TSIS: an R package to infer time-series isoform switch of alternative splicing

User manual

Wenbin Guo

2017-01-04

Table of Contents

[Installation and loading 2](#_Toc471293683)

[Install dependency packages 2](#_Toc471293684)

[Install TSIS package 2](#_Toc471293685)

[Loading 2](#_Toc471293686)

[TSIS workflow 2](#_Toc471293687)

[Prepare the input data 2](#_Toc471293688)

[Score the isoform switch 3](#_Toc471293689)

[Step 1: search the intersections 3](#_Toc471293690)

[Step 2: score the isoform switches 5](#_Toc471293691)

[Filtrate results 7](#_Toc471293692)

[Subset of results 7](#_Toc471293693)

[Scripts for scoring 7](#_Toc471293694)

[Scoring 7](#_Toc471293695)

[Filtering 9](#_Toc471293696)

[Visualization 11](#_Toc471293697)

[Error bar plot 12](#_Toc471293698)

[Ribbon plot 13](#_Toc471293699)

[Shiny app- as easy as mouse click 14](#_Toc471293700)

[References 14](#_Toc471293701)

[Session Info 14](#_Toc471293702)

# Installation and loading

## Install dependency packages

install.packages(c("shiny", "shinythemes","ggplot2","plotly","zoo","gtools"), dependencies=TRUE)

## Install TSIS package

Install [TSIS](https://github.com/wyguo/TSIS) package from github using [devtools](https://cran.r-project.org/web/packages/devtools/index.html) package.

##if devtools is not installed, typing  
#install.packages('devtools')  
  
library(devtools)  
devtools::install\_github("wyguo/TSIS")

## Loading

Once installed, TSIS package can be loaded as normal

library(TSIS)

# TSIS workflow

## Prepare the input data

Three types of dataset are required for TSIS analysis.

* Dataset 1: Time-series isoform expression data with time points and replicates, with rownames of isoforms and colnames of samples. The expression can be in the format of read counts, TPM (transcript per million), isoform expression ratio to genes, etc.
* Dataset 2: Gene and isoform mapping table corresponding to Dataset 1, with first column of gene names and second column of isoform names.
* Dataset 3: Optional. Names of subset of isoforms. Users can output subset of the results by providing a list of isoforms.

The [TSIS](https://github.com/wyguo/TSIS) package provides example datasets of "AtRTD2" of 300 genes and 766 isoforms, with 26 time points, 3 biological replicates and 3 technique replicates. The experiments were designed to investigate the cold response of genome in Arabidopsis. The isoform expression is in TPM format. For the experiments and data quantification details, please see the AtRTD2 paper [(Zhang, et al.,2016)](http://biorxiv.org/content/early/2016/05/06/051938).

##26 time points, 3 biological replicates and 3 technical replicates, in total 234 sample points.   
AtRTD2$data.exp[1:10,1:3]

## T1\_biorep1\_techrep1 T1\_biorep1\_techrep2 T1\_biorep1\_techrep3  
## AT1G13350\_ID1 3.11146e+00 3.82818000 1.179530  
## AT1G13350\_ID2 1.41983e+01 12.61840000 14.078200  
## AT1G13350\_P1 3.55665e+00 3.03272000 4.347510  
## AT1G01020\_P1 1.00711e+01 9.60077000 11.691600  
## AT1G01020\_P5 3.17617e+00 3.65805000 3.113580  
## AT1G01060.2 0.00000e+00 0.00000000 0.000000  
## AT1G01060.3 2.38935e-01 0.64505200 0.390966  
## AT1G01060.4 4.92209e-01 0.26018000 0.277574  
## AT1G01060\_JS31 0.00000e+00 0.00000000 0.000000  
## AT1G01060\_P1 7.82000e-10 0.00000419 0.000000

AtRTD2$mapping[1:10,]

## genes isoforms  
## 1 AT1G13350 AT1G13350\_ID1  
## 2 AT1G13350 AT1G13350\_ID2  
## 3 AT1G13350 AT1G13350\_P1  
## 4 AT1G01020 AT1G01020\_P1  
## 5 AT1G01020 AT1G01020\_P5  
## 6 AT1G01060 AT1G01060.2  
## 7 AT1G01060 AT1G01060.3  
## 8 AT1G01060 AT1G01060.4  
## 9 AT1G01060 AT1G01060\_JS31  
## 10 AT1G01060 AT1G01060\_P1

colnames(AtRTD2$data.exp)[1:10]

## [1] "T1\_biorep1\_techrep1" "T1\_biorep1\_techrep2" "T1\_biorep1\_techrep3"  
## [4] "T1\_biorep2\_techrep1" "T1\_biorep2\_techrep2" "T1\_biorep2\_techrep3"  
## [7] "T1\_biorep3\_techrep1" "T1\_biorep3\_techrep2" "T1\_biorep3\_techrep3"  
## [10] "T2\_biorep1\_techrep1"

Note: The data loaded to the Shiny App must be in \*.csv format for loading convenience. Users can download the [example datasets](https://github.com/wyguo/TSIS/examples) from <https://github.com/wyguo/examples> or by typing the following codes:

AtRTD2.example(dir='data')

where "dir" is the folder to save the data in the working directory. If it does not exist, a new folder will be created with the name.

## Score the isoform switch

### Step 1: search the intersections

The expression for a pair of isoforms and may experience a number isoform switch in the whole time duration. Two methods have been included to search for these switch points where the isoforms reverse relative expression profiles.

* Method 1: use average expression values across time points. Taking average values of the replicates for time points in the input isoform expression data.

##use function TSIS::rowmean to take average values  
data.exp.mean<-rowmean(t(AtRTD2$data.exp),group = paste0('T',rep(1:26,each=9)))  
data.exp.mean<-t(data.exp.mean)  
data.exp.mean[1:10,1:4]

## T1 T2 T3 T4  
## AT1G13350\_ID1 3.10506344 0.77124133 1.541512e+00 0.8334757  
## AT1G13350\_ID2 10.13306333 6.61213111 3.338652e+00 3.6689611  
## AT1G13350\_P1 3.15604667 6.38890444 1.066438e+01 12.4442111  
## AT1G01020\_P1 10.52656222 10.11033444 8.994877e+00 7.4319078  
## AT1G01020\_P5 3.19043556 1.13783543 1.042149e+00 0.6243580  
## AT1G01060.2 0.00997250 0.20825081 3.661990e+01 144.7847778  
## AT1G01060.3 0.24962776 0.34442077 1.862491e+01 87.5777000  
## AT1G01060.4 0.16089389 0.49383611 2.236930e+01 60.3179444  
## AT1G01060\_JS31 0.00000000 0.20677433 1.056589e+01 24.6278444  
## AT1G01060\_P1 0.01771091 0.01468386 3.755556e-10 0.0000000

##example, to find the intersection points of iso1 and iso2  
iso1='AT1G13350\_ID2'  
iso2='AT1G13350\_P1'  
  
##x1 and x2 are the numeric values for two isoforms to search for the intersection points  
##x.points and y.points are the x axis and y axis coordinate values for the time course intersection points.   
ts.intersection(x1=as.numeric(data.exp.mean[iso1,]),x2=as.numeric(data.exp.mean[iso2,]))

## x.points y.points  
## 1 2.029571 6.515333  
## 2 6.666101 5.766361  
## 3 12.934179 4.340596  
## 4 15.332007 4.476348

* Method 2: use nature spline curves to fit the time-series data and find intersection points of the fitted curves for each pair of isoforms. See details in TSIS::ts.spline and [splines](https://stat.ethz.ch/R-manual/R-devel/library/splines/html/ns.html) package.

##use function TSIS::ts.spline to fit the samples with smooth curve.  
##estimate the values at time points 1-26 on the fitted curves  
data.exp.splined<-apply(AtRTD2$data.exp[1:10,],1,  
 function(x) ts.spline(x,t.start = 1,t.end = 26,nrep = 9,df = 18))  
data.exp.splined<-t(data.exp.splined)  
data.exp.splined[,1:4]

## 1 2 3 4  
## AT1G13350\_ID1 3.11363247 2.6067715 2.4019519 2.6170174  
## AT1G13350\_ID2 10.16635881 9.3482893 8.8486133 8.4414049  
## AT1G13350\_P1 3.12193231 3.9573841 3.9078088 4.0557979  
## AT1G01020\_P1 10.55761600 12.5895772 11.6518549 9.3894956  
## AT1G01020\_P5 3.17663729 4.6517338 3.5272867 2.0264991  
## AT1G01060.2 -0.13521055 1.0843218 -1.7424493 0.6367451  
## AT1G01060.3 -1.45580701 13.7685375 -26.6442369 55.6529147  
## AT1G01060.4 -1.07314222 9.9062994 -19.1126700 40.4415144  
## AT1G01060\_JS31 -0.01399179 0.1370844 -0.3032971 0.5933864  
## AT1G01060\_P1 -0.15509151 1.2123755 -1.2824919 -0.6898001

##x1 and x2 are the numeric values for two isoforms to search for the intersection points  
##x.points and y.points are the x axis and y axis coordinate values for the time course intersection points.   
ts.intersection(x1=as.numeric(data.exp.splined[iso1,]),x2=as.numeric(data.exp.splined[iso2,]))

## x.points y.points  
## 1 10.05429 6.572867  
## 2 14.67084 5.740785  
## 3 18.97011 4.495199  
## 4 21.94467 4.572121

### Step 2: score the isoform switches

We defined 5 parameters to score the quality of isoform switch. The first two are the probability/frequency of switch and the sum of average distance before and after switch, used as Score 1 and Score 2 in [iso-kTSP](https://bitbucket.org/regulatorygenomicsupf/iso-ktsp) method (see [Figure 1(A)](#Figure1)) method for two condition comparisons [(Sebestyen, et al., 2015)](http://biorxiv.org/content/early/2014/07/04/006908). To investigate the switches of two isoforms and in two conditions and , Score 1 is defined as

where and are the frequencies/probabilities that the samples of one isoform is greater or less than in the other in corresponding conditions. Score 2 is defined as

where and are the mean distances of samples in conditions and , respectively.

However, the time-series for a pair of isoforms may undergo a number of switches in the time duration. The time duration is divided into intervals with the intersection points determined in [Step 1](#step1). For example, in [Figure 1(B)](#Figure1), the duration of four time points is divided into interval 1 to 3 with the intersection points of switch1 and switch2. To extend the iso-kTSP to TSIS, the samples in each pair of consecutive intervals before and after switch are assimilated as samples in two conditions to implement the calculation of Score 1 and Score 2.

The time-series isoform switches are more complex than the comparisons over two conditions. In addition to Score 1 and Score 2 for each switch point, we defined other 3 parameters as metrics of switch qualities.

* p-value of paired t-test for the two isoform sample differences within each interval. For example, the p-value for interval 2 in [Figure 1(B)](#Figure1) is

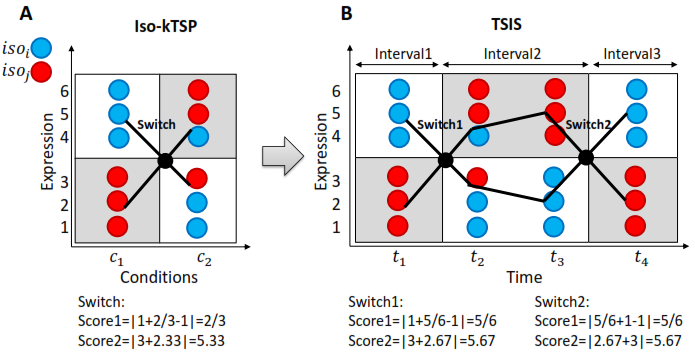
t.test(c(1,1,2,2,3,4),c(3,4,5,5,6,6),paired = T)$p.value

## [1] 5.487226e-05

* Time points number within each interval. For example, there are 1 time point in interval 1 and 3, and 2 time points in interval 2.
* Pearson correlation of two isoforms. For example, the correlation of and is

cor(c(1,2,3,3,5,6,4,5,6,1,2,3),c(4,5,6,1,2,4,1,2,3,4,5,6),method = 'pearson')

## [1] -0.398568



**Figure 1: Isoform switch analysis methods.** Expression data with 3 replicates for each condition/time point is simulated for isoforms and . (A) is the iso-kTSP algorithm for comparisons of two conditions and . The iso-kTSP is extended to time-series isoform switch (TSIS) in figure (B). The time-series with 4 time points is divided into 3 intervals with breaks of isoform switch poitns, which are the intersections of average exprssion of 3 replicates. The intervals are assimlated as the conditions in iso-kTPS. Thereby, the scores for each switch point can been determined based on the intervals before and after switch occurring. Additionally, 3 parameters in interval basis are defined to further filtrate switch results, the p-value of paird t-test for sample differences, the time point number in each interval and the Pearson correlation of two isoforms.

## Filtrate results

A prospective isoform switch should be:

* Have high Score 1 of swtich frequency/probability.
* With proper value of Score 2 the sum of average distances.
* The samples in the intervals before and after switch are statistically different.
* The switch event lasting a few time points in both intervals before and after switch, i.e. the intervals should contain a number of time points.
* For further details, users can investigate the co-expressed isoform pairs with high Pearson correlation. Note: the isoform pairs with high negative correlation may show better switch pattern if look at the time-series plots.

## Subset of results

Users may need to investigate subset of isoforms for specific purpose. Three options have been build-in the TSIS package.

* Users can set the lower and upper boundaries of a region in the time duration to study the switches only within this region.
* Users can provide a name list of isoforms to only show the results cantain the isoforms in the list.
* Users can output subset of results with highest ratio (the proportion of isoforms to the gene) isoforms.

## Scripts for scoring

All the steps of searching intersection points, scoring and filtering are intergrated in two functions TSIS::iso.switch and TSIS::score.filer. We use the datasets TSIS::AtRTD2 in the package as an example to do the analysis. Please go the documentation for function details.

##load the data  
data.exp<-AtRTD2$data.exp  
mapping<-AtRTD2$mapping  
dim(data.exp)

## [1] 766 234

dim(mapping)

## [1] 766 2

### Scoring

Parameters for TSIS::iso.switch function:

* **data.exp, mapping**, input expression data frame and gene-isoform mapping data frame.
* **t.start, t.end, nrep**, start time point, end time point and number of replicates. The time step is assumed to be 1.
* **min.t.points**, pre-filtering, if the time points in all intervals < min.t.points, skip this pair of isoforms.
* **min.distance**, pre-filtering, if the sample distances in the time courses (mean expression or splined value) for intersection search all < min.distance, skip this pair of isoforms.
* **rank**, logical, to use rank of isoform expression for each sample (TRUE) or not (FALSE).
* **spline**, logical, to use spline method (TRUE) or mean expression (FALSE).
* **spline.df**, the degree of freedom used in spline method. See splines::ns for details.
* **verbose**, logical, to track the progressing of runing (TRUE) or not (FALSE).

**Example 1: search intersection points with mean expression**

##Scores  
scores.mean2int<-iso.switch(data.exp=data.exp,mapping =mapping,  
 t.start=1,t.end=26,nrep=9,rank=F,  
 min.t.points =2,min.distance=1,spline =F,spline.df = 9,verbose = F)

## Input genes: 300

## Genes with more than 2 isoforms: 300

## Average isoforms per gene for switch analysis: 2.553

## Step 1: Search for intersection points with Mean expression..

## 479 pairs of isoforms have intersection points.

## Step 2: Calculate scores for isoform switch

## Score 1: Switch frequencies/probabilities

## Score 2: Sum of average sample distances before and after switch.

## Score 3: P-values of sample differences before and after switch

## Score 4: Time points in each intervals

## Score 5: Pearson correlation of isoforms

## Time for analysis: 6.899 secs

## Done!!!

**Example 2: search intersection points with spline method**

##Scores  
scores.spline2int<-suppressMessages(iso.switch(data.exp=data.exp,mapping =mapping,  
 t.start=1,t.end=26,nrep=9,rank=F,  
 min.t.points =2,min.distance=1,spline =T,spline.df = 9,verbose = F))

### Filtering

Parameters for TSIS::score.filter function:

* **scores**, the scores output from TSIS::iso.switch
* **prob.cutoff, dist.cutoff, t.points.cutoff, pval.cutoff, cor.cutoff**, the cut-offs corresponding to switch frequencies/probablities, sum of average distances, p-value and time points cut-offs for both intervals before and after switch and Pearson correlation.
* **data.exp, mapping**, the expression and gene-isoform mapping data.
* **sub.isoform.list**, a vector of isoform names to output subset of the corresponding results.
* **sub.isoform**, logical, to output subset of the results(TRUE) or not (FALSE). If TRUE, sub.isoform.list must be provided.
* **max.ratio**, logical, to show the subset of results with the isoforms of maximum ratios to the genes. If TRUE, data.exp and mapping data must be provided to calculate the isoform ratios to the genes.
* **x.value.limit**, the region of x axis (time) for investigation. If there is no intersection point in this region, the isoform pair is filtered.

**Example 1, general filtering**

##intersection from mean expression  
scores.mean2int.filtered<-score.filter(scores = scores.mean2int,prob.cutoff = 0.5,dist.cutoff = 1,  
 t.points.cutoff = 2,pval.cutoff = 0.01, cor.cutoff = 0.5,  
 data.exp = NULL,mapping = NULL,sub.isoform.list = NULL,  
 sub.isoform = F,max.ratio = F,x.value.limit = c(9,17) )  
  
scores.mean2int.filtered[1:5,]

## iso1 iso2 iso1.mean.ratio iso2.mean.ratio left.interval  
## 1 AT1G07010\_ID1 AT1G07010\_P1 0.3444658 0.6555342 [1,10.4]  
## 2 AT3G61600\_P1 AT3G61600\_P2 0.4587105 0.5412895 [1,9.7]  
## 3 AT1G07010\_ID1 AT1G07010\_P1 0.3444658 0.6555342 (10.4,12.7]  
## 4 AT5G27730\_ID1 AT5G27730\_P1 0.4650146 0.2606082 [1,10.4]  
## 5 AT4G25080.1 AT4G25080.3 0.4249328 0.4579446 [1,10.6]  
## right.invertal x.value y.value left.prob right.prob left.dist  
## 1 (10.4,12.7] 10.384667 32.112522 1.0000000 1.0000000 -28.604855  
## 2 (9.7,26] 9.699607 22.525101 1.0000000 0.9738562 18.251757  
## 3 (12.7,26] 12.667896 73.588727 1.0000000 0.9523810 11.259083  
## 4 (10.4,26] 10.371245 1.249807 0.9444444 1.0000000 -2.706725  
## 5 (10.6,17.9] 10.620866 49.592087 0.9888889 0.9523810 -29.411702  
## right.dist left.pval right.pval left.t.points right.t.points  
## 1 11.259083 5.365298e-30 1.045041e-07 10 2  
## 2 -15.069986 2.149216e-42 1.101754e-68 9 17  
## 3 -35.449155 1.045041e-07 4.451268e-45 2 14  
## 4 3.683429 1.455048e-21 1.352668e-56 10 16  
## 5 9.995878 4.688194e-33 8.124810e-17 10 7  
## prob dist cor  
## 1 1.0000000 39.863939 0.5151262  
## 2 0.9738562 33.321743 -0.7179305  
## 3 0.9523810 46.708239 0.5151262  
## 4 0.9444444 6.390154 -0.5209861  
## 5 0.9412698 39.407580 0.9318253

##intersection from spline method  
scores.spline2int.filtered<-score.filter(scores = scores.spline2int,prob.cutoff = 0.5,  
 dist.cutoff = 1,t.points.cutoff = 2,pval.cutoff = 0.01,  
 cor.cutoff = 0.5,data.exp = NULL,mapping = NULL,  
 sub.isoform.list = NULL,sub.isoform = F,max.ratio = F,  
 x.value.limit = c(9,17) )

**Example 2, only show subset of results according to a isoform list**

##intersection from mean expression  
sub.isoform.list<-AtRTD2$sub.isoforms  
sub.isoform.list[1:10]

## [1] "AT1G13350\_P1" "AT1G01020\_P1" "AT1G01060\_P1" "AT1G01110\_P1"  
## [5] "AT1G01220\_P1" "AT1G01260\_P1" "AT1G01290\_P1" "AT1G01520\_P1"  
## [9] "AT1G01550\_P1" "AT1G01690\_P1"

scores.mean2int.filtered.subset<-score.filter(scores = scores.mean2int,prob.cutoff = 0.5,dist.cutoff = 1,  
 t.points.cutoff = 2,pval.cutoff = 0.01, cor.cutoff = 0.5,  
 data.exp = NULL,mapping = NULL,sub.isoform.list = sub.isoform.list,  
 sub.isoform = T,max.ratio = F,x.value.limit = c(9,17) )  
  
scores.mean2int.filtered.subset[1:5,]

## iso1 iso2 iso1.mean.ratio iso2.mean.ratio  
## 1 AT1G07010\_ID1 AT1G07010\_P1 0.3444658 0.6555342  
## 3 AT1G07010\_ID1 AT1G07010\_P1 0.3444658 0.6555342  
## 21 AT3G54380\_ID8 AT3G54380\_P1 0.3773857 0.4715479  
## 49 AT1G07350\_P1 AT1G07350\_P4 0.1630072 0.1744870  
## 60 AT3G53830\_P1 AT3G53830\_P2 0.3557018 0.2672451  
## left.interval right.invertal x.value y.value left.prob right.prob  
## 1 [1,10.4] (10.4,12.7] 10.38467 32.112522 1.0000000 1.0000000  
## 3 (10.4,12.7] (12.7,26] 12.66790 73.588727 1.0000000 0.9523810  
## 21 [1,9.83] (9.83,15.7] 9.82508 4.461112 1.0000000 0.8148148  
## 49 (11.3,13.5] (13.5,26] 13.45674 1.654753 0.2222222 0.1367521  
## 60 (12.18,15.96] (15.96,22.03] 15.95769 1.974204 0.8148148 0.7460317  
## left.dist right.dist left.pval right.pval left.t.points  
## 1 -28.6048553 11.2590833 5.365298e-30 1.045041e-07 10  
## 3 11.2590833 -35.4491554 1.045041e-07 4.451268e-45 2  
## 21 -5.7669564 2.1463129 5.965631e-38 7.696136e-09 9  
## 49 0.7447684 -1.2927091 6.975744e-03 3.037211e-24 2  
## 60 -1.1612804 0.5480803 1.675155e-04 8.779675e-04 3  
## right.t.points prob dist cor  
## 1 2 1.0000000 39.863939 0.5151262  
## 3 14 0.9523810 46.708239 0.5151262  
## 21 6 0.8148148 7.913269 -0.7146564  
## 49 13 0.6410256 2.037478 0.8167079  
## 60 7 0.5608466 1.709361 0.5962938

### Visualization

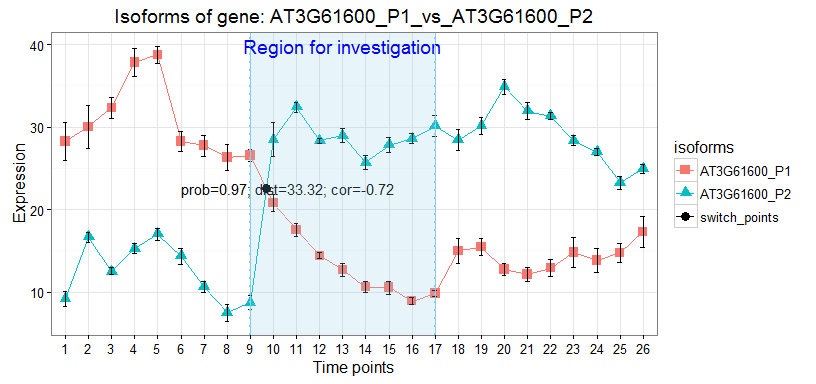
Parameters for TSIS::plotTSIS function:

* **data.exp**, the isoform expression data
* **scores**, the scores output from TSIS::iso.switch or from TSIS::score.filter
* **iso1,iso2**, the names of a pair of isoforms. If not provided, the data.exp must be a two row data frame and the row names of data.exp will be used as iso1 and iso2.
* **gene.name**, the gene name show in the plot title. If not provided, the titile name is shown as iso1\_vs\_iso2
* **y.lab**, the y label of the plot
* **make.plotly**, logical, use plotly::ggplotly (TRUE) to have dynamic plot or ggplot2 to have static plot (FALSE). See the [plotly](https://plot.ly/r/) for details.
* **t.start, t.end, nrep**, start time point, end time point and number of replicates. The time step is assumed to be 1.
* **x.lower.boundary, x.upper.boundary**, the lower and upper boundaries of the time region for investigation
* **show.region**, logical, to show the region (TRUE) for investigation or not (FALSE).
* **show.scores**, logical, to show the score labels on the plot (TRUE) or not (FALSE).
* **line.width, point.size**, the line width and point size for plots
* **error.type, show.errorbar, errorbar.size, errorbar.width**, parameters for error bars. The error.type options are "stderr" for standard error and "sd" for standard deviation.
* **spline, spline.df**, parameters for spline method, corresponding to the settings in TSIS::iso.switch
* **ribbon.plot**, logical, to show ribbon plot (TRUE) or error bar plot (FALSE). See ribbon plot details in [ggplot2::geom\_smooth](http://docs.ggplot2.org/current/geom_smooth.html).

#### Error bar plot

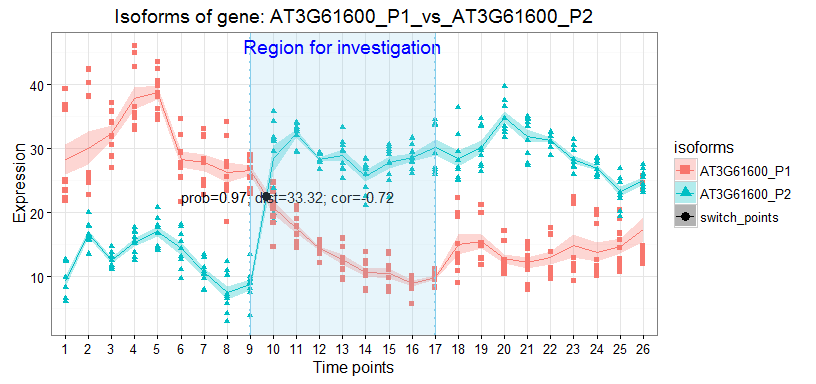
plotTSIS(data2plot = data.exp,scores = scores.mean2int.filtered,iso1 = 'AT3G61600\_P1',  
 iso2 = 'AT3G61600\_P2',gene.name = NULL,y.lab = 'Expression',make.plotly = F,  
 t.start = 1,t.end = 26,nrep = 9,prob.cutoff = 0.5,x.lower.boundary = 9,  
 x.upper.boundary = 17,show.region = T,show.scores = T,  
 line.width =0.5,point.size = 3,error.type = 'stderr',show.errorbar = T,errorbar.size = 0.5,  
 errorbar.width = 0.2,spline = F,spline.df = NULL,ribbon.plot = F )

## Loading required package: ggplot2



#### Ribbon plot

plotTSIS(data2plot = data.exp,scores = scores.mean2int.filtered,iso1 = 'AT3G61600\_P1',  
 iso2 = 'AT3G61600\_P2',gene.name = NULL,y.lab = 'Expression',make.plotly = F,  
 t.start = 1,t.end = 26,nrep = 9,prob.cutoff = 0.5,x.lower.boundary = 9,  
 x.upper.boundary = 17,show.region = T,show.scores = T,error.type = 'stderr',  
 line.width =0.5,point.size = 3,show.errorbar = T,errorbar.size = 0.5,  
 errorbar.width = 0.2,spline = F,spline.df = NULL,ribbon.plot = T )



# Shiny app- as easy as mouse click

All the functions of scoring, filtering, visulisation and saving results have been integrated into a [Shiny app](https://shiny.rstudio.com/). Users can implement the analysis as easy as mouse click. To start the app, simply typing the following code in the R console:

TSIS.app(data.size.max = 100)

where data.size.max is the maximum size limit for unload files in Shiny. Default is 100 (MB).

# References

Chang, W., et al. 2016. shiny: Web Application Framework for R. <https://CRAN.R-project.org/package=shiny>

Sebestyen, E., Zawisza, M. and Eyras, E. Detection of recurrent alternative splicing switches in tumor samples reveals novel signatures of cancer. Nucleic Acids Res 2015;43(3):1345-1356.

Zhang, R., et al. AtRTD2: A Reference Transcript Dataset for accurate quantification of alternative splicing and expression changes in Arabidopsis thaliana RNA-seq data. bioRxiv 2016.

# Session Info

## R version 3.3.1 (2016-06-21)  
## Platform: x86\_64-w64-mingw32/x64 (64-bit)  
## Running under: Windows 7 x64 (build 7601) Service Pack 1  
##   
## locale:  
## [1] LC\_COLLATE=English\_United Kingdom.1252   
## [2] LC\_CTYPE=English\_United Kingdom.1252   
## [3] LC\_MONETARY=English\_United Kingdom.1252  
## [4] LC\_NUMERIC=C   
## [5] LC\_TIME=English\_United Kingdom.1252   
##   
## attached base packages:  
## [1] stats graphics grDevices utils datasets methods base   
##   
## other attached packages:  
## [1] ggplot2\_2.1.0 TSIS\_0.1.0   
##   
## loaded via a namespace (and not attached):  
## [1] Rcpp\_0.12.6 lattice\_0.20-33 gtools\_3.5.0 zoo\_1.7-13   
## [5] digest\_0.6.10 plyr\_1.8.4 grid\_3.3.1 gtable\_0.2.0   
## [9] magrittr\_1.5 scales\_0.4.0 evaluate\_0.10 stringi\_1.1.1   
## [13] reshape2\_1.4.1 rmarkdown\_1.0 labeling\_0.3 splines\_3.3.1   
## [17] tools\_3.3.1 stringr\_1.0.0 munsell\_0.4.3 yaml\_2.1.14   
## [21] colorspace\_1.2-6 htmltools\_0.3.5 knitr\_1.15.1