The Landscape of Isoform Switches in Human Cancers – supplementary text 2: IsoformSwitchAnalyzeR

By Kristoffer Vitting-Seerup & Albin Sandelin

**Table of Contents**

Introduction 1

Analysis of annotated CDS 1

Analysis of accuracy of coding potential predictions via CPAT 2

Prediction of open reading frames (ORFs) 2

Predicting ORF 2

Accuracy of ORF predictions 2

The test for differential isoform usage implemented in IsoformSwitchAnalyzeR 3

Appendix for the test for differential isoform usage implemented in IsoformSwitchAnalyzeR 6

References 6

# Introduction

This supplementary text is meant to supplement the methods section of the original article regarding IsoformSwitchAnalyzeR and the vignette distributed with the R package.

For a step-by-step introduction to the R package IsoformSwitchAnalyzeR we referrer to the R vignette (also given as supplementary text 1).

# Analysis of annotated CDS

We analyzed the annotated CDS in the latest Gencode and RefSeq annotations for Hg19, Hg38, mm9 and mm10 (RefSeq was downloaded from the UCSC genome browser 17th June 2016). We found that a larger fraction of the annotated CDSs in Gencode had undesired properties (Fig. S6A, top). These features include

1. Stop-codons within the annotated CDS
2. Nonsense-start codons – even when considering the 7 non-AUG start codons identified in 1.
3. Out of frame stop-codon

We then removed these features resulting in the HQ dataset we used for benchmarking below (Fig. S6A, bottom).

# Analysis of accuracy of coding potential predictions via CPAT

One potential problem with the coding potential cutoffs provide with the CPAT tools (0.364 for human and 0.44 for mouse) is that they are derived by comparing known genes to random non-coding regions of the genome. This could be a problem since we are analyzing isoforms originating from genes, which often both produce coding and non-coding isoforms. We therefore wanted to investigate the ability to, based on CPAT predictions; distinguish between isoforms annotated as coding (with a CDS) and those annotated as non-coding (without a CDS).

Using the HQ dataset we calculated the accuracy, sensitivity and specificity of all CPAT cutoffs for each of the annotations. From these analyses it is clear to see that the cutoff supplied from CPAT are not optimal when analyzing overlapping coding and non-coding isoforms (Fig. S6B). Based on the notation that real-life transcripts are more like Gencode than RefSeq we therefore suggest the following cutoffs: 0.725 for human and 0.721 for mouse - selected to balance specificity and sensitivity (Fig. S6B).

# Prediction of open reading frames (ORFs)

If you have performed (guided) de-novo isoform reconstruction (isoform deconvolution) the first step of such annotation is to predict Open Reading Frames (ORF). If you did not perform a (guided) de novo isoform reconstruction you should instead use the annotated CDS (Coding Sequence), obtained though one of the implemented methods, see ”Importing Data Into R” section of the vignette.

## Predicting ORF

To predict ORF we have implemented the analyzeORF() function. This function utilize that we know the genomic coordinates of each transcript to extract the transcript nucleotide sequence from a reference genome (supplied via the genomeObject argument). In analyzeORF() four different methods for predicting the ORF, suitable for different purposes and circumstances are implemented. The four methods are:

1. The ‘longest’ method. This method identifies the longest ORF based on finding the canonical start and stop codons in the transcript nucleotide sequence. This approach is what the CPAT tool uses in its analysis of coding potential. This is the default as it is the most common use case and has the highest accuracy in benchmark against known annotated ORFs. See below.
2. The ‘mostUpstream’ method. This method identifies the most upstream ORF based on finding the canonical start and stop codons in the transcript nucleotide sequence.
3. The ‘longestAnnotated’ method. This method identifies the longest ORF downstream of an annotated translation start site. It requires known translational start sites are supplied to the cds argument.
4. The ‘mostUpstreamAnnoated’ method. This method identifies ORF downstream of the most upstream overlapping annotated translation start site. It requires known translational start sites are supplied to the CDS argument.

## Accuracy of ORF predictions

For each transcript in the HQ dataset we compared the CDS annotation with the result of the predicted ORF, using the ’longest’ method. The result shows that the accuracy of the ORF prediction is on average 89.3% for Gencode and 97.5% for RefSeq (Fig. S6C). Interestingly this accuracy increases slightly if you incorporate the CPAT and/or Pfam results (requiring that the isoform should be predicted to be coding or contain a protein domain respectively) (Fig. S6C). Note that the CPAT cutoff used are 0.725 for human and 0.721 for mouse – which as analyzed above might be more suitable cutoffs when analyzing overlapping coding and non-coding isoforms.

# The test for differential isoform usage implemented in IsoformSwitchAnalyzeR

To facilitate isoform switch analysis when sample sizes are small we developed and implemented a statistical test in IsoformSwitchAnalyzeR. The idea behind this test can be summarized as a three-step approach:

1. Use the variance in gene and isoform expression estimates, obtained via biological replicates, to estimate the uncertainty of the isoform usage (e.g. the variance of the IF values).
2. Use the uncertainty of the IF estimate to statistically test whether changes in isoform usage between conditions are different.
3. Calibrate and correct for multiple testing

The advantages of this approach is that unless the samples analyzed are paired one would not know how to pair the IF values from the samples difference conditions and thereby obtain the dIF values. Another advantage is that certain tools, such as CuffDiff (2), can utilize the statistical model to estimate the mean and variance of genes and isoforms whereby the estimates (probably) are more reliable than those estimated directly from the data – especially when small sample sizes are used.

The statistical test implemented can be derived as follows: Consider the isoform fraction value of isoform x (*IFx*) from a given gene, in a setting with *k* biological replicates:

(1)

where *ix* is the mean isoform expression and *g* is the mean gene expression.

Calculating the variance of IF (denoted *var(IF)*, with the subscript x dropped for clarity) corresponds to calculating the variance of the ratio *i* / *g*. If both the variance (*var*) of *i* and *g* (*var(i)* and *var(g)* respectively) are known then *var(IF)* can be estimated using the delta method (relying on truncated Taylor series expansion) which can be written as follows (3,4):

(2)

where *cov(i,g)* refers to the covariance between *i* and *g* and it is assumed that *i* and *g* have a joint normal distribution and that *g* is not near zero (more specifically, that the standard error of g is small compared to g). This can be viewed as an approximation of Fieller’s Theorem. Estimating the *cov(i,g)* is however non-trivial since *i* and *g* by definition are correlated.

Considering a gene g with n isoforms denoted

(3)

the gene expression can be written as:

(4)

where

for (5)

Using this, the covariance estimate can be rewritten as follows (fully derived in supplementary text 4):

(6)

By assuming that the *cov(ix, in)* is zero, the general expression for *varIF* becomes:

(7)

Although this assumption is not necessarily true for all isoforms, it will hold in those cases where there are changes in isoform usage. The variance of *i* and *g* (*var(i)* and *var(g)*) can easily be obtained from any standard measure of variance since:

(8)

(9)

where *k* is the number of replicates, *SD* is the standard deviation and *SE* is the standard error (of the mean).

To avoid testing genes where the mean expression g is too close to zero we have implemented two (unskippable filters) in the isoformSwitchTest() function. The first filter only allows for test of an isoform if the lower boundary of the 95% confidence interval (CI) of the gene expression (*g*), denoted *Lg*, is larger than 0. The lower 95% CI boundary is calculated as:

(10)

Where *t\** is obtained using the 0.975 quantile of a t-distribution with *k-1* degrees of freedom. The second filter only allows for test of an isoform if the standard deviation of the gene expression (*SDg*) is small compared to the mean gene expression (*g*) by requiring that:

(11)

Together these two filters ensure that *g* is not too close to zero. The CI based filter is stricter for cases with small sample sizes whereas the *SD* filter is the stricter for cases with large sample sizes.

To assess the significance of the differences in *IF* values between conditions lets consider two *IF* values *IF1* and *IF2* along with the associated variance *var(IF1)*, *var(IF2)* from two different conditions (indicated by subscript number) with *k1* and *k2* biological replicates. Then the difference in IF values is calculated as the delta IF value (dIF):

(12)

Using the approach of the Welch t-test (5) we can then calculate the standard error of the difference (*SEdIF*) as follows:

(13)

And the (Welch) t-test statistics (*tdIF*) is calculated as:

(14)

The t-test statistics can be compared to a t-distribution with *V* degrees of freedom to obtain a p-value for the change in isoform usage between the two conditions (e.g. a p-value describing the evidence of a isoform switch). The degrees of freedom *V* are estimated (via Welch–Satterthwaite equation) as:

(15)

The test described above is a conservative test, causing the *P*-value distribution to be skewed towards higher *P*-values. This is most likely due to the multiple times that variations are combined (first to obtain *varIF* and since to obtain *vardIF*) each time increasing the variation.

Assuming that the type I error could be accurately calculated the resulting p-value distribution should be uniform in the interval [0,1) except for a possible peak near zero. This also means that it is possible to calibrate p-values from a conservative test by transforming the original p-value so they have the correct uniform distribution under the null hypothesis (6). This approach has been shown to be beneficial for amongst other DESeq a tool for differential expression analysis known to be quite conservative (6).

For these reasons IsoformSwitchAnalyzeR has the option, controled via the “calibratePvalues” parameter, of using the calibration suggested and implemented by Ferguson *et al* (6) to transform the p-values obtained as described above so the distribution of the transformed p-values becomes (approximately) uniform in the interval [0,1). Afterwards the function corrects for multiple testing as follows. If calibratePvalues=TRUE the calibrated p-values are corrected for multiple testing using Benjamini & Hochberg’s False Discovery Rate (FDR) (if calibratePvalues=FALSE the un-calibrated p-values are corrected) whereby isoform switch q-values are obtained.

As suggested by Ferguson et al the p-value correction is only based on highly expressed data, here defined as cases where an isoform are expressed above the 50 percentile of all isoform expression in both conditions compared. Furthermore the correction is, in according with the authors’ suggestion, only performed if the estimated sigma is < 0.9, else the q-values are calculated directly from the untransformed p-values.

Since a gene, per definition, contains changes in isoform usage if the usage of one of the isoform is significantly changes, we define the gene switch q-value to be the minimum of the associated isoform switch q-values.

# Appendix for the test for differential isoform usage implemented in IsoformSwitchAnalyzeR

One can show that

where

for

This result can be obtained as follows as follows

=

=

=

=

=

# References

1. Ivanov, I. P., Firth, A. E., Michel, A. M., Atkins, J. F. & Baranov, P. V. Identification of evolutionarily conserved non-AUG-initiated N-terminal extensions in human coding sequences. 39, 4220–4234 (2011).