MARGE - Mutation Analysis for Regulatory Genomic Elements

Verena M. Link

March 19, 2018

Contents

1 Introduction

MARGE (ADD REFERENCE TO PAPER) is a publicly available, open-source suite of software tools that integrates genome-wide genetic variation with epigenetic data to identify collaborative transcription factor pairs. MARGE is optimized to work with chromatin accessibility assays (such as ATAC-seq or DNase hypersensitivity), as well as transcription factor binding data collected by ChIP-seq. MARGE is based on the software tool HOMER [19] and the original idea of MARGE was developed in Gosselin et al. [9]. Parts of MARGE were also used in Heinz et al. [7].
This documentation shows how to install MARGE and all required additional software. We show some basic pipeline to simplify getting used to the software. Furthermore, we explain the syntax and all parts of MARGE.

Please send any bug reports to verena.m.link@gmail.com

2 Prerequisites

MARGE is implemented in Perl and R [18]. It has been tested on several UNIX systems, including CentOS, RedHat, and Debian with Perl version 5.20 and higher and R version 3.3 and higher. We provide a script that installs MARGE and allows download of pre-processed mutation data from the mouse genome project [11] and the genomes from the 1000 Genome Project [6] used in the MARGE manuscript (ADD REFERENCE). MARGE requires the Perl core modules POSIX, Getopt::Long, Storable and threads, as well as the modules Set::IntervalTree [5], Sys::CPU [15] and Statistics::Basic [16]. It further requires the R packages SeqLogo [4], gridBase [17], lme4 [3], and gplots [20]. It also requires an installed version of gzip. For the motif mutation analysis MARGE requires HOMER [19] (http://homer.ucsd.edu/homer/) to be installed and executable. Without a working installation of HOMER, MARGE's functionality is limited to only visualization and annotation of the data.

3 Installation

3.1 Installation of Prerequisites

To install MARGE, first all prerequisites should be installed. This guide is written for a completely new system (as an example RedHat is chosen). It therefore is possible that your system already has many of the tools installed.
To install these programs globally and also to be able to just copy all commands, the user should the be super user of the system.

3.1.1 Installation of system tools

Assuming you are working on a complete new system, some system tools need to be installed.

- g++ (often called gcc-c++)
- gcc
- make
- zip
- unzip
- wget
- bzip2
- curl
- grep
- sed
- perl
- cpan

For a RedHat system you can install all of the programs with this command:

```
yum install gcc-c++ gcc make zip unzip wget bzip2 curl grep sed perl cpan
```

3.1.2 Installation of Anaconda

There are many different ways to install software, but one of the easiest ways to get bioinformatics software installed on your system without too much hassle, is to use Anaconda [1]. We recommend to install all of the additional software for MARGE, as well as all the R-packages via Anaconda.

```
echo 'export PATH=$PATH:/opt/conda/bin' > /etc/profile.d/conda.sh

wget --quiet https://repo.continuum.io/miniconda/Miniconda3-4.3.14-Linux-x86_64.sh

/bin/bash Miniconda3-4.3.14-Linux-x86_64.sh -b -p /opt/conda
```

```
rm Miniconda3-4.3.14-Linux-x86_64.sh
```

To access conda, export the path:

```
export PATH=$PATH:/opt/conda/bin
```

and also add it to your .bash_profile.
Now the conda channels are added:

```
conda config --add channels conda-forge

conda config --add channels r

conda config --add channels defaults

conda config --add channels bioconda
```

3.1.3 Installation of R-packages

The different R-packages can be installed manually, but we recommend to use Anaconda [1] which optimizes the installation of R-packages. For details on how to install Anaconda, see section 3.1.2.
With this command all necessary R-packages are installed:

```
conda install r-essentials bioconductor-deseq2

conda install bioconductor-edger r-seqinr rstudio
```

3.1.4 Installation of PERL modules

MARGE also needs some Perl modules that are non standard. There are different ways to install Perl modules. We recommend to use cpan as super user and install one module from scratch, but we also provide some instructions on how to use cpanm.

Installation with cpan If you decide to use cpan, only Statistics::Basic [16] and Sys::CPU [15] can be installed with it. In theory you should be able to install Set::IntervalTree [5] with cpan, but it always failed when we tried to do it. Please run these commands as super user

```
cpan Statistics::Basic

cpan Sys::CPU
```

Installation from scratch To install Set::IntervalTree [5] run:

```
wget http://search.cpan.org/CPAN/authors/id/B/BE/BENBOOTH/Set-IntervalTree-0.01.tar.gz
```

```
tar xfvz Set-IntervalTree-0.01.tar.gz

cd Set-IntervalTree-0.01

perl Makefile.PL

make

make install
```

Installation with cpanm You can install cpanm as root, which will install it to /usr/local/bin and if your Perl path is set correctly will make the Perl modules available for all users on your system. This requires you to have sudo access.

```
curl -L https://cpanmin.us | perl - -sudo App::cpanminus
```

You can also install cpanm as a local user.

```
curl -L https://cpanmin.us | perl - App::cpanminus
```

Now you can install three Perl modules (Set::IntervalTree [5], Sys::CPU [15] and Statistics::Basic [16]):

```
cpanm Set::IntervalTree

cpanm Sys::CPU

cpanm Statistics::Basic
```

3.1.5 Installation of bowtie2

For more information about how to install and use bowtie2, please see [13].
The easiest way to install bowtie2 is with Anaconda (for details how to install Anaconda, please see section 3.1.2).

```
conda install -c bioconda bowtie2
```

3.1.6 Installation of STAR

For more information about how to install and use STAR, please see [2].
The easiest way to install STAR is with Anaconda (for details how to install Anaconda, please see section 3.1.2).

```
conda install -c bioconda star
```

3.2 Installation of HOMER

To take advantage of the full potential of MARGE, HOMER [19] (http://homer.ucsd.edu/homer/) needs to be installed. All prerequisites for HOMER have been installed in the previous section 3.1. HOMER can be installed via its own installation script.

```
wget http://homer.ucsd.edu/homer/configureHomer.pl
```

With the next command HOMER will be installed in your current working directory. If you want to install HOMER in another location just substitute homer in the next command with the location you want HOMER to be installed:

```
mkdir homer && mv configureHomer.pl homer
```

```
perl homer/configureHomer.pl -install
```

Follow the instructions on how to add the PATH to your .bash_profile.
To download and install genomes for HOMER to use, run:

```
perl homer/configureHomer.pl -help
```

HOMER has a great documentation (http://homer.ucsd.edu/homer/), that explains all analysis in detail and gives a lot of very helpful tips. In case you are not very familiar with analysis of next-generation sequencing (NGS) data, we highly recommend to check it out!
To visualize data with in the UCSC genome browser [12] using multiWig hubs, a webserver is required. For more details, please see the HOMER documentation (http://homer.ucsd.edu/homer/).
To create bedGraphs (used for visualization) HOMER requires the bedGraphToBigWig tool [10], which can be installed via Anaconda (for details how to install Anaconda, please see section 3.1.2).

```
conda install -c daler bedgraphtobigwig
```

3.3 Installation of MARGE

This section explains how to install MARGE. All prerequisites should have been installed. If MARGE can not find all tools, it will try to install it, but we highly recommend to install everything upfront, to avoid any complications. If your local user has read/write and execute permission on the MARGE installation folder, there is no need to run this as super user.

```
wget http://homer.ucsd.edu/MARGE/MARGE_v1.0.tar.gz
```

```
tar fxvz MARGE_v1.0.tar.gz
```

Move MARGE_v1.0 to your final installation destination before proceeding.

```
cd MARGE_v1.0
```

MARGE will use your current working directory as the directory to save all pro-processed data. If you want to use another path in your configuration, please run configure.pl

with -h or –help.
If configure.pl is only called without any options, MARGE is configured to save all data in the MARGE_v1.0/data folder. In case you would like to have a separate directory for your data, run:

```
perl configure.pl -h
```

or

```
perl configure.pl --help
```

```
Script to configure MARGE installation
Options:
    -install_dir <path to installation directory>
    -homer_path <path to HOMER (only necessary when HOMER not part of $PATH
    -data_folder <path to folder used for storing data> (default in
        install_dir)

    -h|--help: print this help
```

The directory in which MARGE is installed can be changed with -install_dir, and the folder in which all the MARGE data can be found is specified with -data_folder. Furthermore, in case HOMER is installed, but not in the PATH variable, the path can be specified with -homer_path.

To configure and install MARGE in the current working directory, run:

```
perl configure.pl
```

```
perl install.pl
```

Please follow the instruction on how to add MARGE to your PATH variable at the end of the installation script.

4 Dockerfile

We provide a Dockerfile that will install HOMER and MARGE into an Ubuntu environment, containing all necessary environment variables.
We would like to note that it would be better to save all processed data in a Docker Volume. Therefore we recommend to either build the Dockerfile with a Volume specified, so MARGE is configured correctly during the installation, or to change the MARGE config manually every time the build is run.
It is highly recommended to use the first solution and just run build with a variable for a external folder.
Details no all of that: TBA

5 MARGE

All individual scripts can be accessed via the head script MARGE.pl. This documentation will provide a step by step introduction into how to use MARGE, as well as a detailed description of every possible analysis that can be ran.
In order to see an overview of all MARGE scripts, just call the script without any parameters.

5.1 Usage for MARGE

To see usage

```
MARGE.pl
```

```
MARGE USAGE
Download pre-processed data
    MARGE.pl download

Getting started with data processing
    MARGE.pl prepare_files
        This script takes VCF input files and generates MARGE mutation
            files, shifting files and the genomes for the individuals from
            the VCF file.

    MARGE.pl create_genomes
        This script generates the individual genomes for all individuals
            specified when called. It only works when the mutation and
            shifting files were already prepared using MARGE.pl
            prepare_files.

    MARGE.pl update_files
        For some projects (e.g. the 1000 genome project) mutations are
            annotated per chromosomes. Therefore, MARGE.pl prepare_files was
             run per chromosome. In this case some of the MARGE files were
            overwritten and need to be adjusted after finishing MARGE.pl
            prepare_files for all chromosomes.



Access to mutation data
    MARGE.pl genome_interactions
        This script allows to extract sequences for different individuals.
            It also provides protein sequences and alignments. The
            alignments are based on the VCF file and are not generated using
             any alignment algorithms.

    MARGE.pl mutation_bedfiles
        This script outputs a bed file per individual per allele that can
            be uploaded to the UCSC genome browser.

    MARGE.pl mutation_info
        This script counts mutations between different individuals and
            outputs them in a table format. It can either count all
            mutations in comparison to the reference or all private
            mutations

    MARGE.pl extract_sequences
        This script extracts the individual-specific sequences in fasta
            format.

    MARGE.pl annotate_mutations
        This script adds all mutations within specified genomic loci for
            each individual at the end of the input file.
```

```
Data handling
    MARGE.pl shift
        This script shifts the data mapped to the individual genome to the
            reference coordinates. This step is necessary to compare genomic
             loci between different individuals, as well as for
            visualization of data in the different genome browsers.

    MARGE.pl shift_to_strain
        This script shifts the data mapped to the reference genome to the
            coordinates of the individual. Please note that this script will
             be used very rarely. To analyze the data it is better to shift
            all strains data to the reference coordinates, not vice versa!

    MARGE.pl allele_specific_reads
        For heterozygous data, this script only keeps reads that were
            perfectly mapped (to avoid bias introduced by SNPs) and also
            outputs a file with reads that overlap mutations (to call allele
            -specific binding)

    MARGE.pl annotate_allele_specific
        For heterozygous data, this script annotates all genomic loci
            between different individuals. It takes reads that overlap
            mutations for loci with mutations in this individual. In case
            there is no mutation, it takes the perfectly mapped reads for
            this locus and divides it by 2 (because of two alleles)



Mutation analysis
    MARGE.pl denovo_motifs
        HOMER denovo motif analysis extended to integrate the usage of
            different genomes.

    MARGE.pl mutation_analysis
        Pairwise or all versus all analysis of the impact of mutations on
            open chromatin (ATAC-Seq, DNAse Hypersensitivity assays etc.) or
             transcription factor binding.

    MARGE.pl summary_heatmap
        Script to summarize the results from several runs of pairwise
            mutation analyses or all versus all runs (needs at least two
            files as input).

For more information call the different scripts either without any
   parameters or the parameter --help or -h
```

6 Most common pipeline examples

In this section, we will discuss some basic pipelines that can be used in order to analyze data from mouse strains or human individuals. Once the MARGE paper is published, we will also provide a pdf document with all commands that were used in order to analyze the data from this paper.
We will give detailed instructions on how to analyze ChIP-seq and RNA-seq data from different mouse strains, as well as ChIP-seq data from human individuals and data from F1 mouse strains.

6.1 Mouse strain ChIP-seq data

Here we show in detail how to run MARGE on ChIP-seq data from three different inbred strains of mice. We will download some data from GEO to use as a test case (see section 6.1.1). Then we will demonstrate how to generate MARGE indices (see section 6.1.2 - section 6.1.4, and section 8.1) including individualized genomes (see section 8.2). After mapping (see section 6.1.5), we show how to shift the data (see section 6.1.6 and section 10.1), use HOMER on this data set (see section 6.1.7), visualize the data (see section 6.1.8), and then run de novo motif analysis (see section 6.1.9 and section 11.1) and a motif mutation analysis (see section 6.1.10 and section 11.2).

6.1.1 Download data from GEO

For the mouse ChIP-seq data, we download data from Gosselin et al. [7]. We get three PU.1 ChIP-seq datasets in large peritoneal macrophages (LPMs) for C57BL/6J (C57), NOD/ShiLtJ (NOD), and SPRET/EiJ (SPRET) and also the corresponding inputs.
The data can be downloaded via wget <link>:
C57 LPM PU.1 ChIP:

ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByExp/sra/SRX/SRX746/SRX746980/SRR1634688/SRR1634688.sra

C57 LPM Input:

ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByExp/sra/SRX/SRX746/SRX746981/SRR1634689/SRR1634689.sra

NOD LPM PU.1 ChIP:

ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByExp/sra/SRX/SRX747/SRX747019/SRR1634727/SRR1634727.sra

NOD LPM Input:

ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByExp/sra/SRX/SRX747/SRX747020/SRR1634728/SRR1634728.sra

SPRET LPM PU.1 ChIP:

ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByExp/sra/SRX/SRX747/SRX747023/SRR1634731/SRR1634731.sra

SPRET LPM Input:

ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByExp/sra/SRX/SRX747/SRX747024/SRR1634732/SRR1634732.sra

Next, the files are renamed:

```
mv SRR1634688.sra C57_LPM_PU1.sra

mv SRR1634689.sra C57_LPM_Input.sra

mv SRR1634727.sra NOD_LPM_PU1.sra

mv SRR1634728.sra NOD_LPM_Input.sra

mv SRR1634731.sra SPRET_LPM_PU1.sra

mv SRR1634732.sra SPRET_LPM_Input.sra
```

6.1.2 Download genetic variation data

The SNP and InDel files from the mouse genome project [11] can be downloaded with wget <link>:

SNP file:

ftp://ftp-mouse.sanger.ac.uk/REL-1303-SNPs_Indels-GRCm38/mgp.v3.snps.rsIDdbSNPv137.vcf.gz

InDel file:

ftp://ftp-mouse.sanger.ac.uk/REL-1303-SNPs_Indels-GRCm38/mgp.v3.indels.rsIDdbSNPv137.vcf.gz

These two files need to be unzipped:

```
gunzip mgp.v3.snps.rsIDdbSNPv137.vcf.gz

gunzip mgp.v3.indels.rsIDdbSNPv137.vcf.gz
```

6.1.3 Download reference genome

Additionally, the mm10 genome fasta files need to be downloaded:

```
mkdir genome_mm10_clean && cd genome_mm10_clean
```

You can download each chromosome from the UCSC genome browser via rsync:

```
rsync -avzP rsync://hgdownload.cse.ucsc.edu/goldenPath/mm10/chromosomes/chr1.fa.gz .

rsync -avzP rsync://hgdownload.cse.ucsc.edu/goldenPath/mm10/chromosomes/chr2.fa.gz .

⋮

rsync -avzP rsync://hgdownload.cse.ucsc.edu/goldenPath/mm10/chromosomes/chrX.fa.gz .

rsync -avzP rsync://hgdownload.cse.ucsc.edu/goldenPath/mm10/chromosomes/chrY.fa.gz .
```

Now the chromosome files need to be unzipped:

```
gunzip *gz

cd ..
```

6.1.4 Prepare MARGE data with MARGE.pl prepare_files

To prepare the indices for NOD/ShiLtJ and SPRET/EiJ using the SNP and InDel files downloaded from the mouse genome project [11] (see section 6.1.2) run:

```
MARGE.pl prepare_files -ind nodshiltj, spreteij -files mgp.v3.snps.rsIDdbSNPv137.vcf,
mgp.v3.indels.rsIDdbSNPv137.vcf -core 12 -genome genome_mm10_clean
```

MARGE will generate a folder called REFERENCE. This is the genome of C57BL/6J. For further downstream analysis, we rename this folder

```
mv <path to MARGE data folders>/REFERENCE <path to MARGE data folders>/C57BL6J
```

To create the MARGE folders will take some time. You can also download the already pre-processed MARGE folders including the genomes with MARGE.pl download (see section 7.1).

```
MARGE.pl download -MARGE_folders c57bl6j, nodshiltj, spreteij -genome mm10
```

6.1.5 Generate bowtie2 index

In this tutorial the ChIP-seq data is mapped with bowtie2. Please refer to the bowtie 2 manual [13] for details on how to install, map and so on. To simplify the generation of a bowtie2 index, we recommend to concatenate all fasta files into one large file.

```
cat <path to MARGE data folder>/C57BL6J/.*fa >> genome-c57.fa

cat <path to MARGE data folder>/NODSHILTJ/.*fa >> genome-nod.fa

cat <path to MARGE data folder>/SPRETEIJ/.*fa >> genome-spret.fa
```

To generate a bowtie2 index, run:

```
bowtie2-build genome-c57.fa c57bl6j

bowtie2-build genome-nod.fa nodshiltj

bowtie2-build genome-spret.fa spreteij
```

These commands will take some time. MARGE also allows to download already generated bowtie2 indices (see section 7.1).

```
MARGE.pl download -bowtie_index c57bl6j, nodshiltj, spreteij -genome mm10
```

For each command, 6 files are generated (for example for c57bl6j: c57bl6j.1.bt2, c57bl6j.2.bt2, c57bl6j.3.bt2, c57bl6j.4.bt2, c57bl6j.rev.1.bt2, c57bl6j.rev.2.bt2)
We recommend to move these files into a general bowtie2 index folder.

```
mkdir -p bowtie2/index && mv *bt2 bowtie2/index
```

SRA files need to be converted to fastq files with fastq-dump. In case fastq-dump [14] is not installed on your system, it is part of the sra-tools and can be installed via Anaconda [1] (for details on how to install Anaconda, see section 3.1.2):

```
conda install -c bioconda sra-tools
```

Now fastq-dump can be run:

```
fastq-dump C57_LPM_PU1.sra

fastq-dump C57_LPM_Input.sra

fastq-dump NOD_LPM_PU1.sra

fastq-dump NOD_LPM_Input.sra

fastq-dump SPRET_LPM_PU1.sra

fastq-dump SPRET_LPM_Input.sra
```

After generating the indices, the ChIP-seq data can now be mapped:

```
bowtie2 -p 8 -x bowtie2/index/c57bl6j C57_LPM_PU1.fastq > C57_LPM_PU1.sam 2>
C57_LPM_PU1.log

bowtie2 -p 8 -x bowtie2/index/c57bl6j C57_LPM_Input.fastq > C57_LPM_Input.sam 2>
C57_LPM_Input.log

bowtie2 -p 8 -x bowtie2/index/nodshiltj NOD_LPM_PU1.fastq > NOD_LPM_PU1.sam 2>
NOD_LPM_PU1.log

bowtie2 -p 8 -x bowtie2/index/nodshiltj NOD_LPM_Input.fastq > NOD_LPM_Input.sam 2>
NOD_LPM_Input.log

bowtie2 -p 8 -x bowtie2/index/spreteij SPRET_LPM_PU1.fastq > SPRET_LPM_PU1.sam 2>
SPRET_LPM_PU1.log

bowtie2 -p 8 -x bowtie2/index/spreteij SPRET_LPM_Input.fastq > SPRET_LPM_Input.sam
2> SPRET_LPM_Input.log
```

6.1.6 Shift data to reference coordinates

After mapping the data to its individual genome, the data needs to be shifted back to the reference coordinates with MARGE.pl shift (see section 10.1):

```
MARGE.pl shift -files NOD_LPM_PU1.sam, NOD_LPM_Input.sam -ind nodshiltj

MARGE.pl shift -files SPRET_LPM_PU1.sam, SPRET_LPM_Input.sam -ind spreteij
```

6.1.7 HOMER analysis of ChIP data

We will use HOMER for all ChIP-seq analysis, but every peak caller can be used, as long as the output format has the same format HOMER's peak files are in (ID<tab>chr<tab>start<tab>end<tab>strand). For the HOMER analysis, we first

need to create tag directories:

```
makeTagDirectory tag_C57_LPM_PU1 C57_LPM_PU1.sam

makeTagDirectory tag_C57_LPM_Input C57_LPM_Input.sam

makeTagDirectory tag_NOD_LPM_PU1 NOD_LPM_PU1_shifted_from_NODSHILTJ.sam

makeTagDirectory tag_NOD_LPM_Input NOD_LPM_Input_shifted_from_NODSHILTJ.sam

makeTagDirectory tag_SPRET_LPM_PU1 SPRET_LPM_PU1_shifted_from_SPRETEIJ.sam

makeTagDirectory tag_SPRET_LPM_Input SPRET_LPM_Input_shifted_from_SPRETEIJ.sam
```

As next step the peaks are called for each PU.1 ChIP-seq data set:

```
findPeaks tag_C57_LPM_PU1 -style factor -i tag_C57_LPM_Input > peaks_C57_LPM_PU1.txt

findPeaks tag_NOD_LPM_PU1 -style factor -i tag_NOD_LPM_Input > peaks_NOD_LPM_PU1.txt

findPeaks tag_SPRET_LPM_PU1 -style factor -i tag_SPRET_LPM_Input > peaks_SPRET_LPM_PU1.txt
```

6.1.8 Visualizing ChIP-seq data

To have a look at the data, HOMER offers the possibility to create UCSC genome [12] browser hubs. This requires the mm10 genome for HOMER to be available:

```
perl homer/configureHomer.pl -install mm10
```

It furthermore requires a web server, as well as the ENCODE bedtools [10]. In case all of this installed, you can visualize the data:

```
makeMultiWigHub.pl LPM-PU1-strains mm10 -d tag_C57_LPM_PU1 tag_NOD_LPM_PU1 tag_SPRET_LPM_PU1
```

Alternatively, HOMER can also visualize each tag directory as a separate bedGraph:

```
makeUCSCfile tag_C57_LPM_PU1 -o tag_C57_LPM_PU1.ucsc.bedGraph.gz

makeUCSCfile tag_NOD_LPM_PU1 -o tag_NOD_LPM_PU1.ucsc.bedGraph.gz

makeUCSCfile tag_SPRET_LPM_PU1 -o tag_SPRET_LPM_PU1.ucsc.bedGraph.gz
```

To additionally be able to see if there are mutations within the strains, MARGE allows the generation of bed files containing the mutations per strain:

```
MARGE.pl mutation_bedfiles -ind nodshiltj, spreteij
```

This command creates two gzipped bed files (output_mut_nodshiltj_1.bed.gz and output_mut_spreteij_1.bed.gz) which can be uploaded to the UCSC genome browser.

6.1.9 MARGE de-novo motif analysis

One of the most common analysis for ChIP-seq data is de-novo motif analysis. It also provides a great opportunity for further quality control.
In case there is no mm10 genome for HOMER so far, run:

```
perl homer/configureHomer.pl -install mm10
```

To be able to calculate statistics on the motif data, a sequence background is necessary. MARGE and HOMER can create the pre-parsed backgrounds on the run, but this takes some time. It is also possible to download some of these backgrounds with MARGE (see section 7.1):

```
MARGE.pl download -motif_analysis_folders c57bl6j, nodshiltj, spreteij -genome
mm10 -window 200
```

With the peak files called in section 6.1.7 de-novo motif analysis can be performed (for more details, see section 11.1):

```
MARGE.pl denovo_motifs peaks_C57_LPM_PU1.txt mm10 motifs_peaks_C57_LPM_PU1 -fg_ind
c57bl6j
```

```
MARGE.pl denovo_motifs peaks_NOD_LPM_PU1.txt mm10 motifs_peaks_NOD_LPM_PU1 -fg_ind
nodshiltj
```

```
MARGE.pl denovo_motifs peaks_SPRET_LPM_PU1.txt mm10 motifs_peaks_SPRET_LPM_PU1
-fg_ind spreteij
```

The results of the motif analysis are found in motifs_peaks_C57_LPM_PU1, motifs_peaks_NOD_LPM_PU1, and motifs_peaks_SPRET_LPM_PU1).

6.1.10 MARGE motif mutation analysis

Finally, we demonstrate how to run MARGE's motif mutation analysis in order to determine the association between the existence of a transcription factor binding motif and the binding of another transcription factor. We demonstrate how to run the pairwise comparison, as well as the all versus all comparison.

Pairwise comparison For the pairwise comparison, the peak files of the strains have to be merged in a pairwise manner:

```
mergePeaks peaks_C57_LPM_PU1.txt peaks_NOD_LPM_PU1.txt > merge_C57_NOD_LPM_PU1.txt
```

```
mergePeaks peaks_C57_LPM_PU1.txt peaks_SPRET_LPM_PU1.txt > merge_C57_SPRET_LPM_PU1.txt
```

```
mergePeaks peaks_NOD_LPM_PU1.txt peaks_SPRET_LPM_PU1.txt > merge_NOD_SPRET_LPM_PU1.txt
```

These merged files are now annotated with the binding data from their respective tag directories:

```
annotatePeaks.pl merge_C57_NOD_LPM_PU1.txt mm10 -noann -nogene -d tag_C57_LPM_PU1
tag_NOD_LPM_PU1 > ann_merge_C57_NOD_LPM_PU1.txt
```

```
annotatePeaks.pl merge_C57_SPRET_LPM_PU1.txt mm10 -noann -nogene -d tag_C57_LPM_PU1
tag_SPRET_LPM_PU1 > ann_merge_C57_SPRET_LPM_PU1.txt
```

```
annotatePeaks.pl merge_NOD_SPRET_LPM_PU1.txt mm10 -noann -nogene -d tag_NOD_LPM_PU1
tag_SPRET_LPM_PU1 > ann_merge_NOD_SPRET_LPM_PU1.txt
```

Finally, MARGE's pairwise motif mutation analysis can be run on each of the annotated merged peak files. Here we do not demonstrate how to make distance plots or mutation position plots, but in the pdf document showing the MARGE paper commands, more sophisticated analyses are shown (file TBA). The provided command will utilize 8 cores for the analysis. However, if you have less than 8 cores on your machine, MARGE will automatically use all available cores. If you have more, feel free to use all available cores, as it will speed up your analysis considerably.

```
MARGE.pl mutation_analysis -file ann_merge_C57_NOD_LPM_PU1.txt -ind c57bl6j, nodshiltj
-output output-C57-NOD-PU1 -plot plot-C57-NOD-PU1-LPM -core 8
```

```
MARGE.pl mutation_analysis -file ann_merge_C57_SPRET_LPM_PU1.txt -ind c57bl6j,
spreteij -output output-C57-SPRET-PU1 -plot plot-C57-SPRET-PU1-LPM -core 8
```

```
MARGE.pl mutation_analysis -file ann_merge_NOD_SPRET_LPM_PU1.txt -ind nodshiltj,
spreteij -output output-NOD-SPRET-PU1 -plot plot-NOD-SPRET-PU1-LPM -core 8
```

Each analysis will output two pdf files. One pdf shows the S-plot, whereas the other shows a density plot. For more information, please see the Supplemental Data of the MARGE paper (ADD REF).
To visualize and summarize the results of these three runs, a heatmap can be generated:

```
MARGE.pl summary_heatmap -files plot-C57-NOD-PU1-LPM_with_motif.R, plot-C57-SPRET-PU1-LPM_with_motif.R,
plot-NOD-SPRET-PU1-LPM_with_motif.R, -names C57-NOD-PU1, C57-SPRET-PU1, NOD-SPRET-PU1 -sig
-output heatmap-all-pairwise-comparisons.pdf
```

All versus all comparison To run a linear mixed effect model on the data, the peak files for all three strains are merged into one peak file:

```
mergePeaks peaks_C57_LPM_PU1.txt peaks_NOD_LPM_PU1.txt peaks_SPRET_LPM_PU1.txt >
merge_C57_NOD_SPRET_PU1_LPM.txt
```

This file then is annotated with the binding data from all three tag directories:

```
annotatePeaks.pl merge_C57_NOD_SPRET_PU1_LPM.txt mm10 -noann -nogene -d tag_C57_LPM_PU1
tag_NOD_LPM_PU1 tag_SPRET_LPM_PU1 > ann_merge_C57_NOD_SPRET_PU1_LPM.txt
```

Now MARGE can be called to run the motif mutation analysis with three different individuals. We recommend to use several cores to speed up this analysis:

```
MARGE.pl mutation_analysis -file ann_merge_C57_NOD_SPRET_PU1_LPM.txt -ind c57bl6j,
nodshiltj, spreteij -core 8 -GLMM_output GLMM-analysis-C57-NOD-SPRET-LPM-PU1.txt
```

After the analysis is finished, MARGE produces one text file (GLMM-analysis-C57-NOD-SPRET-LPM-PU1.txt) which shows the Benjamini-corrected p-values (from the most significant to the least significant motifs).

6.2 Mouse strain RNA-seq data

MARGE does not provide any functionality to analyze RNA-seq data. However, if RNA-seq was performed on several different mouse strains, the mapping to the individualized genomes still improves the downstream analysis. Therefore, we demonstrate here how to prepare MARGE folders (see section 6.1.3 and section 8.1), STAR indices (see section 6.2.5), and individualized genomes (see section 8.2). After mapping we show how to shift the data back to the reference coordinates (see section 6.2.6 and section 10.1).

6.2.1 Download strain RNA-seq data from GEO

For the mouse RNA-seq data, we download data from Gosselin et al. [7]. We get three RNA-seq datasets in large peritoneal macrophages (LPMs) for C57BL/6J (C57), NOD/ShiLtJ (NOD), and SPRET/EiJ (SPRET).

The data can by downloaded via wget <link>:
C57 LPM polyA RNA

ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByExp/sra/SRX/SRX747/SRX747001/SRR1634709/SRR1634709.sra

NOD LPM polyA RNA

ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByExp/sra/SRX/SRX747/SRX747032/SRR1634740/SRR1634740.sra

SPRET LPM polyA RNA

ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByExp/sra/SRX/SRX747/SRX747033/SRR1634741/SRR1634741.sra

Next, the files are renamed:

```
mv SRR1634709.sra C57-LPM-RNA.sra

mv SRR1634740.sra NOD-LPM-RNA.sra

mv SRR1634741.sra SPRET-LPM-RNA.sra
```

6.2.2 Download genetic variation data

Instructions on how to download genetic variation data from the mouse genome project [11] can be found in section 6.1.2. You can follow the instruction in this section step by step to download the VCF files.

6.2.3 Download reference genome

To see how to download the mm10 reference genome see section 6.1.3. You can follow the instruction in this section step by step to download the reference genome for mm10.

The usage for the script is described in more detail in section 8.2.

6.2.4 Prepare MARGE data with MARGE.pl prepare_files

To see how to generate MARGE folders from the already downloaded genetic variation data (see section 6.1.2), see a step by step instruction in section 6.1.4.
For more details about the usage of this script, see section 8.1.

To download the data directly from MARGE, run:

```
MARGE.pl download -MARGE_folders c57bl6j, nodshiltj, spreteij -genome mm10
```

6.2.5 Generate STAR indices

In case you have not installed STAR previously, it can be installed via Anaconda [1] (for details on how to install Anaconda, see section 3.1.2):

```
conda install -c bioconda star
```

To simplify the generation of a STAR index, we recommend to copy the fasta files from the genomes in a separate folder and rename them.

```
mkdir genome-c57 && cp <path to MARGE data folder>/C57BL6J/.*fa genome-c57

mkdir genome-nod && cp <path to MARGE data folder>/NODSHILTJ/.*fa genome-nod

mkdir genome-spret && cp <path to MARGE data folder>/SPRETEIJ/.*fa genome-spret
```

Now rename all files from chrN_allele_1.fa to chrN.fa

For more information about how to generate STAR indices, please consult the STAR manual [2].
First, we create the output directory for the STAR index:

```
mkdir c57bl6j

mkdir nodshiltj

mkdir spreteij
```

To generate a STAR index, run:

```
STAR --runMode genomeGenerate --runThreadN 8 --genomeDir c57bl6j --genomeFastaFiles
genome-c57/chr1.fa genome-c57/chr2.fa genome-c57/chr3.fa genome-c57/chr4.fa genome-c57/chr5.fa
genome-c57/chr6.fa genome-c57/chr7.fa genome-c57/chr8.fa genome-c57/chr9.fa genome-c57/chr10.fa
genome-c57/chr11.fa genome-c57/chr12.fa genome-c57/chr13.fa genome-c57/chr14.fa genome-c57/chr15.fa
genome-c57/chr16.fa genome-c57/chr17.fa genome-c57/chr18.fa genome-c57/chr19.fa genome-c57/chrX.fa
genome-c57/chrY.fa
```

```
STAR --runMode genomeGenerate --runThreadN 8 --genomeDir nodshiltj --genomeFastaFiles
genome-nod/chr1.fa genome-nod/chr2.fa genome-nod/chr3.fa genome-nod/chr4.fa genome-nod/chr5.fa
genome-nod/chr6.fa genome-nod/chr7.fa genome-nod/chr8.fa genome-nod/chr9.fa genome-nod/chr10.fa
genome-nod/chr11.fa genome-nod/chr12.fa genome-nod/chr13.fa genome-nod/chr14.fa genome-nod/chr15.fa
genome-nod/chr16.fa genome-nod/chr17.fa genome-nod/chr18.fa genome-nod/chr19.fa genome-nod/chrX.fa
genome-nod/chrY.fa
```

```
STAR --runMode genomeGenerate --runThreadN 8 --genomeDir spreteij --genomeFastaFiles
genome-spret/chr1.fa genome-spret/chr2.fa genome-spret/chr3.fa genome-spret/chr4.fa
genome-spret/chr5.fa genome-spret/chr6.fa genome-spret/chr7.fa genome-spret/chr8.fa
genome-spret/chr9.fa genome-spret/chr10.fa genome-spret/chr11.fa genome-spret/chr12.fa
genome-spret/chr13.fa genome-spret/chr14.fa genome-spret/chr15.fa genome-spret/chr16.fa
genome-spret/chr17.fa genome-spret/chr18.fa genome-spret/chr19.fa genome-spret/chrX.fa
genome-spret/chrY.fa
```

Generating these indices will take some time. To download the data from MARGE, run this command (for more details see section 7.1):

```
MARGE.pl download -STAR_index c57bl6j, nodshiltj, spreteij -genome mm10
```

After the index is generate, everything is set up to map the data.
SRA files need to be converted to fastq files with fastq-dump [14]. In case fastq-dump is not installed on your system, it is part of the sra-tools and can be installed via Anaconda [1] (for details on how to install Anaconda, see section 3.1.2):

```
conda install -c bioconda sra-tools
```

Next, sra files are converted to fastq files wiht fastq-dump:

```
fastq-dump C57-LPM-RNA.sra
```

```
fastq-dump NOD-LPM-RNA.sra
```

```
fastq-dump SPRET-LPM-RNA.sra
```

Now the data can be mapped with STAR:

```
STAR --genomeDir c57bl6j --readFilesIn C57-LPM-RNA.fastq --outFileNamePrefix
C57-LPM-RNA- --runThreadN 8
```

```
STAR --genomeDir nodshiltj --readFilesIn NOD-LPM-RNA.fastq --outFileNamePrefix
NOD-LPM-RNA- --runThreadN 8
```

```
STAR --genomeDir spreteij --readFilesIn SPRET-LPM-RNA.fastq --outFileNamePrefix
SPRET-LPM-RNA- --runThreadN 8
```

6.2.6 Shift data to reference coordinates

To shift the data back to the reference coordinates, please run:

```
MARGE.pl shift -files NOD-LPM-RNA-Aligned.out.sam -ind nodshiltj
```

```
MARGE.pl shift -files SPRET-LPM-RNA-Aligned.out.sam -ind spreteij
```

For more information, please see section 10.1.

6.2.7 HOMER analysis of RNA-seq data

To use HOMER for further downstream analysis, first tag directories need to be made:

```
makeTagDirectory tag_C57_LPM_polyA_RNA C57-LPM-RNA-Aligned.out.sam

makeTagDirectory tag_NOD_LPM_polyA_RNA NOD-LPM-RNA-Aligned.out_shifted_from_NODSHILTJ.sam

makeTagDirectory tag_SPRET_LPM_polyA_RNA SPRET-LPM-RNA-Aligned.out_shifted_from_SPRETEIJ.sam
```

6.2.8 Visualizing RNA-seq data

To see how to visualize the RNA-seq data in the UCSC genome browser [12], as well as how to visualize the mutations in the mouse strains, see section 6.1.8 for step by step instructions.
To get a general documentation about how to visualize data with HOMER, check out the HOMER documentation (http://www.homer.ucsd/homer/).
For more information about how to generate bed files for mutations in mouse strains, see section 9.2.

6.3 Human ChIP-seq data

In order to process heterozygous data, several additional steps are required in data pre-processing. Each binding site should be assigned with allele-specific reads, which requires to calculate an allele-specific ratio rather than just counting reads, as done for homozygous data. Therefore, we recommend to generate mapping indices for both alleles separate, which will simplify the mapping process. After that only perfectly aligned reads should be consider. Furthermore, to calculate an allele-specific ratio, only reads overlapping mutations are used to generate that. This documentation will provide a step-by-step instruction on how to deal with heterozygous ChIP-seq data. This can also be generalized for RNA-seq data. As a data set, we will use some PU.1 ChIP-seq data sets in human lymphoblast from ([21]).
In this instruction, we will download data (see 6.3.1). Then we will generate MARGE indices for both alleles (see 6.3.4), create indices for mapping (see 6.3.5) and filter perfectly aligned reads, as well as reads only overlapping mutations (see 6.3.7). Finally, we will annotate peaks with allele-specific reads (see 6.3.8) and run a motif mutation analysis on it (see 6.3.9).

6.3.1 Download human data from GEO database

For this example, we will download PU.1 ChIP-seq data sets from two human individuals from ([21]). We will use data from individual NA06985, as well as NA06986.

```
wget https://www.ebi.ac.uk/arrayexpress/files/E-MTAB-3657/E-MTAB-3657.IP_06985_PU1_120521_6_sorted.bam

wget https://www.ebi.ac.uk/arrayexpress/files/E-MTAB-3657/E-MTAB-3657.IP_06986_PU1_120605_7_sorted.bam
```

Next, we converte these files into fastq files.

```
samtools fastq E-MTAB-3657.IP_06985_PU1_120521_6_sorted.bam > PU1_ChIP_human_NA06985.fastq
```

```
samtools fastq E-MTAB-3657.IP_06986_PU1_120605_7_sorted.bam > PU1_ChIP_human_NA06986.fastq
```
newline

6.3.2 Download human genetic variation data

The SNP and InDel files for the individuals NA06985 and NA06985 can be downloaded from the 1000 genome project ([6]):
ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20130502/
Each chromosome is annotated separately, and can be downloaded like this:

```
for i in 1:22; do wget ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20130502/ALL.chr$i.
phase3_shapeit2_mvncall_integrated_v5a.20130502.genotypes.vcf.gz .; done

wget ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20130502/ALL.chrX.
phase3_shapeit2_mvncall_integrated_v5a.20130502.genotypes.vcf.gz .

wget ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20130502/ALL.chrY.
phase3_shapeit2_mvncall_integrated_v5a.20130502.genotypes.vcf.gz .


for i in 1:22; do gunzip ALL.chr$i.phase3_shapeit2_mvncall_integrated_v5a.20130502.genotypes.vcf.gz;
done

gunzip ALL.chrX.phase3_shapeit2_mvncall_integrated_v5a.20130502.genotypes.vcf.gz

gunzip ALL.chrY.phase3_shapeit2_mvncall_integrated_v5a.20130502.genotypes.vcf.gz
```

Alternatively, you can also download the already pre-processed MARGE mutation files. How to do this is described in section 6.3.4.

6.3.3 Download human reference genome

Additionally, the human hg19 genome needs to be downloaded.

```
mkdir genome_hg19_clean && cd genome_hg19_clean
```

You can download each chromosome from the hg19 genome for the UCSC genome browser via rsync:

```
for i in 1:22; do rsync -avzP rsync://hgdownload.cse.ucsc.edu/goldenPath/hg19/chromosomes/chr$i.fa.gz
.

rsync -avzP rsync://hgdownload.cse.ucsc.edu/goldenPath/hg19/chromosomes/chrX.fa.gz
.

do rsync -avzP rsync://hgdownload.cse.ucsc.edu/goldenPath/hg19/chromosomes/chrY.fa.gz .
```

Now the chromosomes files need to be unzipped:

```
gunzip *gz

cd ..
```

6.3.4 Prepare MARGE data for human with MARGE.pl prepare_files

To prepare the indices for NA06985 and NA06986 using the files downloaded from the 1000 genome project (see section 6.3.2), run prepare_files with the parameter -add:

```
for i in 1:22; do MARGE.pl prepare_files -files ALL.chr$i.phase3_shapeit2_mvncall
_integrated_v5a.20130502.genotypes.vcf -hetero -no-genome -ind NA06985, NA06986 -genome hg19
-add; done
```

```
MARGE.pl prepare_files -files ALL.chrX.phase3_shapeit2_mvncall
_integrated_v5a.20130502.genotypes.vcf -hetero -no-genome -ind NA06985, NA06986 -genome hg19
-add
```

```
MARGE.pl prepare_files -files ALL.chrY.phase3_shapeit2_mvncall
_integrated_v5a.20130502.genotypes.vcf -hetero -no-genome -ind NA06985, NA06986 -genome hg19
-add
```

After this pre-processing is finished, it is important to update the file, that contains the last mutation per chromosome. This file was overwritten during the processing of the single chromosomes, so now we need to run:

```
MARGE.pl update_files -ind NA06985, NA06986
```

Now the genome can be generated for the human individuals NA06985 and NA06986.

```
MARGE.pl create_genomes -genome genome_hg19_clean -ind NA06985, NA06986 -hetero
```

To create the different genomes will to some time. It is also possible to just download the already pre-processed mutation data, as well as the genome data from the webserver:

```
MARGE.pl download -genome hg19 -MARGE_folders NA06985, NA06986
```

6.3.5 Generating heterozygous bowtie2 indices

This commands so far generated one MARGE mutation folder per individual. Each folder contains mutation data for allele 1, as well as allele 2. Because it is easier for downstream analysis to generate mapping indices for each allele individually, we recommend to generate a bowtie2 index for allele 1, as well as allele 2.

```
cat <path to MARGE data folder>/NA06985/*allele_1*.*fa » genome-na06985-allele1.fa
```

```
cat <path to MARGE data folder>/NA06985/*allele_2*.*fa » genome-na06985-allele2.fa
```

```
cat <path to MARGE data folder>/NA06986/*allele_1*.*fa » genome-na06986-allele1.fa
```

```
cat <path to MARGE data folder>/NA06986/*allele_2*.*fa » genome-na06986-allele2.fa
```

To generate a bowtie2 index, run:

```
bowtie2-build genome-na06985-allele1.fa na06985-allele1
```

```
bowtie2-build genome-na06985-allele2.fa na06985-allele2

bowtie2-build genome-na06986-allele1.fa na06986-allele1

bowtie2-build genome-na06986-allele2.fa na06986-allele2
```

These commands will take some time. MARGE also allows to directly download the already pre-processed bowtie2 indices:

```
MARGE.pl download -genome hg19 -bowtie_index NA06985, NA06986
```

For each bowtie2 command 6 files are generated na06985-allele1.1.bt2, na06985-allele1.2.bt2, na06985-allele1.3.bt2, na06985-allele1.4.bt2, na06985-allele1.rev.1.bt2, na06985-allele1.rev.2.bt2). We recommend to move these files into a specific bowtie2 index folder.

```
mkdir -p bowtie2/index && mv *bt2 bowtie2/index
```

After generating the indices or downloading them, the ChIP-seq data can now be mapped. Each ChIP-seq data set should be mapped to allele 1, as well as allele 2.

```
bowtie2 -p 8 -x bowtie2/index/na06985-allele1 PU1_ChIP_human_NA06985.fastq>
PU1_ChIP_human_NA06985_allele_1.sam 2> PU1_ChIP_human_NA06985_allele_1.log

bowtie2 -p 8 -x bowtie2/index/na06985-allele2 PU1_ChIP_human_NA06985.fastq>
PU1_ChIP_human_NA06985_allele_2.sam 2> PU1_ChIP_human_NA06985_allele_2.log

bowtie2 -p 8 -x bowtie2/index/na06986-allele1 PU1_ChIP_human_NA06986.fastq>
PU1_ChIP_human_NA06986_allele_1.sam 2> PU1_ChIP_human_NA06986_allele_1.log

bowtie2 -p 8 -x bowtie2/index/na06986-allele2 PU1_ChIP_human_NA06986.fastq>
PU1_ChIP_human_NA06986_allele_2.sam 2> PU1_ChIP_human_NA06986_allele_2.log
```

6.3.6 Shift human data to reference coordinates

After mapping the data, it has to be shifted to the reference coordinates for downstream analysis:

```
MARGE.pl shift -files PU1_ChIP_human_NA06985_allele_1.sam, PU1_ChIP_human_NA06985_allele_2.sam
-ind NA06985

MARGE.pl shift -files PU1_ChIP_human_NA06986_allele_1.sam, PU1_ChIP_human_NA06986_allele_2.sam
-ind NA06986
```

6.3.7 Filter perfectly aligned reads from mapped data

In order to call allele-specific reads, only perfectly aligned reads can be considered. Mismatches can be due to sequencing errors, but also due to point mutations. To ensure that only correct

reads are considered, all reads that map with mismatches have to be filtered out. Furthermore, in order to calculate an allele-specific ratio for each peak, only reads overlapping mutations have to be used. Therefore, MARGE provides a script that creates a file containing only perfectly aligned reads, as well as a file containing perfectly aligned reads that overlap with mutations between the two alleles.

```
MARGE.pl allele_specific_reads -ind NA06985 -files
PU1_ChIP_human_NA06985_allele_1_shifted_from_NA06985.sam,
PU2_ChIP_human_NA06985_allele_2_shifted_from_NA06985.sam
```

```
MARGE.pl allele_specific_reads -ind NA06986 -files
PU1_ChIP_human_NA06986_allele_1_shifted_from_NA06986.sam,
PU2_ChIP_human_NA06986_allele_2_shifted_from_NA06986.sam
```

This script provides two files per sam file, called "only_muts_<input sam file>" and "perfect_reads_<input sam file". To annotate data allele-specific, HOMER tag directories need to be generated from both files. In this example, there are 8 files after running both commands:

- only_muts_PU1_ChIP_human_NA06985_allele_1_shifted_from_NA06985.sam
- only_muts_PU1_ChIP_human_NA06985_allele_2_shifted_from_NA06985.sam
- only_muts_PU1_ChIP_human_NA06986_allele_1_shifted_from_NA06986.sam
- only_muts_PU1_ChIP_human_NA06986_allele_2_shifted_from_NA06986.sam
- perfect_reads_PU1_ChIP_human_NA06985_allele_1_shifted_from_NA06985.sam
- perfect_reads_PU1_ChIP_human_NA06985_allele_2_shifted_from_NA06985.sam
- perfect_reads_PU1_ChIP_human_NA06986_allele_1_shifted_from_NA06986.sam
- perfect_reads_PU1_ChIP_human_NA06986_allele_2_shifted_from_NA06986.sam

For each file a tag directory is generated:

```
makeTagDirectory PU1_ChIP_human_NA06985_allele_1_only_muts
only_muts_PU1_ChIP_human_NA06985_allele_1_shifted_from_NA06985.sam
```

```
makeTagDirectory PU1_ChIP_human_NA06985_allele_1_perfect_reads
perfect_reads_PU1_ChIP_human_NA06985_allele_1_shifted_from_NA06985.sam
```

```
makeTagDirectory PU1_ChIP_human_NA06985_allele_2_only_muts
only_muts_PU1_ChIP_human_NA06985_allele_2_shifted_from_NA06985.sam
```

```
makeTagDirectory PU1_ChIP_human_NA06985_allele_2_perfect_reads
perfect_reads_PU1_ChIP_human_NA06985_allele_2_shifted_from_NA06985.sam
```

```
makeTagDirectory PU1_ChIP_human_NA06986_allele_1_only_muts
only_muts_PU1_ChIP_human_NA06986_allele_1_shifted_from_NA06986.sam
```

```
makeTagDirectory PU1_ChIP_human_NA06986_allele_1_perfect_reads
perfect_reads_PU1_ChIP_human_NA06986_allele_1_shifted_from_NA06986.sam
```

```
makeTagDirectory PU1_ChIP_human_NA06986_allele_2_only_muts
only_muts_PU1_ChIP_human_NA06986_allele_2_shifted_from_NA06986.sam

makeTagDirectory PU1_ChIP_human_NA06986_allele_2_perfect_reads
perfect_reads_PU1_ChIP_human_NA06986_allele_2_shifted_from_NA06986.sam
```

6.3.8 Allele-specific annotation

In order to annotate peaks in an allele-specific manner, peaks are called on the tag directories of the perfectly aligned reads:

```
findPeaks PU1_ChIP_human_NA06985_allele_1_perfect_reads -style factor > peaks_PU1_NA06985_allele_1.txt

findPeaks PU1_ChIP_human_NA06985_allele_2_perfect_reads -style factor > peaks_PU1_NA06985_allele_2.txt

findPeaks PU1_ChIP_human_NA06986_allele_1_perfect_reads -style factor > peaks_PU1_NA06986_allele_1.txt

findPeaks PU1_ChIP_human_NA06986_allele_2_perfect_reads -style factor > peaks_PU1_NA06986_allele_2.txt
```

Next, the peak files that should be annotated in a allele-specific manner should be merged. MARGE can annotate a file with allele-specific reads from only one individual, or as many individuals as defined. In this example, we merge all four peak files and then annotate it.

```
mergePeaks peaks_PU1_NA06985_allele_1.txt peaks_PU1_NA06985_allele_2.txt
peaks_PU1_NA06986_allele_1.txt peaks_PU1_NA06986_allele_2.txt > merge_PU1_ChIP_NA06985_and_NA06986.txt
```

Finally, we annotate this file with allele-specific reads:

```
MARGE.pl annotate_allele_specific -tag_perfect PU1_ChIP_human_NA06985_allele_1_perfect_reads,
PU1_ChIP_human_NA06985_allele_2_perfect_reads, PU1_ChIP_human_NA06986_allele_1_perfect_reads,
PU1_ChIP_human_NA06986_allele_2_perfect_reads -tag_mut PU1_ChIP_human_NA06985_allele_1_only_muts,
PU1_ChIP_human_NA06985_allele_2_only_muts, PU1_ChIP_human_NA06986_allele_1_only_muts,
PU1_ChIP_human_NA06986_allele_2_only_muts -ind NA06985, NA06986 -merged_file
merge_PU1_ChIP_NA06985_and_NA06986.txt -no_resize -output
allele_specific_annotated_merge_PU1_ChIP_NA06985_and_NA06986.txt -hetero
```

The output file allele_specific_annotated_merge_PU1_ChIP_NA06985_and_NA06986.txt annotates each peak with all allele-specific reads. The ratio for the allele-specificity was calculated based on the perfectly aligned reads overlapping mutations. The peaks were then assigned with all reads from the perfectly aligned reads according to the ratio calculated.

6.3.9 Run MARGE motif mutation analysis on heterozygous data

Finally, MARGE can run motif mutation analysis on heterozygous data sets. Using the file that was annotated allele-specific in the previous section (6.3.8), MARGE can be run with following command:

```
MARGE.pl mutation_analysis -file allele_specific_annotated_merge_PU1_ChIP_NA06985_and_NA06986.txt
-ind NA06985, NA06986 -hetero -GLMM_output motif_mutation_analysis_PU1_NA06985_vs_NA06986.txt
```

The results of this analysis can be visualized in a summary heatmap:

```
MARGE.pl summary_heatmap -files motif_mutation_analysis_PU1_NA06985_vs_NA06986.R,
motif_mutation_analysis_PU1_NA06985_vs_NA06986.R -method all -sig
```

Please note, that this script requires two input files (because R requires two rows in order to draw a heatmap), so please specify the same file twice.

6.4 F1 mouse data

Currently, there is no published F1 mouse data available. As soon as there is data available for download, we will add this section. In the meantime, please refer to the mouse example pipelines for general instructions of how to create/download the pre-processed data. For the downstream analysis after mapping, please refer to the human heterozygous data section. However, there are some minor differences between processing F1 mouse data and heterozygous human data. Here, we explain the most important differences.

6.4.1 Mapping and filtering of F1 data

Instead of mapping the data to the two alleles for each human individual, the data should be mapped to both parental genomes. After mapping the perfectly aligned reads, as well as the reads overlapping mutations have to be filtered out. This can be done by using MARGE.pl allele_specific_reads. Please note, that instead of specifying the individual with -ind <individual> you have to specify both parental genomes with -inds <parent 1>, <parent 2>. Furthermore, the -hetero parameter should not be used, because both parental genomes are homozygous.
After filtering, tag directories should be created for all files containing only mutations, as well as all perfectly aligned reads.

6.4.2 Allele-specific annotation of F1 mice

After creating the tag directories, peaks can be annotated in an allele-specific manner by using MARGE.pl annotate_allele_specific. Please use the -F1 parameter. Furthermore, do not use the -hetero parameter.

7 Download pre-processed data

7.1 MARGE.pl download

To use MARGE, some data needs to be pre-processed. Section 8 has detailed instructions on how to pre-process the data, but MARGE allows to download all pre-processed data that was used while MARGE was developed. There are several different pre-processing steps depending on what part of the pipeline is used.

1. MARGE folders: these folders contain the pre-processed mutation data, the individual genomes and the shift vectors. These folders are always needed when using MARGE and we highly recommend to download the folders for all individuals that are used in the analysis

2. STAR indices: STAR indices for individualized genomes were generated to facilitate mapping of the data to the correct genome with STAR.

3. bowtie indices: bowtie2 indices for individualized genomes were generated to facilitate mapping of the data to the correct genome with bowtie2.

4. motif background: pre-pre-processedd HOMER motif backgrounds are needed in order to run MARGE.pl denovo_motifs.

5. motif distribution: List of all occurrences of motif in the whole genome. This list is needed in order to make distance plots with MARGE.pl mutation_analysis.

7.1.1 Usage for MARGE download

To see usage

```
MARGE.pl download
```

```
Script to download already pre-processed data for MARGE
The script offers downloads for MARGE folders and individualzed genomes, as
    well as background scans for motifs and pre-processed files for motif
   analysis per strain (200bp)


Usage:
    -genome <genome>: The genome for which the motifs were scanned for (e.g
        . mm10/hg19)
    -MARGE_folders <list of folders>: Downloads the list of all folders
        specified in the list.
        If set to all: downloads all available folders
        If empty: lists all available folders
    -motif_bg <list of motifs>: Downloads all the motif background scans
        for the motifs listed here.
        If set to all: downloads all available motifs
        If empty: lists all motif backgrounds for genome specified in
            motif_genome
    -motif_analysis_folders <list of folders> : Downloads all the pre-
        processed motif analysis backgrounds specified in the list.
        If set to all: downloads all available pre-processed motif
            backgrounds
        If empty: lists all available pre-processed motif backgrounds
    -motif_window <bp>: Motif window for which files should be downloaded (
        default in HOMER motif analysis - 200bp)
    -STAR_index <list of individuals/strains>: Downloads the STAR indices
        for all individuals specified.
        If set to all: downloads all available STAR indices
        If empty: lists all available STAR indices
    -bowtie_index <list of individuals/strains>: Downloads the bowtie
        indices for all individuals specified.
        If set to all: downloads all available bowtie indices
        If empty: lists all available bowtie indices
    -complete <list of individuals/strains>: Downloads all data per strain
        (MARGE folder, motif background, STAR index and bowtie index>
        If set to all: downloads all available folders (NOT RECOMMENDED!)
        If empty: Lists all available MARGE folders

List of all content to download:
    -MARGE_folders_list: lists all available MARGE folders
    -motif_analysis_list: lists all available motif analysis backgrounds
    -bowtie_index_list: lists all available bowtie indices
    -STAR_index_list: lists all available STAR indices
    -motif_bg_list: lists all available backgrounds for distance plots
```

7.1.2 Example for MARGE download

In order to process ChIP-seq data from two commonly used mouse strains (BALB/cJ and NOD/ShiLtJ) and compare them to the reference mouse strain (C57BL/6J) the MARGE folders, as well as the bowtie2 indices need to be downloaded.

```
MARGE.pl download -MARGE_folders balbcj, nodshiltj, c57bl6j -bowtie_index balbcj,
nodshiltj, c57bl6j -genome mm10
```

If you want to see all available indices for bowtie2 you can run:

```
MARGE.pl download -bowtie_index
```

or

```
MARGE.pl download -bowtie_index_list
```

8 Getting started with data processing

This section explains this part of the MARGE usage:

```
Getting started with data processing
        MARGE.pl prepare_files
                This script takes VCF input files and generates MARGE
                   mutation files, shifting files and the genomes for the
                   individuals from the VCF file

        MARGE.pl create_genomes
                This script generates the individual genomes for all
                   individuals specified when called. It only works when
                   the mutation and shifting files were already prepared
                   using MARGE.pl prepare_files

        MARGE.pl update_files
        For some projects (e.g. the 1000 genome project) mutations are
           annotated per chromosomes. Therefore, MARGE.pl prepare_files was
            run per chromosome. In this case some of the MARGE files were
           overwritten and need to be adjusted after finishing MARGE.pl
           prepare_files for all chromosomes.
```

Whenever a genome is used that is not part of MARGE's pre-processed data sets, the data has to be pre-processed by the user. MARGE can only process natural genetic variation in VCF format. MARGE offers some very basic VCF file format filtering steps, but it is highly recommended to use VCFtools [8]. For some sequencing projects like the mouse genome project [11], SNPs and InDels are annotated in separate files, whereas other projects like the 1000 Genome project [6] provides one large file with SNPs and InDels. When SNPs and InDels are provided separately, MARGE merges them as a first step. If a combined file is provided then the first processing step is skipped. In cases where SNPs overlap deletions or insertions within one genomic background the SNP is filtered out and the longer mutation is kept. MARGE also simplifies the annotation of the variants per genotype. In cases where more than one possible mutation occurs in a particular genomic location (e.g. two different genotypes have two different mutations in comparison to the reference genome), the mutation is not always annotated as the shortest mutation per genotype. Figure 2a in the MARGE paper (ADD REFERENCE) exemplifies this. The genetic variant for genotype2 is annotated as GTT -> GTTGTT. MARGE processes each genotype separately and therefore calculates the shortest genetic variation for each genotype (in this case

T -> TGTT). Besides processing the VCF files and generating separate MARGE folders per individual, MARGE can also generate the individualized genomes based on a reference, which is required for further downstream analysis and also to generate indices for mapping tools. In case MARGE folders were generated chromosome by chromosome, MARGE needs to update some files after all pre-processing is down using MARGE.pl update_files.

8.1 MARGE.pl prepare_files

This script can prepare all folders necessary for MARGE's downstream functionalities. If this script is run per chromosome separately, the -add parameter needs to be used. After finishing all pre-processing, MARGE has to update some files for all MARGE folders. Please run MARGE.pl update_files. All files need to be ASCII text files. This script does NOT unzip any files.

8.1.1 Usage for MARGE.pl prepare_files

To see usage

```
MARGE.pl prepare_files
```

```
Usage:

Input files:
    -files <file>: Files with mutations in VCF format
        either one file containing SNPs and InDels or two files (one with
           SNPs, one with InDels)
        when using two files the order is SNP_file, InDel_file (either
           comma separated or with white space)

Additional parameters
    -ind <list of individuals>: comma separated list of strains to include
       - when empty every strain in VCF file is considered
    -hetero: Assumes phenotype is heterozygous - default: homozygous
    -genome: path to fasta files per chromosome: input directory of the
       reference genome per chromosome in fasta file format
    -no-genome: Does not create strains specific genomes (default: off)
    -ref <name>: Software generates a reference genome folder with all
       necessary files for further analysis - Folder is called REFERENCE,
       if nothing is specified here
    -force: Overwrites existing folder
    -add: Adds data to existing folder - if file exists it is overwritten
    -core <number of cores>: default 1

Config parameters - these parameters are defined in the config, but can be
   changed
    CAUTION: If these parameters are changed, the config file either needs
       to be adjusted or they need to be defined for every script
    -dir: output directory for mutation files for strains folders - default
       : folder specified in config file
    -genome_dir: output directory for fasta genome file per strain -
       default: folder specified in config file

Filter VCF files:
    For a more sophisticated filtering - use vcftools
    -filter - Filters out all mutations that did not pass all filters
    -same - Filters out all mutations that are homozygous
```

```
        when position is homozygous and -same is not defined the first
            allele is taken without any further evaluation


    -h | --help: shows help
```

If SNP and InDels are annotated in different files and two files are used for MARGE, it is important that the order is correct. The first file contains the SNPs, the second file contains InDels. Furthermore, the files need to be sorted by chromosome and chromosome position.
In order to generate the individualized genomes, MARGE requires the path to one fasta file per chromosome.
It is possible to change the directory where MARGE outputs the MARGE folders and the genomes. However, this is NOT recommended. If you decided you want to use this parameter, please note that you either have to change the path in the config file in order to access the data or you have to specify the path every time you run any MARGE script that needs to access the MARGE folders and genomes.

8.1.2 Example for MARGE.pl prepare_files

To prepare files for two strains of mice (BALB/cJ and NOD/ShiLtJ) with two separate files from the mouse genome project run:

```
MARGE.pl prepare_files -files mgp.v3.snps.rsIDdbSNPv137.vcf, mgp.v3.indels.rsIDdbSNPv137.vcf
-ind balbcj, nodshiltj -genome /data/genomes/mm10/ -core 6
```

To prepare files for human individuals (e.g. NA19865 and NA19868) from the 1000 genome projects per chromosome, run:

```
MARGE.pl prepare_files -files ALL.chr1.phase3_shapeit2_mvncall_integrated_v5a.20130502.genotypes.vcf
-ind na19865, na19868 -hetero

MARGE.pl prepare_files -files ALL.chr2.phase3_shapeit2_mvncall_integrated_v5a.20130502.genotypes.vcf
-ind na19865, na19868 -hetero -add

⋮

MARGE.pl prepare_files -files ALL.chr22.phase3_shapeit2_mvncall_integrated_v5a.20130502.genotypes.vcf
-ind na19865, na19868 -hetero -add
```

To keep MARGE files up to date, also run:

```
MARGE.pl update_files -ind na19865, na19868 -hetero
```

8.2 MARGE.pl create_genomes

In order to extract individual specific sequences, as well as to generate indices for mapping software, a individualized genome is necessary. MARGE can create these individualized genomes, either while processing VCF files, or separately with this script.

8.2.1 Usage for MARGE.pl create_genomes

To see usage

```
MARGE.pl create_genomes
```

```
Usage:
    -genome <genome>: Path to folder with fasta files per chromosome
    -ind <individuals>: one or several individuals (comma separated)
    -data_dir <path to folder with individuals mutations> (default defined
        in config)
    -genome_dir <directory where genomes are saved>: Script automatically
        creates a directory for each strain (default in config)
    -ref <name for reference>: creates folder for reference genome (if
        nothing is specified folder is called REFERENCE
    -hetero: Genome is heterozygous (Default: homozygous)
```

It is not recommended to change the data_dir or out parameter. If you decide to change it, either also change the config file or keep in mind that you have to specify this folder every time MARGE needs to access this data.

8.2.2 Example for MARGE.pl create_genomes

In this example the genomes can be found in /data/genomes. However, this is an arbitrary path and changes are very high, that the fasta file for your reference genome are in another location. Many genomes for reference species can be downloaded from the UCSC genome browser of other whole genome projects.
To create genomes for the two strains of mice from the previous example, run:

```
MARGE.pl create_genomes -genome /data/genomes/mm10/ -ind balbcj, nodshiltj
```

In order to create the genomes for the human individuals from the previous example run:

```
MARGE.pl create_genomes -genome /data/genomes/hg19/ -ind na19865, na19868
```

8.3 MARGE.pl update_files

This script only needs to be used when the MARGE folders were generated from single files per chromosome, using the -add parameter in MARGE.pl prepare_files. However, if you did this, the script is mandatory, otherwise, MARGE will throw errors in the downstream analysis.

8.3.1 Usage for MARGE.pl update_files

To see usage

```
MARGE.pl update_files
```

```
This script updates the last_shift and lookup tables
It should be used when mutation files for each individual were generated
   using a different file per chromosome
The -add parameter must have been used
The last_shift and lookup tables only contain the last entry, so it needs
   to be updated and all chromosomes need to be added


Usage:
   -ind <list of individuals>
   -hetero: Strain is heterozygous (Default: homozygous)
   -data_dir: Directory where mutation files are located - default: folder
       specified in config file
```

8.3.2 Example for MARGE.pl update_files

In our last example, the human MARGE files were created from files separate per chromosome. Therefore, the MARGE folders need to be updated:

```
MARGE.pl update_files -ind na19865, na19868 -hetero
```

9 Access to mutation data

This section explains this part of the MARGE usage:

```
Access to mutation data
   MARGE.pl genome_interactions
       This script allows to extract sequences for different individuals.
          It also provides protein sequences and alignments. The
          alignments are based on the VCF file and are not generated using
           any alignment algorithms

   MARGE.pl mutation_bedfiles
       This script outputs a bed file per individual per allele that can
          be uploaded to the UCSC genome browser

   MARGE.pl mutation_info
       This script counts mutations between different individuals and
          outputs them in a table format. It can either count all
          mutations in comparison to the reference or all private
          mutations

   MARGE.pl extract_sequences
       This script extracts the individual-specific sequences in fasta
          format.

   MARGE.pl annotate_mutations
       This script adds all mutations within specified genomic locations
          for each individual at the end of the input file
```

This part of MARGE allows the user to interact with the individualized genomes. It allows to extract and align sequences between different individuals, generate UCSC genome browser [12] mutation bed files. It also counts different mutations between individuals and can either extract the original sequences for each individual or annotate peaks or genes with mutations per specified individual.

9.1 MARGE.pl genome_interactions

This script allows the user to align nucleotide sequences, protein sequences etc. based on the annotation in the VCF files. This script does NOT run any local or global alignment algorithms (like Smith-Waterman or Needleman-Wunsch). Therefore, the output is not the best alignment, but the "true" alignment based on the VCF files (which obviously can be flawed, but nothing we can do about that in this part of the analysis). The script allows to extract sequences based on gene names (if a corresponding file is provided), as well as RefSeq IDs (also if the file is provided).

9.1.1 Usage for MARGE.pl genome_interactions

To see usage

```
MARGE.pl genome_interactions
```

```
Usage

    -method <gene|protein|both|genomic|genomic_protein|genomic_both|
        align_gene|align_protein|align_both|align_genomic_gene|
        align_genomic_protein|align_genomic_both>
         gene: outputs the gene nucleotide sequence based on RefSeq of Gene
            ID annotation
         protein: outputs the amino acid sequence of a gene based on RefSeq
            or Gene ID annotation
         both: outputs the nucleotide and amino acid sequence of a gene
            based on RefSeq or GENE ID annotation
             File with RefSeq or Gene ID coordinates have to be specified
                with -refseq_file or _gene_file
             IDs for genes have to be specified with -gene or -transcript
         genomic: outputs the nucleotide sequence for any specified locus
             genomic locus has to be specified with -chr, -start, -end and
                if desired -strand
         genomic_protein: outputs the amino acid sequence of the specified
            locus
         genomic_both: outputs the nucleotide and amino acid sequence of
            specified locus
         align_gene: outputs the alignment of nucleotide gene sequence based
             on mutation annotations in VCF file
         align_protein: outputs the alignment of protein sequence. For this
            the nucleotide sequences are aligned based on mutations from the
             VCF file and then translated into amino acid sequences
         align_both: outputs the alignment of the nucleotide and amino acid
            sequence for a gene based on the mutations in VCF file
         alig_genomic_gene: outputs the alignment the nucleotide sequence of
             a locus specified with -chr, -start, -end (and -strand)
         align_genomic_protein: outputs the alignment of the amino acid
            sequence of a locus specified with -chr, -start, -end (and -
            strand). For this the nucleotide sequences are aligned based on
            mutations from the VCF file and then translated into amino acid
            sequences
         align_genomic_both: outputs the alignment of the nucleotide and
            amino acid sequence for a locus specified with -chr, -start, -
            end (and -strand)
```

```
    -ind <individuals>: Comma-separated list of individuals

Parameters for genes
    -genome: Genome
    -refseq_file: File with RefSeq IDs (check HOMER format)
    -gene_file: File with Gene IDs (check HOMER format)
    -gene <gene name> (Comma separated list)
    -transcript <RefSeq transcript name> (Comma separated list)

Parameters for genomic locations
    -start <pos>: Start position of genomic region (Comma separated list -
       same length as end, chr, and strand)
    -end <pos>: Stop position of genomic region (Comma separated list -
       same length as start, chr, and strand)
    -chr <chromosome>:Chromosome for genomic region (Comma separated list -
        same length as start, end, and strand)
    -strand <+|->: Default: + (Comma separated list - same length as start,
        end, and chr)

Additional parameters:
    -hetero: Data is heterozygous
    -data_dir: Path to data directory for individuals - default specified
       in config
    -genome_dir: Path to fastq files for individuals - default specified in
        config
```

9.1.2 Example for MARGE.pl genome_interactions

In order to see an alignment we randomly pick a genomic location (e.g. chr11:62,873,925-62,873,985).

```
MARGE.pl genome_interactions -genome mm10 -method align_genomic_gene -chr 11 -start
62873925 -end 62873985 -ind c57bl6j, balbcj, nodshiltj
```

This command creates following output

```
$ MARGE.pl genome_interactions -genome mm10 -method align_genomic_gene -chr
    11 -start 62873925 -end 62873985 -ind c57bl6j, balbcj, nodshiltj
Loading shift vectors
C57BL6J - allele 1
ACACACACACACACACACACACACACACACACACACACACGGGGGGGGGGGGAGCGGGAGAGT
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
BALBCJ - allele 1
ACACACACACACACACACACACACACACACACACACACACGGGGGGGGGGGGAGCGGGAGAGT
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
NODSHILTJ - allele 1
ACACACACACACACACACACACACACACAC-------CGGGGGGGGGGGAGCGGGAGAGT
||||||||||||||||||||||||||||| - - - - - - - ||||||||||||||||||||||
$
```

9.2 MARGE.pl mutation_bedfiles

This script generates bed files with all individual specific mutations relative to the reference. This file can be uploaded to the UCSC genome browser [12] to have a graphical representation of the differences in the individuals while look at the data. For heterozygous data, two bed files are generated - one per allele.

9.2.1 Usage for MARGE.pl mutation_bedfiles

To see usage

```
MARGE.pl mutation_bedfiles
```

```
Usage:
    -ind <individuals>: Comma-separated list of individuals
    -data_dir <data folder>: Default specified in config

    output files are named output_mut_<individual>_allele_<allele>.bed
    output files are gzipped and are ready to be uploaded to the UCSC
       genome browser
```

9.2.2 Example for MARGE.pl mutation_bedfiles

To create mutation bedfiles for the two strains of mice from the previous example, run:

```
MARGE.pl mutation_bedfiles -ind balbcj, nodshiltj
```

For the human data, the same command is run:

```
MARGE.pl mutation_bedfiles -ind na06985, na0698
```

9.3 MARGE.pl mutation_info

This script allows to compare the number of mutations between different individuals. It can either count all mutations or all private mutations. Please note: For heterozygous data, the private counts are not implemented. This is a very complicated problem and no good solution was found so far. Therefore, for heterozygous data, only the general mutations can be counted.

9.3.1 Usage for MARGE.pl mutation_info

To see usage

```
MARGE.pl mutation_info
```

```
Usage:
    -ind <individuals>: Comma separated list to count mutations
    -data_dir <directory>: data folder: default specified in config file
    -private: Counts number of private mutations
```

9.3.2 Example for MARGE.pl mutation_info

To see the mutations between our three strains of mice, run:

```
MARGE.pl mutation_info -ind c57bl6j, balbcj, nodshiltj
```

The command creates following output:

```
$ MARGE.pl mutation_info -ind c57bl6j, balbcj, nodshiltj
Reading in mutations for C57BL6J
Reading in mutations for BALBCJ
Reading in mutations for NODSHILTJ
Strain comparison SNPs InDels
C57BL6J vs BALBCJ 4,158,340 240,320
C57BL6J vs NODSHILTJ 4,734,324 272,463
BALBCJ vs NODSHILTJ 4,079,736 286,421
$
```

To just see all private mutations between the three strains of interest; run:

```
MARGE.pl mutation_info -ind c57bl6j, balbcj, nodshiltj -private
```

The command creates following output:

```
$ MARGE.pl mutation_info -ind c57bl6j, balbcj, nodshiltj -private
Reading in mutations for C57BL6J
Reading in mutations for BALBCJ
Reading in mutations for NODSHILTJ
Strain private SNPs private InDels
C57BL6J 0 0
BALBCJ 1,750,238 123,240
NODSHILTJ 2,326,206 155,399
$
```

To get the mutations for the heterozygous human individuals NA06985 and NA06986, run:

```
MARGE.pl mutation_info -ind na06985, na06986
```

This command creates following output:

```
$ MARGE.pl mutation_info -ind NA06985, NA06986
Reading in mutations for NA06985
Reading in mutations for NA06986
Strain comparison   SNPs homo - homo    SNPS homo - hetero  SNPs hetero -
   homo  SNPs hetero - hetero    InDels homo - homo  InDels homo - hetero
      InDels hetero - homo    InDels hetero -hetero
NA06985 vs NA06986  961539  394449  1409480 438 162507  64785   200226
   2040
$
```

9.4 MARGE.pl extract_sequences

To analyze the sequences between the strains in more detail with external software, MARGE provides a script to extract the specific individual sequences. The script requires an input file in HOMER peak format:

```
seq-1   chr10   100439234   100439334   +
seq-2   chr2    73492345    73492445    +
seq-3   chr19   48249523    48249723    -
```

The output file is in fasta format and each sequences is specified as:
<chr>_<start>_<end>_<individual>_<allele>.
If the data is homozygous, the allele is always 1.

9.4.1 Usage for MARGE.pl extract_sequences

To see usage

```
MARGE.pl extract_sequences
```

```
Usage:
    -ind <individuals>: Comma-separated list of individuals
    -file <file>: File with genomic coordinates to pull the sequences
    -output <file>: Name of the output files (Default: sequences.txt)
    -id: Uses peak ID as identifier for sequences - can only be used when
        only one individual was specified
    -hetero: Data is heterozygous (Default: homozygous)

Additional parameters:
    -data_dir <directory>: default defined in config
    -genome_dir <directory>: default defined in config
```

9.4.2 Example for MARGE.pl extract_sequences

To get all sequences for our three strains of interest for the example peak file, run:

```
MARGE.pl extract_sequences -file example.file -ind c57bl6j, balbcj, nodshiltj
```

This command creates an output file called sequences.txt:

```
>chr10_100439234_100439334_BALBCJ_1
CACCTCAAATGGGTTTTGCTGTTAGTTGTCTCTTAGTTTCTTCCTTTACCCAACTCTGTCCTTCCCTCCTCATTTA

>chr10_100439234_100439334_C57BL6J_1
CACCTCAAATGGGTTTTGCTGTTAGTTGTCTCTTAGTTTCTTCCTTTACCCAACTCTGTCCTTCCCTCCTCATTTA

>chr10_100439234_100439334_NODSHILTJ_1
CACCTCAAATGGGTTTTGCTGTTAGTTCTCTTAGTTTCTTCCTTTACCCAACTCTGTCCTTCCCTCCTCATTTAAT

>chr19_48249523_48249723_BALBCJ_1
GAAGTTGAACTCTGACCTCTACACATGAAACTTGGCATATGCAAACACACAAAACACCTGTATACACACATACTCA

>chr19_48249523_48249723_C57BL6J_1
GAAGTTGAACTCTGACCTCTACACATGAAACTTGGCATATGCAAACACACAAAACACCTGTATACACACATACTCA

>chr19_48249523_48249723_NODSHILTJ_1
GAAGTTGAACTCTGACCTCTACACATGAAACTTGGCATATGCAAACACACAAAACACCTGTATACACACATACTCA

>chr2_73492345_73492445_BALBCJ_1
TTGGGTCACTGCCTCACAATGTTGGGTCAGGCTTTTTTCCTTGTCTTTTTCTTTTTGTTTTTTTTTTATATATGT

>chr2_73492345_73492445_C57BL6J_1
TTGGGTCACTGCCTCACAATGTTGGGTCAGGCTTTTTTCCTTGTCTTTTTCTTTTTGTTTTTTTTTTATATATGT

>chr2_73492345_73492445_NODSHILTJ_1
TTGGGTCACTGCCTCACAATGTTGGGTCAGGCTTTTTTCCTTGTCTTTTTCTTTTTGTTTTTTTTTTATATATGT
```

For some analyses it is more convenient to have a file with the peak ID and the sequences right next to it. MARGE supports this format, but it is called with only one individual:

```
MARGE.pl extract_sequences -file example.file -ind balbcj -id -output BALB_seqs.txt
```

This command creates an output file called BALB_seqs.txt:

```
seq-3
    GAAGTTGAACTCTGACCTCTACACATGAAACTTGGCATATGCAAACACACAAAACACCTGTATACACACATA

seq-1
    CACCTCAAATGGGTTTTGCTGTTAGTTGTCTCTTAGTTTCTTCCTTTACCCAACTCTGTCCTTCCCTCCTCAT

seq-2
    TTGGGTCACTGCCTCACAATGTTGGGTCAGGCTTTTTTCCTTGTCTTTTTCTTTTTGTTTTTTTTTTATATA
```

For the human data, each sequences is extracted for each allele:

```
MARGE.pl extract_sequences -file example.file -ind na19685, na19686 -hetero
```

This creates an output file called sequences.txt:

```
>chr10_100439234_100439334_NA06985_1
TTTAAGGTATGGTTTTGGAGTATACGTATATATGTTTTCAAAGGTCGACTCAGTTTGCTCTCAATACTTTCTACTC

>chr10_100439234_100439334_NA06985_2
TTTAAGGTATGGTTTTGGAGTATACGTATATATGTTTTCAAAGGTCGACTCAGTTTGCTCTCAATACTTTCTACTC

>chr10_100439234_100439334_NA06986_1
TTTAAGGTATGGTTTTGGAGTATACGTATATATGTTTTCAAAGGTCGACTCAGTTTGCTCTCAATACTTTCTACTC

>chr10_100439234_100439334_NA06986_2
TTTAAGGTATGGTTTTGGAGTATACGTATATATGTTTTCAAAGGTCGACTCAGTTTGCTCTCAATACTTTCTACTC

>chr19_48249523_48249723_NA06985_1
ACTTGAACCCAGGAGGCAGAGGTTGCAGTGAGCCGAGATCGTGCCATTGCACTCCAGCCTGGACTACAGAGAAAGA

>chr19_48249523_48249723_NA06985_2
ACTTGAACCCAGGAGGCAGAGGTTGCAGTGAGCCGAGATCGTGCCATTGCACTCCAGCCTGGACTACAGAGAAAGA

>chr19_48249523_48249723_NA06986_1
ACTTGAACCCAGGAGGCAGAGGTTGCAGTGAGCCGAGATCGTGCCATTGCACTCCAGCCTGGACTACAGAGAAAGA

>chr19_48249523_48249723_NA06986_2
ACTTGAACCCAGGAGGCAGAGGTTGCAGTGAGCCGAGATCGTGCCATTGCACTCCAGCCTGGACTACAGAGAAAGA

>chr2_73492345_73492445_NA06985_1
GCAGGAGCCCAGGAGATTCACTGCCTCAGACCTGTGTCCCCCTGCCCAGCCTTCCGGGACCTACCTGAGAGCCGGC

>chr2_73492345_73492445_NA06985_2
GCAGGAGCCCAGGAGATTCACTGCCTCAGACCTGTGTCCCCCTGCCCAGCCTTCCGGGACCTACCTGAGAGCCGGC

>chr2_73492345_73492445_NA06986_1
GCAGGAGCCCAGGAGATTCACTGCCTCAGACCTGTGTCCCCCTGCCCAGCCTTCCGGGACCTACCTGAGAGCCGGC
```

```
>chr2_73492345_73492445_NA06986_2
GCAGGAGCCCAGGAGATTCACTGCCTCAGACCTGTGTCCCCCTGCCCAGCCTTCCGGGACCTACCTGAGAGCCGGC
```

9.5 MARGE.pl annotate_mutations

This script annotates HOMER files with all mutations found per individual specified. This is a very fast and easy way to check which genomic loci of interest have natural genetic variation between the individuals.

9.5.1 Usage for MARGE.pl annotate_mutations

To see usage

```
MARGE.pl annotate_mutations
```

```
Usage:
    -file <input file>: coordinates have to be the reference coordinates
    -output <output file>: default <file name>_snps.txt
    -ind <individuals>: one or several individuals (comma separated)
    -hetero: Strains are heterozygous (Default: homozygous)


Options for RNA-seq:
    Annotation of exon-specific mutations for RNA-seq takes a while
    -genome <genome> (e.g. mm10/hg38 - to lookup exon and refseq
       annotations)
    -exons: Just looks for mutations within the exons
    -refseq_file <file>: File with RefSeq IDs (default in HOMER path
       defined in config)
    -gene_file <file>: File with Gene IDs (default in HOMER path defined in
        config)

Additional parameters
    -data_dir <directory>: default defined in config
```

9.5.2 Example for MARGE.pl annotate_mutations

To annotate the previously used file with mutations in our two strains different from the reference, run:

```
MARGE.pl annotate_mutations -file example.file -ind balbcj, nodshiltj
```

This command generates an output file called example.file_snps.txt:

```
ID      chromosome      start   end     strand  BALBCJ - 1      NODSHILTJ -
    1
seq-3   chr19   48249523        48249723        -
seq-1   chr10   100439234       100439334       +               100439260:
    TTG->T
seq-2   chr2    73492345        73492445        +
```

To annotate this file with the heterozygous human data, run:

```
MARGE.pl annotate_mutations -file example.file -ind na06985, na06986 -hetero -output
human_example.file
```

This command creates an output file called human_example.file:

```
ID      chromosome      start   end     strand  NA06985 - 1     NA06985 - 2
        NA06986 - 1     NA06986 - 2
seq-3   chr19   48249523        48249723        -       48249609:T->C
seq-1   chr10   100439234       100439334       +       100439259:T->C
    100439259:T->C  100439259:T->C  100439259:T->C
seq-2   chr2    73492345        73492445        +
```

10 Data handling

This section explains this part of the MARGE usage:

```
Data handling
    MARGE.pl shift
        This script shifts the data mapped to the individual genome to the
            reference coordinates. This step is necessary to compare genomic
             loci between different individuals, as well as for
            visualization of data in the different genome browsers

    MARGE.pl shift_to_strain
        This script shifts the data mapped to the reference genome to the
            coordinates of the individual. Please note that this script will
             be used very rarely. To analyze the data it is better to shift
            all strains data to the reference coordinates, not vice versa!

    MARGE.pl allele_specific_reads
        For heterozygous data, this script only keeps reads that were
            perfectly mapped (to avoid bias introduced by SNPs) and also
            outputs a file with reads that overlap mutations (to call allele
            -specific binding)

    MARGE.pl annotate_allele_specific
        For heterozygous data, this script annotates all genomic loci
            between different individuals. It takes reads that overlap
            mutations for loci with mutations in this individual. In case
            there is no mutation, it takes the perfectly mapped reads for
            this locus and divides it by 2 (because of two alleles)
```

The handling of the data after mapping is one of the most important steps in the pipeline. We recommend to map the data to individualized genomes and then shift them back to the reference coordinates. Furthermore, for heterozygous data, we recommend to only consider perfectly aligned reads, to avoid incorrect alignment due to SNPs. We also recommend to annotate heterozygous data with this script in order to get the best possible annotation. The allele-specific methods are very preliminary and better, more sophisticated tools should be developed. However, so far no publicly available tool was able to handle InDels. Therefore, we recommend to use the MARGE scripts until some statistic based methods are published.

10.1 MARGE.pl shift

The data should be mapped to individualized genomes. For many genomes, the STAR and bowtie2 indices can be downloaded via MARGE.pl download (see section 7.1). Otherwise, we

recommend to generate the genome with MARGE.pl create_genomes (see section 8.2) and then additionally generate the indices for your mapping software. In order to be able to compare data mapped to different individualized genomes, the coordinates need to be shifted back to the reference coordinates.
For heterozygous data, we highly recommend to create a mapping index per allele, and then map the data to both alleles.

10.1.1 Usage for MARGE.pl shift

To see usage

```
MARGE.pl shift
```

```
Usage:

    -dir <directory with files to shift>: has to be the same individual
    -files <list with files>: comma separated list
    -ind: one or several individuals - comma separated
        If only one individual is specified all files are shifted from this
            individual
        If several individuals are specified it has to match the number of
            files specified!
    -keep-alleles: Keeps the allele annotation in the chromosome name if
        present (example chr1_allele_1). Default: change to only chromosome
        name


Format to shift (default = sam):
    -sam: shifts sam file (result file from mapping with bowtie or STAR)
    -homer: shifts HOMER peak and RNA files
    -tag: shifts tag directory
    -bed: shifts bed file
    -bismark: shifts cytosine report output from BISMARK
    -hic: shifts HiC tag directories
    -bedpe: shifts bedpe files (from PLAC-Seq pipeline)

Additional parameters:
    -data_dir <directory>: default defined in config


    -h | --help: prints help
```

MARGE can shift several different formats. If needed, this script can be extended by the authors. Currently MARGE can shift sam files, HOMER peak files, as well as HOMER tag directories and HOMER HiC tag directories. It also can shift bed files, and bismark cytosine reports. Furthermore, it can shift bedpe files, which are bedfiles used for HiC interaction annotations.
When files are specified with -dir, MARGE lists files according to the file format it is supposed to shift. It lists:

1. For sam files: all files ending in .sam
2. For tag directories: all directories within the specified directory. If other directories are in there, MARGE will throw errors while trying to shift them.
3. For HOMER files: all files in the specified directory. If other files are in this directory, MARGE will throw errors while trying to shift them.

4. For bed files: all files ending in .bed
5. For gtf files: all files ending in .gtf
6. For bismark: This functionality is not offered for bismark output files. Please specify the files directly with -files.
7. for bedpe files: all files ending in .bedpe
8. for HiC tag directories: all directories within the specified directory. If other directories are in there, MARGE will throw errors while trying to shift them.

MARGE creates an output file with the shifted data, which is always called <input_file_name>_shifted_from_<individual> and the corresponding file ending.

10.1.2 Example for MARGE.pl shift

After mapping RNA-seq data for BALB/cJ and generating a sam file, the file in our example is called RNA-seq-balbcj.sam. To shift this file back, run:

```
MARGE.pl shift -files RNA-seq-balbcj.sam -ind balbcj
```

To shift RNA-seq data for the human individuals (files called RNA-seq-na06985.sam and RNA-seq-na06986.sam), run:

```
MARGE.pl shift -files RNA-seq-na06985.sam, RNA-seq-na06986.sam -ind NA06985, NA06986
```

The data is automatically shifted according to the allele it is on. If -keep-alleles is not set, the allele annotation is removed while shifting. This is recommended, in order to visualize the data and compare allele specific binding/expression. We do NOT recommend to use -keep-alleles. In case you do, be advised, that MARGE will not be able to analyze your data. MARGE needs both alleles to have to same chromosome name, otherwise it interprets the data not as two alleles from the same chromosome, but data from two different chromosomes.

10.2 Usage for MARGE.pl shift_to_strain

In very rare cases, it is useful to shift files from the reference genome to the genomic coordinates of one or several individual or strains used in the analysis. This is not a good way to do general analysis, and we highly recommend to shift all data to the reference because it makes the analysis easier. In case you want to shift from the reference to the individual, however, this script is useful.
The usage is the same as for MARGE.pl shift, only the shifting is the other direction. All output files are labeld with <intput>_shifted_to_individual.

10.2.1 Usage for MARGE.pl shift_to_strain

To see usage

```
MARGE.pl shift_to_strain
```

```
CAUTION:
This script is very rarely used. It shifts files FROM the reference genome
   TO the individual genome.
It is NOT recommended to run the analysis that way!
Please make sure you really want to shift in this direction!


Usage:

    -dir <directory with files to shift>: has to be the same individual
    -files <list with files>: comma separated list
    -ind: one or several individuals - comma separated
        If only one individual is specified all files are shifted from this
            individual
        If several individuals are specified it has to match the number of
           files specified!
    -keep-alleles: Keeps the allele annotation in the chromosome name if
       present (example chr1_allele_1). Default: change to only chromosome
       name


Format to shift (default = sam):
    -sam: shifts sam file (result file from mapping with bowtie or STAR)
    -homer: shifts HOMER peak and RNA files
    -tag: shifts tag directory
    -bed: shifts bed file
    -bismark: shifts cytosine report output from BISMARK
    -hic: shifts HiC tag directories
    -bedpe: shifts bedpe files (from PLAC-Seq pipeline)
    -gtf: shifts gtf file

Additional parameters:
    -data_dir <directory>: default defined in config


    -h | --help: prints help
```

10.3 MARGE.pl allele_specific_reads

Heterozygous data should be mapped twice independently to both alleles. In order to be able to call allele-specific reads with high confidence, we recommend to only consider reads that mapped without any mismatch. This makes sure that SNPs can not introduce biases. Furthermore, to annotate allele-specific reads for further downstream analysis, MARGE also creates a separate file with all reads that are overlapping mutations. These reads can later be used to estimate the influence of the genetic variance onto gene expression or transcription factor binding/chromatin accessibility.

10.3.1 Usage for MARGE.pl allele_specific_reads

To see usage

```
MARGE.pl allele_specific_reads
```

```
Usage:
    -ind <individual>: Individual we look for muts versus reference
```

```
    -inds <individuals>: Two individuals we look for muts against each
        other
    -files <files>: Comma seperated list of files
    -method <bowtie|star>: define mapping method (default: bowtie)
    -hetero: Data is heterozygous (Default: homozygous)

Additional parameters:
    -v: verbose mode - progress monitoring
    -data_dir <directory>: default defined in config
    -genome_dir <directory>: default defined in config
```

If -ind is specified, MARGE will check for the existence of mutations against the reference. If -inds is specified, MARGE will check for the existence of mutations against the two individuals. It is generally easier to compare the mutations directly to the reference and there are not many cases in which it is necessary to compare to each other directly.
Please note that MARGE assumes that the files are shifted back to the reference coordinates. There will not be any error message if the file is not shifted, but the results will be incorrect!
Currently the script is only able to handle the sam output formats generated by bowtie2 or STAR. This will probably get extended to additional formats in the near future.
Please be patient. Usually sam files are very big files. Therefore, the script takes some time to check for the occurrences of mutations for each read.

10.3.2 Example for MARGE.pl allele_specific_reads

In this case we have two individual human genomes, which both have mutations annotated for two alleles. Therefore, for these two individuals MARGE.pl allele_specific_reads should be run:

```
MARGE.pl allele_specific_reads -files RNA-seq-na06985_shifted_from_NA06985.sam
-ind na06985
```

and

```
MARGE.pl allele_specific_reads -files RNA-seq-na06985_shifted_from_NA06986.sam
-ind na06986
```

These two commands will generate four output files called:

1. perfect_RNA-seq-na06985_shifted_from_NA06985.sam
2. only_muts_RNA-seq-na06985_shifted_from_NA06985.sam
3. perfect_RNA-seq-na06986_shifted_from_NA06986.sam
4. only_muts_RNA-seq-na06986_shifted_from_NA06986.sam

10.4 MARGE.pl annotate_allele_specific

This script annotates allele-specific binding for heterozygous data. To use this script MARGE.pl allele_specific_reads has to be ran first. Next, HOMER tag directories need to be generated from the two output files (perfect_<file>, and only_muts_<file>). Currently, this script only works with ChIP-seq peaks. It can theoretically also be applied to RNA-seq data. It then considers the complete gene (including all introns). To just run it for exons, each exon needs to be provided as a separate genomic loci. In the future exon files might be incorporated to simplify this procedure.
The script annotates each genomic locus individual specific. For each genomic locus MARGE

checks if there is a mutation between the individual and the reference. In case a mutation is found in both alleles of the individual, MARGE checks whether the mutations are different or not. In case there is no mutation, or the mutation of the alleles are the same, MARGE annotates this genomic locus with half the reads mapped to this position (half, because the reads from both alleles mapped to this locus). In case there is an allele-specific mutation, MARGE calculates the percentage of reads per allele based only on reads overlapping mutations.. This is an estimate for the allele-specificity of this region. MARGE then assigns the fraction of the perfectly aligned reads accordingly.

10.4.1 Usage for MARGE.pl annotate_allele_specific

To see usage

```
MARGE.pl annotate_allele_specifc
```

```
Usage:
    -tag_perfect <list of tag directories from perfectly aligned reads>
    -tag_mut <list of tag directories from reads overlapping mutations only
       >
    -ind <list of individuals>
    -hetero: individuals are heterozygous (Default: homozygous)
    -bp <basepairs>: read lengths (default: 50bp)
    -input_perfect <list of tag directores from input - either input per
       individual, or input per allele>
    -peak_files <list of peak files>: Input peak files - if not provided
       script calls peaks with HOMER
    -ann_perfect <list of peak files annotated with perfect mapped reads>
    -ann_mut <list of peak files annotated with mut mapped reads>
    -merged_file <merged peak file>
    -no_resize: does not resize peak file
    -no_ann: does not annotate individual peak files
    -resize: only resizes peaks without any further annotation
    -output_resize <output file name>: output file for resized coordinates
       (default: merged_peaks_resized.txt)
    -output <output file>: default annotated_heterozygous_peaks.txt
    -data_dir <directory>: default defined in config
    -F1: Annotate allele-specific based on a F1 mouse strain. In this case,
        please specify only 2 genomes - both parental mouse strains. These
       genomes are then used as different alleles. This does only work for
       two genomes, not more!
```

The script requires HOMER tag directories for all perfectly aligned reads, as well as all reads that are spanning a mutation (in comparison to the reference!). It also requires a list of all individuals. When the parameter -hetero is set, the list can either be as long as the specified tag directories, or half as long. This script will call peaks on each perfect tag directory if no peak files are specified. It can consider inputs (-input_perfect). This list can again either be as long as the specified tag directories, or half as long. If peak files are defined, the script will skip the peak finding step.
After peak finding, all peak files are merged into one peak file. However, this means many very noisy peaks will be in this file and there is no external way to control this file. To be more stringent a merged peak file that is pre-processed can be used with -merged_file. In this case, no peak calling and no merging is performed.
The script resizes the peak to center +/- bp. This makes sure that the peaks do not get too big. For a high quality ChIP-seq or ATAC-seq data set without much noise, we recommend to

set -no_resize.
The script can also annotate allele-specific reads for F1 strains of mice, when -F1 is set.

10.4.2 Example for MARGE.pl annotate_allele_specific

To annotate a peak file from two PU.1 ChIP-seq experiments for NA06985 and NA06986, the data was first mapped, shifted and then the allele-specific reads and mutation-spanning reads were extracted. In this example we call these four files containing the perfectly aligned reads perfect_ChIP-PU1-NA06985-allele-1_shifted.sam, perfect_ChIP-PU1-NA06985-allele-2_shifted.sam, perfect_ChIP-PU1-NA06986-allele-1_shifted.sam, and perfect_ChIP-PU1-NA06986-allele-2_shifted.sam. The reads that were perfectly aligned and also contained a mutation are called only_muts_ChIP-PU1-NA06985-allele-1_shifted.sam, only_muts_ChIP-PU1-NA06985-allele-2_shifted.sam, only_muts_ChIP-PU1-NA06986-allele-1_shifted.sam, and only_muts_ChIP-PU1-NA06986-allele-2_shifted.sam. From these sam files we generate tag directories called tag_<sam file name>. There is no input file, and we run the script with peak calling, but without resizing:

```
MARGE.pl annotate_allele_specific -tag_perfect tag_perfect_ChIP-PU1-NA06985-allele-1_shifted.sam,
tag_perfect_ChIP-PU1-NA06985-allele-2_shifted.sam, tag_perfect_ChIP-PU1-NA06986-allele-1_shifted.sam,
tag_perfect_ChIP-PU1-NA06986-allele-2_shifted.sam -tag_mut tag_only_muts_ChIP-PU1-NA06985-allele-1_shifted.sam,
tag_only_muts_ChIP-PU1-NA06985-allele-2_shifted.sam, tag_only_muts_ChIP-PU1-NA06986-allele-1_shifted.sam,
tag_only_muts_ChIP-PU1-NA06986-allele-2_shifted.sam -no_resize -ind NA06985, NA06986 -hetero
```

This command generates one output file called annotated_heterozygous_peaks.txt with allele-specific annotation for all heterozygous individuals specified in -ind.

11 Mutation analysis

This section explains this part of the MARGE usage:

```
Mutation analysis
    MARGE.pl denovo_motifs
        HOMER denovo motif analysis extended to integrate the usage of
            different genomes

    MARGE.pl mutation_analysis
        Pairwise or all versus all analysis of the impact of mutations on
            open chromatin (ATAC-Seq, DNAse Hypersensitivity assays etc.) or
             transcription factor binding

    MARGE.pl summary_heatmap
        Script to summarize the results from several runs of pairwise
            mutation analyses or all versus all run
```

This part of the MARGE pipeline allows to analyze the effect of natural genetic variation on transcription factor binding and accessibility of chromatin. MARGE provides a script to perform de-novo motif analysis. This script is a slightly altered version of HOMER's de-novo motif analysis. The only difference is, that MARGE's version allows to extract genome sequences for the individuals, rather than only the reference genome, and therefore provides a better motif analysis results for different individuals. It further can extract sequences for another individual as background. MARGE also offers an analysis that calculates if mutations in a motif are significantly associated with changes in transcription factor binding between two individuals.

For more than two individuals a linear mixed model is used to model the association between the existence or strength of a collaborative transcription factor binding motif and the binding of the transcription factor that was assessed by ChIP-seq (or the openness of chromatin assessed by ATAC-seq, respectively). For more information please see the MARGE publication (ADD REFERENCE ONCE PUBLISHED).
To summarize the analysis results, MARGE provides a script to generate heatmaps. However, this script needs at least two analysis in order to work. MARGE is not able to generate a heatmap from only one sample (mainly because R does not create these heatmaps without errors).

11.1 MARGE.pl denovo_motifs

This script is an extension of HOMER's findMotifsGenomeWide.pl. It uses exactly the same syntax, but additionally allows to specify an individual which sequences are extracted (rather than the reference). It also allows to specify an individual for a specific background, allowing the comparisons of sequences from two different individuals in a de-novo motif analysis.
A detailed information about how this algorithm works can be found in the documentation of HOMER (`http://homer.ucsd.edu/homer/ngs/peakMotifs.html`). This documentation just gives a short overview of the MARGE specific additional parameters.
The de-novo motif analysis requires a pre-processed background. It can generate this background if it can not find it, but it takes a while. MARGE provides pre-processed backgrounds with 200bp windows for a subset of mouse strains. These backgrounds can be downloaded with:

```
MARGE.pl download -motif_analysis_folders
```

11.1.1 Usage of MARGE.pl denovo_motifs

To see usage

```
MARGE.pl denovo_motifs
```

```
Program will find de novo and known motifs in regions in the genome

Usage: findMotifsGenome.pl <pos file> <genome> <output directory> [
    additional options]
Example: findMotifsGenome.pl peaks.txt mm8r peakAnalysis -size 200 -len
    8

Possible Genomes:
        -- or --
    Custom: provide the path to genome FASTA files (directory or single
        file)
        Heads up: will create the directory preparsed/ in same location
            .

Basic options:
    -mask (mask repeats/lower case sequence, can also add 'r' to genome
        , i.e. mm9r)
```

```
    -bg <background position file> (genomic positions to be used as
        background, default=automatic)
        removes background positions overlapping with target positions
        -chopify (chop up large background regions to the avg size of
            target regions)
    -len <x>[,<x>,<x>...] (motif length, default=8,10,12) [NOTE: values
        greater 12 may cause the program
        to run out of memory - in these cases decrease the number of
            sequences analyzed (-N),
        or try analyzing shorter sequence regions (i.e. -size 100)]
    -size <x> (fragment size to use for motif finding, default=200)
        -size <x,x> (i.e. -size -100,50 will get sequences from -100 to
            +50 relative from center)
        -size given (uses the exact regions you give it)
    -S <x> (Number of motifs to optimize, default: 25)
    -mis <x> (global optimization: searches for strings with x
        mismatches, default: 2)
    -norevopp (don't search reverse strand for motifs)
    -nomotif (don't search for de novo motif enrichment)
    -rna (output RNA motif logos and compare to RNA motif database,
        automatically sets -norevopp)

MARGE specific options:
    -fg_ind <individual>: Individual used in the foreground
    -bg_ind <individual>: Individual used in the background
    -compare_motif_file <motif file>: File used to compare de novo
        motifs to
     Default: /data/MARGE_v1.0//config/homer.motifs
    -genome_dir <dir>: directory with fasta files for individuals (
        default defined in config)
    -data_dir <dir>: directory with MARGE folders for individuals (
        default defined in config)

Scanning sequence for motifs
    -find <motif file> (This will cause the program to only scan for
        motifs)

Known Motif Options/Visualization
    -mset <vertebrates|insects|worms|plants|yeast|all> (check against
        motif collects, default: auto)
    -basic (just visualize de novo motifs, don't check similarity with
        known motifs)
    -bits (scale sequence logos by information content, default: doesn'
        t scale)
    -nocheck (don't search for de novo vs. known motif similarity)
    -mcheck <motif file> (known motifs to check against de novo motifs,
    -noknown (don't search for known motif enrichment, default: -known)
    -mknown <motif file> (known motifs to check for enrichment,
    -nofacts (omit humor)

Sequence normalization options:
    -gc (use GC% for sequence content normalization, now the default)
    -cpg (use CpG% instead of GC% for sequence content normalization)
    -noweight (no CG correction)
     Also -nlen <x>, -olen <x>, see homer2 section below.

Advanced options:
    -h (use hypergeometric for p-values, binomial is default)
```

```
        -N <x> (Number of sequences to use for motif finding, default=max
            (50k, 2x input)
        -local <x> (use local background, x of equal size regions around
            peaks to use i.e. 2)
        -redundant <x> (Remove redundant sequences matching greater than x
            percent, i.e. -redundant 0.5)
        -maxN <x> (maximum percentage of N's in sequence to consider for
            motif finding, default: 0.7)
        -maskMotif <motif file1> [motif file 2]... (motifs to mask before
            motif finding)
        -opt <motif file1> [motif file 2]... (motifs to optimize or change
            length of)
        -rand (randomize target and background sequences labels)
        -ref <peak file> (use file for target and background - first
            argument is list of peak ids for targets)
        -oligo (perform analysis of individual oligo enrichment)
        -dumpFasta (Dump fasta files for target and background sequences
            for use with other programs)
        -preparse (force new background files to be created)
        -preparsedDir <directory> (location to search for preparsed file
            and/or place new files)
        -keepFiles (keep temporary files)
        -fdr <x> (Calculate empirical FDR for de novo discovery x=number of
             randomizations)

    homer2 specific options:
        -homer2 (use homer2 instead of original homer, default)
        -nlen <x> (length of lower-order oligos to normalize in background,
             default: -nlen 3)
            -nmax <x> (Max normalization iterations, default: 160)
            -neutral (weight sequences to neutral frequencies, i.e. 25%,
                6.25%, etc.)
        -olen <x> (lower-order oligo normalization for oligo table, use if
            -nlen isn't working well)
        -p <x> (Number of processors to use, default: 1)
        -e <x> (Maximum expected motif instance per bp in random sequence,
            default: 0.01)
        -cache <x> (size in MB for statistics cache, default: 500)
        -quickMask (skip full masking after finding motifs, similar to
            original homer)
        -minlp <x> (stop looking for motifs when seed logp score gets above
             x, default: -10)

    Original homer specific options:
        -homer1 (to force the use of the original homer)
        -depth [low|med|high|allnight] (time spent on local optimization
            default: med)
```

There are only four MARGE specific options. The fg_ind has to be defined in order to run this script. However, the bg_ind is optional and should only be used when a specific background is specified (-bg - see Basic Options), which is from another individual. Furthermore, the motif file that is used to compare denovo motifs with already known motifs is changed. It uses the same motif file as MARGE uses in the motif mutation analysis to have consistent results. The genome that is specified in the HOMER command is mandatory and is always the reference genome (e.g. mm10/hg19/hg38).

11.1.2 Example of MARGE.pl denovo_motifs

To find motifs for a BALB/cJ PU.1 ChIP-seq sample, first peaks need to be called from the shifted tag directory for BALB/cJ (output file called peaks_balbcj_pu1.txt). Then this file is used to find denovo motifs with the extended MARGE version.

```
MARGE.pl denovo_motifs peaks_balbcj_pu1.txt mm10 motif_analysis_BALB_PU1 -fg_ind
balbcj
```

This command runs denovo motif analysis (and also known motif analysis - please check HOMER documentation for more details) and writes the output in a folder called motif_analysis_BALB_PU1.
In case we want to run this analysis, but with the PU.1 ChIP-seq results from NOD/ShiLtJas background, the command is:

```
MARGE.pl denovo_motifs peak_balbcj_pu1.txt mm10 motif_analysis_BALB_vs_NOD_PU1
-fg_ind balbcj -bg peak_nodshiltj_pu1.txt -bg_ind nodshiltj
```

The output of this command can be found in a folder called motif_analysis_BALB_vs_NOD.

11.2 MARGE.pl mutation_analysis

The mutation analysis script offers a pairwise comparison of the effect of natural genetic variation on transcription factor binding or chromatin accessibility. For more details, please see the MARGE publication (ADD REFERENCE).
It is also possible to run an analysis with more than two individuals (all versus all). In this case a linear mixed effects model is utilized. A detailed description of the algorithm can be found in the MARGE publication.

11.2.1 Usage of MARGE.pl mutation_analysis

To see usage

```
MARGE.pl mutation_analysis
```

```
Usage:

Required parameters:
    -file <file>: annotated peak file (including tag counts)
    -ind <individuals>: Comma-separated list of individuals - Order must
       overlay with order in annotated peak file

Peak file manipulation:
    -center: centers peaks on TF specified in -AB (automatically analyzes
       sequences with motif in center - set -far_motif and/or -no_motif for
        analysis of these sequences)
    -center_dist: Distance to peak center that motif can be away from (max)
        to still be centered on (default: 25bp)
    -region: Size of the region used to look for other motifs (Default: 0 -
        peak is not extended)
    -tg <minimal tag count>: Filters out all peaks with less than x tag
       counts
```

```
Additional parameters for pairwise comparison:
    -output <output name>: Name of the output files
    -plots <plot name>: Output name of the plots
    -AB <antibody>: Antibody that was used for this ChIP (to exclude
        mutations in this motif from analysis in the secound round). It is
        also required for centering peaks
    -motif: analyzes sequences with TF motif (only works with -center -
        without centering on TF it is not possible to group the sequences in
         with motif/ far motif/ no motif)
    -no_motif: analyzes sequences without TF motif
    -far_motif: analyzes sequences with TF motif that are more that n bp
        away from peak center (default: 25 - different value can be defined
        in -center_dist)
    -mut_only: just keeps peaks where one individuals is mutated

Additional output for pairwise comparison:
    -tf_direction <list with TF>: (comma separated list) for each of these
        transcription factor 3 output files are printed (all peaks where
        mutation and loss of binding are in the same direction (
        same_direction_<TF>.txt), all peaks with mutations that are between
        significant fold change (direction_between_foldchanges_<TF>.txt) and
         all peaks where mutation and loss of binding are in the opposite
        direction (opposite_direction_<TF>.txt) - all: all motifs
    -tf_dir_name <prefix>: Prefix for the naming of the files created by
        tf_direction option

Plot options for pairwise comparison:
    -mut_pos: Also analyzes the position of the motif that is mutated
    -dist_plot: Plots distance relationships between TF and motif
        candidates
        -effect: Just plots distance relationship for peaks that are
            affected
    -genome <genome>: Genome (Used for distribution plots to generate the
        background distributions)

Additional parameters for all versus all comparison:
    -GLMM_output <name>: Name of the output file from the GLMM analysis (
        default = GLMM_results.txt)
    -measure [Motif|Motif_score]: For all versus all analysis. Either the
        existance of the motif (Motif) or the exact motif score (Motif_score
        ) is considered in the GLMM (default: Motif)

Optional parameters - only change them if you are what you are doing:
    -no_correction: Turns off multiple testing correction
    -motif_diff <difference>: Difference in the motif score to count it as
        mutated motif (for pairwise analysis) - can either be a percentage
        or absolute number (please use % for percentage - default 50% - also
         turned on when only using -delta)
    -delta_tag: Does not use fold change but delta of the tag counts
        between the individuals
    -delta_threshold <threshold>: Difference between tag counts that is
        counted as significant (default: 100)
    -fc_pos <threshold>: Fold change threshold to count peaks as strain
        specific vs not (Default: 2fold)
    -overlap <complete|half|
```

```
Additional options:
    -hetero: Data is heterozygous (Default: homozygous)
    -core <number of cores for model calculation> (default 4): Is only used
        when more than 2 individuals are defined
    -keep: keep temporary files

Parameters to overwrite default paths:
    -TF <file with PWM for transcription factors>: (default in config)
    -data_dir <directory>: default defined in config
    -genome_dir <directory>: default defined in config
```

MARGE's mutation analysis requires an annotated file (for heterozygous data, we recommend to annotate the file with MARGE.pl annotate_allele_specific), as well as all individuals that are in the annotated file. For two individuals, MARGE will run a pairwise comparisons. When more than two individuals are specified, MARGE will automatically run an all versus all comparison utilizing a linear mixed model.
This script allows some basic peak manipulations. It can center the peaks on the motif that is specified in -AB. We recommend to only use this parameter for ChIP-seq and center the peaks on the transcription factor binding motif of the assessed factor. center_dist specifies how far a motif can be the current center of the motif to still be used for centering. Otherwise, the peak will be filtered out. With region the size of the peak can be manipulated. Furthermore, to remove very noisy peaks, it is possible to filter out all peaks with less than a specified number of tags.
MARGE offers some additional parameters. The option -output should always be used when more than one analysis is run at the same time to ensure that the different analyses do not override each others files. -plots specifies the name of the output plots. When -AB is set MARGE runs two rounds of analysis. In the second round all peaks with a mutation in the transcription factor binding motif specified in -AB are removed. This allows to assess the impact of collaborative factors on the binding of the transcription factor without considering any changes in the main transcription factor that was chipped for. When -motif is specified, MARGE analyzes peaks with the motif in the center. With the parameter -far_motif all peaks with the motif outside the center are analyzed and with -no_motif all peaks without any motif are analyzed.
For further downstream analysis, MARGE can also output all peaks with mutations in a collaborative transcription factor binding motif where the loss of the motif also leads to a loss of binding (same direction), the loss of the motif does not affect the binding (direction_between_foldchange) and the loss of the motif increases the binding (opposite_direction). The name of these output files can be changed with the paramater -tf_dir_name.
MARGE can also generate additional plots. More details to these plots can be found in the supplemental section of the MARGE publication. In short, the parameter -mut_pos outputs a detailed analysis of the positions in the binding motif that are mutated and also the effect of the mutation. -dist_plot creates a plot that shows the distribution of the TF of interest around the measured transcription factor. For this plot -center and -AB need to be specified. Furthermore, the genome needs to be scanned for the motif of interest and the motif that was chipped for in order to generate a background distribution. For the motifs available in the MARGE motif file, all motif distributions can be downloaded for the mm10 genome with MARGE download -motif_bg. To just plot the distribution for motif mutations that have an effect on binding -effect can be set. Furthermore, for this analysis a background genome needs to be specified (mm10/hg19 etc.). The scan on the background was only done on the reference genome.
For the all versus all comparison (GLMM) the output of the analysis can be specified with -GLMM_output. MARGE also can consider either differences in the Motif score (-measure Motif_score) or the existence of the motif (-measure Motif), which is the default.

There are several additional options MARGE offers. We recommend to not change any of them unless you are very certain about what you are doing. MARGE corrects p-values with a Benjamini correction, which can be turned off by -no_correction. Furthermore, instead of using a hard cutoff for when a motif is considered present or not, MARGE can analyze differences between the motifs between two individuals with -motif_diff, either specified in percentage or as absolute number.
Instead of using the fold change to call differences in binding, MARGE can also calculate the difference (tag count ind1 - tag count ind2) and define a threshold to call them different of similar in the analysis. Furthermore, MARGE considers all peaks bound by more than 2-fold difference as different and considers these peaks to calculate the p-values. However, this can be changed with -fc_pos. If a more stringent analysis is desired, the fold change can for example be set to 4.
MARGE considers motifs mutated, if they do not overlap perfectly between the individuals. However, sometimes InDels cause motifs to shift slightly in their position, but does actually not affect the motif by itself. With -overlap motifs can overlap by n base pairs in order to be called the same motif. If -overlap complete is set, only perfectly overlapping motifs are considered as the same motif.
When analyzing heterozygous data, please specify -hetero. MARGE expects two annotations per individual next to each other (for allele 1 and allele 2). Therefore, the file needs to have twice as many annotations as individuals and the alleles of the individuals have to be next to each other in the file.

11.2.2 Example of MARGE.pl mutation_analysis

To run a motif mutation analysis between the PU.1 ChIP-seq samples for BALB/cJ and NOD/ShiLtJ from the previous example the peak files first need to be merged and annotated with their tag counts. Then run:

```
MARGE.pl mutation_analysis -file ann_merge_BALB_NOD_PU1.txt -ind balbcj, nodshiltj
```

This will create two plots (output_mut_with_motif_density.pdf and output_mut_with_motif.pdf). It is recommended to change the output name for the plots, to not overwrite your plots by accident. Therefore, a better command is:

```
MARGE.pl mutation_analysis -file ann_merge_BALB_NOD_PU.1.txt -ind balbcj, nodshiltj
-plot plot_BALB_NOD_PU1
```

Now the plots are called plot_BALB_NOD_PU1_with_motif_density.pdf and plot_BALB_NOD_PU1_with_motif.pdf
To run an all versus all comparison of the human data we processed previously in these examples use the output file from MARGE.pl annotate_allele_specific and run:

```
MARGE.pl mutation_analysis -file output_annotate_allele_specific_na06985_na06986.txt
-ind na06985, na06986 -hetero -GLMM_output GLMM_NA06985_NA06986.txt
```

This command will run a linear mixed model to model and score the relationship between the binding of the transcription factor and the existence of the motifs and create an output file called GLMM_NA06985_NA06986.txt.

11.3 MARGE.pl summary_heatmap

To create a heatmap to summarize several different MARGE analysis runs, MARGE offers a script to generate a heatmap.

11.3.1 Usage of MARGE.pl summary_heatmap

To see usage

```
MARGE.pl summary_heatmap
```

```
-files <comma separated list of files>
-names <comma separated list of names>
-sig: only significant motifs (threshold p-value 0.001)
-threshold <threshold>: threshold for significant (default: 0.001)
-output <name>: output name for R file and pdf (default summary_heatmap
    .R)
-title <title>: Title of the heatmap plot (default: output name for R
    file)
-method <pairwise|all>: Method that was used to generate output files (
    default pairwise)
```

This script only works with 2 or more files, which can be specified in -files. Please note, you have to specify the R output files, not the pdfs. The files have the same name as the plots (you can specify with -plot), but end with .R instead of .pdf. If -names is not set, the file names are used to label the heatmap. For long filenames we recommend to set -names. With -sig only significant motifs are considered. The threshold is set to a p-value < 0.001 and can be changed with -threshold.
It is important to specify -method all when a heatmap for the GLMM output should be generated, otherwise, MARGE will throw errors.

11.3.2 Example of MARGE.pl summary_heatmap

After running a pairwise MARGE mutation_analysis for C57BL/6J versus BALB/cJ, C57BL/6J versus NOD/ShiLtJ and BALB/cJ versus NOD/ShiLtJ we can summarize this with:

```
MARGE.pl summary_heatmap -files plot_C57_BALB_PU1_with_motif.R, plot_C57_NOD_PU1_with_motif.R,
plot_BALB_NOD_PU1_with_motif.R -names C57_BALB, C57_NOD, BALB_NOD -sig -output summary_heatmap_strains_PU1.pdf
```

This command will generate a heatmap called summary_heatmap_strains_PU1.pdf with all significant motifs in all pairwise comparisons between BALB/cJ, C57BL/6J, and NOD/ShiLtJ.
To generate a summary heatmap for the GLMM analysis of the human data, we also need two files. In this case we summarize the output from the GLMM analysis using -measure Motif and -measure Motif_score:

```
MARGE.pl summary_heatmap -files GLMM_NA06985_NA06986.txt, GLMM_NA06985_NA06986_motif_score.txt
-names GLMM_motif, GLMM_motif_score -sig -method all
```

This command will create an output file called summary_heatmap.pdf with all significant motifs from the GLMM analysis based on the motif existence or the motif score.

12 MARGE paper analysis

The MARGE paper is not completed yet. Once the paper is published, we will publish another pdf document with all commands and data sets to recreate the complete MARGE paper analysis. We will provide a link here as soon as the file is available.

References

[1] Inc.) 2017 Continuum Analytics, Inc. (dba Anaconda. Anaconda 3.0. `http://anaconda.com`.

[2] Dobin A, Davis CA, Schlesinger F, Drenkow J, Zaleski C, Jha S, Batut P, Chaisson M, and Gingeras TR. Star: ultrafast universal rna-seq aligner. *Bioinformatics*, 2013.

[3] Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015.

[4] Oliver Bembom. *seqLogo: Sequence logos for DNA sequence alignments*, 2016. R package version 1.38.0.

[5] B. Booth. Perl module set::intervaltree. `http://search.cpan.org/~benbooth/Set-IntervalTree/`, 2014.

[6] The 1000 Genomes Project Consortium. A global reference for human genetic variation. *Nature*, 526:68–74, 2015.

[7] Gosselin D, Link VM, Romanoski CE, Fonseca GJ, Eichenfield DZ, Spann NJ, Stender JD, Chun HB, Garner H, Geissmann F, and Glass CK. Environment drives selection and function of enhancers controlling tissue-specific macrophage identities. *Cell*, 159:1327–40, 2014.

[8] Petr Danecek, Adam Auton, Goncalo Abecasis, Cornelis A. Albers, Eric Banks, Mark A. DePristo, Robert E. Handsaker, Gerton Lunter, Gabor T. Marth, Stephen T. Sherry, Gilean McVean, Richard Durbin, and 1000 Genomes Project Analysis Group. The variant call format and vcftools. *Bioinformatics*, 27:2156—-2158, 2011.

[9] S. Heinz, C. E. Romanoski, C. Benner, K. A. Allison, M. U. Kaikkonen, L. D. Orozco, and C. K. Glass. Effect of natural genetic variation on enhancer selection and function. *Nature*, 503:487—-492, 2013.

[10] W. J. Kent A. S. Zweig G. Barber A. S. Hinrichs D. Karolchik. Bigwig and bigbed: enabling browsing of large distributed datasets. *Bioinformatics*, 26:2204–2207, 2010.

[11] Thomas M. Keane, Leo Goodstadt, Petr Danecek, Michael A. White, Kim Wong, Binnaz Yalcin, Andreas Heger, Avigail Agam, Guy Slater, Martin Goodson, Nicholas A. Furlotte, Eleazar Eskin, Christoffer Nellåker, Helen Whitley, James Cleak, Deborah Janowitz, Polinka Hernandez-Pliego, Andrew Edwards, T. Grant Belgard, Peter L. Oliver, Rebecca E. McIntyre, Amarjit Bhomra, Jérôme Nicod, Xiangchao Gan, Wei Yuan, Louise van der Weyden, Charles A. Steward, Sendu Bala, Jim Stalker, Richard Mott, Richard Durbin, Ian J. Jackson, Anne Czechanski, José Afonso Guerra-Assunçao, Leah Rae Donahue, Laura G. Reinholdt, Bret A. Payseur, Chris P. Ponting, Ewan Birney, Jonathan Flint, and David J. Adams. Mouse genomic variation and its effect on phenotypes and gene regulation. *Nature*, 477:289–294, 2011.

[12] W. James Kent, Charles W. Sugnet, Terrence S. Furey, Krishna M. Roskin, Tom H. Pringle, Alan M. Zahler, and David Haussler. The human genome browser at ucsc. *Genome Research*, 2002.

[13] Ben Langmead, Cole Trapnell, Mihai Pop, and Steven L Salzberg. Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome Biology*, 10, 2009.

[14] Rasko Leinonen, Hideaki Sugawara, , and Martin Shumway on behalf of the International Nucleotide Sequence Database Collaboration. The sequence read archive. *Nucleic Acids Research*, 2011.

[15] M. Koderer M. Sanford. Perl module sys::cpu. `http://search.cpan.org/~mzsanford/Sys-CPU/CPU.pm`, 2013.

[16] P. Miller. Perl module statistics::basic. `http://search.cpan.org/~jettero/Statistics-Basic-1.6611/`, 2014.

[17] Paul Murrell. *gridBase: Integration of base and grid graphics*, 2014. R package version 0.4-7.

[18] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0.

[19] Heinz S1, Benner C, Spann N, Bertolino E, Lin YC, Laslo P, Cheng JX, Murre C, Singh H, and Glass CK. Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and b cell identities. *Molecular Cell*, 38:576–89, 2010.

[20] Gregory R. Warnes, Ben Bolker, Lodewijk Bonebakker, Robert Gentleman, Wolfgang Huber Andy Liaw, Thomas Lumley, Martin Maechler, Arni Magnusson, Steffen Moeller, Marc Schwartz, and Bill Venables. *gplots: Various R Programming Tools for Plotting Data*, 2016. R package version 3.0.1.

[21] S. M. Waszak, O. Delaneau, A. R. Gschwind, H. Kilpinen, S. K. Raghav, R. M. Witwicki, A. Orioli, M. Wiederkehr, N. I. Panousis, A. Yurovsky, L. Romano-Palumbo, A. Planchon, D. Bielser, I. Padioleau, G. Udin, S. Thurnheer, D. Hacker, N. Hernandez, A. Reymond, B. Deplancke, and E. T. Dermitzakis. Population variation and genetic control of modular chromatin architecture in humans. *Cell*, 162:1039–50, 2015.