HBA-DEALS Documentation

Release 0.9

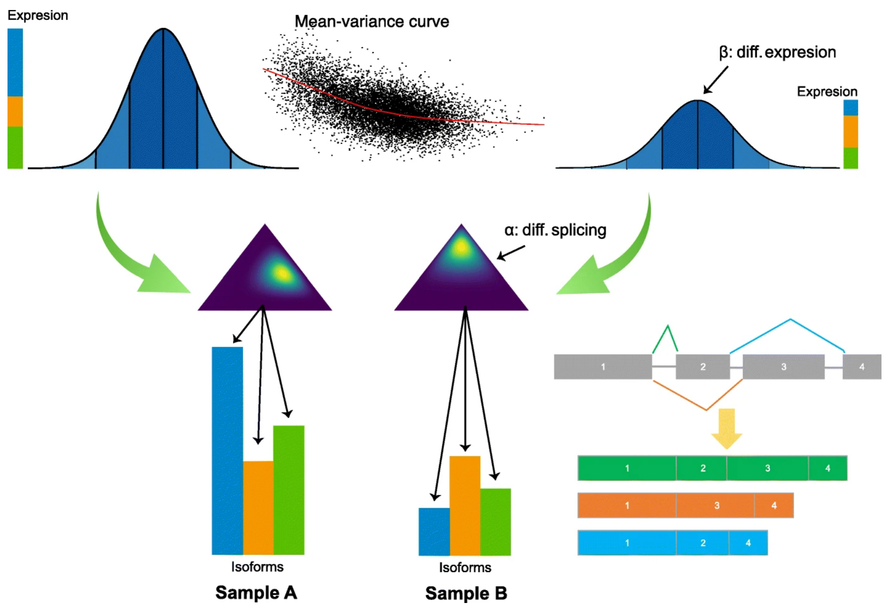
Guy Karlebach, Peter Robinson

Mar 25, 2022

CONTENTS:

HBA-DEALS simultaneously characterizes differential expression and splicing in cohorts.

HBA-DEALS is based on a hierarchical Bayesian model of the absolute expression levels of the gene and its isoforms. HBA-DEALS assumes that the data are available from *n* RNA-seq samples and that the sequence reads have been mapped to isoforms. The *n* samples are divided into two cohorts *n1* and *n2* (e.g., cases and controls). The output of any isoform quantification tool, including but not limited to Salmon, RSEM, Kallisto, and StringTie, can be used as the input for HBA-DEALS.



This documentation explains how to setup and run HBA-DEALS.

CHAPTER ONE

SETTING UP HBA-DEALS

HBA-DEALS is an R package that has been tested with R version 4.1. Source code can be downloaded from the HBA-DEALS GitHub page.

To install HBA-DEALS and its Prerequisites, run the following from the R console:

Listing 1: setting up HBA-DEALS

```
install.packages('edgeR')
install.packages('devtools')
library(devtools)
install_github('https://www.github.com/TheJacksonLaboratory/HBA-DEALS')
```

If any error messages are printed, make the required modifications and re-run the commands.

CHAPTER

TWO

COHORT RNA-SEQ DATA

HBA-DEALS is designed for the analysis of RNA-seq cohort data. The nature of the cohorts is arbitrary but will commonly be labeled as cases and controls or group 1 vs. group2.

In general, there are many ways to obtain such data including generating RNA-seq data or downloading it from an appropriate site. For this tutorial, we download a dataset from the NCBI Sequence Read Archive (SRA), which is the primary archive of NGS datasets.

2.1 Downloading from SRA

We will use the SRA Toolkit.

We will use 8 RNA-seq samples from the dataset SRP149366, which investigated estrogen responsive transcriptome of estrogen receptor positive normal human breast cells in 3D cultures.

control	estradiol
SRR7236472 SRR7236473 SRR7236474 SRR7236475	SRR7236480 SRR7236481 SRR7236482 SRR7236483

First install `fasterq-dump` on your system accorrding to the SRA toolkit instructions. The download page may be helpful.

Then execute the following command from the shell.

Listing 1: Downloading RNA-seq files with the SRA Toolkit

```
for srr in SRR7236472 SRR7236473 SRR7236474 SRR7236475 SRR7236480 SRR7236481 SRR7236482␣
↪SRR7236483; do \
    prefetch $srr
    fasterq-dump -t tmp/ --split-files --threads 8 --outdir tutorial/ $srr
done
```

If necessary, change the `--threads` argument according to the resources of your system. The downloaded `*.fastq` files will be written to the `tutorial` directory. Other directories that were created (e.g., `SRR7236472`, which contains the file `SRR7236472.sra`) can be deleted.

This step will typically take a few hours.

2.2 Cleaning the reads

fastp performs quality control, adapter trimming, quality filtering, per-read quality pruning and many other operations with a single scan of the FASTQ data.

fastp can be downloaded at its GitHub site, which also has installation instructions for various platforms. If you have a debian or Ubuntu system, then the easiest installation is just

```
sudo apt install fastp
```

After you have installed fastp, run the following command in the shell.

Listing 2: Cleaning FASTQ files with fastp

```
for srr in SRR7236472 SRR7236473 SRR7236474 SRR7236475 SRR7236480 SRR7236481 SRR7236482␣
↪SRR7236483; do \
    fastp -i tutorial/${srr}_1.fastq -I tutorial/${srr}_2.fastq -o tutorial/${srr}_
↪trimmed_1.fastq -O tutorial/${srr}_trimmed_2.fastq; \
done
```

This will create for each .fastq file that was downloaded a file by the same name with an added "_trimmed_" in its name. At this point, you can delete the original `*.fastq` files if desired.

CHAPTER

THREE

INFERRING TRANSCRIPT READCOUNTS

For this tutorial, we will use the isoform quantification tool RSEM to generate isoform counts. A typical run of RSEM consists of just two steps. First, a set of reference transcript sequences are generated and preprocessed for use by later RSEM steps. Second, a set of RNA-Seq reads are aligned to the reference transcripts and the resulting alignments are used to estimate abundances and their credibility intervals.

3.1 Installing rsem

Download the `RSEM-1.3.3.tar.gz` file, uncompress it, and build it.

```
tar xfvz RSEM-1.3.3.tar.gz
cd RSEM-1.3.3
make
```

You can follow this with `sudo make install`, softlink to the required executables, or add the path of this directory to the path when you execute the RSEM commands below. The latter option is as follows.

```
export PATH=/home/<user>/<somepath>/RSEM-1.3.3/:$PATH
```

3.2 Installing STAR

RSEM requires Spliced Transcripts Alignment to a Reference (STAR) to be installed and available in the path. STAR can be downloaded from its GitHub site. For instance, download the statically compiled executable (`STAR_Linux_x86_64_static.zip`), unpack it, and put it in the PATH similarly to the above.

3.3 Preparing the reference genome

First, run the following commands to download and unzip the human genome and gene annotation:

```
wget https://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_39/GRCh38.p13.
↪genome.fa.gz
gunzip GRCh38.p13.genome.fa.gz
wget https://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_39/gencode.v39.
↪annotation.gtf.gz
gunzip gencode.v39.annotation.gtf.gz
```

Create a reference using RSEM by executing the following commands (note that we make a directory for the output of ths RSEM script).

```
mkdir ref_rsem
rsem-prepare-reference --gtf gencode.v39.annotation.gtf --num-threads 8 --star GRCh38.
↪p13.genome.fa ref_rsem/ref_rsem
```

Note that the RSEM steps are computationally intensive and would typically be done in a high-performance computing environment rather than on a laptop.

3.4 Counting isoforms

RSEM is a software package for estimating gene and isoform expression levels from RNA-Seq data.

```
for srr in SRR7236472 SRR7236473 SRR7236474 SRR7236475 SRR7236480 SRR7236481 SRR7236482␣
↪SRR7236483; do
    rsem-calculate-expression --star --paired-end --no-bam-output --num-threads 8␣
↪tutorial/${srr}_trimmed_1.fastq tutorial/${srr}_trimmed_2.fastq ref_rsem/ref_rsem rsem/
↪$srr;
done
```

This will produce outfiles with names such as `SRR7236472.isoforms.results` with the following table structure. The HBA-DEALS data ingest script will use this file as input.

transcript_id	gene_id	length	effective_length	expected_count	TPM	FPKM	IsoPct
ENST00000373020.9	ENSG00000000003.15	3768	3546.60	1265.04	26.44	20.76	77.37
ENST00000494424.1	ENSG00000000003.15	820	598.61	0.00	0.00	0.00	0.00
ENST00000496771.5	ENSG00000000003.15	1025	803.60	20.42	1.88	1.48	5.51

CHAPTER

FOUR

RUNNING HBA-DEALS

We will now read the expected counts from the rsem directory and create a count matrix, which we will pass as input to HBA-DEALS.

The following code creates the counts matrix

```
countsData=NULL  #The counts matrix
labels=c()   #The sample labels, 1 for control and 2 for case
for (srr in c('SRR7236472','SRR7236473','SRR7236474','SRR7236475')) {
    if (is.null(countsData))
        countsData=read.table(paste0('rsem/',srr,'.isoforms.results'),header=TRUE)[c(2,
↪1)]
    next.file=read.table(paste0('rsem/',srr,'.isoforms.results'),header=TRUE)
    countsData=cbind(countsData,next.file$expected_count)
   labels=c(labels,1)
}
for (srr in c('SRR7236480','SRR7236481','SRR7236482','SRR7236483')) {
   next.file=read.table(paste0('rsem/',srr,'.isoforms.results'),header=TRUE)
   countsData=cbind(countsData,next.file$expected_count)
   labels=c(labels,2)
}
countsData=countsData[rowSums(countsData[,-c(1,2)]>0)>=ncol(countsData)-2,] #remove low-
↪count isoforms
num.iso=unlist(lapply(countsData$gene_id,function(x){sum(countsData$gene_id %in% x)}))
countsData=countsData[num.iso>1,]  #keep only genes with at least two isoforms
```

The following code performs the HBA-DEALS analysis.

```
library(hbadeals)
library(edgeR)
res=hbadeals(countsData = countsData,labels = labels,n.cores = 32,isoform.level=TRUE,
↪mcmc.iter=100000,mcmc.warmup=10000,mtc=TRUE,
        lib.size=colSums(countsData[,-c(1,2)])*calcNormFactors(as.matrix(countsData[,-
↪c(1,2)]),method='TMM'))
write.table(res,'hbadeals_output.txt', sep='\t',quote = F,col.names = T,row.names = F)
```

4.1 HBA-DEALS arguments

- **countsData: A table of gene names, transcript names and transcript counts in each sample. At least two transcripts must**
  each gene.
- labels: An ordered vector of 1's and 2's. Its length is ncol(countsdata)-2. Each entry indicates whether the corresponding sample/column of countsData belongs to the first experimental condition or the second.
- n.cores: The number of cores to use in the calculation. It is recommended to dedicate as many cores as possible.
- isoform.level: if `true`, return 1-probability of differential proportion for each transcript be returned. If `FALSE` (the default) return 1-probability of differential splicing for each gene.
- mcmc.iter: The number of iterations of the MCMC algorithm after warmup.
- mcmc.warmup: The number of warmup iterations of the MCMC algorithm.
- hierarchy: Determines whether a hierarchical model will be used (hierarchy='yes'), a flat one (hierarchy='no') or will the decision will be made automatically (hierarchy='auto')
- lib.size: A numeric vector containing total library sizes for each sample. If not provided, the default is column-wise count totals.
- mtc: A logical argument (default FALSE) that indicates whether the output probabilities should be corrected for multiple comparisons.

4.2 Output format

The HBA-DEALS output file contains 4 columns. The first column is the gene name, the second is the transcript name, the third is the fold change, and the fourth is 1-probability of differential expression or proportion(splicing), which is the posterior error probability (PEP). Entries that refer to expression have 'Expression' in their second column. If isoform.level is FALSE, entries that refer to differential splicing of the gene will have 'Splicing' in their second column entry. The fold change for expression is given as log2 fold change, and for splicing as fold change.

Gene	Isoform	ExplogFC/FC	P
ENSG00000000419	Expression	0.00360856329234098	0.98668
ENSG0000000041	ENST00000371588	0.996357449743301	0.95506
ENSG00000196141	Expression	-2.74059622402359	0
ENSG00000196233	ENST00000371103	0.10984085543658	0.04972