User Manual for BayesIso (V1.2) Demo (MATLAB)

Introduction

Bayesian identification of differentially expressed isoforms using a novel joint model (BayesIso) is a computational approach for the identification of differentially expressed isoforms through a sampling procedure. As a joint model, BayesIso, with differential state of the isoforms as model parameters, is developed to account for the intrinsic complicate variability of RNA-seq data at isoform level, taking into consideration that different isoforms of the same gene may have different variability. In specific, within-sample variability and between-sample variability of each transcript are modeled by a Poisson-Lognormal model and a Gamma-Gamma model, respectively. Under the Bayesian framework, the differential state of each transcript as well as other model parameters are estimated jointly through a Markov Chain Monte Carlo (MCMC) process.

Citation

Xu Shi, Xiao Wang, Lu Jin, Leena Halakivi-Clarke, Robert Clarke, Andrew F. Neuwald, and Jianhua Xuan, "Bayesian identification of differentially expressed isoforms using a novel joint model of RNA-seq data".

Requirement

The Matlab package of BayesIso method was tested under Windows7/10 64bit MATLAB 2012b (or later) and Ubuntu 10.04 64bit MATLAB 2012b (or later). The package also includes several external Matlab packages and functions for string operation and numerical calculations, which are:

'string' package: includes functions such as strsplit.m;

alogam.m: computes the logarithm of the Gamma function;

Usage

- ***Pre-processing of annotation information***

The processed annotation information for human isoform structure is provided by 'Isof_structure_hg19.mat' in folder 'annotation_info'. The details to generate the annotation information are as follows.

The original annotation information for human isoform structure in 'bed' format was downloaded from UCSC table browser (http://genome.ucsc.edu/cgibin/hgTables). For the implementation of BayesIso, the annotation information was processed by: 1. The overlapped exons were split into non-overlaped 'sub-exons'; 2. Genes were sorted according to the number of involved isoforms. Then, the processed annotation information for human was saved as 'hg19.gene.post.bed' in the folder 'annotation_info', which can be further processed by 'generate_isoform_structure.m' to formulation annotation information in '.mat' format.

- ***Simulation study***

We provide a script demo_BayesIso_sim.m to generate a demo simulation experiment and run BayesIso. There are two options to run BayesIso with or without covariates settings. The option is controlled by a variable 'HAS_COV'. If HAS_COV = 0, the demo will run without covariates. If HAS_COV=1, the demo will run with covariates. The simulation data is generated and analyzed according to the following steps.

1. Simulate over-dispersed RNA-seq counts

With the processed annotation information for isoform structure ('isof_structure_forDemo.mat') which is a subset of the annotation information saved in 'Isof_structure_hg19.mat', we simulate over-dispersed RNA-seq counts on a randomly selected a set of genes with the parameter settings in Table 1.

Table 1. Parameter setting for simulation data

Parameter	Value	Description
tau	1.78	Within-sample over-dispersion parameter
alpha	1	Gamma shape parameter for β
alpha0	0.5	Gamma shape parameter for λ
v	0.1	Gamma rate parameter for λ
J1	10	Number of samples in phenotype 1
J2	10	Number of samples in phenotype 2
Num_genes	200	Number of genes to be simulated
K	2	Number of isoforms per gene
DEGPCENT	0.5	Percentage of differentially expressed isoforms

If HAS_COV is set to 1, then an additional covariate matrix C will be needed to simulate and analyze the data. In this demo, a random covariate matrix including age, sex and treatment will be generated. The contribution of covariates to the true expression of isoforms is controlled by not exceeding a cutoff of 1%. The cutoff can be tuned by changing parameter th_C.

2. Run BayesIso using:

[beta_e1, d_e] = Func_BayesIso_multi(Y0, X0, I,G,K,J1,J2,Result_dir,C)

The inputs to the *Func_BayesIso_multi* function are:

G	Number of genes.
Y0	$G \times 1$ Cell array. Each cell is a matrix of simulated counts for a gene, where each row corresponds to an exon and each column corresponds to a sample.
X0	$G \times 1$ Cell array. Each cell is a matrix indicating isoform structure of a gene, where each row corresponding to an exon; each column corresponds to an isoform; and each element is the length of the exon if involved in the corresponding isoform.
I	$G \times 1$ Vector indicating the number of exons for the selected genes.
K	Number of isoforms per gene.
J1	Number of samples in phenotype 1
J2	Number of samples in phenotype 2
Result_dir	Directory to save the results.
C	(optional) Covariates matrix. If provided, BayesIso will run in covariates mode

The outputs to the *Func_BayesIso_multi* function are:

beta_e1	Estimated abundance for each isoform
d_e	Estimated probability that the isoform is differentially expressed
result.mat*	Intermediate results from the sampling procedure.

3. Evaluate the performance

The ROC curve of the demo experiment is shown by Fig. 1. The experiment without covariates has an AUC of 0.9285. The experiment with covariates has an AUC of 0.8780. Fig. 2 shows the histogram of estimated probability for differential expression.

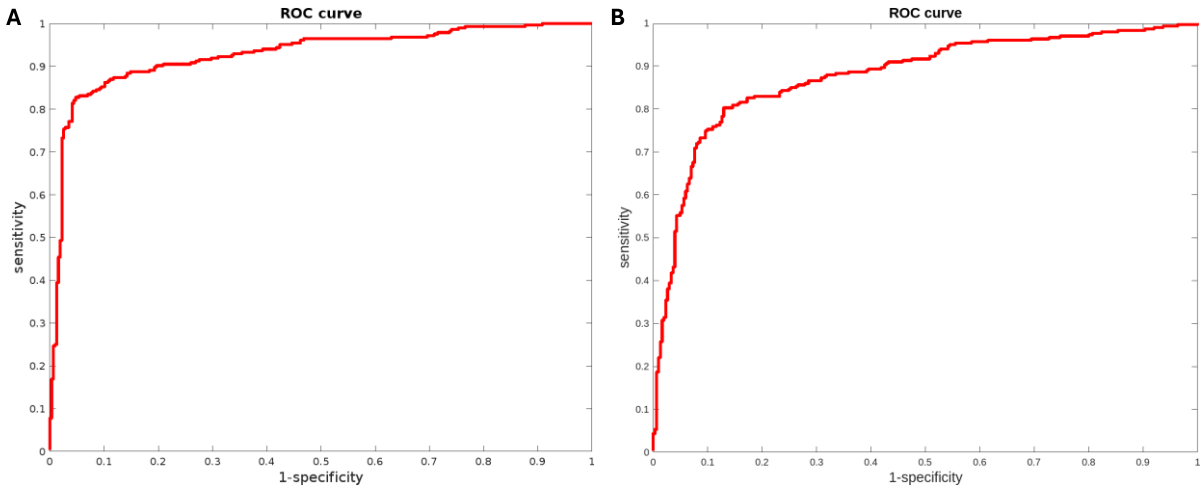

Fig. 1. ROC curve of demo experiment for differential isoform identification. (A) without covariates and (B) with covariates.

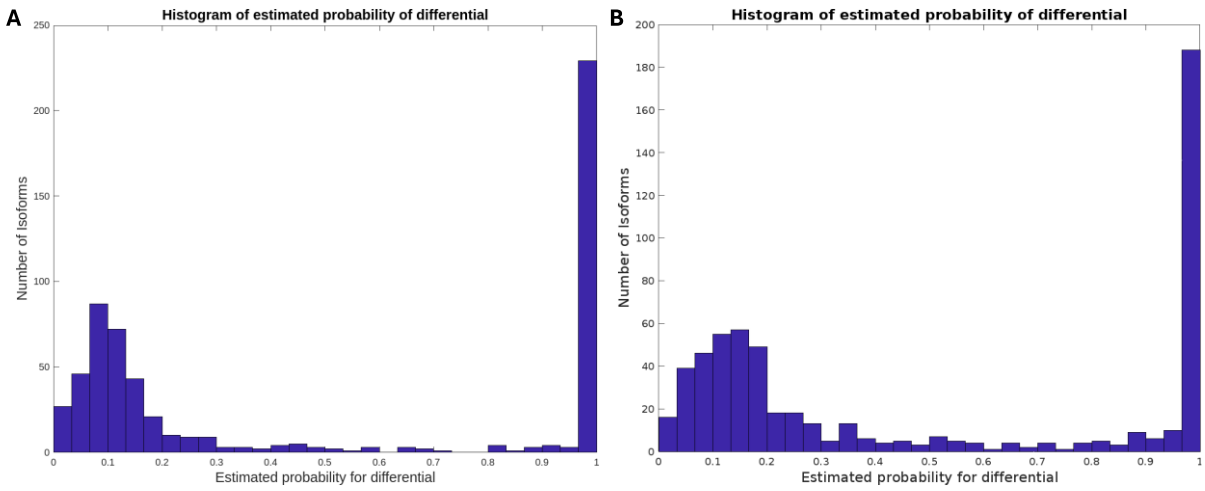

Fig. 2. Histogram of estimated probability for differential expression. (A) without covariates and (B) with covariates.

- ***Pipeline for real data analysis***

To implement BayesIso for real data analysis, the raw RNA-seq data need to be first pre-processed to generate the read count information by the following steps.

Step 1. Alignment of sequencing reads.

The sequencing reads are first aligned to the reference genome using tools such as 'TopHat', and the aligned reads of each sample are saved in the *.bam file.

Step 2. Generating read count 'cov' files.

For each sample, given the annotation file of transcript structure in the 'bed' format, the read count information can be generated by the provided 'perl' scripts, and saved in the 'cov' files.

Single-end RNA-seq data:

```
perl CountReadsSingle.pl -i *.bam -o *.cov -b annotation_info/hg19.gene.post.bed
```

Paired-end RNA-seq data:

```
perl CountReadsPair.pl -i *.bam -o *.cov -b annotation_info/hg19.gene.post.bed
```

Step 3. Generating the summary file of sample information

Sample information including the filenames of the read count 'cov' files, the phenotype information, and the library size, are summarized in the sample_info.txt file, and the format is shown by Table 2. The library size can be obtained by

```
samtools view -c *.bam
```

Table 2 Format of the sample_info.txt file

Filenames	Phenotype	LibSize
Sample1.cov	1	1000000
Sample2.cov	1	1000000
Sample3.cov	2	1000000
Sample4.cov	2	1000000
…	….	…

Step 4. Implementation of BayesIso

With sample_info.txt as input, BayesIso is implemented on real data by running *run_BayesIso.m*. The results are saved in the 'Results_BayesIso.txt' file with the following information.

GeneName	Isoform_ID	Prob(differential)	Fold change (cond1/cond2)	Mean_beta (cond1)	Mean_beta (cond2)
ABL1	ABL1	1	0.1918	1.567	8.168
…	…	…	…	…	…