BEST PRACTICES

Imprinting Detection

The workflow for imprinting detection is created for identification of imprinted SNPs in multiple samples. In the R MAGE package, analysis on count files created with the MAGE pre-processing pipeline is performed. It uses count files per chromosome and sample files, also created in the pre-processing, as input.

The MAGE package can be downloaded from GitHub (MAGE_0.1.0.tar.gz on https://github.com/Biobix/MAGE) and installed as:
install.packages("MAGE_0.1.0.tar.gz", repos = NULL, type="source")

The analysis outlined here for imprinting detection can be performed using the R script MAGE_imprinting.R

Input

Count files per chromosome from the pre-processing workflow and files with samples ids, also obtained from the pre-processing workflow, are used as input. The count files contain allelic counts per SNP and sample and list the chromosome, position, standard alleles, type of variation, dbSNP id, gene name, A/T/C/G count and sample name. Sample files list sample names and their sample number. In the R scripts, a working directory where the count files are located has to be specified, as well a working directory where the sample file is located. The working directory for the results files also has to be provided. Lastly, the file name of the counts files and the samples file has to be given. Afterwards, the count files are read and a hash with a dataframe listing count data per position is created in R.

Prior filtering

Functions: standard_alleles, prior_filter

To ensure detection of a robust set of imprinted SNPs, prior filtering of SNPs is recommended. Firstly, all insertions and deletions are removed. As later in the pipeline genotyping is done using two reference alleles, only two allelic counts based on the dbSNP standard alleles are retained for each position, implemented in the “standard_alleles” function. For each position, a dataframe containing at least columns of allelic counts ("A", "T", "C", "G") and the dbSNP reference alleles ("ref_alleles") should be present. If no dbSNP reference alleles are available, "A/T/C/G" can be used as standard alleles.

The function “prior_filter” can be used to filter loci on median coverage, number of samples and prior estimate of the allele frequency. Also in this step, samples without reference or variant counts are filtered and a Bayes approach filters erroneous samples. The dataframe resulting from the standard_alleles function can be used as input.

Estimate sequencing error rate, inbreeding and allele frequency

Functions: estimate_parameters

As estimates of sequencing error rates, inbreeding and allele frequencies are necessary in the model for imprinting, an expectation-maximisation (EM) algorithm is used for parameter estimation. The function “estimate_paramaters” estimates allele frequency and sequencing error rate per SNP from the reference and variant allelic counts with an EM approach. Also inbreeding coefficients per position are calculated from the observed (i.e. the heterozygous probabilities) and expected (calculated using the estimated allele frequencies) number of heterozygous samples. The output of “estimate_parameters” is a list with the allele frequency, sequencing error rate, inbreeding coefficient, called genotypes per samples, genotype probabilities per sample and the number of EM iterations per SNP. SNPs can be filtered on allele frequency and position-specific sequencing error rate. Afterwards, the median inbreeding and sequencing error rate over the retained positions can be calculated.

Symmetry goodness-of-fit

Functions: symmetry_gof

Only high quality SNPs are interesting for analysis and hence loci are filtered on goodness-of-fit. Based on reference and variant counts and allele frequencies, a chi-squared test for symmetry of each SNP is performed with the function “symmetry_gof”. The function returns a p-value of the chi-squared test and only SNPs with a non-significant p-value are retained for further analysis.

Imprinting analysis

Functions: lrt_i, median_imprinting

For each SNP, imprinting detection can be done with the function “lrt_i”. Reference and variant counts, allele frequency, sequencing error rate and inbreeding coefficient are necessary as input. Maximum likelihood estimation for determining the degree of imprinting and a likelihood ratio test for detection of significant imprinting is performed. Also the goodness-of-fit of the model to data is calculated. The function returns a list with the estimated degree of imprinting, the test statistic of the likelihood ratio test, the p-value of the likelihood ratio test and the goodness-of-fit statistic.

To only retain a robust set of imprinted SNP, the median degree of imprinting is afterwards calculated with the function “median_imprinting” based on reference and variant counts, allele frequency and inbreeding for each position. A results dataframe with a position per row and all interesting parameters is then created. At least the columns "position", "gene", "p”, "estimated.i", "allele.frequency", "dbSNP", "est_SE", "GOF", "med_impr", "est_inbreeding" are necessary.

Retain and plot imprinted SNPs

Functions: final_filter, plot_imprinting

Lastly, only the significantly imprinted SNPs have to be retained from all results. This can be done with the function “final_filter”. The function needs a dataframe with imprinting results and a hash with count data per position. Also various filter values, a results working directory and which files should be created have to be provided. Text files with all results and/or significant results are created. For the latter SNPs are filtered on goodness-of-fit, degree of (median) imprinting and adjusted p-value. Also count files of standard and variant allele per samples and per position for all or all significant SNPs are created.

Imprinted SNPs are afterwards plotted with the function “plot_imprinting” based on reference and variant counts, allele frequency, degree of imprinting, sequencing error rate and degree of inbreeding.

Detection of differential imprinting

The workflow for detection of differential imprinting is created for identification of imprinted SNPs deregulated in multiple case samples. In the R MAGE package, analysis on count files created with the MAGE pre-processing pipeline is performed. It uses count files per chromosome and sample files, also created in the pre-processing, as input.

This workflow is performed after the detection of imprinting described previously. When a set of imprinted SNPs is detected, case data is analysed in comparison to the control data. The imprinting results files (from MAGE_imprinting.R) are here necessary as input in the R script (MAGE_LOI.R).

The MAGE package can be downloaded from GitHub (MAGE_0.1.0.tar.gz on https://github.com/Biobix/MAGE) and installed as:
install.packages("MAGE_0.1.0.tar.gz", repos = NULL, type="source")

Input

Count files per chromosome from the pre-processing workflow and files with sample ids, also obtained from the pre-processing workflow, are used as input for case data. The count files contain allelic counts per SNP and sample and list the chromosome, position, standard alleles, type of variation, dbSNP id, gene name, A/T/C/G count and sample name. Sample files list sample names and their sample number. In R, a case hash with a dataframe listing count data per position is created. For control data, the results dataframe and control hash created in the imprinting detection workflow are necessary for further analysis.

Working directories where the sample files are located have to be specified, as well working directories where the case count files and control imprinting files are located. The working directory for the results files also has to be provided

Filtering of case data

Functions: standard_alleles

When a set of imprinted SNPs is detected in control data, the same positions can be analysed in case data. The count files from the pre-processing workflow are read and a hash with all positions is created. Again, only two alleles per position are retained with the function “standard_alleles”. As we are only interested in differential imprinting of SNPs imprinted in control data, only those positions are analysed in case data. Afterwards, the reference and variant allele are set equal to the control reference and variant allele.

Detection of differential imprinting

Functions: binomial_logistic, combine_p_gene, plot_histo

When reference and variant allelic counts are determined for control and case data, differential imprinting can be identified using the function “binomial_logistic”. It uses reference and variant counts from the control data and case data to create a logistic regression indicating whether or not the SNP is featured by differential imprinting (significant p-value indicates differential imprinting). P-values of differential imprinting analyses can be combined per gene using the function “combine_p_gene”. Histograms of control and case data are made with the function “plot_histo” using reference and variant counts and a results working directory.

Detection of imprinting and differential imprinting

The workflow for detection of (differential) imprinting is created for identification of imprinted SNPs in multiple control samples and their deregulation in multiple case samples. In the R MAGE package, analysis on count files created with the MAGE pre-processing pipeline is performed. It uses count files per chromosome and sample files, also created in the pre-processing, as input.

The first part of this workflow is exactly the same as the workflow for imprinting detection. When a set of imprinted SNPs is detected, case data is analysed in comparison to the control data, outlined in MAGE_imprinting_LOI.R.

The MAGE package can be downloaded from GitHub (MAGE_0.1.0.tar.gz on https://github.com/Biobix/MAGE) and installed as:
install.packages("MAGE_0.1.0.tar.gz", repos = NULL, type="source")

Input

Count files per chromosome from the pre-processing workflow and files with samples ids, also obtained from the pre-processing workflow, are used as input for control data as well as case data. The count files contain allelic counts per SNP and sample and list the chromosome, position, standard alleles, type of variation, dbSNP id, gene name, A/T/C/G count and sample name. Sample files list sample names and their sample number. In R, a control and case hash with a dataframe listing count data per position is created.

For the analysis, working directories where the count files for control and case samples are located have to be specified, as well a working directories where the sample files are located. The working directory for the results files also has to be provided.

Prior filtering

Functions: standard_alleles, prior_filter

To ensure detection of a robust set of imprinted SNPs in control samples, prior filtering of SNPs is recommended. Firstly, all insertions and deletions are removed. As later in the pipeline genotyping is done using two reference alleles, only two allelic counts based on the dbSNP standard alleles are retained for each position, implemented in the “standard_alleles” function. For each position, a dataframe containing at least columns of allelic counts ("A", "T", "C", "G") and the dbSNP reference alleles ("ref_alleles") should be present. If no dbSNP reference alleles are available, "A/T/C/G" can be used as standard alleles.

The function “prior_filter” can be used to filter loci on median coverage, number of samples and prior estimate of the allele frequency. Also in this step, samples without reference or variant counts are filtered and a Bayes approach filters erroneous samples. The dataframe resulting from the standard_alleles function can be used as input.

Estimate sequencing error rate, inbreeding and allele frequency

Functions: estimate_parameters

As estimates of sequencing error rates, inbreeding and allele frequencies are necessary in the model for imprinting, an expectation-maximisation (EM) algorithm is used for parameter estimation. The function “estimate_paramaters” estimates allele frequency and sequencing error rate per SNP from the reference and variant allelic counts with an EM approach. Also inbreeding coefficients per position are calculated from the observed (i.e. the heterozygous probabilities) and expected (calculated using the estimated allele frequencies) number of heterozygous samples. The output of “estimate_parameters” is a list with the allele frequency, sequencing error rate, inbreeding coefficient, called genotypes per samples, genotype probabilities per sample and the number of EM iterations per SNP. SNPs can be filtered on allele frequency and position-specific sequencing error rate. Afterwards, the median inbreeding and sequencing error rate over the retained positions can be calculated.

Symmetry goodness-of-fit

Functions: symmetry_gof

Only high quality SNPs are interesting for analysis and hence loci are filtered on goodness-of-fit. Based on reference and variant counts and allele frequencies, a chi-squared test for symmetry of each SNP is performed with the function “symmetry_gof”. The function returns a p-value of the chi-squared test and only SNPs with a non-significant p-value are retained for further analysis.

Imprinting analysis

Functions: lrt_i, median_imprinting

For each SNP, imprinting detection can be done with the function “lrt_i”. Reference and variant counts, allele frequency, sequencing error rate and inbreeding coefficient are necessary as input. Maximum likelihood estimation for determining the degree of imprinting and a likelihood ratio test for detection of significant imprinting is performed. Also the goodness-of-fit of the model to data is calculated. The function returns a list with the estimated degree of imprinting, the test statistic of the likelihood ratio test, the p-value of the likelihood ratio test and the goodness-of-fit statistic.

To only retain a robust set of imprinted SNP, the median degree of imprinting is afterwards calculated with the function “median_imprinting” based on reference and variant counts, allele frequency and inbreeding for each position. A results dataframe with a position per row and all interesting parameters is then created. At least the columns "position", "gene", "p”, "estimated.i", "allele.frequency", "dbSNP", "est_SE", "GOF", "med_impr", "est_inbreeding" are necessary.

Retain and plot imprinted SNPs

Functions: final_filter, plot_imprinting

Lastly, only the significantly imprinted SNPs have to be retained from all results. This can be done with the function “final_filter”. The function needs a dataframe with imprinting results and a hash with count data per position. Also various filter values, a results working directory and which files should be created have to be provided. Text files with all results and/or significant results are created. For the latter SNPs are filtered on goodness-of-fit, degree of (median) imprinting and adjusted p-value. Also count files of standard and variant allele per samples and per position for all or all significant SNPs are created.

Imprinted SNPs are afterwards plotted with the function “plot_imprinting” based on reference and variants counts, allele frequency, degree of imprinting, sequencing error rate and degree of inbreeding.

Filtering of case data

Functions: standard_alleles

When a set of imprinted SNPs is detected in control data, the same positions can be analysed in case data. The count files from the pre-processing workflow are read and a hash with all positions is created. Again, only two alleles per position are retained with the function “standard_alleles”. As we are only interested in differential imprinting of SNPs imprinted in control data, only those positions are analysed in case data. Afterwards, the reference and variant allele are set equal to the control reference and variant allele.

Detection of differential imprinting

Functions: binomial_logistic, combine_p_gene, plot_histo

When reference and variant allelic counts are determined for control and case data, differential imprinting can be identified using the function “binomial_logistic”. It uses reference and variant counts from the control data and case data to create a logistic regression indicating whether or not the SNP is featured by differential imprinting (significant p-value indicates differential imprinting). P-values of differential imprinting analyses can be combined per gene using the function “combine_p_gene”. Histograms of control and case data are made with the function “plot_histo” using reference and variant counts and a results working directory.