (2005).

10 Rice, P., Longden, I. & Bleasby, A. EMBOSS: the European Molecular Biology Open Software Suite. *Trends in genetics : TIG* **16**, 276-277 (2000).

11 Smit AFA, H. R., Green P. RepeatMasker Open-3.0. (1996-2010).

12 Genovese, G., Handsaker, R. E., Li, H., Kenny, E. E. & McCarroll, S. A. Mapping the human reference genome's missing sequence by three-way admixture in Latino genomes. *American journal of human genetics* **93**, 411-421, doi:10.1016/j.ajhg.2013.07.002 (2013).