

# Analyses for JSLHR version

2020-10-17

## Contents

History: . . . . .	1
TODO: . . . . .	1
Read data in . . . . .	1
Data post-processing . . . . .	2
Results . . . . .	4
Descriptive analyses . . . . .	4
Main analyses . . . . .	7
Child level descriptors . . . . .	8
Additional analyses . . . . .	11

## History:

- 2020-08-05 final first version
- 2020-10-10 (rehaul),
- latest minor edits 2020-10-17

## TODO:

- pipeline is still not transparent
- there are duplicate files across raw and derived data
- there are a bunch of files with similar names
- README is old
- a note said read demo data created by AC from info in paper - should be replaced with real demo data
- add code to print out the results paragraphs
- corage: - adjust margins and remove title repetition?
- chunk-based analyses for age

## Read data in

```
# read datasets

demo_data=read.csv("../Derived_Data/demo-data.tsv",sep="\t")
data_ang <- read.csv("../Derived_Data/classifications_PU_zoon_final17.csv",header=T,sep=",")
data_td <- read.csv("../Derived_Data/classifications_PU_zoon_final.csv")
data_all<-rbind(data_ang, data_td)

#add filenames to demo data
demo_data_fn <- demo_data %>%
  left_join(select(data_all, filename, ChildID), by = c("ChildID"))
```

```

demo_data_fn<-unique(demo_data_fn)

#remove the word mixed that takes up space and is unnecessary
data_all$Zoon_classif=factor(gsub("Mixed_", "", as.character(data_all$Zoon_classif), fixed=T))
#relevel the factor so that it's easier to read
data_all$Zoon_classif=factor(data_all$Zoon_classif, levels=c("Canonical", "Non-Canonical",
# create lab column with easier to read correspondance
data_all$lab<-as.character(data_all$Major_Choice)
data_all$lab[data_all$lab=="Non-canonical syllables"]<-"Non-Canonical"
data_all$lab[data_all$lab=="Canonical syllables"]<-"Canonical"
data_all$lab[data_all$lab %in% c("Don't mark", "None")]<-"Junk"
data_all$lab=factor(data_all$lab, levels=levels(data_all$Zoon_classif))
#apply same factor levels as zooniverse so that we can do symmetrical confusion matrices

#add binomials for linguistic proportion
data_all$lab_ling=ifelse(data_all$lab %in% c("Canonical", "Non-Canonical"), 1, 0)
data_all$zoo_ling=ifelse(data_all$Zoon_classif %in% c("Canonical", "Non-Canonical"), 1, 0)
data_all$lab_ling[data_all$lab=="Junk"]<-NA
data_all$zoo_ling[data_all$lab=="Junk"]<-NA

#add binomials for canonical proportion
data_all$lab_can=data_all$zoo_can=NA
data_all$lab_can[data_all$lab=="Canonical"]<-1
data_all$lab_can[data_all$lab=="Non-Canonical"]<-0
data_all$zoo_can[data_all$Zoon_classif=="Canonical"]<-1
data_all$zoo_can[data_all$Zoon_classif=="Non-Canonical"]<-0

```

## Data post-processing

We collected a total of 169,628 judgments provided for 33,880 500-ms chunks, corresponding to 11,984 LENA segments. Nearly a fifth of chunks did not have at least 3 labels in agreement out of the 5 Zooniverse labels ( $N = 6,585$ , 19% of all chunks). Of the chunks without a majority agreement, 4341 (66%) contained one or two Junk judgements (out of 5), 6523 (99.9%) had at least two matching judgements (the threshold used for lab-annotated segments), and only 61 (0.01%) had 5 different judgements. Future work may explore different ways of setting the minimal requirement for convergence, but for further analyses here, we focused on the 81% of chunks that did have at least 3 labels in agreement; this represented 136,703 labels for 27,295 chunks, corresponding to 11,593 LENA segments. As the segments average 1.12 seconds in length, this means about 3.8 hours of audio data were annotated by 8 different annotators (3 in the laboratory, 5 on Zooniverse).

```

# remove non-majority labels
#TODO

# we map the mixed
data_all$Zoon_classif[data_all$Zoon_classif=="Laughing_Canonical"]<-"Canonical"
data_all$Zoon_classif[data_all$Zoon_classif=="Laughing_Non-Canonical"]<-"Non-Canonical"
data_all$Zoon_classif[data_all$Zoon_classif=="Laughing_Non-Canonical_Crying"]<-"Non-Canonical"
data_all$Zoon_classif[data_all$Zoon_classif=="Laughing_Crying"]<-"Crying"
data_all$Zoon_classif[data_all$Zoon_classif=="Non-Canonical_Crying"]<-"Non-Canonical"
data_all$Zoon_classif[data_all$Zoon_classif=="Non-Canonical_Laughing_Crying"]<-"Non-Canonical"
data_all$Zoon_classif[data_all$Zoon_classif=="Crying_Canonical"]<-"Canonical"
# +
data_all$Zoon_classif[data_all$Zoon_classif=="Canonical_Crying"]<-"Canonical"
data_all$Zoon_classif[data_all$Zoon_classif=="Canonical_Laughing"]<-"Canonical"

```

```

data_all$Zoon_classif[data_all$Zoon_classif=="Laughing_Canonical_Crying"]<-"Non-Canonical"

data_all$Zoon_classif[data_all$Zoon_classif=="Crying_Laughing"]<-"Crying"
data_all$Zoon_classif[data_all$Zoon_classif=="Crying_Canonical_Laughing"]<-"Canonical"
data_all$Zoon_classif[data_all$Zoon_classif=="Crying_Laughing_Non-Canonical"]<-"Non-Canonical"
data_all$Zoon_classif[data_all$Zoon_classif=="Crying_Non-Canonical"]<-"Non-Canonical"
data_all$Zoon_classif[data_all$Zoon_classif=="Crying_Non-Canonical_Laughing"]<-"Non-Canonical"
data_all$Zoon_classif[data_all$Zoon_classif=="Non-Canonical_Laughing"]<-"Non-Canonical"

#and reset the factors for cleanliness
data_all$Zoon_classif=factor(data_all$Zoon_classif)
data_all$lab=factor(data_all$lab)
sample_data<-cbind(data_all$lab,data_all$Zoon_classif)

table_zoon=round(table(data_all$Zoon)/sum(table(data_all$Zoon))*100)
table_lab=round(table(data_all$lab)/sum(table(data_all$lab))*100)

#get the ns by child, then calculate the linguistic ratio & canonical ratio, separately for zooniverse
ztab=table(data_all$filename,data_all$Zoon_classif)
z_lp=rowSums(ztab[,c("Canonical", "Non-Canonical")])/rowSums(ztab[,which(colnames(ztab) %in% c("Junk"))])
z_cp=ztab[,c("Canonical")]/rowSums(ztab[,c("Canonical", "Non-Canonical")])

ltab=table(data_all$filename,data_all$lab)
l_lp=rowSums(ltab[,c("Canonical", "Non-Canonical")])/rowSums(ltab[,which(colnames(ztab) %in% c("Junk"))])
l_cp=ltab[,c("Canonical")]/rowSums(ltab[,c("Canonical", "Non-Canonical")])

#and also get junk from zooniverse
z_junk=ztab[,c("Junk")]/rowSums(ztab)

#put all the proportions together
if(sum(rownames(ztab)==rownames(ltab))==dim(ztab)[1]) proportions=cbind(rownames(ztab),z_lp,z_cp,l_lp,l_cp,z_junk)
colnames(proportions)[1]<-"filename"

#ages=aggregate(data_all$Age,by=list(data_all$ChildID),mean) #this is a weird way of adding ages, since
#improvement: now we merge with a demo data tab, but note this is merged with child id, so the problem

# Created demo_data with filenames. Use filenames instead of childIDs to merge proportions and demo data

merge(proportions,demo_data_fn,by="filename")->proportions
colnames(proportions)[dim(proportions)[2]]<-"Age"

#cbinding results in text, so we numerize the proportions
for(thisvar in c("z_lp","z_cp","l_lp","l_cp","z_junk")) proportions[,thisvar]=as.numeric(as.character(proportions[,thisvar]))

#add median junk
proportions$median_junk<-ifelse(proportions$z_junk<median(proportions$z_junk),"lower_junk","higher_junk")

summary(proportions)

```

```

##          filename      z_lp      z_cp
## 20180206_110905_009463: 1  Min.   :0.5927  Min.   :0.02882
## 20180419_111712_022875: 1  1st Qu.:0.8058  1st Qu.:0.05891
## 20180530_180405_024879: 1  Median :0.8785  Median :0.14065

```

```
## 20180530_181101_022875: 1 Mean :0.8511 Mean :0.19128
## 20180808_111325_024882: 1 3rd Qu.:0.9313 3rd Qu.:0.28454
## 20180906_133011_022875: 1 Max. :0.9740 Max. :0.51084
## (Other) :14
## l_lp l_cp z_junk ChildID
## Min. :0.3880 Min. :0.01488 Min. :0.04329 1111_1 : 1
## 1st Qu.:0.9130 1st Qu.:0.06609 1st Qu.:0.09418 1151_1 : 1
## Median :0.9578 Median :0.11115 Median :0.12230 1801_1 : 1
## Mean :0.9131 Mean :0.19129 Mean :0.13302 2881_1 : 1
## 3rd Qu.:0.9787 3rd Qu.:0.26529 3rd Qu.:0.16368 2931_1 : 1
## Max. :0.9968 Max. :0.60000 Max. :0.27706 2991_1 : 1
## (Other):14
## Diagnosis Sex Age median_junk
## AngelmanSyndrome:10 F: 7 Min. : 4.67 Length:20
## Low-RiskControl :10 M:13 1st Qu.:12.13 Class :character
## Median :17.50 Mode :character
## Mean :25.04
## 3rd Qu.:43.46
## Max. :53.26
##
```

```
merge(data_all,proportions,all=T)->data_all
```

## Results

### Descriptive analyses

In this section, we provide descriptive analyses of our dataset. According to lab annotators, 15% of segments were canonical, 56% non-canonical, 2% laughing, and 5% crying, with the remaining 22% being categorized as “Don’t code”. Zooniverse data revealed a similar distribution: 15% canonical, 60% non-canonical, 4% laughing, 9% crying, 13% junk. Next, we inspected the relationship between age and child-level derived metrics, of which we had two: i) Linguistic proportion = (“Canonical”+“Non-Canonical”)/“All vocalizations” (i.e., we remove junk), and ii) Canonical proportion = “Canonical”/(“Canonical”+“Non-Canonical”) (i.e. we remove junk + non-linguistic vocalizations). See Figure XX for results.

Figure XX. Correlations between child-level descriptors and age as a function of metric (linguistic ratio in the top row, canonical ratio in the bottom row), annotation method, and child group (Red as TD, and black as AS)

```
prettynames=c("Linguistic Proportion (Zooniverse)","Canonical Proportion (Zooniverse)",
              "Linguistic Proportion (Lab)","Canonical Proportion (Lab)" )
names(prettynames)<-c("z_lp","z_cp","l_lp","l_cp")
mycols=c("black","red")
names(mycols)<-c("Low-RiskControl","AngelmanSyndrome")
mypch=c(4,20)
names(mypch)<-c("Low-RiskControl","AngelmanSyndrome")

lp_range=range(proportions[,c("z_lp","l_lp")])
cp_range=range(proportions[,c("z_cp","l_cp")])

#jpeg("../Results/corage.jpg",width=20,height=20,units="cm",res=300)
layout(matrix(c(1:4), 2, 2, byrow = F))
par(mar = c(4, 4, 1, 1))
```

```

#create matrix to hold r results
myr=matrix(NA,nrow=4,ncol=2)
rownames(myr)<-c("z_lp","z_cp","l_lp","l_cp")
colnames(myr)<-c("AS","LR")
mylb=myhb=myn=myp=myr #this is an ugly way of creating 5 matrices

for(thisvar in c("z_lp","z_cp","l_lp","l_cp")) {

  if(thisvar %in% c("z_lp","l_lp")) myrange=lp_range else myrange=cp_range

  plot(proportions[,thisvar]~proportions$Age, pch=mypch[proportions$Diagnosis],xlab="Age (months)",ylab="Proportion",
        col=mycols[proportions$Diagnosis],
        ylim=myrange)
  abline(lm(proportions[,thisvar]~proportions$Age,subset=c(proportions$Diagnosis=="AngelmanSyndrome")),lty=2)

  #get cor properties
  thiscor=cor.test(proportions[proportions$Diagnosis=="AngelmanSyndrome",thisvar],proportions$Age[proportions$Diagnosis=="AngelmanSyndrome"])
  myr[thisvar,"AS"]=round(thiscor$estimate,3)
  mylb[thisvar,"AS"]=round(thiscor$conf.int[1],3)
  myhb[thisvar,"AS"]=round(thiscor$conf.int[2],3)
  myn[thisvar,"AS"]=round(thiscor$parameter,3)
  myp[thisvar,"AS"]=round(thiscor$p.value,3)

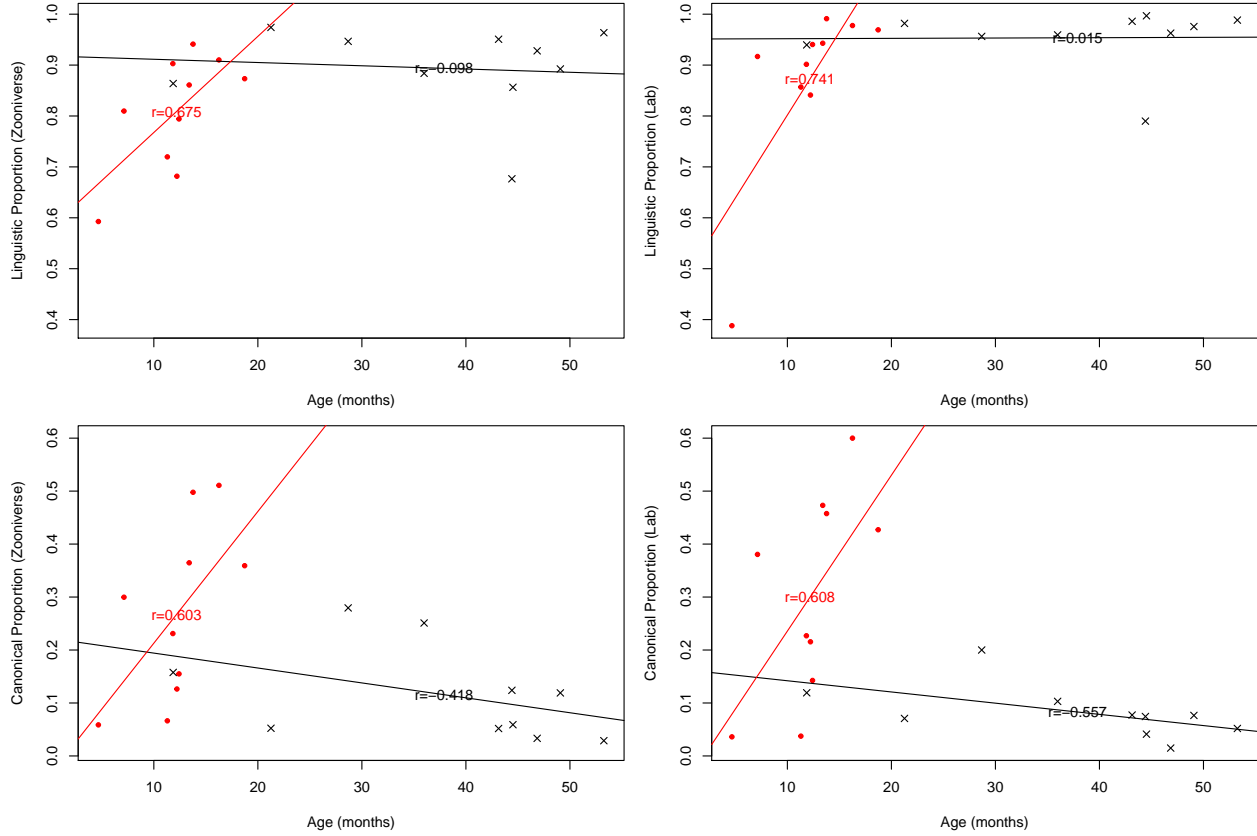
  #add r to figure
  text(mean(proportions$Age[proportions$Diagnosis=="AngelmanSyndrome"]),mean(proportions$Age[proportions$Diagnosis=="AngelmanSyndrome"]),
        myr[thisvar,"AS"],mylb[thisvar,"AS"],myhb[thisvar,"AS"],myn[thisvar,"AS"],myp[thisvar,"AS"],cex=0.8)

  #repeat for LR
  abline(lm(proportions[,thisvar]~proportions$Age,subset=c(proportions$Diagnosis!="AngelmanSyndrome")),lty=2)

  #get cor properties
  thiscor=cor.test(proportions[proportions$Diagnosis!="AngelmanSyndrome",thisvar],proportions$Age[proportions$Diagnosis!="AngelmanSyndrome"])
  myr[thisvar,"LR"]=round(thiscor$estimate,3)
  mylb[thisvar,"LR"]=round(thiscor$conf.int[1],3)
  myhb[thisvar,"LR"]=round(thiscor$conf.int[2],3)
  myn[thisvar,"LR"]=round(thiscor$parameter,3)
  myp[thisvar,"LR"]=round(thiscor$p.value,3)

  #add r to figure
  text(mean(proportions$Age[proportions$Diagnosis!="AngelmanSyndrome"]),mean(proportions$Age[proportions$Diagnosis!="AngelmanSyndrome"]),
        myr[thisvar,"LR"],mylb[thisvar,"LR"],myhb[thisvar,"LR"],myn[thisvar,"LR"],myp[thisvar,"LR"],cex=0.8)
}

```



*#dev.off()*

Descriptive analyses on the laboratory annotations showed that correlations between the Linguistic proportion and age differed across the groups. There was a near-zero relationship among the older children diagnosed with Angelman Syndrome  $r(8) = 0.015$ , CI  $[-0.621, 0.638]$ ,  $p=0.968$ ; and a significant association among younger low-risk control children  $r(8) = 0.741$ , CI  $[0.209, 0.935]$ ,  $p=0.014$ . The Canonical proportion exhibited non-significant developmental decreases among older children diagnosed with Angelman Syndrome  $r(8) = -0.557$ , CI  $[-0.879, 0.112]$ ,  $p=0.094$ ; and marginal developmental increases among low-risk control  $r(8) = 0.608$ , CI  $[-0.035, 0.895]$ ,  $p=0.062$ .

Using the Zooniverse annotations, we found that the association with age was very weak for children diagnosed with Angelman Syndrome  $r(8) = -0.098$ , CI  $[-0.685, 0.567]$ ,  $p=0.788$ ; whereas younger low-risk control children showed a significant increase with age  $r(8) = 0.675$ , CI  $[0.079, 0.916]$ ,  $p=0.032$ . Similarly, there were non-significant developmental decreases in the Canonical among children with Angelman Syndrome  $r(8) = -0.418$ , CI  $[-0.829, 0.288]$ ,  $p=0.23$ ; and marginal developmental increases among low-risk control children,  $r(8) = 0.603$ , CI  $[-0.043, 0.893]$ ,  $p=0.065$ .

Given that these derived metrics are defined at the child level, there was one stage in our pre-processing that may not have been necessary, whereby we collapsed judgments across chunks associated to the same segment. We therefore repeated our analyses but deriving our proportions for the Zooniverse data not from the segment-level composite, but rather the individual chunk-level annotations. We found that the Linguistic proportion was not strongly associated with age in the Angelman Syndrome group  $r(8) = -.04$ , CI  $[-.6, .6]$ ,  $p=.91$ , but increased with age in the low-risk control group  $r(8) = .71$ , CI  $-.15, .92$ ,  $p=.02$ . Similarly, the Canonical proportion did not exhibit the same pattern across the groups, with developmental decreases found among children with Angelman Syndrome  $r(8) = -.35$ , CI  $-.80, .35$ ,  $p=.32$ ; and developmental increases among low-risk control  $r(8) = .57$ , CI  $-.08, .88$ ,  $p=.83$ .

## Main analyses

Next, we discuss the correspondence between citizen science classifications and the laboratory gold standard, at the level of individual clips. Results were visualized and assessed with a confusion matrix. We report a Precision plot (Fig. 1) and a Recall plot (Fig. 2): the diagonal elements show the number of correct segment-level classifications for each class while the off-diagonal elements show non-matching classifications.

```
mygac=gac(data = sample_data, kat = 5, weight = c("unweighted"),
  conf.level = 0.95)

mycf=confusionMatrix(data_all$lab, data_all$Zoon_classif, dnn = c("Lab","Zooniverse"))
conf_tab=mycf$table
# this package uses sensitivity & specificity
#Sensitivity=recall
#Specificity=precision
#mycf

colsums=colSums(conf_tab)
my_conf_tab=conf_tab
for(i in 1:dim(conf_tab)[2]) my_conf_tab[,i]=my_conf_tab[,i]/colsums[i]
colSums(my_conf_tab)

##      Canonical Non-Canonical      Crying      Laughing      Junk
##      1          1          1          1          1

prop_cat=data.frame(my_conf_tab*100) #generates precision because columns
prop_cat$id=paste(prop_cat$Lab,prop_cat$Zooniverse)
colnames(prop_cat)[3]<-"pr"
data.frame(conf_tab)->stall
stall$id=paste(stall$Lab,stall$Zooniverse)
stall=merge(stall,prop_cat[c("id","pr")])
prec_plot = ggplot(data = stall, mapping = aes(y = Lab, x=Zooniverse)) +
  geom_tile(aes(fill= rescale(pr)), colour = "white") +
  geom_text(aes(label = paste(round(pr,"%")), vjust = -1,size=2) +
  geom_text(aes(label = Freq, vjust = 1,size=1) +
  scale_fill_gradient(low = "white", high = "red", name = "Percentage") +
  theme(legend.position = "none") +
  xlab("Zooniverse") + ylab("Lab") +
  ggtitle("Precision")+ theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

# repeat for recall
rowsums=rowSums(conf_tab)
my_conf_tab=conf_tab
for(i in 1:dim(conf_tab)[2]) my_conf_tab[,i]=my_conf_tab[,i]/rowsums[i]
rowSums(my_conf_tab)

##      Canonical Non-Canonical      Crying      Laughing      Junk
##      0.9213192      2.8808609      0.5824913      0.6608792      1.6885824

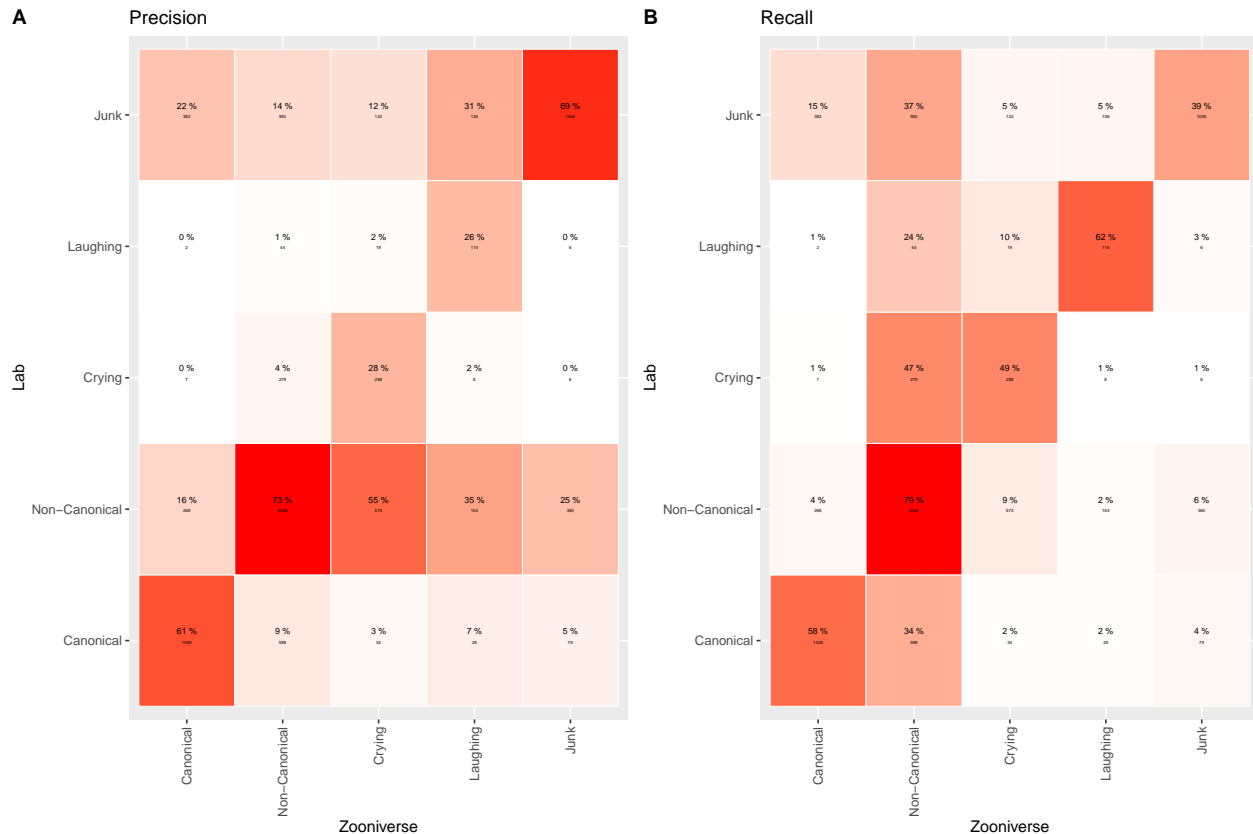
prop_cat=data.frame(conf_tab/rowSums(conf_tab)*100) #generates recall because rows
prop_cat$id=paste(prop_cat$Lab,prop_cat$Zooniverse)
colnames(prop_cat)[3]<-"rec"
data.frame(conf_tab)->stall
stall$id=paste(stall$Lab,stall$Zooniverse)
stall=merge(stall,prop_cat[c("id","rec")])
rec_plot = ggplot(data = stall, mapping = aes(y = Lab, x=Zooniverse)) +
```

```

geom_tile(aes(fill= rescale(rec)), colour = "white") +
geom_text(aes(label = paste(round(rec), "%")), vjust = -1, size=2) +
geom_text(aes(label = Freq), vjust = 1, size=1) +
scale_fill_gradient(low = "white", high = "red", name = "Percentage") +
theme(legend.position = "none") +
xlab("Zooniverse") + ylab("Lab") +
ggtitle("Recall") + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

ggarrange(prec_plot, rec_plot,
  labels = c("A", "B"),
  ncol = 2, nrow = 1)

```



From both visualizations, it appears that performance is moderate to good, with an overall accuracy of 65%, CI = [64,66], a kappa of 0.426, and a Gwet's AC1 coefficient of 0.587, CI = [0.576,0.597].

## Child level descriptors

```

layout(matrix(c(1:2), 1, 2, byrow = F))
par(mar = c(4, 4, 1, 1))

#create matrix to hold r results
myr=matrix(NA,nrow=2,ncol=3)
rownames(myr)<-c("lp","cp")
colnames(myr)<-c("AS","LR","all")
mylb=myhb=myn=myp=myr #this is an ugly way of creating 5 matrices

```



```

lims=range(c(proportions[, "z_lp"], proportions[, "l_lp"]))

plot(proportions[, "z_lp"]~proportions[, "l_lp"], xlab=prettynames["l_lp"], ylab=prettynames["z_lp"],
      xlim=lims, ylim=lims,
      col=mycols[proportions$Diagnosis], pch=mypch[proportions$Diagnosis])
lines(c(0,1), c(0,1), lty=2, col="darkgray")

abline(lm(proportions[, "z_lp"]~proportions[, "l_lp"], subset=c(proportions$Diagnosis=="AngelmanSyndrome"))

#get cor properties
thiscor=cor.test(proportions[proportions$Diagnosis=="AngelmanSyndrome", "z_lp"], proportions[proportions$Diagnosis=="AngelmanSyndrome", "l_lp"])
myr[1, "AS"]=round(thiscor$estimate, 3)
mylb[1, "AS"]=round(thiscor$conf.int[1], 3)
myhb[1, "AS"]=round(thiscor$conf.int[2], 3)
myn[1, "AS"]=round(thiscor$parameter, 3)
myp[1, "AS"]=round(thiscor$p.value, 3)

#repeat for LR
abline(lm(proportions[, "z_lp"]~proportions[, "l_lp"], subset=c(proportions$Diagnosis!="AngelmanSyndrome"))

#get cor properties
thiscor=cor.test(proportions[proportions$Diagnosis!="AngelmanSyndrome", "z_lp"], proportions[proportions$Diagnosis!="AngelmanSyndrome", "l_lp"])
myr[1, "LR"]=round(thiscor$estimate, 3)
mylb[1, "LR"]=round(thiscor$conf.int[1], 3)
myhb[1, "LR"]=round(thiscor$conf.int[2], 3)
myn[1, "LR"]=round(thiscor$parameter, 3)
myp[1, "LR"]=round(thiscor$p.value, 3)

#repeat for all cor

thiscor=cor.test(proportions$z_lp, proportions$l_lp)
myr[1, "all"]=round(thiscor$estimate, 3)
mylb[1, "all"]=round(thiscor$conf.int[1], 3)
myhb[1, "all"]=round(thiscor$conf.int[2], 3)
myn[1, "all"]=round(thiscor$parameter, 3)
myp[1, "all"]=round(thiscor$p.value, 3)

# repeat whole for cp

lims=range(c(proportions[, "z_cp"], proportions[, "l_cp"]))

plot(proportions[, "z_cp"]~proportions[, "l_cp"], xlab=prettynames["l_cp"], ylab=prettynames["z_cp"],
      xlim=lims, ylim=lims,
      col=mycols[proportions$Diagnosis], pch=mypch[proportions$Diagnosis])
lines(c(0,1), c(0,1), lty=2, col="darkgray")

abline(lm(proportions[, "z_cp"]~proportions[, "l_cp"], subset=c(proportions$Diagnosis=="AngelmanSyndrome"))

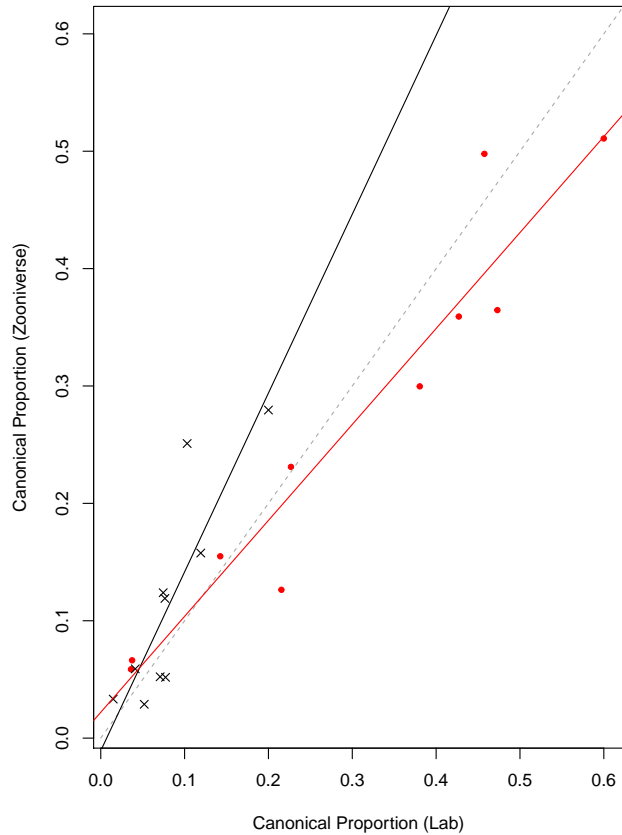
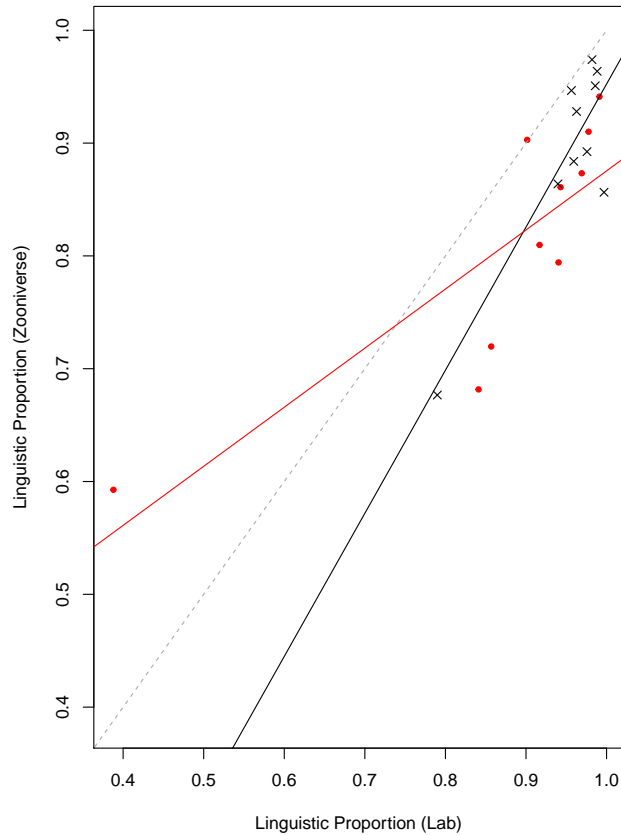
#get cor properties
thiscor=cor.test(proportions[proportions$Diagnosis=="AngelmanSyndrome", "z_cp"], proportions[proportions$Diagnosis=="AngelmanSyndrome", "l_cp"])
myr[2, "AS"]=round(thiscor$estimate, 3)
mylb[2, "AS"]=round(thiscor$conf.int[1], 3)

```

```
myhb[2, "AS"] = round(thiscor$conf.int[2], 3)
myn[2, "AS"] = round(thiscor$parameter, 3)
myp[2, "AS"] = round(thiscor$p.value, 3)
```

*#repeat for LR*

```
abline(lm(proportions[, "z_cp"] ~ proportions[, "l_cp"], subset=c(proportions$Diagnosis != "AngelmanSyndrome"))
```



*#get cor properties*

```
thiscor = cor.test(proportions[proportions$Diagnosis != "AngelmanSyndrome", "l_cp"], proportions[proportions$Diagnosis != "AngelmanSyndrome", "z_cp"])
myr[2, "LR"] = round(thiscor$estimate, 3)
mylb[2, "LR"] = round(thiscor$conf.int[1], 3)
myhb[2, "LR"] = round(thiscor$conf.int[2], 3)
myn[2, "LR"] = round(thiscor$parameter, 3)
myp[2, "LR"] = round(thiscor$p.value, 3)
```

*#repeat for all cor*

```
thiscor = cor.test(proportions$z_cp, proportions$l_cp)
myr[2, "all"] = round(thiscor$estimate, 3)
mylb[2, "all"] = round(thiscor$conf.int[1], 3)
myhb[2, "all"] = round(thiscor$conf.int[2], 3)
myn[2, "all"] = round(thiscor$parameter, 3)
myp[2, "all"] = round(thiscor$p.value, 3)
```

Although the classification at the clip level is only moderately accurate, what we are ultimately interested in is whether citizen scientists' classifications are able to provide a reliable snapshot of childrens' individual

development. Looking at all 20 children together, we found a strong positive correlation  $r(18) = 0.811$ , CI [0.575,0.923],  $p=0$ ] between Linguistic proportion by child from the Zooniverse and the lab annotators' data. When we split by participant group, correlations remain high: for Angelman Syndrome  $r(8) = 0.877$ , CI [0.552,0.971],  $p=0.001$ ; low-risk control  $r(8) = 0.823$ , CI [0.401,0.957],  $p=0.003$ .

Similarly, a strong positive correlation is found in the Canonical Proportion  $r(18) = 0.937$ , CI [0.844,0.975],  $p=0$ . When we split by participant group, correlations remain high although we do note they are somewhat smaller for the older children with Angelman Syndrome:  $r(8) = 0.858$ , CI [0.498,0.966],  $p=0.001$ ; than the low-risk control children  $r(8) = 0.96$ , CI [0.833,0.991],  $p=0$ .

When we use chunk-level judgments to derive the our child-level metrics, we observe very similar levels of correlation: linguistic proportion overall  $r(18) = .84$ , CI .63 .93,  $p<0.001$  [Angelman Syndrome  $r(8) = .82$ , CI .41 .95,  $p<0.01$ ; low-risk control  $r(8) = .84$ , CI .44 0.96,  $p<0.01$ ]; canonical proportion overall  $r(18) = .96$ , CI .90 .98,  $p<0.001$  [Angelman Syndrome  $r(8) = .85$ , CI .49 .85,  $p<0.01$ ; low-risk control  $r(8) = .97$ , CI .89 .99,  $p<0.001$ ].

```
# one outlier in ling prop is removed

ling_all=glmer(lab_ling~zoo_ling*Diagnosis*Age + (1|ChildID),data=data_all,family="binomial",subset=c(C

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00742676 (tol = 0.002, component 1)

#plot(lin_mod)
# doesn't converge

ling_AS=glmer(lab_ling~zoo_ling*Age + (1|ChildID),data=data_all,family="binomial",subset=c(Diagnosis=="Ang
#only effect of zoo

ling_LR=glmer(lab_ling~zoo_ling*Age + (1|ChildID),data=data_all,family="binomial",subset=c(Diagnosis!="Ang
#both main and interaction are sig

can_all=glmer(lab_can~zoo_can*Diagnosis*Age + (1|ChildID),data=data_all,family="binomial")

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00215928 (tol = 0.002, component 1)

#doesn't converge either

can_AS=glmer(lab_can~zoo_can*Age + (1|ChildID),data=data_all,family="binomial",subset=c(Diagnosis=="Ang
#only effect of zoo

can_LR=glmer(lab_can~zoo_can*Age + (1|ChildID),data=data_all,family="binomial",subset=c(Diagnosis!="Ang
#both main and interaction are sig
```

## Additional analyses

We additionally explored under what conditions Zooniverse judgments more closely aligned with laboratory judgments. In previous work using a similar method, for instance, data from all three children from one dataset were often labeled as “Junk” (i.e., not a child’s vocalization), and the data points from this corpus stood out when the authors attempted to integrate results with other corpora [Cychosz]. A high proportion of “Junk” may indicate that automated segmentation was errorful for those children, and may be a sign that the rest of the data could be compromised as well.

```

junkXdiag=t.test(proportions$z_junk ~ proportions$Diagnosis)

mycols=c("blue","darkgreen")
mypch=c(10,18)
names(mycols)=names(mypch)=c("lower_junk","higher_junk")

layout(matrix(c(1:2), 1, 2, byrow = F))
par(mar = c(4, 4, 1, 1))

#create matrix to hold r results
myr=matrix(NA,nrow=2,ncol=2)
rownames(myr)<-c("lp","cp")
colnames(myr)<-c("lower_junk","higher_junk")
mylb=myhb=myn=myp=myr #this is an ugly way of creating 5 matrices

lims=range(c(proportions[, "z_lp"], proportions[, "l_lp"]))

plot(proportions[, "z_lp"]~proportions[, "l_lp"], xlab=prettynames["l_lp"], ylab=prettynames["z_lp"],
      xlim=lims, ylim=lims,
      col=mycols[proportions$median_junk], pch=mypch[proportions$median_junk])
lines(c(0,1),c(0,1),lty=2,col="darkgray")

abline(lm(proportions[, "z_lp"]~proportions[, "l_lp"], subset=c(proportions$median_junk=="lower_junk")),

#get cor properties
thiscor=cor.test(proportions[proportions$median_junk=="lower_junk", "z_lp"], proportions[proportions$median_junk=="lower_junk", "l_lp"],
  myr[1, "lower_junk"]=round(thiscor$estimate,3)
  mylb[1, "lower_junk"]=round(thiscor$conf.int[1],3)
  myhb[1, "lower_junk"]=round(thiscor$conf.int[2],3)
  myn[1, "lower_junk"]=round(thiscor$parameter,3)
  myp[1, "lower_junk"]=round(thiscor$p.value,3)

#repeat for LR
abline(lm(proportions[, "z_lp"]~proportions[, "l_lp"], subset=c(proportions$median_junk!="lower_junk")),

      #get cor properties
thiscor=cor.test(proportions[proportions$median_junk!="lower_junk", "z_lp"], proportions[proportions$median_junk!="lower_junk", "l_lp"],
  myr[1, "higher_junk"]=round(thiscor$estimate,3)
  mylb[1, "higher_junk"]=round(thiscor$conf.int[1],3)
  myhb[1, "higher_junk"]=round(thiscor$conf.int[2],3)
  myn[1, "higher_junk"]=round(thiscor$parameter,3)
  myp[1, "higher_junk"]=round(thiscor$p.value,3)

      # repeat whole for cp

lims=range(c(proportions[, "z_cp"], proportions[, "l_cp"]))

plot(proportions[, "z_cp"]~proportions[, "l_cp"], xlab=prettynames["l_cp"], ylab=prettynames["z_cp"],
      xlim=lims, ylim=lims,
      col=mycols[proportions$median_junk], pch=mypch[proportions$median_junk])
lines(c(0,1),c(0,1),lty=2,col="darkgray")

```

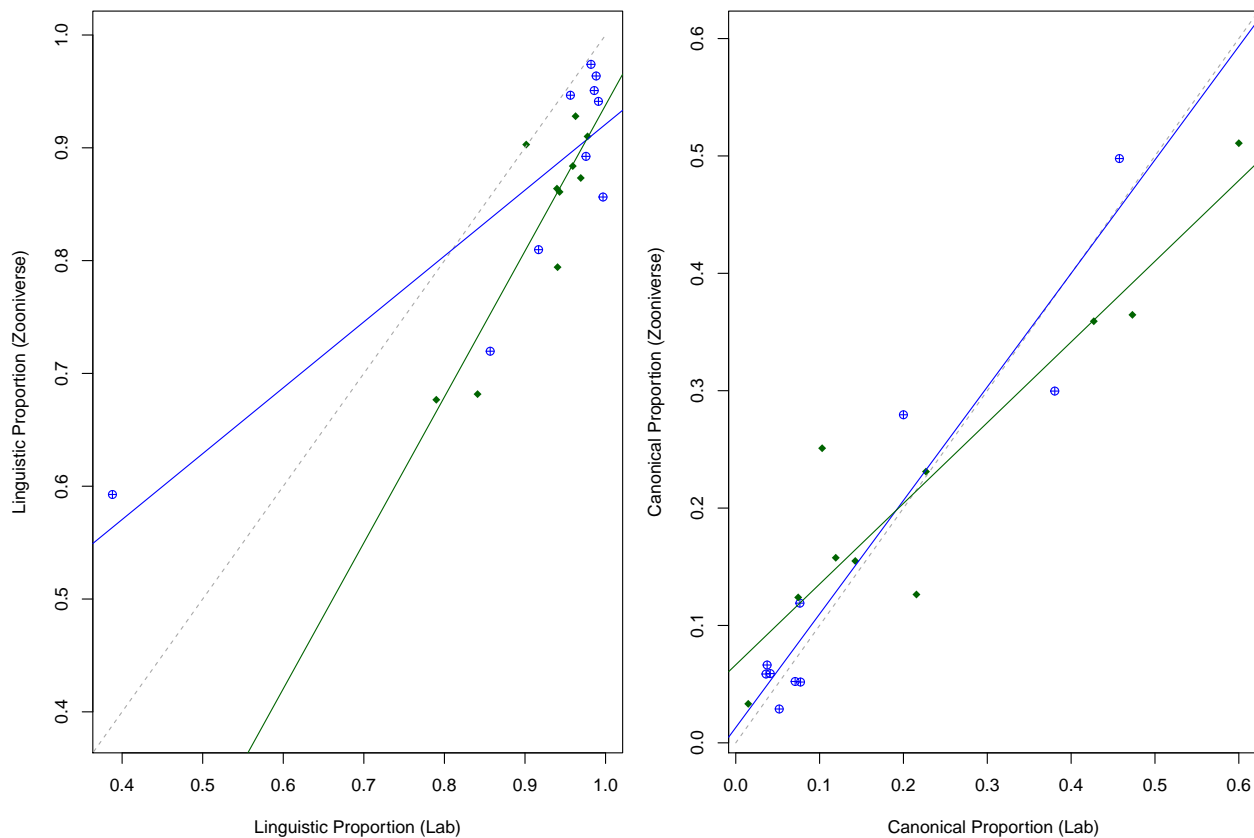
```

abline(lm(proportions[, "z_cp"] ~ proportions[, "l_cp"], subset=c(proportions$median_junk=="lower_junk")),

#get cor properties
thiscor=cor.test(proportions[proportions$median_junk=="lower_junk", "z_cp"], proportions[proportions$median_junk=="lower_junk", "l_cp"])
myr[2, "lower_junk"] = round(thiscor$estimate, 3)
mylb[2, "lower_junk"] = round(thiscor$conf.int[1], 3)
myhb[2, "lower_junk"] = round(thiscor$conf.int[2], 3)
myn[2, "lower_junk"] = round(thiscor$parameter, 3)
myp[2, "lower_junk"] = round(thiscor$p.value, 3)

#repeat for higher_junk
abline(lm(proportions[, "z_cp"] ~ proportions[, "l_cp"], subset=c(proportions$median_junk!="lower_junk")),

```



```

#get cor properties
thiscor=cor.test(proportions[proportions$median_junk!="lower_junk", "l_cp"], proportions[proportions$median_junk!="lower_junk", "z_cp"])
myr[2, "higher_junk"] = round(thiscor$estimate, 3)
mylb[2, "higher_junk"] = round(thiscor$conf.int[1], 3)
myhb[2, "higher_junk"] = round(thiscor$conf.int[2], 3)
myn[2, "higher_junk"] = round(thiscor$parameter, 3)
myp[2, "higher_junk"] = round(thiscor$p.value, 3)

```

We investigated this hypothesis by calculating the proportion of their data labeled as “Junk” for each individual child. There was no significant difference in the proportions of their data labeled as “Junk” for older children with Angelman Syndrome ( $M = 0.126$ ) compared to the younger low risk children ( $M = 0.14$ ): Welch’s  $t(13.3654471) = -0.516$ ,  $p = 0.614$ . We therefore collapsed across groups for this exploratory analysis, and split the 20 children using a median split on the proportion of their data labeled as “Junk”. Results

were similar across these two post-hoc subgroups. For Linguistic Proportion, the correlation across lab and Zooniverse data for the lower junk group was  $r(8) = 0.871$ , CI [0.534,0.969],  $p=0.001$ ; and for the higher junk group it was  $r(8) = 0.872$ , CI [0.539,0.97],  $p=0.001$ . For Canonical Proportion, the correlation across lab and Zooniverse data for the lower junk group was  $r(8) = 0.957$ , CI [0.822,0.99],  $p=0$ ; and for the higher junk group it was  $r(8) = 0.931$ , CI [0.729,0.984],  $p=0$ .

- take all 5 Zoo judgments or 3 or 1 (may need to run the latter two a couple of times to check for potential variance – if variance is small, we’ll just say so, if it’s large then we’ll have to write a loop and report CIs)
- look at what happens to TD Zoo-PU correlation when you undersample 100 vocs to 80 and 60 and to the same number of non-junk vocs as in AS