Splicing analysis toolkit (Spanky) v.0.1.2

Overview

Spanky is a set of tools to facilitate analysis of alternative splicing from RNA-SEQ data. Drawing from multiple published and novel analysis methods, Spanky compiles quantitative and qualitative information about junction alignments and splicing events. The basic functions of Spanky are to assess confidence in junction alignments, and to quantify and compare splicing events.

Getting started

Prerequisites

Spanky requires the following python packages (Python will attempt to install them for you):

- pyfasta
- pysam
- numpy
- Biopython
- scikits.statsmodels
- fisher

Required for all functions:

- samtools http://samtools.sourceforge.net/
- Cufflinks (*Spanky uses the gtf_to_sam utility to create a sam representation of a gtf reference*) http://cufflinks.cbcb.umd.edu

Required for splicing analysis:

- AStalavista (or a precomputed splicing event file) http://genome.crg.es/astalavista/

Installation

Files are located here:
ftp://helix.nih.gov/pub/sturgill
Decompress the program files spanky.0.1.2.tar.gz
Then use the python install process:

```
sudo python setup.py install
```

==========

Manual

Contents:

- Junction alignment evaluation
- Simulations
- Splicing analysis
    - Quantification of inclusion / exclusion paths
    - Comparison of splicing events between samples
    - Comparison of junctions between samples
- Utilities and extras
    - Merging junction results
    - Quick junction coverage calculator
    - Generate models for simulation

==========

Junction alignment evaluation

Spanky takes an alignment file in BAM format as input and calculates a variety of confidence metrics about spliced alignments. The BAM file can be from any aligner, but has only been tested on BAM files produced by Tophat (Trapnell et. al, http://tophat.cbcb.umd.edu)

Usage:

spankyjunc -i alignments.bam -g genemodels.gtf -f genome.fa [Options]

Options:

```
-i    input alignments
-o    output directory        [default: junctions_out]
-g    reference gtf           (Required)
-f    reference fasta         (Required)
-m    mode                    ('all', 'eval', 'quant')
```

```
                                [Default: all]
-f    filter                    ('T','F')
                                [Default: T]
```

Details:

Modes:

Spanky offers three bundles of analyses to perform. The 'quant' mode performs quantitation only, and the 'eval' mode generates more quality criteria to help evaluate the validity of the junction. The 'all' mode performs all available analyses

Fields reported:

cov: Junction coverage where minimum anchor size is >= 8

unfilt_cov: Total junction coverage, including small anchors

annostatus: A code for the annotation status of the join
- an = annotated
- un.ad = unannotated join, annotated donor
- un.aa = unannotated join, annotated acceptor
- un.ad.ad = unannotated join of annotated donor and acceptor

geneassignL: Gene assignment of "Left" edge

geneassignR: Gene assignment of "Right" edge

geneassign: Gene assignment of entire join

gmcode: A code for the gene model where this join resides. Code is 'num transcripts.num introns.num starts'
eg: 3t.21i.3s A gene with 3 transcripts, 21 introns, and 3 start sites

regcode: A code for the type of regulation annotated in the gene model
- ap = alternative promoters
- as = alternative splicing
- asap = alternative promoters and splicing
- con = constitutive, multi-exon
- se = single exon

offsets: Number of unique starting positions for junction alignments

entropy: Entropy calculation described in Graveley et.al, 2011

hamming3: Edit distance between the 5' intron sequence and the 3' exon anchor

hamming5: Edit distance between the 3' intron sequence and the 5' exon anchor

dinucleotide: Donor and acceptor motifs (eg 'GT..AG')

MAXmmes: Maximum of the minimum match on either side for all alignments to this junction (Wang L., et al., 2010)

MAXminanc: Maximim of the minimum anchor sizes or all alignements to this junction.

lirt: Intron read-through on the "Left" side

rirt: Intron read-through on the “Right” side
irt: Total intron read-through
datrans: Donor to acceptor transition probability
dncov: Coverage for all joins connected to this donor
ancov: Coverage for all joins connected to this acceptor

Filtering:
By default, some low-confidence junctions are filtered out (not reported). Filtering can be turned off so that all junctions are reported. Junction alignments with short anchors are handled differently. Junction alignments where the anchor size is less than 8 are reported, along with a total coverage that counts shorter anchors. Junctions that are filtered are those that have any of these traits:

1. Have intron size > 50kb
2. Have non-canonical donor/acceptors
3. Have intron size < 42
4. Have ambiguous gene assignment

Simulations

This will generate simulated reads for every transcript in the provided reference. The number of reads simulated is based on input values of either reads per kilobase (rpk) or coverage (cov). In neither of these is provided, all transcripts are simulated to 1x coverage. Optionally, a text file can be provided that specifies a value for rpk or cov for each transcript. Artificial mismatches are inserted according to models based on data from external controls (NIST) or from *D. melanogaster* RNA-SEQ (dm3). The default model is weighted random.

Usage:

sim_transcripts -g genemodels.gtf -f genome.fa [options]

Example:

```
sim_transcripts -o all_tx_sim -g fb_rel5.25.gtf -f dm3.fa -cov 1
(Generates simulated reads to 1x coverage for all transcripts)
```

Options:

```
-o    output directory        [default: sims_out]
-g    reference gtf           (Required)
-f    reference fasta         (Required)
-bp   read length             [default: 76]
-cov  coverage
-rpk  reads per kb
-m    mismatch model          ('NIST' or 'dm3')
                              [default: Weighted random]
                              (Can also be 'NIST' or 'dm3')
-s    start pos. model        ('weighted' or 'random')
                              [default: random]
-t    optional transcript file
      (see below)
```

Details:

Optional transcript file:

A text file of transcripts and abundances to simulate can be supplied separately. This is essential if you only want to simulate a few transcripts, or to simulate different proportions. This file has two columns: transcript ids and either coverages or RPK. Column headers must be present

Example:

```
txid            rpk
FBtr0081760     100
FBtr0081761     100
FBtr0081759     200
```

or:

```
txid            cov
FBtr0081760     5
FBtr0081761     5
FBtr0081759     10
```

Splicing analysis

Spanky uses splicing definitions from AStalavista. From this, Spanky extracts junction coverages that define each event, which is partitioned into inclusion and exclusion paths. Comparisons are made using Fisher's exact tests. This general approach was first developed by Eric Wang and extended by A. Brooks et. al, 2011 (see Refs). The differential splicing testing used is similar to JuncBASE (See http:// compbio.berkeley.edu/proj/juncbase/).

In addition, Spanky calculates some metrics that are not dependent on defined events.

Quantification of inclusion / exclusion paths
The script 'spankysplice' quantifies inclusion/exclusion paths

Usage:

spankysplice -j juncs.all -c isoform FPKMs -g genemodels.gtf -f genome.fa -a events [options]

Example:

```
spankysplice -o female_repsmerged_asta -j female_repsmerged.juncs -c testdata/female_cuff/
isoforms.fpkm_tracking -f testdata/fasta/myref.fa -g testdata/annotation/genemodels.gtf -a testdata/
annotation/genemodels_splices.out
```

Options:

```
-o    output directory        [default: astatools_out]
-g    reference gtf           (Required)
-j    junction results tab
-a    events annotation       (From AStalvista)
-f    reference fasta
-c    transcript FPKMs        (From Cufflinks)
```

Comparison of splicing events between samples
The script 'splicecomp' compares splicing events between samples

Usage:

splicecomp -a [sample a] -b [sample b] -g genemodels.gtf

Example:

```
splicecomp -a female_repsmerged_asta/asta.out -b male_repsmerged_asta/asta.out -o F_vs_M_splicecomp
```

Options:

```
-o    output directory        [default: splicecomp_out]
-g    reference gtf           (Required)
-a    asta table for sample A (Output of spankysplice)
-b    asta table for sample B (Output of spankysplice)
```

Comparison of junctions between samples
The script 'splicecomp' compares splicing events between samples

Usage:

junccomp -a [sample a] -b [sample b] -g genemodels.gtf

Example:

```
junccomp -a female_repsmerged.juncs -b male_repsmerged.juncs -o F_vs_M_junccomp
```

Options:

```
-o    output directory        [default: junccomp_out]
-g    reference gtf           (Required)
-a    junc table for sample A (Output of spankyjunc)
-b    junc table for sample B (Output of spankyjunc)
```

Complete pipeline:

A pipeline with a sample run is in the file analysis_commands.sh.
Make a working directory somewhere, and decompress the test data from there:
testdata.tar.gz
Also copy analysis_commands.sh to this directory and run it. It will run the following commands:

```
ECHO "[***************] Starting female spankyjunc runs"
spankyjunc -m all -o female_rep1 -i testdata/female_r1.bam -g testdata/annotation/genemodels.gtf -f testdata/fasta/
myref.fa
spankyjunc -m all -o female_rep2 -i testdata/female_r2.bam -g testdata/annotation/genemodels.gtf -f testdata/fasta/
myref.fa
```

```
ECHO "[***************] Starting male spankyjunc runs"
spankyjunc -m all -o male_rep1 -i testdata/male_r1.bam -g testdata/annotation/genemodels.gtf -f testdata/fasta/myref.fa
spankyjunc -m all -o male_rep2 -i testdata/male_r2.bam -g testdata/annotation/genemodels.gtf -f testdata/fasta/myref.fa
ECHO "[***************] Merging the replicates together"
merge_jtabs_all female_rep1/juncs.all,female_rep2/juncs.all > female_repsmerged.juncs
merge_jtabs_all male_rep1/juncs.all,male_rep2/juncs.all > male_repsmerged.juncs
ECHO "[***************] Parsing asta output"
spankysplice -o female_repsmerged_asta -j female_repsmerged.juncs -c testdata/female_cuff/isoforms.fpkm_tracking -f
testdata/fasta/myref.fa -g testdata/annotation/genemodels.gtf -a testdata/annotation/genemodels_splices.out
spankysplice -o male_repsmerged_asta -j male_repsmerged.juncs -c testdata/male_cuff/isoforms.fpkm_tracking -f testdata/
fasta/myref.fa -g testdata/annotation/genemodels.gtf -a testdata/annotation/genemodels_splices.out
ECHO "[***************] Doing pairwise comparison"
splicecomp -a female_repsmerged_asta/asta.out -b male_repsmerged_asta/asta.out -o F_vs_M_splicecomp
junccomp -a female_repsmerged.juncs -b male_repsmerged.juncs -o F_vs_M_junccomp
```

Utilities and extras

Merge multiple junction results tables

merge_jtabs_all
Merges the junction tables together for each replicate within a samples

Example:
```
merge_jtabs_all female_rep1/juncs.all,female_rep2/juncs.all > female_repsmerged.juncs
```

Options:
```
arg[0] is comma separated list of files
```

Quick junction coverage calculator

quickjunc
Quantifies only junction coverages, evaluates only anchor size

Example:
```
quickjunc -i alignments.bam -a 8
```

Wang, E. T., Sandberg, R., Luo, S., Khrebtukova, I., Zhang, L., Mayr, C., Kingsmore, S. F., et al. (2008). Alternative isoform regulation in human tissue transcriptomes. Nature, 456(7221), 470-6.

Wang, L., Xi, Y., Yu, J., Dong, L., Yen, L., & Li, W. (2010). A statistical method for the detection of alternative splicing using RNA-seq. PloS one, 5(1), e8529.