Response to Reviewer #3 and simulations
=======================================

We sincerely thank the reviewer for his/her clearly thorough and comprehensive critique of our manuscript.

```{r init,echo=FALSE,eval=TRUE,message=FALSE,warning=FALSE}
# Load dependencies
library(ggplot2)
library(reshape2)
library(stringr)
library(dplyr)
library(distr)
#library(cummeRbund)

# Helper functions
JSdist<-function(mat,...){
  res<-matrix(0,ncol=dim(mat)[2],nrow=dim(mat)[2])

#       col_js <- matrix(0,ncol=dim(mat)[2],nrow=1)
#       for(i in 1:dim(mat)[2]){
#           col_js[,i] <- shannon.entropy(mat[,i])
#   }
        col_js<-apply(mat,MARGIN=2,shannon.entropy)
    #print(col_js)
        colnames(res)<-colnames(mat)
        rownames(res)<-colnames(mat)
        for(i in 1:dim(mat)[2]){
                for(j in i:dim(mat)[2]){
                        a<-mat[,i]
                        b<-mat[,j]
                        JSdiv<-shannon.entropy((a+b)/2)-(col_js[i]+col_js[j])*0.5
                        res[i,j] = sqrt(JSdiv)
                        res[j,i] = sqrt(JSdiv)
                }
        }
        res<-as.dist(res,...)
        attr(res,"method")<-"JSdist"
        res
}


JSdistVec<-function(p,q){
  JSdiv<-shannon.entropy((p+q)/2)-(shannon.entropy(p)+shannon.entropy(q))*0.5
        JSdist<-sqrt(JSdiv)
        JSdist
}


makeprobsvec<-function(p){
  phat<-p/sum(p)
        phat[is.na(phat)] = 0
        phat
}

shannon.entropy <- function(p) {
        if (min(p) < 0 || sum(p) <=0)
                return(Inf)
        p.norm<-p[p>0]/sum(p)
        -sum( log10(p.norm)*p.norm)
}

maxSpecificity<-function(p){
  probs<-makeprobsvec(p)
  specs<-c()
  for(i in 1:length(p)){
    q<-rep(0,length(p))
    q[i]<-1
    specs<-c(specs,1-JSdistVec(p,q))
  }
  return (max(specs))
```

}

```

## Data import
```{r readTable,cache=TRUE}
sigGenes<-read.csv("sigGenes.csv")[,c(1:13,16)]
sigGenes.melt<-melt(sigGenes)
sigGenes.melt<-cbind(sigGenes.melt,as.data.frame(str_split_fixed(sigGenes.melt$variable,"_",2)))
colnames(sigGenes.melt)<-c("gene_id","gene_type","condition","fpkm","time","cell")

sigGenes.melt$fpkm<-log10(sigGenes.melt$fpkm+1)

head(sigGenes.melt)
```

Inline response

> Molyneaux et al. present transcriptional profiles for three neuronal cell types (ScPN, CPN, CThPN) at four developmental stages (E15, E16, E18, P1) using a recently published FACS immunostaining approach (Hrvatin, 2014) followed by RNA-seq. In addition to protein coding genes, the authors assembled lncRNA genes and quantified their expression using a previously published pipeline (Cabili, 2011) and conclude that non-coding genes are expressed in a more cell type-specific fashion than protein coding genes and so are particularly important for neuronal specification and maturation.
> I have significant concerns about the results used to justify this claim as well as other aspects of the analysis, and so cannot recommend publication in Neuron.

> I reanalyzed the author's set of differentially expressed genes and found that the observed lncRNA cell type-specificity is an artifact of their low expression levels.
> Although not apparent from the relative expression heatmaps shown in the manuscript, the authors' lncRNA genes are expressed at much lower levels than protein genes (Fig 1. red vs black)

While we acknowledge that the reviewers critiques represent important concerns, we unambiguously disagree with the statement that the observed lncRNA cell type-specificity is an artifact and will attempt to demonstrate below. We do agree with the reviewer that the expression levels of the significant lncRNAs are an order of magnitude lower than those for significant protein coding genes. This was detailed in the manuscript, however, we agree that it is a useful visual point to provide to the reader. To that end, we will include in supplement the following figure describing the distribution of expression estimates for lncRNAs and protein coding genes at each time point.

```{r expression_Barplots,fig.width=10}
p<-ggplot(sigGenes.melt)
p + geom_boxplot(aes(x=time,group=interaction(time,gene_type),y=fpkm,fill=gene_type)) + theme_bw() +
scale_fill_manual(values=c("red","grey40"))
```
This difference is consistent with what has been demonstrated for lncRNAs in numerous prior studies.

> If one creates a sample of protein genes whose expression level distribution matches that of lncRNAs (Fig 1, grey), the apparent difference in cell type specificity (Fig 2. red vs black) almost completely disappears (Fig 2, grey). That is, **lncRNAs have nearly the same cell type specificity as protein coding genes expressed at similar levels**.
> The authors conclusion that lncRNAs play an "outsized role in later aspects of neuronal subtype development" is mistaken because they did not perform this simple control.

We find this approach and the resulting conclusion somewhat disingenuous for several reasons:

1. While the reviewer is correct in principle, there is one important caveat. As (s)he indicates, by sampling protein coding genes from a comparable distribution, the maximum specificity scores do in fact decrease. However, there remains a reproducible difference between the CDFs of the sampled protein coding genes and the bootstrapped lncRNAs; indicated by a visible separation of the estimated 95% confidence intervals between the grey and red lines (Figure 2).
2. The reviewers concern roots from the assumption that expression estimates at the low end are *noise*, and that specificity in this regime is less meaningful as a high specificity score can arise from the abberent transcription of a low abundance gene in only one condition. Yet there are examples of low-abundance protein coding genes with strong specificity that are known to influence this particular cellular system including:
  - Bmp2
  - Cdk2
  - Cdk1
  - Egr2
  - Egr3
```

3. The reviewer is disregarding the fact that all of the significant genes included in this assay were selected
4. An more appropriate test of our claim would be to sample lncRNAs that are in the same regime of expression as protein coding genes and then assess relative specificity (see below).

We take exception to the statement by the reviewer that lncRNAs have nearly the same cell type specificity as protein coding genes expressed at similar levels. In particular, the reviewers own sampling identifies a separation between low-expressing protein coding genes and lncRNAs at each time point, suggesting that even the sampled protein-coding genes are not as specific as the signifcant lncRNAs.

To address this concern, we provide the following analysis:

> Figure 1: Maximum expression level observed across the three neurons at each developmental stage. I sampled from all protein coding genes (black, n=8,058) to obtain a new set of proteins (grey, n=806) whose expression levels matched that of lncRNAs (red, n=806).

Here we will attempt to recreate Fig. 1 from the reviewers simulation

```{r Fig1_recap,fig.width=8,warning=FALSE}
# Find max fpkm and max specificity for each gene at each condition
geneSummary<-sigGenes.melt %.%
  group_by(gene_id,gene_type,time) %.%
  summarize(maxFPKM=max(fpkm),maxSpec=maxSpecificity(fpkm))

# Starting with E15
# Density estimates
E15.lnc.dens<-density(subset(geneSummary,time=="E15" & gene_type!="Protein coding")$maxFPKM)
E15.PC.dens<-density(subset(geneSummary,time=="E15" & gene_type=="Protein coding")$maxFPKM)

# Learn Empirical distribution for lncRNA genes
E15.lnc.D<-DiscreteDistribution(subset(geneSummary,time=="E15" & gene_type!="Protein coding")$maxFPKM)
E15.PC.D<-DiscreteDistribution(subset(geneSummary,time=="E15" & gene_type=="Protein coding")$maxFPKM)

# Create weighted probabilities on PC gene FPKM values from which to sample
PC.weights.on.lnc.D<-p(E15.lnc.D)(subset(geneSummary,time=="E15" & gene_type=="Protein
coding")$maxFPKM,lower.tail=FALSE)
PC.probs.on.lnc.D<-PC.weights.on.lnc.D/sum(PC.weights.on.lnc.D)

#Sample from PC genes to match lncRNA distribution
samp_PC<-sample(subset(geneSummary,time=="E15" & gene_type=="Protein
coding")$maxFPKM,replace=TRUE,size=10000,prob=PC.probs.on.lnc.D)

# My Sampling function
mySample<-function(geneSummarySubset,n,EmpDist){
  w<-p(EmpDist)(geneSummary$maxFPKM,lower.tail=FALSE)
  probs<-w/sum(w)
  samp<-sample(geneSummary$maxFPKM,replace=TRUE,size=n,prob=probs)
  return(samp)
}

# Sanity Check plot
#plot(density(samp_PC),ylim=c(0,2.5),main="")
plot(density(mySample(subset(geneSummary,time=="E15" & gene_type=="Protein
coding"),n=806,EmpDist=E15.lnc.D)),ylim=c(0,2.0))
lines(density(r(E15.lnc.D)(806)),col="blue")
lines(E15.lnc.dens,col="red")
legend(x=1.0,y=2.0,legend=c("lncRNA distribution","Random draws from learned dist","Sampled PC genes
from learned dist"),col=c("red","blue","black"),lty=1)

# Bootstrap
nBoot<-1000

fit1 <- E15.lnc.dens

fit2 <- replicate(nBoot, { x <- mySample(subset(geneSummary,time=="E15" & gene_type!="Protein
coding"),806,E15.lnc.D);
        density(x, from=min(fit1$x), to=max(fit1$x))$y } )

fit3 <- apply(fit2, 1, quantile, c(0.025,0.975))

plot(density(r(E15.lnc.D)(1000)),col="blue", ylim=range(fit3))
polygon( c(fit1$x, rev(fit1$x)), c(fit3[1,], rev(fit3[2,])), col=rgb(.4,.4,.8,0.3), border=FALSE)
```

```
#lines(density(r(E15.lnc.D)(806)),col="blue")
#lines(fit1,col="red")


```

```{r sampling_viz,fig.width=15}
# Weights by bin
p<-ggplot(geneSummary)
p + geom_density(aes(x=maxFPKM,color=gene_type)) + facet_grid(.~time) + theme_bw() +
scale_color_manual(values=c("red","black")) + coord_equal(2)
```

> Figure 2: Maximum specificity observed for all differentially expressed genes (correspond to manuscript Fig 2D). Specificity scores were computed using the author's routines in cummeRbund. 95% confidence intervals are shown, calculated by resampling protein genes (n=1,000) and bootstrapping lncRNA genes (n=1,000).

```{r Fig2_recap,fig.width=10,warning=FALSE,message=FALSE}

p +
geom_point(mapping=aes(x=maxFPKM,y=maxSpec),color="black",alpha=0.3,data=subset(geneSummary,gene_type==
 coding")) +
geom_point(mapping=aes(x=maxFPKM,y=maxSpec),color="red",data=subset(geneSummary,gene_type!="Protein
coding")) + facet_grid(.~time) + theme_bw() + coord_equal(8)

p + geom_smooth(aes(x=maxFPKM,y=maxSpec,color=gene_type,fill=gene_type)) + facet_grid(.~time) +
theme_bw() + scale_color_manual(values=c("red","black")) + scale_fill_manual(values=c("red","grey50"))
+ coord_equal(6)
```

Assuming that gene must have fpkm greater than cutoff at given timepoint
```{r CDF_fpkmMax,fig.width=10,warning=FALSE}
fpkmCutoff<-2

p + stat_ecdf(aes(x=maxSpec,color=gene_type),data=subset(geneSummary,maxFPKM>=fpkmCutoff)) +
facet_grid(.~time) + theme_bw()  + scale_color_manual(values=c("red","black")) + coord_equal(0.4)
```

But this isn't fair because they are assembled by consensus using ALL conditions, so we need to find the genes with fpkm > cutoff in ANY timepoint.
```{r CDF_fpkmMax_any,fig.width=10,warning=FALSE}

fpkmMax<-apply(sigGenes[,c(2:13)],1,max)
fpkmIdx<-fpkmMax>=fpkmCutoff
gene_ids.above.cutoff<-sigGenes$gene_id[fpkmIdx]

p + stat_ecdf(aes(x=maxSpec,color=gene_type),data=geneSummary[geneSummary$gene_id %in%
gene_ids.above.cutoff,]) + facet_grid(.~time) + theme_bw()  +
scale_color_manual(values=c("red","black")) + coord_equal(0.4)
```

> Together with the challenges posed by de novo assembly and quantification of low expressed genes (Steijger et al., Nat Methods 2013), the above control forces a more cautious interpretation of the authors results: new lncRNA calls have a significant error rate and their low expression levels are driving the apparent cell type specificity. These effects preclude the assignment of "critical roles in developmental patterning" to lncRNAs.

> More generally, the authors do not describe or quantify error modes throughout the study. For example:

> 1. What is the contamination level of the purification procedure? The authors mention in passing that 5 glial and inhibitory neuron markers are not expressed in the purified samples, but they neither quantify this observation nor give a sense of the estimated contamination level.
>
>   For example, the authors claim the Gad1 level is only concerning in E15 CPN (7.5 FPKM), and explain that this is likely due to migrating interneurons. However, Gad1 also expresses at 2.2 FPKM at E16 CThPN, 2.3 in E18 ScPN, and 1.9 in P1 CThPN; these levels are within the range of their lncRNA calls (Fig 1 above).
>
>   A more worrying example, not described by the authors, is the interneuron marker somatostatin (Sst) estimated at 18.4 FPKM in P1 CThPN and 12.3 in E15 CPN.

> 2. What is the lower level of detection of the author's RNA-seq procedure, particularly in the context of contamination in the purification procedure? That is, at what FPKM level do the authors stop believing the estimates?
> 3. The authors do not discuss error modes and rates of their lncRNA assembly strategy. This information is critical for interpreting the paper's results, especially as transcript assembly is a hard problem (Steijger, 2013).
> 4. How well do the estimated lncRNA and protein expression levels correlate between replicates? Given their expression levels, I suspect the precision of lncRNA calls is significantly poorer than protein levels.
> 5. The authors approach to predicting transcriptional regulators of gene co-expression clusters is not described in sufficient detail. These results are based on a series of predictions built on top of other predictions. Error modes and rates are not described for any of these steps. Predicting regulatory motifs from expression data is notoriously difficult and error-prone.
>
> Their observed results are consistent with random expectation. For example, the authors find that 42% of their predicted significant motifs have differential expression support. This value is in fact less than expected by chance, as the authors list of differentially expressed genes contains 48% of all annotated transcription factors (668 of 1,378 TFs in AnimalTFDB).

> The authors claim that the remaining predictions that lack differential expression support "may have influence over cluster-specific genes at other developmental times or other cellular contexts". A more likely explanation is a high error rate in the predicted binding sites.

Conclusions

Session Information
```{r sessionInfo()}
sessionInfo()
```