nn-vs-svm

AnhVu

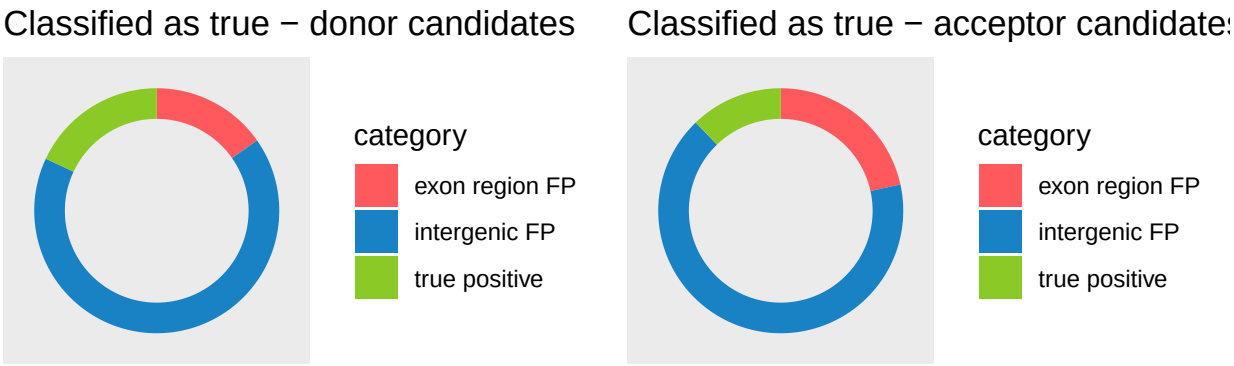
Comparison of SVM and NN models (on *Thega1* phylum)

This section will be devoted to comparing NN and SVM splice site models. In general validation, the models ended up toe to toe in sense of both precision and recall. The tests however do not capture the real nature of the task, which is intron cutting for the purposes of enhancing BLAST search results. Good results here are measured in a slightly more complex manner, as we will see shortly.

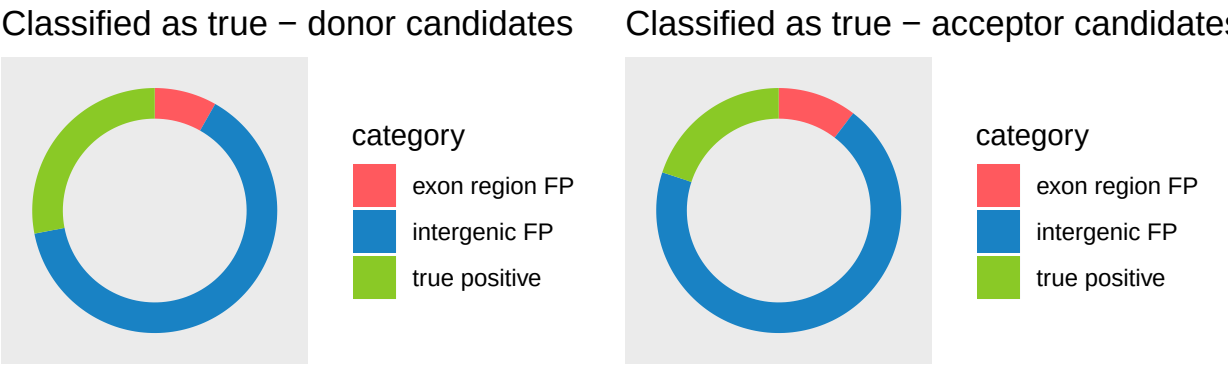
Intron cutting is a classification task. And as in all such tasks, we measure how we successful we are at solving such a task by means of metrics such as precision and recall. In this case however, we need to adjust (or rather extend) the notion of these metrics, especially precision. In intron classiciation and cutting, not all false positives "hurt" the same - in fact, some false positives are even beneficial. This is because we deal with two types of false positives introns here - those which are in intergenic regions and those that are inside exons. The former do not concern us at all. If we cut into the intergenic region, it is only for good, as BLAST will recieve smaller data at the cost of nothing. Intra-exon cuts are on the other hand the "real" false positives, as they will lower the quality of BLAST hits.

One way to compare the two models is therefore to look at the composition of examples, to which the models say they are introns (or splice sites). How many of them are true positives? And if they are false positives, how many of them will actually cause damage? If we compare SVM and NN splice site models in this fashion, we will reveal, that they are not quite the same, as they may seem after the first round testing. We will be particularly interested in ratios between correctly classified donors/acceptors (green) and falsely classified candidates, that lie inside an exon (red).

Splice site classification (SVM models)

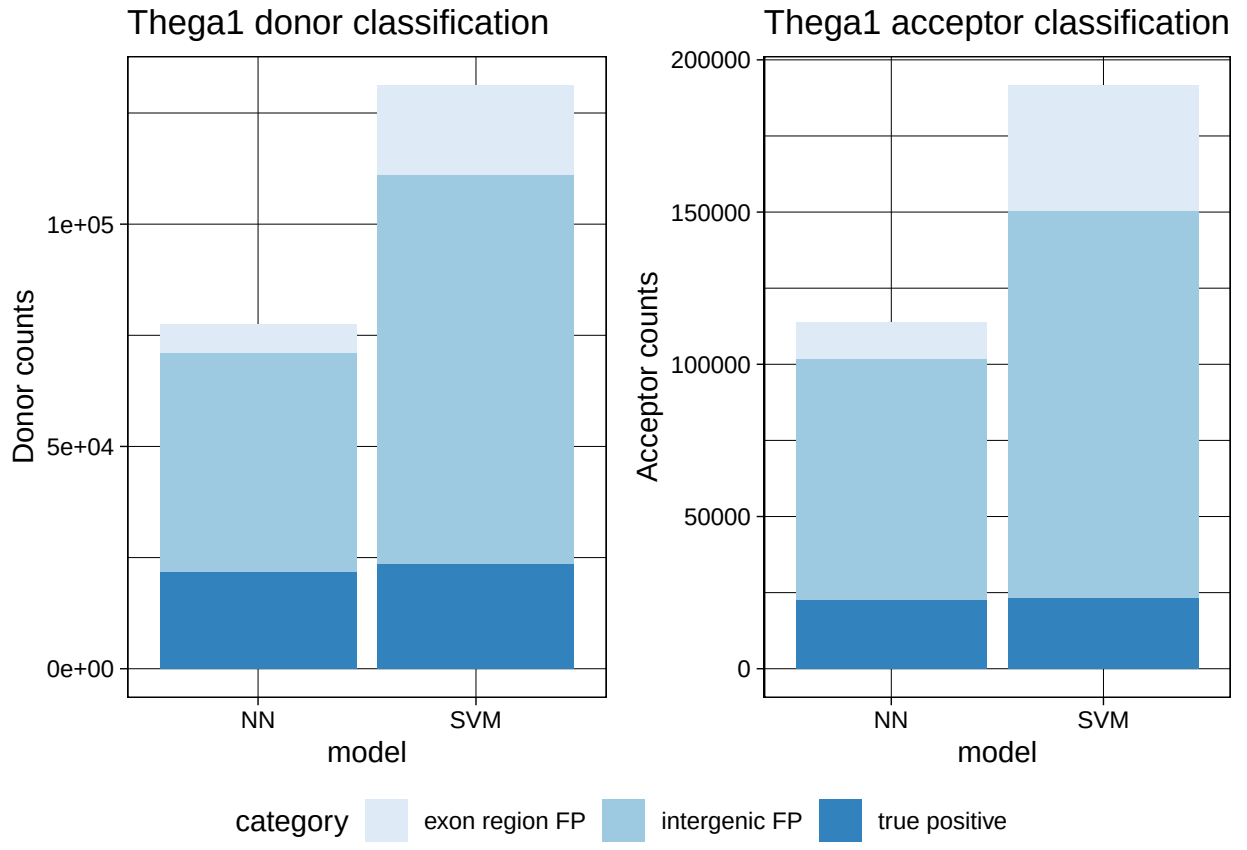


Splice site classification (NN models)



Looking at the two pairs of graphs and the proportions of green-red regions we can clearly see, that NN models have significantly better ratio between correctly classified splice site candidates and incorrectly classified intra-exon ones. The graphs however capture only proportions, not absolute numbers. NN models may easily detect less correct splice sites (even though the general testing indicates this is not the case).

Plots comparing two models in absolute numbers however show this is indeed not the case. The number of true positives is comparable (with a slight advantage of SVM models for donor site). NN models on the other produce substantialy less false positives. What is even more important, this difference is more prominent in the intra-exon cases - the types of FP actually causing damage.

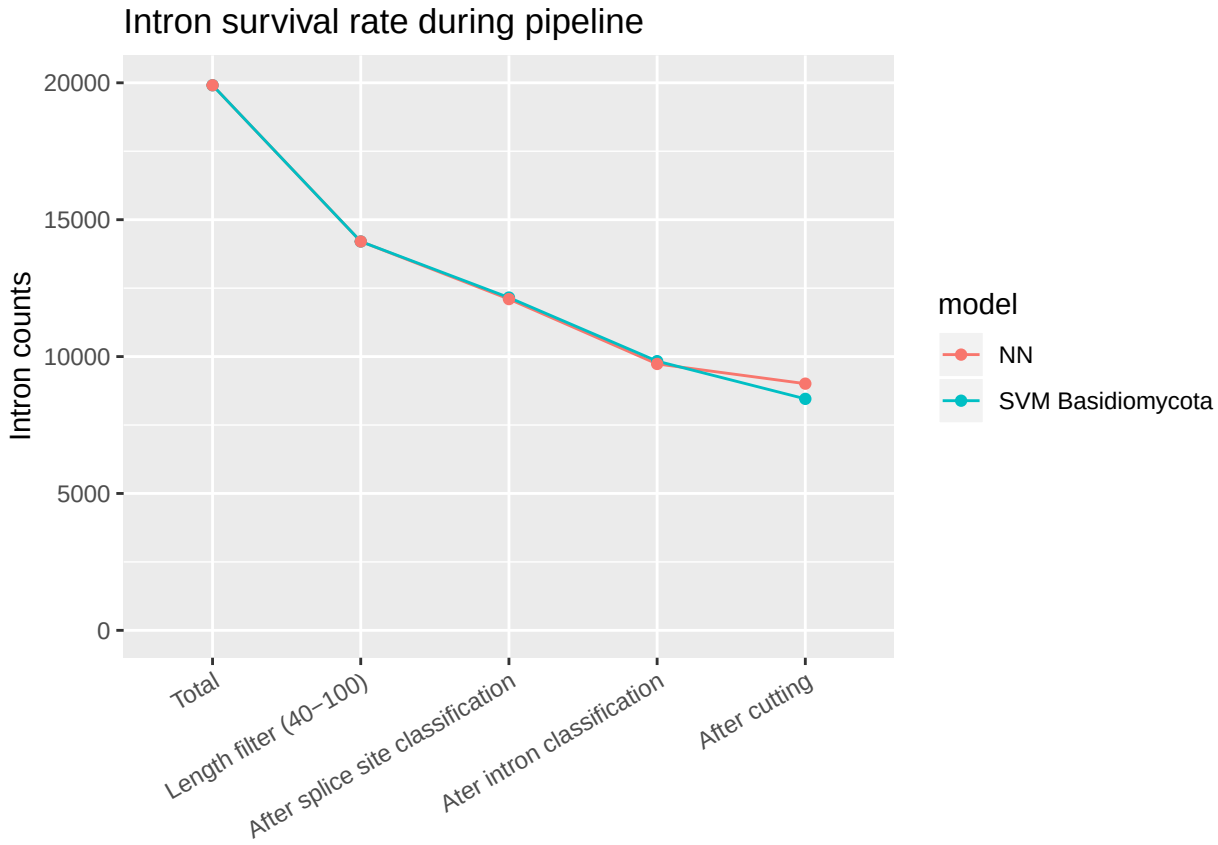


Comparison of SVM and NN models (on *Kocim1* phylum)

```
# Number taken from the output of "intragen/intragen_splice_sites_fp.py" script
kocim.donor.class.metrics <- c(39159, 22171, 3499)
kocim.acceptor.class.metrics <- c(99443, 82582, 29948)
```

Intron classification (*Kocim1*)

Intron cutting and purging is done in steps. Each steps looses a portion of introns due to some filtration or non-perfect recall. Let's visualize, how many introns are lost in each step. The first stage is the number of introns we start with - all introns on the positive strand. Next, we will loose those introns, which are not in the length range of 40-100. Undetected donors and acceptor will also cause some introns to be lost. Intron classification is an apparent source of losses and some are lost during cutting as well. This final decrease is due to incorrect decisions when dealing with candidate overlaps - we simply chose the wrong candidate.

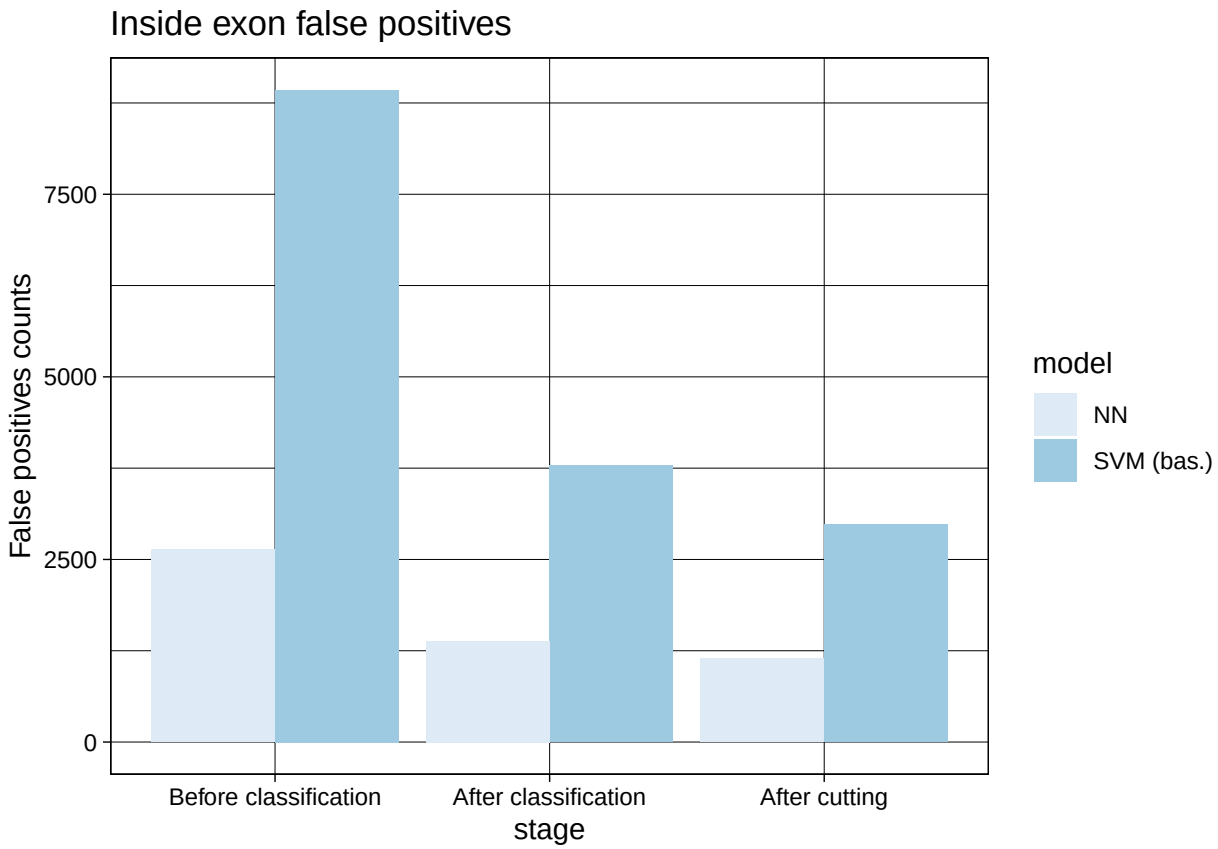



```
kocim.NN.fp.all <- c(20141, 10103, 9495)
kocim.NN.fp.intragen <- c(2640, 1383, 1143)
kocim.NN.fp.intergen <- kocim.NN.fp.all - kocim.NN.fp.intragen

kocim.SVM.fp.all <- c(41001, 18228, 13934)
kocim.SVM.fp.intragen <- c(8927, 3792, 2975)
kocim.SVM.fp.intergen <- kocim.SVM.fp.all - kocim.SVM.fp.intragen

dat_svm <- build_plot_df(kocim.SVM.fp.intergen, kocim.SVM.fp.intragen)
```

```
svm.plot <- ggplot(data=dat_svm, aes(x=stage, y=fp_counts, fill=fp_type)) +
    geom_bar(stat="identity") +
    ggtitle("SVM false positives") +
    ylab("False positives counts") +
    theme(axis.title.x=element_blank(), axis.text.x = element_text(angle = 30, hjust = 1))
# ------------------------------------------------------------------------------------------
dat_nn <- build_plot_df(kocim.NN.fp.intergen, kocim.NN.fp.intragen)

nn.plot <- ggplot(data=dat_nn, aes(x=stage, y=fp_counts, fill=fp_type)) +
    geom_bar(stat="identity") +
    ggtitle("Neural net false positives") +
    ylab("False positives counts") +
    ylim(0,40000) +
    theme(axis.title.x=element_blank(), axis.text.x = element_text(angle = 30, hjust = 1))

ggarrange(svm.plot, nn.plot, nrow=1, common.legend = TRUE, legend="bottom")
```


```
dat3 <- data.frame(
  stage=rep(progress.groups, times=2),
  model=rep(c("SVM (bas.)", "NN"), each=3),
  intragen_fp=c(kocim.SVM.fp.intragen, kocim.NN.fp.intragen)
  )

dat3$stage <- factor(dat3$stage, levels = progress.groups)

exon.fp.plot <- ggplot(data=dat3, aes(x=stage, y=intragen_fp, fill=model)) +
    geom_bar(stat="identity", position = position_dodge()) +
```

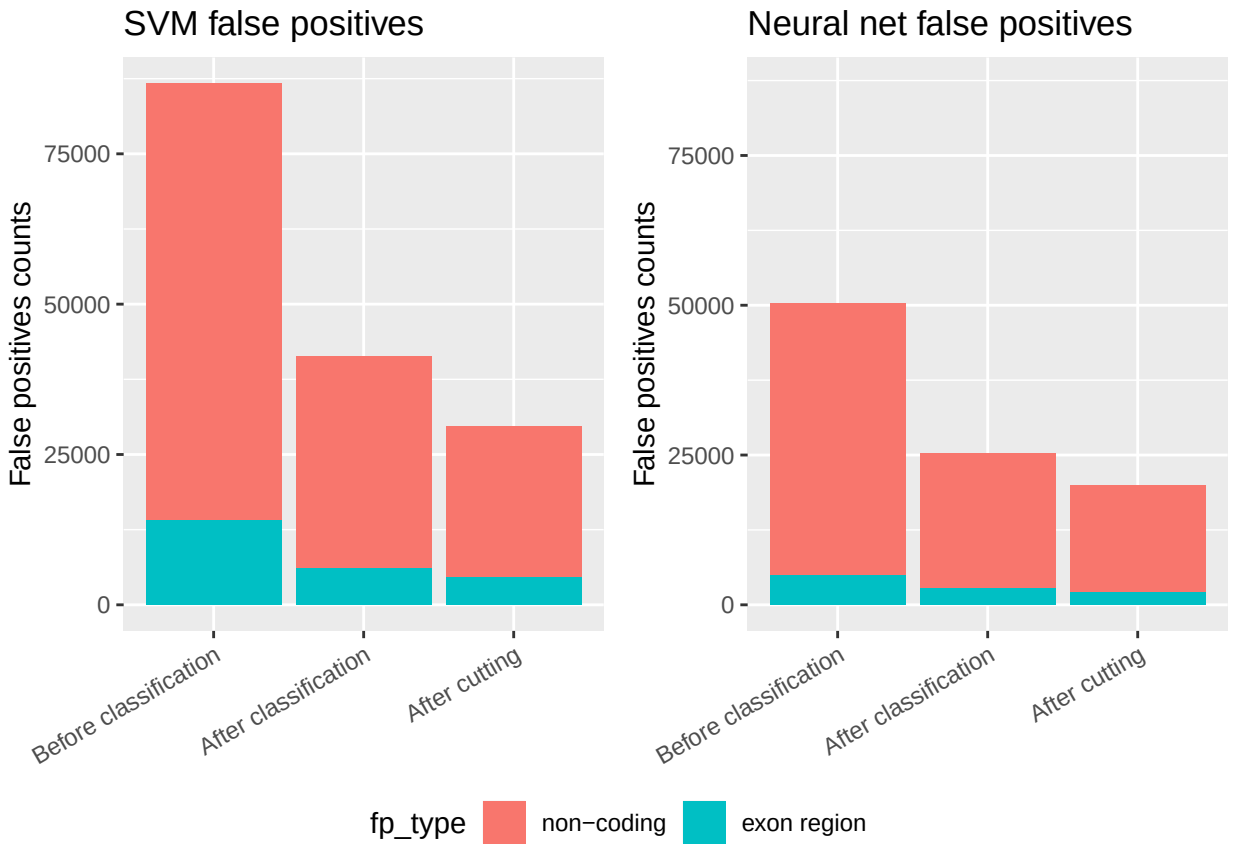
```
    ggtitle("Inside exon false positives") +
    ylab("False positives counts") +
    theme(axis.title.x=element_blank(), axis.text.x = element_text(angle = 30, hjust = 1)) +
    scale_fill_brewer() + theme_linedraw()
exon.fp.plot
```



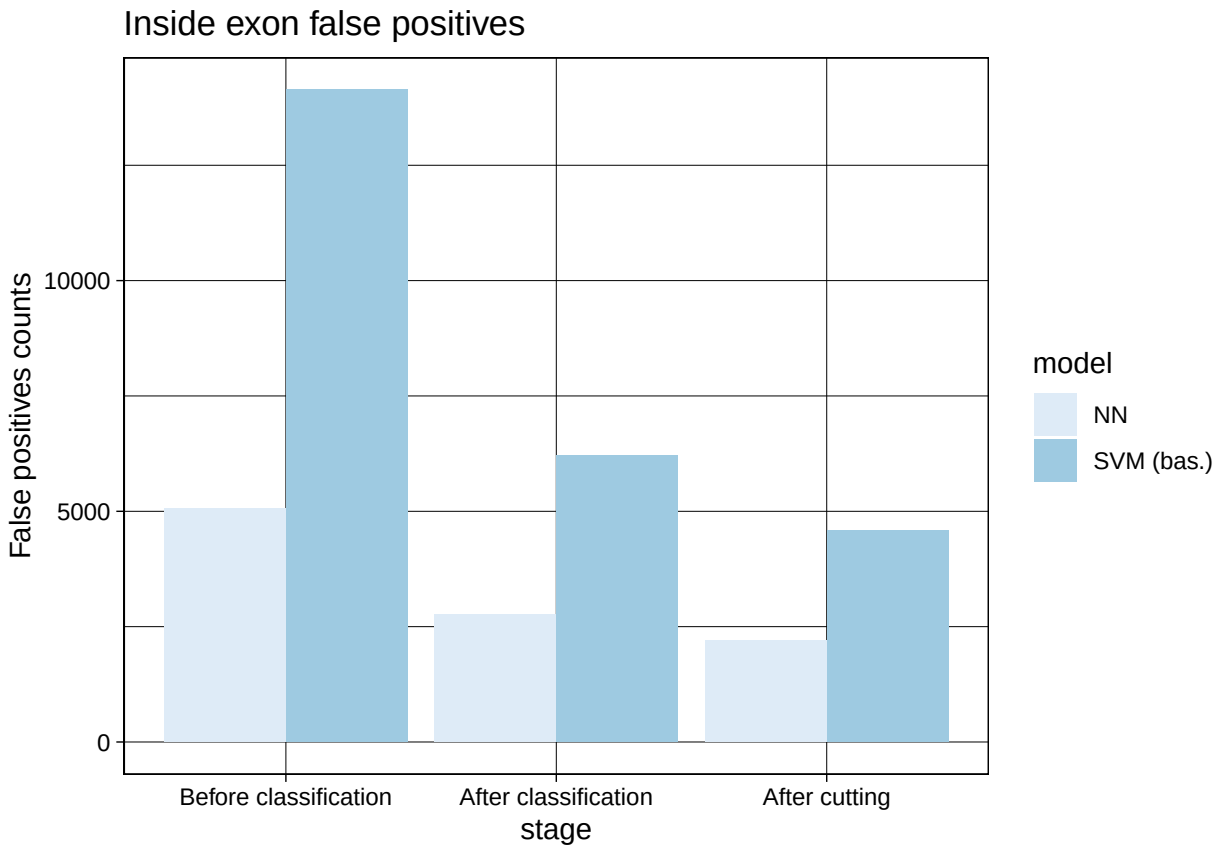
Thega1 experiments

```
thega.NN.fp.all <- c(50331, 25251, 20008)
thega.NN.fp.intragen <- c(5071, 2759, 2195)
thega.NN.fp.intergen <- thega.NN.fp.all - thega.NN.fp.intragen

thega.SVM.fp.all <- c(86685, 41416, 29680)
thega.SVM.fp.intragen <- c(14131, 6204, 4577)
thega.SVM.fp.intergen <- thega.SVM.fp.all - thega.SVM.fp.intragen

dat_svm_thega <- build_plot_df(thega.SVM.fp.intergen, thega.SVM.fp.intragen)

svm.plot <- ggplot(data=dat_svm_thega, aes(x=stage, y=fp_counts, fill=fp_type)) +
    geom_bar(stat="identity") +
    ggtitle("SVM false positives") +
    ylab("False positives counts") +
    theme(axis.title.x=element_blank(), axis.text.x = element_text(angle = 30, hjust = 1))
# ------------------------------------------------------------------------------------------------
dat_nn_thega <- build_plot_df(thega.NN.fp.intergen, thega.NN.fp.intragen)
```

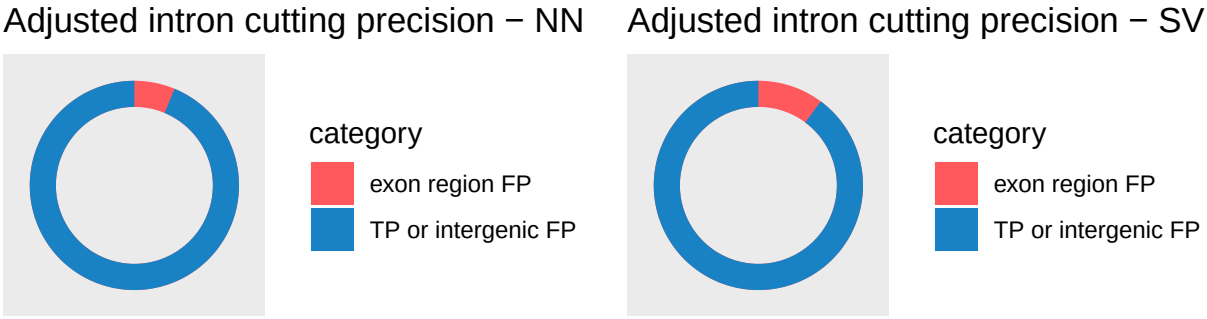
```
nn.plot <- ggplot(data=dat_nn_thega, aes(x=stage, y=fp_counts, fill=fp_type)) +
    geom_bar(stat="identity") +
    ggtitle("Neural net false positives") +
    ylab("False positives counts") +
    ylim(0,87000) +
    theme(axis.title.x=element_blank(), axis.text.x = element_text(angle = 30, hjust = 1))

ggarrange(svm.plot, nn.plot, nrow=1, common.legend = TRUE, legend="bottom")
```




```
dat3 <- data.frame(
  stage=rep(progress.groups, times=2),
  model=rep(c("SVM (bas.)", "NN"), each=3),
  intragen_fp=c(thega.SVM.fp.intragen, thega.NN.fp.intragen)
  )

dat3$stage <- factor(dat3$stage, levels = progress.groups)

exon.fp.plot <- ggplot(data=dat3, aes(x=stage, y=intragen_fp, fill=model)) +
    geom_bar(stat="identity", position = position_dodge()) +
    ggtitle("Inside exon false positives") +
    ylab("False positives counts") +
    theme(axis.title.x=element_blank()) +
    scale_fill_brewer() + theme_linedraw()
exon.fp.plot
```




```
classified_true_data_nn <- c(exon_fp = 2195, total_classified_true =34963)
p_nn <- plot_adjusted_intron_precision_piechart(classified_true_data_nn, "NN")

classified_true_data_svm <- c(exon_fp = 4577, total_classified_true =45015)
p_svm <- plot_adjusted_intron_precision_piechart(classified_true_data_svm, "SVM")

grid.arrange(p_nn, p_svm, ncol = 2)
```



Comparison of models