Identification of Allele-Specific Expression - the ISoLDE package

Christelle Reynès[1,2]*, Marine Rohmer[3], Guilhem Kister[1,2], Tristan Bouschet[1], Annie Varrault[1], Emeric Dubois[3], Stéphanie Rialle[3], Laurent Journot[1] and Robert Sabatier[1,2]

[1] Institute of Functional Genomics, CNRS UMR5203, INSERM U661, University of Montpellier, Montpellier, France.
[2] Laboratory of Biostatistics, Informatics and Pharmaceutical Physics, UFR Pharmacy, University of Montpellier, Montpellier, France.
[3] Montpellier GenomiX facility, UMS 3426 BioCampus, Montpellier, France
*Christelle.Reynes (at) igf.cnrs.fr

March 29, 2016

Abstract

The allele-specific expression of genes is typically questioned through RNA sequencing analyses. However, statistical modelling of data is still under question. This BioConductor package proposes a novel statistical method designed to test for the parent or strain specific gene expression in the context of reciprocal crosses. This method, called ISoLDE for Integrative Statistics of alleLe Dependent Expression, is a robust non-parametric test based on a novel criterion whose distribution is directly learnt from the data through resampling. The main option is to use bootstrap resampling to estimate criterion distribution. Alternatively, for datasets with only two replicates in each cross, empirical thresholds are applied to the criterion. This vignette introduces a typical workflow with ISoLDE and details the main theoretical aspects of the method.

ISoLDE **version**: 0.99.0

Contents

1 List of terms used in this vignette

ASE	Allele Specific Expression.
ASR	Allele Specific Read.
raw count	An ASR count obtained such as described in the 3.2.1 section.
normalized count	An ASR count obtained such as described in the 3.2.1 section and then normalized.
0 count	A value of 0 in an ASR count data file or in a dataframe.

2 Introduction: what ISoLDE does

2.1 Background

In diploid cells, genetic and epigenetic factors influence the relative expression levels of the two alleles of a gene. The preferred allele may depend on the chromosome parental origin as for imprinted genes, but other imbalances such as strain bias may occur. To study allele specific expression (ASE), RNA-seq has become the standard technology but how to statistically analyze those data is still debated.

ISoLDE is a new non-parametric statistical method for identifying genes with allele-specific expression. ISoLDE is dedicated to stranded RNA-seq experiment on hybrid samples resulting from reciprocal parental crosses. ISoLDE has the new and useful advantage of statistically identifying both biased and unbiased genes allowing some genes to be undetermined (see [Reynès *et al.*(2016)] for more information). It aims at freeing itself from approximate modelling by the use of non parametric statistics whose distribution is directly learnt from the data through resampling. To this goal, a specific criterion was designed to take into account data specificities and make the best of biological replicates information.

ISoLDE identifies parental or strain expression biases. It requires pre-processed data that consist of a matrix of allele-specific read (ASR) counts for every gene. Details on how to obtain such counts are provided in section 3.2.1.
Normalization of data is strongly recommended, what can be achieved by other BioConductor packages such as edgeR or DESeq (ISoLDE does NOT provide data normalization).
ISoLDE yields both graphical and textual outputs, containing lists of parental biased genes or strain biased genes according to what you want to study.

2.2 Workflow overview

The following figure shows the workflow overview of the ISoLDE package.

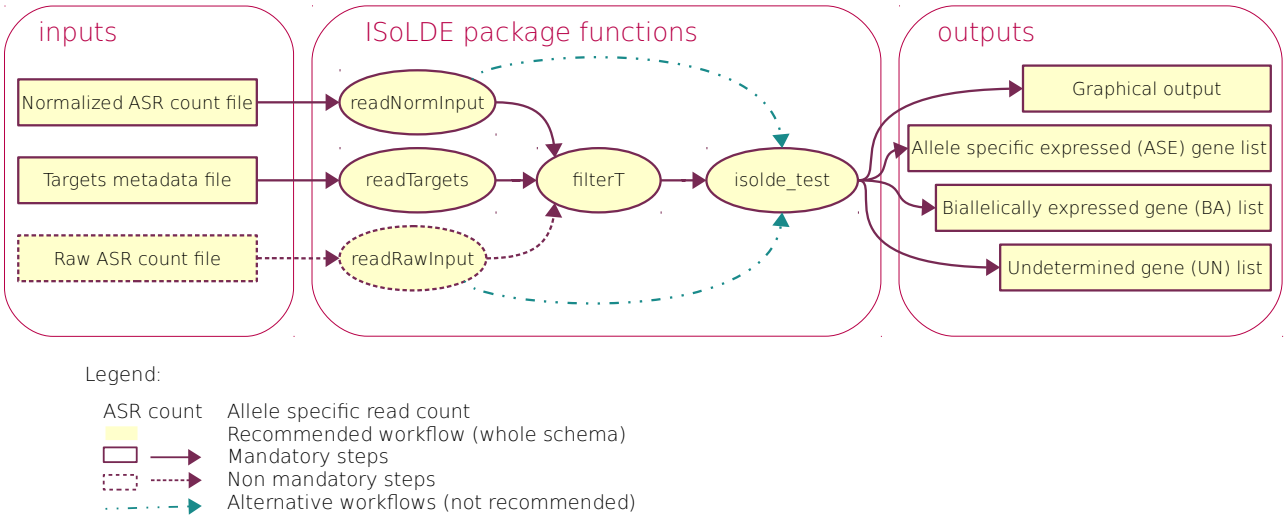

Figure 1: ISoLDE workflows

2.3 The ISoLDE package functions

The `ISoLDE` package includes the five functions described below.
Three functions are available for reading, loading and checking your input data:

- `readRawInput` checks and loads into a dataframe the input file containing raw ASR counts so that it can be input into `filterT`. This is only required to use the `filterT` function and if your normalized data no longer contain 0 counts that are required for the `filterT` function.
- `readNormInput` checks and loads into a dataframe the input file containing normalized ASR counts so that it can be input into `filterT` and `isolde_test`.
- `readTarget` checks and loads into a dataframe your target input file.

Main functions:

- `filterT` filters lowly expressed genes according to a data driven threshold, before any statistical analysis. This step is not mandatory but strongly recommended.
- `isolde_test` performs the statistical test (two possible options according to the number of available biological replicates) and outputs lists of genes according to their ASE status.

3 Standard Workflow

3.1 Quick start

These are the typical steps of application of ISoLDE on filtered and normalized RNAseq data (tab-delimited text files) where reads are assigned according to their parental origin (see section 3.2.1 for more details).

```
> library(ISoLDE)
> rawASRcounts <- readRawInput(raw_file = "my_raw_file.txt")
> normASRcounts <- readNormInput(norm_file = "my_norm_file.txt")
> target <- readTarget(target_file = "my_target_file.txt", asr_counts = rawASRcounts)
> filteredASRcounts <- filterT(rawASRcounts = rawASRcounts, normASRcounts = normASRcounts,
target = target, bias="parental")
> res <- isolde_test(bias = "parental", asr_counts = filteredASRcounts, target = target)
```

For strain origin, the user has to use the `strain` value for argument `bias` in both `filterT` and `isolde_test`.

3.2 Detailed use case of ISoLDE

3.2.1 Preliminary bioinformatics steps

ISoLDE works with pre-processed RNA-seq data obtained from hybrid species resulting from reciprocal crosses of two parental strains.
ISoLDE requires at least two biological replicates for each cross, but the more replicates, the more reliable the results. Moreover, ASE cannot be questioned if RNA sequencing depth is too low and statistical significance is improbable for low expressed transcripts.

Typical preliminary bioinformatics steps are:

- Building a hybrid reference sequence set with IUPAC ambiguous codes at SNP locations. Tools like `Novoalign` (Novocraft, http://www.novocraft.com/main/index.php), with `Novoutil` IUPAC, can be used in this scope.
- Alignment of reads from the RNA-sequencing experiment on this hybrid reference sequence set.
- Counting of the different bases at SNP positions according to their allelic origin (e.g. using `SAMtools/mpileup`).
- Annotation of SNPs to sum allele-specific sense read bases across a gene or a transcript (e.g. using `Annovar`).

Important: The parental origin of chromosome X genes can be assessed only with female samples. To avoid genes that are not meaningful to test with ISoLDE, remove the chromosome X genes from your resulting data files if your samples are not all females.

More details on preliminary steps are available in [Babak (2012)].

In this vignette, "raw counts" is used to name allele-specific sense read bases. The count file thus obtained is called raw count file or raw ASR count file. For example, for three biological replicates in each cross, the raw count file must contain 12 columns: three replicates $\times$ two reciprocal crosses $\times$ two allelic origins (parental or strain). An example of such a design is shown in section 3.2.2.

We strongly recommend use normalized counts before performing any statistical test. In our example, counts have been normalized with the edgeR BioConductor package using the RLE normalization factor.

3.2.2 Input data

The first mandatory input is an ASR counts file, depending on whether you intend to filter your ASR counts or not, and whether you work on normalized data or not. As well as normalizing, we strongly encourage to filter your data before statistical analysis. The raw count file is only used in the filterT function to locate genes (or transcripts) having 0 counts in at least one column. If your normalized data still contain 0 counts, then the raw count file is not necessary.

- **Case 1 (recommended): with the ISoLDE filtering step on normalized data:** The filterT function uses raw data to determine a threshold for filtering and then applies the filter on your normalized data. In this case, input data are both raw and normalized ASR count files.
- **Case 2 (not recommended): with the filtering step on raw data:** Input data is only a raw ASR count file. Warning: in this case, you work with non normalized data.
- **Case 3 (not recommended): without the ISoLDE filtering step:** Input data is an ASR count file which has been either filtered by yourself or not, and either normalized or not.

Each ASR count file must have one line per feature (gene or transcript).
Each ASR count file must have two columns per biological sample (one for each allelic origin, such as described in section 3.2.1). For example, for three biological replicates in each cross, the raw count file must contain 12 columns: three replicates × two reciprocal crosses × two allelic origins (parental or strain). Columns are delimited with a character (e.g. tabulation).
While having row names (gene or transcript names) is quite obvious, column names are not mandatory.
An example of normalized ASR count file obtained after loading is shown later.

Target file

The other mandatory input is a metadata file that we call "target file". It describes the experiment design and each column of the ASR count file contents.

It consists of three delimited columns describing your input data (raw and / or normalized ASR count file(s) with the same structure if both are provided). Each line of the target file corresponds to a column of the ASR count file. **Lines of target file MUST be in the same order as the columns in the input data.**
The first line corresponds to the column names: sample, parent and strain.
Then, each line contains the three corresponding values, separated by a tabulation or any character.

Details of the three columns:
sample: the sample (biological replicate) name. Two lines per sample name are expected (one for

the maternal origin and one for the paternal origin).
parent: the parental origin of the ASR count. Two possible values: maternal or paternal.
strain: the strain origin of the ASR count. Exactly two different values are expected in the whole column.

Note: spaces and the ":" character are forbidden in the sample and strain columns.

Here is the target file of our example. As you can see, the same sample name appears twice, once for the maternal origin and once for the paternal origin. Do not use different names for the same biological sample.

```
sample parent strain
sample1 maternal BL/6
sample2 maternal BL/6
sample3 maternal BL/6
sample4 maternal BL/6
sample5 maternal JF1
sample6 maternal JF1
sample7 maternal JF1
sample1 paternal JF1
sample2 paternal JF1
sample3 paternal JF1
sample4 paternal JF1
sample5 paternal BL/6
sample6 paternal BL/6
sample7 paternal BL/6
```

Reading data

ISoLDE proposes its own functions to load your input data as a data.frame. Each function includes some specific checks according to ISoLDE requirements (hence their use is recommended).

Assuming the raw ASR count file is called "my_raw_file.txt", the normalized ASR count file "my_norm_file.txt" and the target file "my_target_file.txt", reading input files simply consists of:

```
> rawASRcounts <- readRawInput(raw = "my_raw_file.txt")
> normASRcounts <- readNormInput(norm = "my_norm_file.txt")
> target <- readTarget(target_file = "my_target_file.txt", asr_counts = rawASRcounts)
```

Three data frames are obtained. The structure of the normASRcounts data frame of our example is given below:

```
> head(normASRcounts)
       sample1      sample2     sample3     sample4    sample5      sample6
gene_1 299.7457552 219.3375221 244.5973016  238.414208 171.1630330  257.307921
gene_2  20.4372106  24.5894083  16.8346321    9.691634  12.3617746   16.321771
gene_3   0.9732005   0.9835763   0.9902725    0.000000   1.9018115    0.000000
gene_4 108.9984564  87.5382936  86.1537054   92.070528 113.1577829  139.215107
gene_5   0.9732005   0.0000000   2.9708174    0.000000   0.9509057    3.840417
gene_6  22.3836116   8.8521870  16.8346321   13.568288  10.4599631   18.241979
        sample7    sample1     sample2      sample3    sample4       sample5
gene_1  213.834348 283.2013467 194.7481138  216.869672 193.8326895   216.80651
gene_2   16.448796  35.0352181  20.6551030   24.756812  18.4141055    21.87083
gene_3    1.935152   0.9732005   0.9835763    0.000000   0.9691634     0.00000
gene_4  150.941893 121.6500630  79.6696829   83.182888 118.2379406    74.17065
gene_5    0.000000   0.0000000   0.0000000    0.000000   0.0000000     0.00000
gene_6    6.773034  17.5176091  11.8029160    6.931907  10.6607979    16.16540
       sample6      sample7
gene_1 265.9488588  265.115889
gene_2  27.8430213   27.092135
gene_3   0.9601042    2.902729
gene_4  95.0503142  117.076725
gene_5   1.9202084    0.000000
gene_6  23.0425004   18.383948
```

Filtering data

Filtering is recommended to avoid considering genes without enough information, and thus to avoid a too strong effect of multiple test correction.
In the `filterT` function, the filter threshold is defined according to the maximum number of counts for genes having at least 66% of replicates as zero counts in data, for each parental (or strain origin).

Thus, the `filterT` method needs either the raw or normalized input file to have genes (or transcripts) having zero values in at least one column. After normalization, zero values are often changed into non integer values. That is why if your normalized file still contains genes having zero values in at least one column, you do not need to provide the raw count file, else both raw and normalized data files should be provided. If you want to analyze raw data, only raw ASR counts can be provided.

Note that in any case a minimal filtering step will always be performed while applying the `isolde_test` function. It consists of eliminating all genes not satisfying these two conditions:
- at least one of the two medians (of paternal or maternal ASR counts) is different from 0;
- there is at least one ASR count (different from 0) in each cross.

The `filterT` function outputs two dataframes: `removedASRcounts` containing genes that did not satisfied the two conditions, and `filteredASRcounts` containing genes that successfully pass the filtering step.

```
> res_filterT <- filterT(rawASRcounts = rawASRcounts, normASRcounts =
normASRcounts, target = target)
> filteredASRcounts <- res_filterT$filteredASRcounts
```

Now we have a dataframe `filteredASRcounts` containing normalized and filtered ASR counts on which to run the statistical test.

3.2.3 ASE analysis

Depending on how many biological replicates are available for both crosses, `ISoLDE` will use the bootstrap or the threshold method. The main and recommended option is to use bootstrap resampling to estimate criterion distribution. Alternatively, for datasets with only two replicates from at least one cross, empirical thresholds are applied to the criterion (see section 4 for details). The default behaviour of the `isolde_test` function is to adapt to the number of replicates per cross: when only two replicates are available for at least one cross, the threshold method is used, if more than two replicates are available for both crosses, the more accurate bootstrap method is applied. When less than two replicates are available in the dataset, `isolde_test` can not be run. The bootstrap method is more robust than the threshold one because it can take into account more information from the replicates, but one may desire to perform the threshold method for comparison purpose. Then, one can set the method parameter to `threshold`. Note that the contrary is not possible (one can not force the bootstrap method if no more than two replicates per cross are available).

Below are two examples of `ISoLDE` use.

Parental bias, bootstrap method:
Here is the code to identify genes with parent-of-origin dependant expression, using the bootstrap method on our example:

```
 > res <- isolde_test(bias = "parental",
asr_counts = filteredASRcounts,
target = target)
```

Strain bias, threshold method:
If you only have two biological replicates in each cross, here is the code to look for strain bias in gene expression with the threshold method:

```
 > res <- isolde_test(bias = "strain",
method = "threshold",
asr_counts = filteredASRcounts,
target = target}
```

3.2.4 Output data

ISoLDE returns R objects and can produce both graphical and textual outputs.

Object output The object output of `isolde_test` consists of three different `data.frame`:

- listASE is a `dataframe` with one row per gene (or transcript) identified as having an allelic bias and five columns:
  - `names` contains gene (or transcript) names such as `asr_counts` row names,
  - `criterion` contains the criterion value (see [Reynès *et al.*(2016)]),
  - `diff_prop` is the criterion numerator which reflects the difference between proportions of either parents or strain origins,
  - `variability` is the criterion denominator which quantifies the gene (or transcript) variability between replicates,
  - `origin` specifies the bias direction either "P" or "M" for parental bias or one of specified strain names for strain bias.
- listBA is a `dataframe` with one row per gene (or transcript) identified as biallelically expressed and four columns corresponding to the first four ones in listASE.
- listUN is a `dataframe` with one row per gene (or transcript) with undetermined status (when `ISoLDE` can not affirm that expression of these genes is biased or biallelic) and six columns. The first five columns are the same as `listASE`, the last one may contain three values:
  - `FLAG_consistency` for genes with no statistical evidence of neither bias nor biallelic expression but whose parental or strain bias is always in the same direction across replicates,
  - `FLAG_significance` for genes with statistical evidence of bias but with discrepancies in bias direction across replicates,
  - `NO_FLAG` for other undetermined genes.

```
> head(res$listASEtot)
name criterion diff_prop variability origin
gene_6196 12.167896 -1.0000000 0.006754124 P
gene_561 11.072286 -0.9979916 0.008124173 P
gene_6174 8.776794 -0.9967999 0.012898647 P
gene_4729 8.582397 -0.9961772 0.013472742 P
gene_2891 8.235402 -1.0000000 0.014744511 P
gene_6959 8.100851 -0.9926529 0.015015284 P
```

Graphical output

A graphical output is generated by default. This graph allows to locate genes according to criterion values. Looking for maternal or paternal expression biases, we obtain:

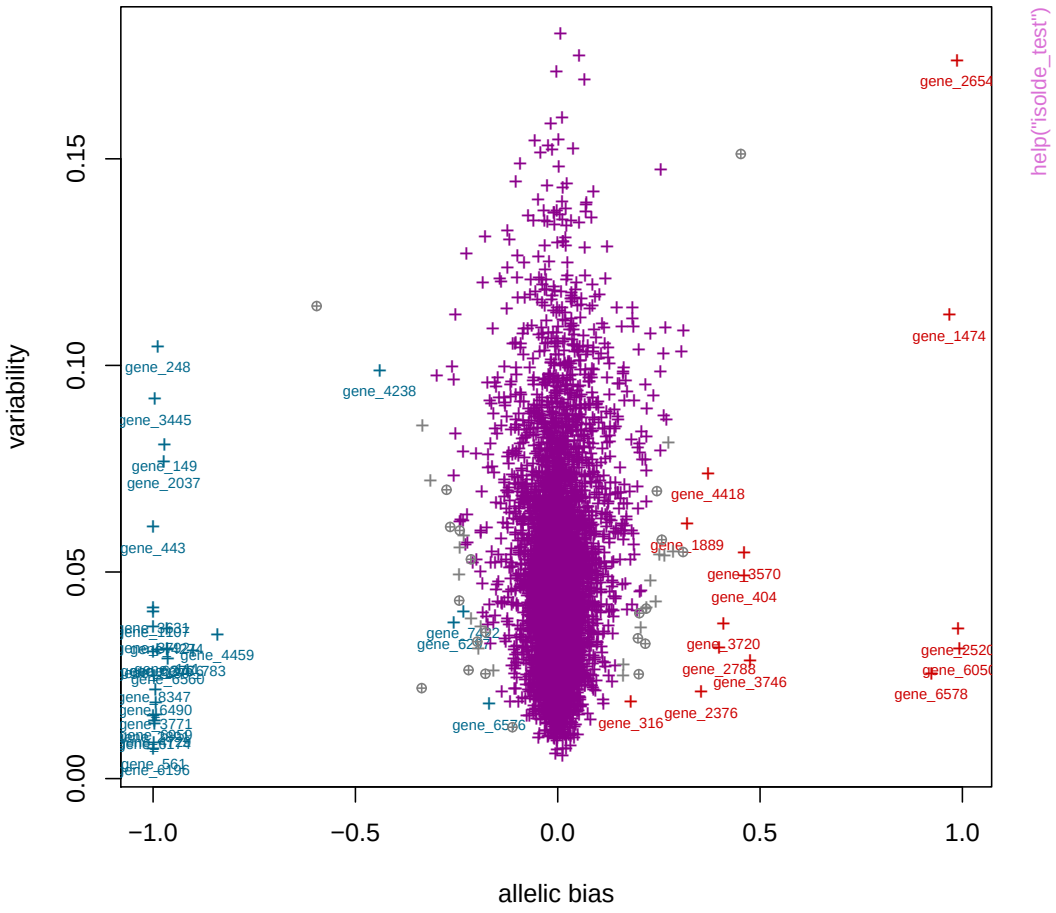

Figure 2: ISoLDE - Graphical output

Legend

+ **purple**	biallelically expressed genes.
+ **blue**	paternally expressed genes.
+ **red**	maternally expressed genes.
+ **grey**	undetermined genes.
⊕ **grey surrounded**	undetermined genes with either consistency or significance flag.

When performing the threshold method, additional grey dashed curves represent the two criterion empirical thresholds used for the analysis.

Textual output When `text` argument is set to `TRUE` (default), three tab-delimited text files are produced:

- the "BA" file contains what is in listBA object (see 3.2.4),
- the "ASE" file contains what is in listASE object (see 3.2.4),
- the "UN" file contains what is in listUN object (see 3.2.4).

4 Theoretical aspects of ISoLDE algorithm

In previous studies, ASE status has been tested through different ways of modelling data. ISoLDE aims at both defining an appropriate criterion taking into account the data specificities and better taking into account replicates.

4.1 Criterion choice

The so far used methods to identify genes with ASE rely on classical statistical methods such as z-test or chi-square test with a global use of replicates whose reads are often summed before applying the chosen test (see for example [Babak (2012), DeVeale *et al.*(2012)]). In our method, the goal was to adapt usual statistics to the data specificities. In particular, in usual z-test, the denominator accounts for the samples variability but based on a binomial behaviour which underestimates RNAseq data variability. Moreover, as few replicates are most of time available, the use of classical variance is inappropriate. Hence, we chose the MAD (Median Absolute Deviation, [Hampel (1974)]) to quantify samples variability. Finally, as sequencing depth has to be taken into account (the results concerning a gene having many reads are more reliable than those concerning few reads), the MAD has been divided by the median number of reads in the sample. Thus, variability estimation is a robust version of coefficient of variation using MAD instead of standard deviation and median instead of mean.
The next section will now focus on how to put thresholds on this criterion. The aim is to be able to define both genes with ASE and biallelically expressed genes and to keep the possibility that some genes neither BA nor allele dependent expressed remain undetermined.

4.2 Threshold definition

To define thresholds, two situations are considered: either there are more than two biological replicates in both reciprocal crosses or not.

4.2.1 Situation 1: more than two biological replicates in both crosses

When enough information is available, the method aims at taking advantage of it by using bootstrap resampling. For each gene and each biological replicate, the total number of reads is divided up between maternal and paternal origin according to the current question:

- when genes with ASE are being identified, in order to generate the null hypothesis distribution of the criterion, the reads are *equally* allocated to maternal and paternal origins (or strain origins) using the same distribution of proportions as what is observed between replicates within one cross. Indeed, within a given cross, differences between maternal and paternal (or strain) origins are expected to only account for biological noise.

- when biallelically expressed genes are being identified, in order to generate the null hypothesis distribution of the criterion, a bias ratio is randomly chosen and the reads are distributed according to those proportions.

In both cases, the resampling is performed many times (the default value is 5000). Then, the distributions obtained for each gene are used to compute empirical p-values which are corrected using usual Benjamini-Hochberg FDR correction for multiple tests [Benjamini *et al.*(1995)].

4.2.2 Situation 2: only two biological replicates in at least one cross

In this situation, there is too few information to obtain reliable distributions under null hypothesis from resampling. Predefined thresholds will be chosen and applied. This choice is based on a consensus of ten different datasets including two or more replicates for each cross.
The ten datasets are the following ones:

- five datasets obtained in our labs including the two *in vivo* datasets used and detailed in [Reynès *et al.*(2016)] and [Bouschet *et al.*(2016)] and three *in vitro* experiments containing only two biological replicates;
- Hasin-Brumshtein's data from [Hasin-Brumshtein *et al.*(2014)] studying two replicates experiments on the reciprocal crosses of C57BL/6J with DBA/2J and concerning mouse adipose tissue;
- Babak's data from [Babak *et al.* (2008)] containing four replicates of E9.5 mouse embryos of reciprocal crosses of CAST/Eij and C57BL/6J;
- three datasets from Lorenc *et al.* [Lorenc *et al.*(2014)] concerning three mouse tissues (vomeronasal organ, hypothalamus and liver) obtained by reciprocally crossing WSB and PWD strains with three to six biological replicates for each cross.

See [Reynès *et al.*(2016)] for more details.

References

[Babak *et al.* (2008)] Babak T., DeVeale B., Armour C., Raymond C., Cleary M., Van Der Kooy D., Johnson J. & Lim L. P. **Global survey of genomic imprinting by transcriptome sequencing.** *Current biology, vol. 18-22, 1735–1741, Elsevier.*

[Babak (2012)] Babak T. **Identification of imprinted Loci by transcriptome sequencing.** *Genomic Imprinting, 79–88, Springer.*

[Benjamini *et al.*(1995)] Benjamini Y. & Hochberg Y. (1995) **Controlling the false discovery rate: a practical and powerful approach to multiple testing.** *Journal of the Royal Statistical Society. Series B (Methodological), JSTOR, 289–300.*

[Bouschet *et al.*(2016)] Bouschet T., Dubois E., Reynès C., Kota S. K., Rialle S., Maupetit-Méhouas S., Pezet M., Le Digarder A., Nidelet S., Demolombe V., Cavelier P., Meusnier C., Maurizy C., Sabatier R., Feil R., Arnaud P., Journot L. and Varrault A. (2016) **In vitro corticogenesis from embryonic stem cells recapitulates the in vivo epigenetic control of imprinted gene expression.** *Under submission.*

[DeVeale *et al.*(2012)] DeVeale B., Van Der Kooy D., & Babak T. **Critical evaluation of imprinted gene expression by RNA–Seq: a new perspective.** *PLoS genetics, vol 8 - 3, e1002600, Public Library of Science.*

[Hampel (1974)] Hampel F. R. **The influence curve and its role in robust estimation.** *Journal of the American Statistical Association, vol 69 - 346, 383–393, Taylor & Francis.*

[Hasin-Brumshtein *et al.*(2014)] Hasin-Brumshtein Y., Hormozdiari F., Martin L., Van Nas A., Eskin E., Lusis A. J, Drake T. A. **Allele-specific expression and eQTL analysis in mouse adipose tissue** *BMC genomics, vol 15-1-471, BioMed Central Ltd*

[Lorenc *et al.*(2014)] Lorenc A., Linnenbrink M., Montero I, Schilhabel M. B. & Tautz D. **Genetic differentiation of hypothalamus parentally biased transcripts in populations of the house mouse implicate the Prader-Willi syndrome imprinted region as a possible source of behavioral divergence.** *Molecular biology and evolution, SMBE, msu257.*

[Reynès *et al.*(2016)] Reynès C., Kister G., Rohmer M., Bouschet T., Varrault A., Dubois E., Rialle S., Journot L. & Sabatier R. (2016) **ISoLDE: a new method for identification of allelic imbalance.** *Under submission.*