

iDiffIR Documentation

Release 0.3.1

Michael Hamilton

February 03, 2015

CONTENTS

1 Introduction **2**
- 1.1 Intron retention . . . 2
- 1.2 Differential intron retention . . . 2

2 Method **4**

3 Get iDiffIR **5**
- 3.1 Required packages . . . 5
- 3.2 Download . . . 5
- 3.3 Install . . . 5

4 Tutorial **6**
- 4.1 Align reads . . . 6
- 4.2 Filtering alignments . . . 6
- 4.3 Getting read depths . . . 7
- 4.4 Running **iDiffIR** . . . 8
- 4.5 Interpreting iDiffIR output . . . 9

5 Contact **11**

Bibliography **12**

Welcome to the home of **iDiffIR**, a method for **i**dentifying **diff**erential **i**ntron **r**etention from RNA-seq.

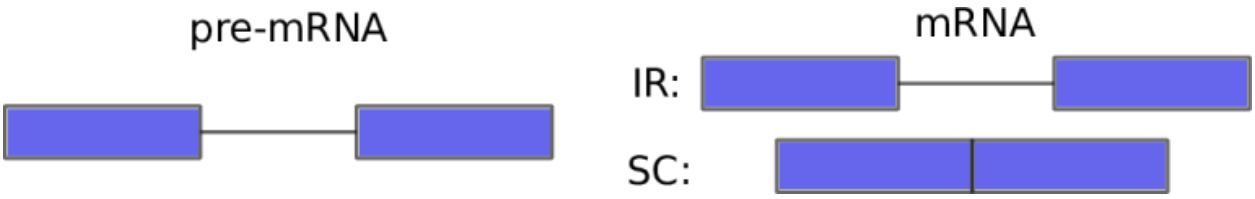
CHAPTER

ONE

INTRODUCTION

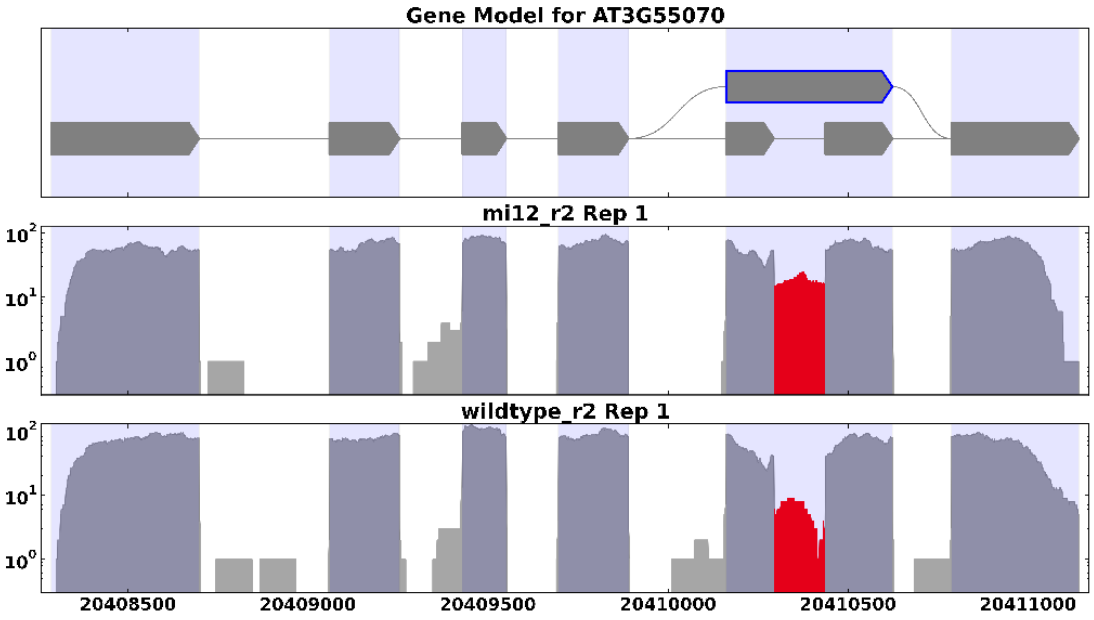
1.1 Intron retention

Intron retention (IR) is an alternative splicing (AS) form where the intronic that separate the exonic components that make up mature mRNA remain in the final transcript. On the left below is a figure of a pre-mRNA showing exons as shaded rectangles separated by a single exon (black line). On the right are two possible outcomes: 1) the intron remains, leading to an IR event or 2) the intron is spliced out, resulting in a constitutively spliced (CS) form.



1.2 Differential intron retention

The figure below shows an example of differential intron retention (IR). The top track shows a splice graph depiction for the gene while the second and third tracks show read depths for RNA-seq alignments from two different conditions. The areas in red corresponds to the depths of the mRNA region detected as a differential IR event. While other introns exhibit some level of expression, there exists insufficient evidence that these introns are differentially retained.


Gene Model for AT3G55070
mi12_r2 Rep 1
wildtype_r2 Rep 1
10^2
10^1
10^0
20408500
20409000
20409500
20410000
20410500
20411000

CHAPTER

TWO

METHOD

iDiffIR uses a $\log FC$ statistic for the average read depth across an intron in conjuction with a gene-wise measure of error from exonic read depths.

CHAPTER

THREE

GET IDIFFIR

3.1 Required packages

The following packages are required for iDiffIR core functions. Version numbers correspond to those tested during development.

- SpliceGrapher (v0.2.4)
- SciPy (v0.14.0)
- Pysam (v0.8.1)
- matplotlib (v1.3.1)
- NumPy (v1.9.0)

3.2 Download

current release: v0.3.1

iDiffIR is hosted on bitbucket https://bitbucket.org/comp_bio/idiffir

3.3 Install

```
$ tar zxvf iDiffIR_<version>.tgz
$ cd iDiffIR_<version>.tgz
$ python setup.py install
```

Note: To install in a user directory, use the option:

```
--home=/Path/To/Local/Library
```

CHAPTER

FOUR

TUTORIAL

This tutorial is meant as a complete walk-through for identifying differential IR from RNA-seq data. The process is as follows:

1. Align reads
2. Filter alignments
3. Get read depths
4. Analyze depths for differential IR

4.1 Align reads

iDiffIR accepts any sorted, indexed BAM file for single- or paired-end reads. It may be helpful to use an aligner that provides an "NH" tag (such as STAR or TopHat) to identify unique alignments; however, there are other ways of removing multi-reads (reads mapping to more than one genomic location).

4.2 Filtering alignments

It is good practice to remove multi-reads and filter spliced alignments for false-positive junctions before beginning transcriptome analysis of RNA-Seq. Assuming your alignments are in a file named aligned.bam, run the following to extract only uniquely aligned reads:

```
$samtools view -h aligned.bam | grep -we 'NH:i:1' -e '^@*' > unique.sam
```

You may wish to filter out potential false-positive splice junctions using SpliceGrapher[SG] **sam_filter.py** from uniquely aligned reads. While the presence of false-positive splice junctions may not adversely affect identifying differential IR analysis, it is a critical step for detecting all other forms of AS (exon skipping, alternative acceptor/donor events).

```
Usage: sam_filter.py SAM/BAM-file classifiers [options]

Removes false-positive spliced alignments from a SAM or BAM file.
The classifers may be given as a zip-file or as a list of classifier
configuration (.cfg) files.

Options:
  -h, --help  show this help message and exit
  -C CIGAR    File for storing unrecognized CIGAR strings [default: none]
  -f FASTA    Reference genome FASTA [default: none]
  -F FPSITES  File for storing false-positive SAM alignment records [default:
```

```
              none]
  -m MODEL    Gene model file (GTF/GFF3 format) [default: none]
  -o OUTPUT   Output file [default: stdout]
  -r REPORT   Write classifier scores to file [default: none]
  -v          Verbose mode [default: False]
  -z          Apply gzip compression to output [default: False]
```

4.3 Getting read depths

4.3.1 Creating indexed, sorted BAM files

If your reads are in the SAM format, you can convert it to a indexed, sorted BAM file using **convertSam.py**.

```
usage: convertSam.py [-h] [-o BAMFILE] [-p PROCS] [-m MEMORY] [-v] samfile

Generate sorted BAM and index files for given SAM file

positional arguments:
  samfile               Samfile to convert

optional arguments:
  -h, --help            show this help message and exit
  -o BAMFILE, --outfile BAMFILE
                        Name of converted BAM file [default=<sambase>.bam]
  -p PROCS, --procs PROCS
                        Number of processors to use for BAM sorting (default
                        1)
  -m MEMORY, --memory MEMORY
                        Max memory (in GBs) for each processor used for BAM
                        sorting (default 2)
  -v, --verbose         Print verbose output
```

If your alignments are in a sorted BAM file, an index can be generated using:

```
$samtools index <BAMFILE>
```

Both methods will produce .bai file, indicating that the BAM file has been indexed.

4.3.2 Getting chromosomal read depths

Now we are prepared to generate chromosomal read depths from the alignments using **getDepths.py**.

```
usage: getDepths.py [-h] [-o OUTDIR] [-v] bamfile_in

Get chromosomal read depths and junctions

positional arguments:
  bamfile_in            Name of sorted, indexed BAM file

optional arguments:
  -h, --help            show this help message and exit
  -o OUTDIR, --output-dir OUTDIR
                        output file directory name
  -v, --verbose         verbose output
```

Note: It is *strongly* recommended to use a clear naming system for the output directories (**-o** option) generated from **getDepths.py**. For instance if you have RNA-seq between an wildtype and mutant with 2 replicates, you may want to use a naming convention such as:

```
$getDepths.py -o wt_rep1_depths wt_rep1/filtered.bam
$getDepths.py -o wt_rep2_depths wt_rep2/filtered.bam
$getDepths.py -o mt_rep1_depths mt_rep1/filtered.bam
$getDepths.py -o mt_rep2_depths mt_rep2/filtered.bam
```

4.4 Running iDiffIR

```
$ idiffir.py --help
usage: idiffir.py [-h] [-v] [-n] [-l FACTORLABEL FACTORLABEL] [-o OUTDIR] [-s]
                  [-k KRANGE KRANGE] [-c COVERAGE] [-d DEXPTHRESH]
                  [-f FDRLEVEL] [-g GRAPHDIRS] [-m {BF,BH,QV}] [-e {IR,SE}]
                  genemodel factor1Dirs factor2Dirs

Identify differentially expressed introns.

positional arguments:
  genemodel             gene model file: NAME.gtf[.gz] | NAME.gff[.gz]
  factor1Dirs           colon-separated list of directories: PATH-TO-
                        REPLICATE_1[:PATH-TO-REPLICATE_2,...]
  factor2Dirs           colon-separated list of directories: PATH-TO-
                        REPLICATE_1[:PATH-TO-REPLICATE_2,...]

optional arguments:
  -h, --help            show this help message and exit
  -v, --verbose         verbose output [default is quiet running]
  -n, --noplot          Do not plot figures [default is to make figures]
  -l FACTORLABEL FACTORLABEL, --factorlabel FACTORLABEL FACTORLABEL
                        factor labels, example: -f Mutant Wildtype
  -o OUTDIR, --output-dir OUTDIR
                        output file directory name
  -s, --shrink_introns  shrink introns for depth plots [default is no
                        shrinking]
  -k KRANGE KRANGE, --krange KRANGE KRANGE
                        kmin kmax; [default is to search for kmax]
  -c COVERAGE, --coverage COVERAGE
                        coverage cutoff, default = 0.99
  -d DEXPTHRESH, --dexpThresh DEXPTHRESH
                        differential gene expression threshold, [default = 10]
  -f FDRLEVEL, --fdrlevel FDRLEVEL
                        FDR test level, [default = 0.05]
  -g GRAPHDIRS, --graph-dirs GRAPHDIRS
                        colon-separated list of directories to recursively
                        search for SpliceGrapher predictions
  -m {BF,BH,QV}, --multTest {BF,BH,QV}
                        Multiple testing adjustment method BF: Bonferroni, BH:
                        Benjamini-Hochberg, QV: q-values [default = QV]
  -e {IR,SE}, --event {IR,SE}
                        AS event to test, IR: Intron Retention, SE: Exon
                        Skipping [default = IR]
```

While **iDiffIR** offers many options, default values should work for most cases. Using the above naming convention, a

way to run **iDiffIR** would be:

```
$idiffir.py -l Mutant Wildtype -o my_result my_genome.gff mt_rep1_depths:mt_rep2_depths wt_rep1_depth
```

The **-l** option provides titles for the reads and will be displayed on the splice graphs of genes containing significant differential IR events. You may wish to use the **-d** option to lower the gene expression fold-change threshold. Introns in genes with a gene expression fold-change above this threshold will not be tested. While **iDiffIR** does adjust for differential gene expression, in some extreme cases it may be necessary.

4.5 Interpreting iDiffIR output

iDiffIR builds an output directory as follows:

```
$ tree my_result
  my_result
  |-- figures
  |   |-- gene1.pdf
  |   |-- gene2.pdf
  |   |-- mva.pdf
  |   |-- pvalues.pdf
  |-- figuresLog
  |   |-- gene1.pdf
  |   |-- gene2.pdf
  |-- lists
      |-- allDIRGenes.txt
      |-- allIntrons.txt

  3 directories, 8 files
```

Two figures directories **figures** and **figuresLog** are created and contain splice graphs figures of genes with at least one significant differential IR event. The **figuresLog** directory contains graphs in $\log$ scale which can be helpful in cases where intronic expression is orders of magnitude less than the exonic expression within the gene. The **lists** directory contain two important files. **allDIRGenes.txt** contain the gene IDs for all genes containing a significant differential IR event. This is should allow convenient downstream analyses, such as GO term enrichment. Finally, **allIntrons.txt** is a tab-delimited file containing intronic coordinates and statistics for all tested introns. The fields of each row are:

1. **geneID** the gene's identifier
2. **lowExonCoords** the coordinates of the lowest (W.R.T. genomic position) flanking exon
3. **intronCoords** the coordinates of the intronic region tested
4. **highExonCoords** the coordinates of the highest (W.R.T. genomic position) flanking exon
5. **pValue** the p-value of the tested intron (using a 2-sided Z-score test).
6. **adjPValue** the multiple testing adjusted p-value
7. **logFoldChange** the $\log$-fold change of the tested intron (w.r.t. the first-given condition)
8. **intronExp** the expression of the intron, computed as $\frac{1}{2}\log(x_1 + x_2)$, where x_1, x_2 are the average read depth of the intron in condition 1 and condition 2, respectively.
9. **statistic** the test statistic (before z-score conversion)
10. **bestA** the pseudo-count value (a) that minimizes the p-value
11. **known** whether this intron is known as a retained intron

Todo

Add differential exon skipping

Todo

Add MISO script usage for testing Alt 5', 3'

Todo

Add simulation

CHAPTER

FIVE

CONTACT

iDiffIR is developed by Mike Hamilton at Colorado State University.

Bug reports and feature requests can be submitted through bitbucket.

- *search*

[SG] Rogers, MF, Thomas, J, Reddy, AS, Ben-Hur, A (2012). SpliceGrapher: detecting patterns of alternative splicing from RNA-Seq data in the context of gene models and EST data. *Genome Biol.*, 13, 1:R4.