

# Gold standard inspection

## Contents

<b>Inspection of the existing data from Hollink et al.</b>	<b>1</b>
Datasets from Hollink et al. . . . .	1
Inspecting new dataframes . . . . .	4
Quick visualization of Tools . . . . .	5
Eliminating duplicates . . . . .	7
<b>Visualizing the data</b>	<b>18</b>

## Inspection of the existing data from Hollink et al.

Loading libraries

```
library(readr)
library(tidyverse)
library(dplyr)
library(readxl)
library(ggplot2)
library("Rgraphviz")
```

### Datasets from Hollink et al.

```
annotator1_tools <- read_delim("handtool_labelled.csv", col_names =
  c('lvl1_synset', 'lvl2_synset', 'lvl3_synset',
    'lvl4_synset', 'lvl5_synset', 'lvl6_synset',
    'isBasic', 'isFamiliar', 'isUnsure', 'note',
    'inHistWords', 'note2'), skip=1, delim=';')

annotator1_fruits <- read_delim("ediblefruit_labelled.csv", col_names =
  c('lvl1_synset', 'lvl2_synset', 'lvl3_synset',
    'lvl4_synset', 'lvl5_synset', 'lvl6_synset',
    'isBasic', 'isFamiliar', 'isUnsure', 'note',
    'inHistWords'), skip=1, delim=';')

annotator1_instruments <- read_delim("musicalinstrument_labelled.csv", col_names =
  c('synonyms', 'glossary', 'lvl1_synset',
    'lvl2_synset', 'lvl3_synset', 'lvl4_synset',
    'lvl5_synset', 'lvl6_synset', 'lvl7_synset',
```

```

                                'lvl8_synset','isBasic','isFamiliar','isUnsure',
                                'notes'), skip=1, delim=';')

annotator2_tools <- read_delim("handtool_a2.csv",
                                col_names=c('synonyms','glossary','lvl1_synset',
                                'lvl2_synset','lvl3_synset','lvl4_synset',
                                'lvl5_synset','lvl6_synset','isBasic','origin',
                                'isFamiliar','isUnsure','notes'), skip=1, delim=',')

annotator3_tools <- read_delim("handtool_a3.csv",
                                col_names=c('synonyms','glossary','lvl1_synset',
                                'lvl2_synset','lvl3_synset','lvl4_synset',
                                'lvl5_synset','lvl6_synset','isBasic','isFamiliar',
                                'isUnsure','notes'), skip=1, delim=',')

```

## Cleaning up dataframe layout

- reduce hierarchy lvl columns to 2 dim

```

#Function to create new columns for synsets
red_lvls <- function(ds, synlen = 6, dflen){
  ds_copy <- ds
  synsets <- c()
  tree_depth <- c()
  tree <- c("", "", "", "", "", "", "", "")
  ancestors <- c()

  for (i in 1:length(ds$lvl1_synset)){
    if(!is.na(ds[["lvl1_synset"]][i])){
      syn <- toString(ds[["lvl1_synset"]][i])
      synsets <- append(synsets, syn)
      tree_depth <- append(tree_depth, 1)

      tree[1] <- syn
      ancestors <- append(ancestors, "root")
    }
    else if(!is.na(ds[["lvl2_synset"]][i])){
      syn <- toString(ds[["lvl2_synset"]][i])
      synsets <- append(synsets, syn)
      tree_depth <- append(tree_depth, 2)

      tree[2] <- syn
      ancestors <- append(ancestors, tree[1])
    }
    else if(!is.na(ds[["lvl3_synset"]][i])){
      syn <- toString(ds[["lvl3_synset"]][i])
      synsets <- append(synsets, syn)
      tree_depth <- append(tree_depth, 3)

      tree[3] <- syn
      ancestors <- append(ancestors, tree[2])
    }
  }
}

```

```

}
else if(!is.na(ds[["lv14_synset"]][i])){
  syn <- toString(ds[["lv14_synset"]][i])
  synsets <- append(synsets, syn)
  tree_depth <- append(tree_depth, 4)

  tree[4] <- syn
  ancestors <- append(ancestors, tree[3])
}
else if(!is.na(ds[["lv15_synset"]][i])){
  syn <- toString(ds[["lv15_synset"]][i])
  synsets <- append(synsets, syn)
  tree_depth <- append(tree_depth, 5)

  tree[5] <- syn
  ancestors <- append(ancestors, tree[4])
}
else if(!is.na(ds[["lv16_synset"]][i])){
  syn <- toString(ds[["lv16_synset"]][i])
  synsets <- append(synsets, syn)
  tree_depth <- append(tree_depth, 6)

  tree[6] <- syn
  ancestors <- append(ancestors, tree[5])
}
else if(!is.na(ds[["lv17_synset"]][i]) & synlen > 6){
  syn <- toString(ds[["lv17_synset"]][i])
  synsets <- append(synsets, syn)
  tree_depth <- append(tree_depth, 7)

  tree[7] <- syn
  ancestors <- append(ancestors, tree[6])
}
else if(!is.na(ds[["lv18_synset"]][i]) & synlen > 7){
  syn <- toString(ds[["lv18_synset"]][i])
  synsets <- append(synsets, syn)
  tree_depth <- append(tree_depth, 8)

  tree[8] <- syn
  ancestors <- append(ancestors, tree[7])
}
#needs optional lvl 7 and 8, see optional parameters
else{
  print("Some error must have occurred.")
  print(ds[['synonyms']][i])
}
}

ds_copy$synsets <- synsets
ds_copy$tree_depth <- tree_depth
ds_copy$ancestors <- ancestors
ds_copy
}

```

```

#adding all the new columns
annotator1_fruits <- red_lvls(annotator1_fruits)
annotator1_tools <- red_lvls(annotator1_tools)
annotator1_instruments <- red_lvls(annotator1_instruments, 8)

annotator2_tools <- red_lvls(annotator2_tools)
annotator3_tools <- red_lvls(annotator3_tools)

#Now we remove the unnecessary old columns from the dfs
#tools
bl_tools <- data.frame(matrix(ncol = 6, nrow = 164))
colnames(bl_tools) <- c('synsets', 'tree_depth', 'ancestors',
                       'judgmt_a1', 'judgmt_a2', 'judgmt_a3')

bl_tools$synsets <- annotator1_tools$synsets
bl_tools$tree_depth <- annotator1_tools$tree_depth
bl_tools$ancestors <- annotator1_tools$ancestors
bl_tools$judgmt_a1 <- annotator1_tools$isBasic

#add judgements of other 2 annotators
bl_tools$judgmt_a2 <- annotator2_tools$isBasic
bl_tools$judgmt_a3 <- annotator3_tools$isBasic

#instruments

bl_instruments <- data.frame(matrix(ncol = 4, nrow = 192))
colnames(bl_instruments) <- c('synsets', 'tree_depth', 'ancestors',
                              'judgmt_a1')

bl_instruments$synsets <- annotator1_instruments$synsets
bl_instruments$tree_depth <- annotator1_instruments$tree_depth
bl_instruments$ancestors <- annotator1_instruments$ancestors
bl_instruments$judgmt_a1 <- annotator1_instruments$isBasic

#fruits

bl_fruits <- data.frame(matrix(ncol = 4, nrow = 197))
colnames(bl_fruits) <- c('synsets', 'tree_depth', 'ancestors',
                        'judgmt_a1')

bl_fruits$synsets <- annotator1_fruits$synsets
bl_fruits$tree_depth <- annotator1_fruits$tree_depth
bl_fruits$ancestors <- annotator1_fruits$ancestors
bl_fruits$judgmt_a1 <- annotator1_fruits$isBasic

```

## Inspecting new dataframes

```
head(bl_tools)
```

```
##               synsets tree_depth               ancestors
```

```
## 1 [Synset('hand_tool.n.01'),          1          root
## 2     [Synset('awl.n.01'),            2 [Synset('hand_tool.n.01'),
## 3     [Synset('bradawl.n.01')],        3     [Synset('awl.n.01'),
## 4 [Synset('scriber.n.01')]],           3     [Synset('awl.n.01'),
## 5     [Synset('bevel.n.02')],          2 [Synset('hand_tool.n.01'),
## 6     [Synset('bodkin.n.03')],         2 [Synset('hand_tool.n.01'),
##   judgmt_a1 judgmt_a2 judgmt_a3
## 1         h         h         h
## 2         b         b         b
## 3         1         1         1
## 4         1         1         1
## 5         b         b         b
## 6         1         b         1
```

```
head(bl_fruits)
```

```
##                                synsets tree_depth
## 1     [Synset('edible_fruit.n.01')]          1
## 2         [Synset('ackee.n.01')]              2
## 3     [Synset('anchovy_pear.n.02')]          2
## 4         [Synset('apple.n.01')]              2
## 5     [Synset('cooking_apple.n.01')]         3
## 6 [Synset('bramley's_seedling.n.01')]        4
##                                ancestors judgmt_a1
## 1                                root          h
## 2     [Synset('edible_fruit.n.01')]          b
## 3     [Synset('edible_fruit.n.01')]          b
## 4     [Synset('edible_fruit.n.01')]          b
## 5         [Synset('apple.n.01')]             1
## 6 [Synset('cooking_apple.n.01')]            1
```

```
head(bl_instruments)
```

```
##                                synsets tree_depth
## 1 [Synset('musical_instrument.n.01')]        1
## 2     [Synset('barrel_organ.n.01')]          2
## 3         [Synset('bass.n.07')]              2
## 4     [Synset('bass_fiddle.n.01')]          3
## 5     [Synset('bass_guitar.n.01')]          3
## 6         [Synset('bass_horn.n.01')]         3
##                                ancestors judgmt_a1
## 1                                root          h
## 2 [Synset('musical_instrument.n.01')]        b
## 3 [Synset('musical_instrument.n.01')]        h
## 4         [Synset('bass.n.07')]             1
## 5         [Synset('bass.n.07')]             1
## 6         [Synset('bass.n.07')]             b
```

## Quick visualization of Tools

```
p <- ggplot(bl_tools, mapping = aes(x = tree_depth, fill = judgmt_a1)) +
  geom_bar() +
  scale_x_continuous(breaks = c(1, 2, 3, 4, 5, 6)) +
  labs(x="Tree depth", y="Count of synsets",
       title="Count of tools by depth",
       subtitle="Colored by annotator judgement.",
       caption="Figure 1.: Tools by depth, colored by judgement.")
p
```

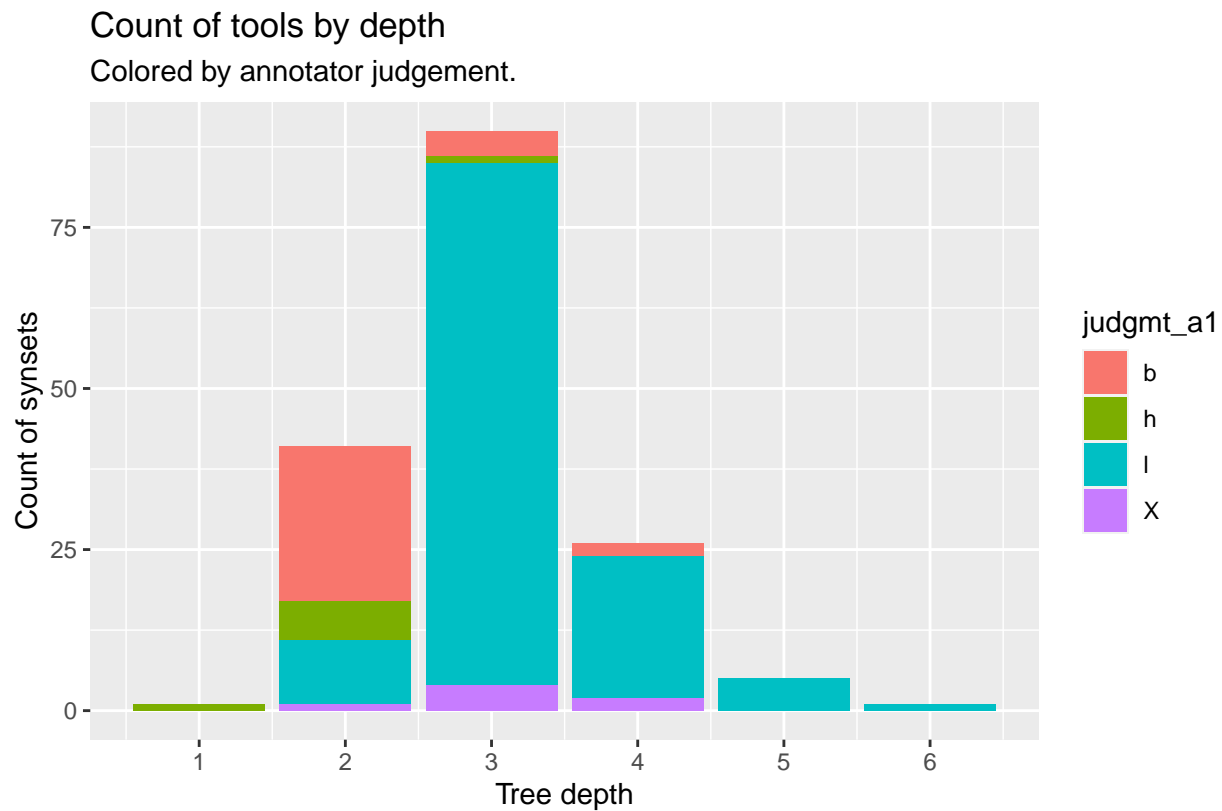


Figure 1.: Tools by depth, colored by judgement.

Datasets from Niamh Henry

```
all_ediblefruit <- read_csv("./niamh/labels/all_ediblefruit_labelled.csv")
all_handtool <- read_csv("./niamh/labels/all_handtool_labelled.csv")
all_instrument <- read_csv("./niamh/labels/all_musicalinstruments_labelled.csv")
all_furniture <- read_csv("./niamh/labels/GoldStandardFurniture_merged.csv")
all_garments <- read_xlsx("./niamh/labels/GoldStandardGarments_merged.xlsx")
```

Now I can complete the two dataframes `bl_fruits` and `_instruments` with the missing annotations. \* Check if same order \* copy over annotations columns (2) \* copy over glossary and synonyms (3)

```
issameorder1 <- (all_ediblefruit$A1_Label == bl_fruits$judgmt_a1) #True

bl_fruits$judgmt_a2 <- all_ediblefruit$Label
bl_fruits$judgmt_a3 <- all_ediblefruit$A3_Label
```

```

bl_fruits$synonyms <- all_ediblefruit$Synonyms
bl_fruits$glossary <- all_ediblefruit$Glossary

issameorder2 <- all_instrument$Label == bl_instruments$judgmt_a1 #True

bl_instruments$judgmt_a2 <- all_instrument$A2_Label
bl_instruments$judgmt_a3 <- all_instrument$A3_Label
bl_instruments$synonyms <- all_instrument$Synonyms
bl_instruments$glossary <- all_instrument$Glossary

bl_tools$synonyms <- all_handtool$Synonyms
bl_tools$glossary <- all_handtool$Glossary

```

Having both the data from Niamh and Hollink, I finally have all of the information. Niamh added the synonyms and glossary to her dataframe and she has the 2 missing anotations from the two other annotators.

By visualizing the trees (below) I found that some rows in the dataset are duplicates. There are also cases where the synset column has duplicates, but some other information in the same column is different, e.g. ancestor. Removing pure duplicates is easy, but removing semi-duplicates with differing annotations is going to be tougher. I will probably have to make a list of all semi-duplicates and handpick the candidate to keep.

## Eliminating duplicates

### Function definitions

```

find_total_dups <- function(df){
  dups <- data.frame()
  dups_bools <- duplicated(df)
  dups_pos <- which(dups_bools)
  for (x in dups_pos){
    dups <- rbind(dups, df[which(df$synonyms == df$synonyms[x]),])
  }

  dups
}

find_semi_dups <- function(df){
  dups <- data.frame()

  dup_bools1 <- duplicated(df$synsets)
  dup_bools2 <- duplicated(df$synonyms)
  dup_bools <- dup_bools1 | dup_bools2
  dup_pos <- which(dup_bools)
  dup <- c()
  for (x in dup_pos){
    dup <- unique(append(dup, which(df$synonyms == df$synonyms[x])))
  }
  df[dup,]
}

```

```

#fruits
fruits_has_total_dup <- find_total_dups(bl_fruits) #NULL
fruits_has_semi_dup <- find_semi_dups(bl_fruits) #2

#tools
tools_has_total_dup <- find_total_dups(bl_tools) #NULL
tools_has_semi_dup <- find_semi_dups(bl_tools) #14

#instruments
instruments_has_total_dup <- find_total_dups(bl_instruments) #yes, 3
instruments_has_semi_dup <- find_semi_dups(bl_instruments) # many

```

Now that I can detect complete or semi duplicates, I will handpick the rows to be eliminated from the original data.

## Fruits

```
print.data.frame(fruits_has_total_dup)
```

```
## data frame with 0 columns and 0 rows
```

The fruits dataset has 0 total duplicates.

```
print.data.frame(fruits_has_semi_dup[,c("synsets", "ancestors", "judgmt_a1", "judgmt_a2", "judgmt_a3")])
```

```
##               synsets               ancestors judgmt_a1
## 42  [Synset('currant.n.01')] [Synset('berry.n.01')]      1
## 105 [Synset('currant.n.03')] [Synset('raisin.n.01')]      1
## 148 [Synset('medlar.n.03')] [Synset('edible_fruit.n.01')]    b
## 149 [Synset('medlar.n.04')] [Synset('edible_fruit.n.01')]    b
##      judgmt_a2 judgmt_a3
## 42           1         b
## 105           1         1
## 148           b         1
## 149           b         X
```

There are 2 instances of semi-duplicate concepts. ##### The currant Currant has also been listed as a subordinate of raisin, additionally to berry. This seems superfluous as the fruit is still the same fruit after drying. People only tend to call all kinds of dried berries 'raisin'. I will remove the raisin subordinate (105).

**The medlar** There does not seem to be any difference between both instances, except for the second one having a higher digit at the end of the name and the third annotator giving two different judgements. I will remove the one with the X judgement (149).

## Removal



```
fruits_rm <- c(105, 149)
bl_fruits_new <- bl_fruits[-fruits_rm,]
```

## Tools

```
print.data.frame(tools_has_total_dup)
```

```
## data frame with 0 columns and 0 rows
```

Tools has no total duplicates.

```
print.data.frame(tools_has_semi_dup[,c("synsets", "ancestors", "judgmt_a1", "judgmt_a2", "judgmt_a3")])
```

##		synsets	ancestors
## 33		[Synset('hand_shovel.n.01'),	[Synset('hand_tool.n.01'),
## 109		[Synset('hand_shovel.n.01'),	[Synset('shovel.n.01'),
## 34		[Synset('coal_shovel.n.01')],	[Synset('hand_shovel.n.01'),
## 110		[Synset('coal_shovel.n.01')],	[Synset('hand_shovel.n.01'),
## 35		[Synset('entrenching_tool.n.01')],	[Synset('hand_shovel.n.01'),
## 111		[Synset('entrenching_tool.n.01')],	[Synset('hand_shovel.n.01'),
## 36		[Synset('spade.n.02'),	[Synset('hand_shovel.n.01'),
## 112		[Synset('spade.n.02'),	[Synset('hand_shovel.n.01'),
## 37		[Synset('ditch_spade.n.01')],	[Synset('spade.n.02'),
## 113		[Synset('ditch_spade.n.01')],	[Synset('spade.n.02'),
## 38		[Synset('garden_spade.n.01')],	[Synset('spade.n.02'),
## 114		[Synset('garden_spade.n.01')],	[Synset('spade.n.02'),
## 39		[Synset('spud.n.02')],	[Synset('hand_shovel.n.01'),
## 115		[Synset('spud.n.02')],	[Synset('hand_shovel.n.01'),
##	judgmt_a1	judgmt_a2	judgmt_a3
## 33	X	X	1
## 109	1	1	1
## 34	X	X	1
## 110	1	1	1
## 35	X	X	1
## 111	1	1	1
## 36	X	X	1
## 112	1	b	1
## 37	X	X	1
## 113	1	1	1
## 38	X	X	1
## 114	1	1	1
## 39	X	X	1
## 115	1	1	1

Hand Shovel RM 33

Coal Shovel RM 34

Entrenching Tool RM 35

Spade RM 36

Ditch Spade RM 37

Garden Spade RM 38

Spud RM 39

## Removal

```
tools_rm <- c(33:39)
bl_tools_new <- bl_fruits[-tools_rm,]
```

## Instruments

```
print.data.frame(instruments_has_total_dup[,c("synsets", "ancestors", "judgmt_a1", "judgmt_a2", "judgmt_a3")])
```

##		synsets	ancestors	judgmt_a1
## 27		[Synset('baby_grand.n.01')]	[Synset('grand_piano.n.01')]	1
## 59		[Synset('baby_grand.n.01')]	[Synset('grand_piano.n.01')]	X
## 107		[Synset('baby_grand.n.01')]	[Synset('grand_piano.n.01')]	X
## 28		[Synset('concert_grand.n.01')]	[Synset('grand_piano.n.01')]	1
## 60		[Synset('concert_grand.n.01')]	[Synset('grand_piano.n.01')]	X
## 108		[Synset('concert_grand.n.01')]	[Synset('grand_piano.n.01')]	X
## 22		[Synset('spinet.n.02')]	[Synset('harpsichord.n.01')]	1
## 31		[Synset('spinet.n.01')]	[Synset('upright.n.02')]	1
## 63		[Synset('spinet.n.01')]	[Synset('upright.n.02')]	X
## 94		[Synset('spinet.n.02')]	[Synset('harpsichord.n.01')]	X
## 111		[Synset('spinet.n.01')]	[Synset('upright.n.02')]	X
##	judgmt_a2	judgmt_a3		
## 27	1	1		
## 59	1	X		
## 107	1	X		
## 28	1	1		
## 60	1	X		
## 108	1	X		
## 22	1	1		
## 31	1	1		
## 63	1	X		
## 94	1	X		
## 111	1	X		

Baby Grand RM 59, 107

Concert Grand RM 60, 108

Spinnet RM 63, 94, 111

```
print.data.frame(instruments_has_semi_dup[,c("synsets", "ancestors", "judgmt_a1", "judgmt_a2", "judgmt_a3")
```

```
##                                synsets
## 22      [Synset('spinet.n.02')]
## 31      [Synset('spinet.n.01')]
## 63      [Synset('spinet.n.01')]
## 94      [Synset('spinet.n.02')]
## 111     [Synset('spinet.n.01')]
## 13      [Synset('synthesizer.n.02')]
## 32      [Synset('synthesizer.n.02')]
## 25      [Synset('piano.n.01')]
## 57      [Synset('piano.n.01')]
## 105     [Synset('piano.n.01')]
## 26      [Synset('grand_piano.n.01')]
## 58      [Synset('grand_piano.n.01')]
## 106     [Synset('grand_piano.n.01')]
## 27      [Synset('baby_grand.n.01')]
## 59      [Synset('baby_grand.n.01')]
## 107     [Synset('baby_grand.n.01')]
## 28      [Synset('concert_grand.n.01')]
## 60      [Synset('concert_grand.n.01')]
## 108     [Synset('concert_grand.n.01')]
## 29      [Synset('mechanical_piano.n.01')]
## 61      [Synset('mechanical_piano.n.01')]
## 109     [Synset('mechanical_piano.n.01')]
## 30      [Synset('upright.n.02')]
## 62      [Synset('upright.n.02')]
## 110     [Synset('upright.n.02')]
## 4       [Synset('bass_fiddle.n.01')]
## 71      [Synset('bass_fiddle.n.01')]
## 19      [Synset('clavichord.n.01')]
## 91      [Synset('clavichord.n.01')]
## 20      [Synset('clavier.n.02')]
## 92      [Synset('clavier.n.02')]
## 21      [Synset('harpsichord.n.01')]
## 93      [Synset('harpsichord.n.01')]
## 23      [Synset('virginal.n.01')]
## 95      [Synset('virginal.n.01')]
## 5       [Synset('bass_guitar.n.01')]
## 99      [Synset('bass_guitar.n.01')]
## 96      [Synset('dulcimer.n.01')]
## 116     [Synset('dulcimer.n.02')]
## 6       [Synset('bass_horn.n.01')]
## 120     [Synset('bass_horn.n.01')]
## 7       [Synset('euphonium.n.01')]
## 121     [Synset('euphonium.n.01')]
## 8       [Synset('helicon.n.01')]
## 122     [Synset('helicon.n.01')]
## 17      [Synset('accordion.n.01')]
```

## 133	[Synset('accordion.n.01')]		
## 24	[Synset('organ.n.05')]		
## 141	[Synset('organ.n.05')]		
## 9	[Synset('bombardon.n.02')]		
## 180	[Synset('bombardon.n.02')]]]		
##		ancestors	judgmt_a1 judgmt_a2
## 22	[Synset('harpsichord.n.01')]	1	1
## 31	[Synset('upright.n.02')]	1	1
## 63	[Synset('upright.n.02')]	X	1
## 94	[Synset('harpsichord.n.01')]	X	1
## 111	[Synset('upright.n.02')]	X	1
## 13	[Synset('electronic_instrument.n.01')]	b	b
## 32	[Synset('keyboard_instrument.n.01')]	X	b
## 25	[Synset('keyboard_instrument.n.01')]	b	b
## 57	[Synset('percussion_instrument.n.01')]	X	h
## 105	[Synset('stringed_instrument.n.01')]	X	b
## 26	[Synset('piano.n.01')]	1	1
## 58	[Synset('piano.n.01')]	X	b
## 106	[Synset('piano.n.01')]	X	1
## 27	[Synset('grand_piano.n.01')]	1	1
## 59	[Synset('grand_piano.n.01')]	X	1
## 107	[Synset('grand_piano.n.01')]	X	1
## 28	[Synset('grand_piano.n.01')]	1	1
## 60	[Synset('grand_piano.n.01')]	X	1
## 108	[Synset('grand_piano.n.01')]	X	1
## 29	[Synset('piano.n.01')]	1	1
## 61	[Synset('piano.n.01')]	X	b
## 109	[Synset('piano.n.01')]	X	1
## 30	[Synset('piano.n.01')]	1	1
## 62	[Synset('piano.n.01')]	X	b
## 110	[Synset('piano.n.01')]	X	1
## 4	[Synset('bass.n.07')]	1	b
## 71	[Synset('bowed_stringed_instrument.n.01')]	X	b
## 19	[Synset('keyboard_instrument.n.01')]	1	b
## 91	[Synset('stringed_instrument.n.01')]	X	b
## 20	[Synset('keyboard_instrument.n.01')]	h	h
## 92	[Synset('stringed_instrument.n.01')]	X	h
## 21	[Synset('clavier.n.02')]	b	b
## 93	[Synset('clavier.n.02')]	X	b
## 23	[Synset('harpsichord.n.01')]	1	1
## 95	[Synset('harpsichord.n.01')]	X	1
## 5	[Synset('bass.n.07')]	1	1
## 99	[Synset('guitar.n.01')]	X	1
## 96	[Synset('stringed_instrument.n.01')]	b	b
## 116	[Synset('zither.n.01')]	1	1
## 6	[Synset('bass.n.07')]	b	b
## 120	[Synset('brass.n.02')]	X	b
## 7	[Synset('bass_horn.n.01')]	1	1
## 121	[Synset('bass_horn.n.01')]	X	1
## 8	[Synset('bass_horn.n.01')]	1	1
## 122	[Synset('bass_horn.n.01')]	X	1
## 17	[Synset('keyboard_instrument.n.01')]	b	b
## 133	[Synset('free-reed_instrument.n.01')]	X	b
## 24	[Synset('keyboard_instrument.n.01')]	b	b

## 141	[Synset('wind_instrument.n.01')	X	b
## 9	[Synset('bass.n.07')	1	b
## 180	[Synset('shawm.n.01')	X	1
##	judgmt_a3		
## 22	1		
## 31	1		
## 63	X		
## 94	X		
## 111	X		
## 13	b		
## 32	X		
## 25	b		
## 57	X		
## 105	X		
## 26	1		
## 58	X		
## 106	X		
## 27	1		
## 59	X		
## 107	X		
## 28	1		
## 60	X		
## 108	X		
## 29	1		
## 61	X		
## 109	X		
## 30	1		
## 62	X		
## 110	X		
## 4	b		
## 71	X		
## 19	b		
## 91	X		
## 20	b		
## 92	X		
## 21	1		
## 93	X		
## 23	1		
## 95	X		
## 5	b		
## 99	1		
## 96	b		
## 116	X		
## 6	b		
## 120	X		
## 7	1		
## 121	X		
## 8	1		
## 122	X		
## 17	b		
## 133	X		
## 24	b		
## 141	X		
## 9	1		

**## 180** X

**Synthesiser"** RM 32

**Piano** RM 57, 105

**Grand Piano** RM 58, 106

**Mechanical Piano** RM 61, 109

**Upright** RM 32, 110

**Bass fiddle** RM 71

**Clavichord** RM 91

**clavier** RM 92

**Harpsichord** RM 93

**Virginal** RM 95

**Bass Guitar** RM 5 Because visually closer to guitar

**Dulcimer** RM /

**Bass Horn** RM 120

**Euphonium** RM 121

**Helicon** RM 122

**Accordion** RM 133

**Organ** RM 141

**Bombardon** RM 180

In the case where two annotators annotated 'x', that duplicate was preferred for removal.

**Removal**

```
instruments_rm <- c(59, 107, 60, 108, 63, 94, 111)

bl_instruments_new <- bl_instruments[-instruments_rm,]
```

Get only entries with 100% agreement on subordinate level NEEDS IMPROVEMENT

```
#fruits
```

```
fruits_full_agree <- bl_fruits %>%
  filter(judgmt_a1 == judgmt_a2) %>%
  filter(judgmt_a2 == judgmt_a3) %>%
  group_by(judgmt_a1)
```

```
fruits_full_agree
```

```
## # A tibble: 157 x 8
## # Groups:   judgmt_a1 [3]
##   synsets      tree_depth ancestors  judgmt_a1 judgmt_a2 judgmt_a3 synonyms
##   <chr>          <dbl> <chr>      <chr>      <chr>      <chr>      <chr>
## 1 [Synset('~ 1 root      h          h          h          edible_f~
## 2 [Synset('~ 2 [Synset('~ b          b          b          ackee ak~
## 3 [Synset('~ 2 [Synset('~ b          b          b          anchovy~
## 4 [Synset('~ 2 [Synset('~ b          b          b          apple
## 5 [Synset('~ 3 [Synset('~ l          l          l          cooking~
## 6 [Synset('~ 4 [Synset('~ l          l          l          Bramley'~
## 7 [Synset('~ 4 [Synset('~ l          l          l          Lane's_P~
## 8 [Synset('~ 4 [Synset('~ l          l          l          Newtown~
## 9 [Synset('~ 4 [Synset('~ l          l          l          Rome_Bea~
## 10 [Synset('~ 3 [Synset('~ l          l          l          crab_app~
## # ... with 147 more rows, and 1 more variable: glossary <chr>
```

```
fruits_full_agree_subordinate <- fruits_full_agree %>%
  filter(judgmt_a1 == "l")
```

```
fruits_full_agree_subordinate
```

```
## # A tibble: 93 x 8
## # Groups:   judgmt_a1 [1]
##   synsets      tree_depth ancestors  judgmt_a1 judgmt_a2 judgmt_a3 synonyms
##   <chr>          <dbl> <chr>      <chr>      <chr>      <chr>      <chr>
## 1 [Synset('~ 3 [Synset('~ l          l          l          cooking~
## 2 [Synset('~ 4 [Synset('~ l          l          l          Bramley'~
## 3 [Synset('~ 4 [Synset('~ l          l          l          Lane's_P~
## 4 [Synset('~ 4 [Synset('~ l          l          l          Newtown~
## 5 [Synset('~ 4 [Synset('~ l          l          l          Rome_Bea~
## 6 [Synset('~ 3 [Synset('~ l          l          l          crab_app~
## 7 [Synset('~ 3 [Synset('~ l          l          l          eating_a~
## 8 [Synset('~ 4 [Synset('~ l          l          l          Baldwin
## 9 [Synset('~ 4 [Synset('~ l          l          l          Cortland
## 10 [Synset('~ 4 [Synset('~ l          l          l          Delicious
## # ... with 83 more rows, and 1 more variable: glossary <chr>
```

```

fruits_full_agree_superordinate <- fruits_full_agree %>%
  filter(judgmt_a1 == "h")

fruits_full_agree_basic <- fruits_full_agree %>%
  filter(judgmt_a1 == "b")

fruit_string <- c("The fruit dataframe of 197 concepts has ",
  length(fruits_full_agree$judgmt_a1),
  " rows where all annotators agree, and of those ",
  length(fruits_full_agree_subordinate$judgmt_a1),
  " are of the subordinate level, ",
  length(fruits_full_agree_superordinate$judgmt_a1),
  " are superordinate and ",
  length(fruits_full_agree_basic$judgmt_a1),
  "are basic.")

```

*#instruments*

```

instruments_full_agree <- bl_instruments %>%
  filter(judgmt_a1 == judgmt_a2) %>%
  filter(judgmt_a2 == judgmt_a3) %>%
  group_by(judgmt_a1)

instruments_full_agree

```

```

## # A tibble: 126 x 8
## # Groups:   judgmt_a1 [3]
##   synsets    tree_depth ancestors  judgmt_a1 judgmt_a2 judgmt_a3 synonyms
##   <chr>          <dbl> <chr>      <chr>      <chr>      <chr>      <chr>
## 1 [Synset(~      1 root      h          h          h          musical_i~
## 2 [Synset(~      2 [Synset('~ h          h          h          bass
## 3 [Synset(~      3 [Synset('~ b          b          b          bass_horn~
## 4 [Synset(~      4 [Synset('~ l          l          l          euphonium
## 5 [Synset(~      4 [Synset('~ l          l          l          helicon b~
## 6 [Synset(~      2 [Synset('~ h          h          h          electroni~
## 7 [Synset(~      3 [Synset('~ b          b          b          synthesiz~
## 8 [Synset(~      3 [Synset('~ b          b          b          theremin
## 9 [Synset(~      2 [Synset('~ b          b          b          jew's_har~
##10 [Synset(~      2 [Synset('~ h          h          h          keyboard_~
## # ... with 116 more rows, and 1 more variable: glossary <chr>

```

```

instruments_full_agree_subordinate <- instruments_full_agree %>%
  filter(judgmt_a1 == "l")

instruments_full_agree_subordinate

```

```

## # A tibble: 62 x 8
## # Groups:   judgmt_a1 [1]
##   synsets    tree_depth ancestors  judgmt_a1 judgmt_a2 judgmt_a3 synonyms
##   <chr>          <dbl> <chr>      <chr>      <chr>      <chr>      <chr>
## 1 [Synset(~      4 [Synset('~ l          l          l          euphonium
## 2 [Synset(~      4 [Synset('~ l          l          l          helicon b~
## 3 [Synset(~      5 [Synset('~ l          l          l          spinet

```



```
## 4 [Synset(~      5 [Synset('~ 1      1      1      virginal ~
## 5 [Synset(~      4 [Synset('~ 1      1      1      grand_pia~
## 6 [Synset(~      5 [Synset('~ 1      1      1      baby_gran~
## 7 [Synset(~      5 [Synset('~ 1      1      1      concert_g~
## 8 [Synset(~      4 [Synset('~ 1      1      1      mechanica~
## 9 [Synset(~      4 [Synset('~ 1      1      1      upright u~
## 10 [Synset(~      5 [Synset('~ 1      1      1      spinet
## # ... with 52 more rows, and 1 more variable: glossary <chr>
```

```
instruments_full_agree_superordinate <- instruments_full_agree %>%
  filter(judgmt_a1 == "h")

instruments_full_agree_basic <- instruments_full_agree %>%
  filter(judgmt_a1 == "b")

instru_string <- c("The instruments dataframe of 192 concepts has",
  length(instruments_full_agree$judgmt_a1),
  " rows where all annotators agree, and of those ",
  length(instruments_full_agree_subordinate$judgmt_a1),
  " are of the subordinate level, ",
  length(instruments_full_agree_superordinate$judgmt_a1),
  " are superordinate and ",
  length(instruments_full_agree_basic$judgmt_a1),
  "are basic.")

#tools

tools_full_agree <- bl_tools %>%
  filter(judgmt_a1 == judgmt_a2) %>%
  filter(judgmt_a2 == judgmt_a3) %>%
  group_by(judgmt_a1)

tools_full_agree
```

```
## # A tibble: 133 x 8
## # Groups:   judgmt_a1 [3]
##   synsets    tree_depth ancestors   judgmt_a1 judgmt_a2 judgmt_a3 synonyms
##   <chr>          <dbl> <chr>      <chr>      <chr>      <chr>      <chr>
## 1 [Synset(~      1 root      h          h          h          hand_tool
## 2 [Synset(~      2 [Synset('h~ b          b          b          awl
## 3 [Synset(~      3 [Synset('a~ 1          1          1          bradawl ~
## 4 [Synset(~      3 [Synset('a~ 1          1          1          scribe ~
## 5 [Synset(~      2 [Synset('h~ b          b          b          bevel be~
## 6 [Synset(~      2 [Synset('h~ 1          1          1          bodkin t~
## 7 [Synset(~      2 [Synset('h~ b          b          b          file
## 8 [Synset(~      3 [Synset('f~ 1          1          1          blunt_fi~
## 9 [Synset(~      3 [Synset('f~ 1          1          1          flat_file
## 10 [Synset(~      3 [Synset('f~ 1          1          1          nailfile
## # ... with 123 more rows, and 1 more variable: glossary <chr>
```

```
tools_full_agree_subordinate <- tools_full_agree %>%
  filter(judgmt_a1 == "1")
```

```
tools_full_agree_subordinate
```

```
## # A tibble: 106 x 8
## # Groups:   judgmt_a1 [1]
##   synsets      tree_depth ancestors  judgmt_a1 judgmt_a2 judgmt_a3 synonyms
##   <chr>          <dbl> <chr>      <chr>      <chr>      <chr>      <chr>
## 1 [Synset('~      3 [Synset('~ 1      1      1      bradawl ~
## 2 [Synset('~      3 [Synset('~ 1      1      1      scribe ~
## 3 [Synset('~      2 [Synset('~ 1      1      1      bodkin t~
## 4 [Synset('~      3 [Synset('~ 1      1      1      blunt_fi~
## 5 [Synset('~      3 [Synset('~ 1      1      1      flat_file
## 6 [Synset('~      3 [Synset('~ 1      1      1      nailfile
## 7 [Synset('~      3 [Synset('~ 1      1      1      rasp woo~
## 8 [Synset('~      3 [Synset('~ 1      1      1      rat-tail~
## 9 [Synset('~      3 [Synset('~ 1      1      1      round_fi~
##10 [Synset('~      3 [Synset('~ 1      1      1      taper_fi~
## # ... with 96 more rows, and 1 more variable: glossary <chr>
```

```
tools_full_agree_superordinate <- tools_full_agree %>%
  filter(judgmt_a1 == "h")

tools_full_agree_basic <- tools_full_agree %>%
  filter(judgmt_a1 == "b")

tool_string <- c("The tools dataframe of 164 concepts has ",
  length(tools_full_agree$judgmt_a1),
  " rows where all annotators agree, and of those ",
  length(tools_full_agree_subordinate$judgmt_a1),
  " are of the subordinate level, ",
  length(tools_full_agree_superordinate$judgmt_a1),
  " are superordinate and ",
  length(tools_full_agree_basic$judgmt_a1),
  "are basic.")
```

## Visualizing the data

```
tree_viz_df <- function(synsets, synonyms, ancestors, judg1, judg2, judg3, depth, maxdepth){
  #first third of box
  y <- c()
  x <- c()
  ymin <- c()
  ymax <- c()

  x1min <- c()
  x1max <- c()
  z1 <- c()
  #second third
  x2min <- c()
  x2max <- c()
}
```

```

z2 <- c()
#last third
x3min <- c()
x3max <- c()
z3 <- c()
#labels and lines
label <- c()
label_raw <- c()
linedest_x <- c()
linedest_y <- c()

for (i in 0:(maxdepth-1)){
  counter <- 0
  depth_len <- length(which(depth==i+1))
  for (j in 0:(length(depth)-1)){
    if (i+1 == depth[j+1]){

      x <- append(x, i)
      x1min <- append(x1min, i-0.4)
      x1max <- append(x1max, i-0.13)
      z1 <- append(z1, judg1[j+1])

      x2min <- append(x2min, i-0.13)
      x2max <- append(x2max, i+0.13)
      z2 <- append(z2, judg2[j+1])

      x3min <- append(x3min, i+0.13)
      x3max <- append(x3max, i+0.4)
      z3 <- append(z3, judg3[j+1])

      new_label <- str_to_sentence(
        str_replace_all(
          str_split(
            synonyms[j+1], " ")[[1]][1], "_", " ")
        label <- append(label, new_label)
        dupli_bool <- any(synsets[j+1] == label_raw)
        label_raw <- append(label_raw, synsets[j+1])

      if(i == 0){
        linedest_x <- append(linedest_x, 0)
        linedest_y <- append(linedest_y, 0)

        y <- append(y, 0)
        ymin <- append(ymin, -0.45)
        ymax <- append(ymax, 0.45)
      }
    } else{
      ancest_pos <- max(which(match(label_raw,c(ancestors[j+1]))==1))
      ancest_y <- y[ancest_pos]
      ancest_x <- x3max[ancest_pos]
      linedest_x <- append(linedest_x, ancest_x)
      linedest_y <- append(linedest_y, ancest_y)
    }
  }
}

```

```

    if(dupli_bool){
      dupli_pos <- which(match(label_raw,c(synsets[j+1]))==1)[1]
      y <- append(y, y[dupli_pos])
      ymin <- append(ymin, ymin[dupli_pos])
      ymax <- append(ymax, ymax[dupli_pos])
    }
    else {
      y <- append(y, (depth_len/2)-counter)
      ymin <- append(ymin, (depth_len/2)-counter-0.45)
      ymax <- append(ymax, (depth_len/2)-counter+0.45)
    }

  }
  counter <- counter +1
}
}
}
tree_df <- data.frame(
  y=y,
  x=x,

  ymin=ymin,
  ymax=ymax,
  label=label,
  xend= linedest_x,
  yend= linedest_y,

  x1min=x1min,
  x1max=x1max,
  z1=z1,

  x2min=x2min,
  x2max=x2max,
  z2=z2,

  x3min=x3min,
  x3max=x3max,
  z3=z3
)
tree_df
}

make_tree_viz <- function(tree_data){
  tree <- ggplot(tree_data) +
    geom_segment(aes(x=x1min,
                     y=y,
                     xend=xend,
                     yend=yend,
                     color=z3)) +
    geom_rect(aes(xmin=x1min,
                  xmax=x1max,
                  ymin=ymin,
                  ymax=ymax,

```

```

        fill = z1),
        colour = "white") +
geom_rect(aes(xmin=x2min,
              xmax=x2max,
              ymin=ymin,
              ymax=ymax,
              fill = z2),
          colour = "white") +
geom_rect(aes(xmin=x3min,
              xmax=x3max,
              ymin=ymin,
              ymax=ymax,
              fill = z3),
          colour = "white") +
theme(axis.ticks.y = NULL,
      axis.ticks.x = NULL,
      legend.position = "top",
      panel.grid = NULL,
      line = NULL,
      axis.line = element_blank(),
      title = NULL,
      panel.grid.major = NULL,
      panel.background = NULL,
      panel.grid.minor = NULL) +
geom_text(aes(label=label,
              x=x,
              y=y,
              fontface="bold"))

tree
}

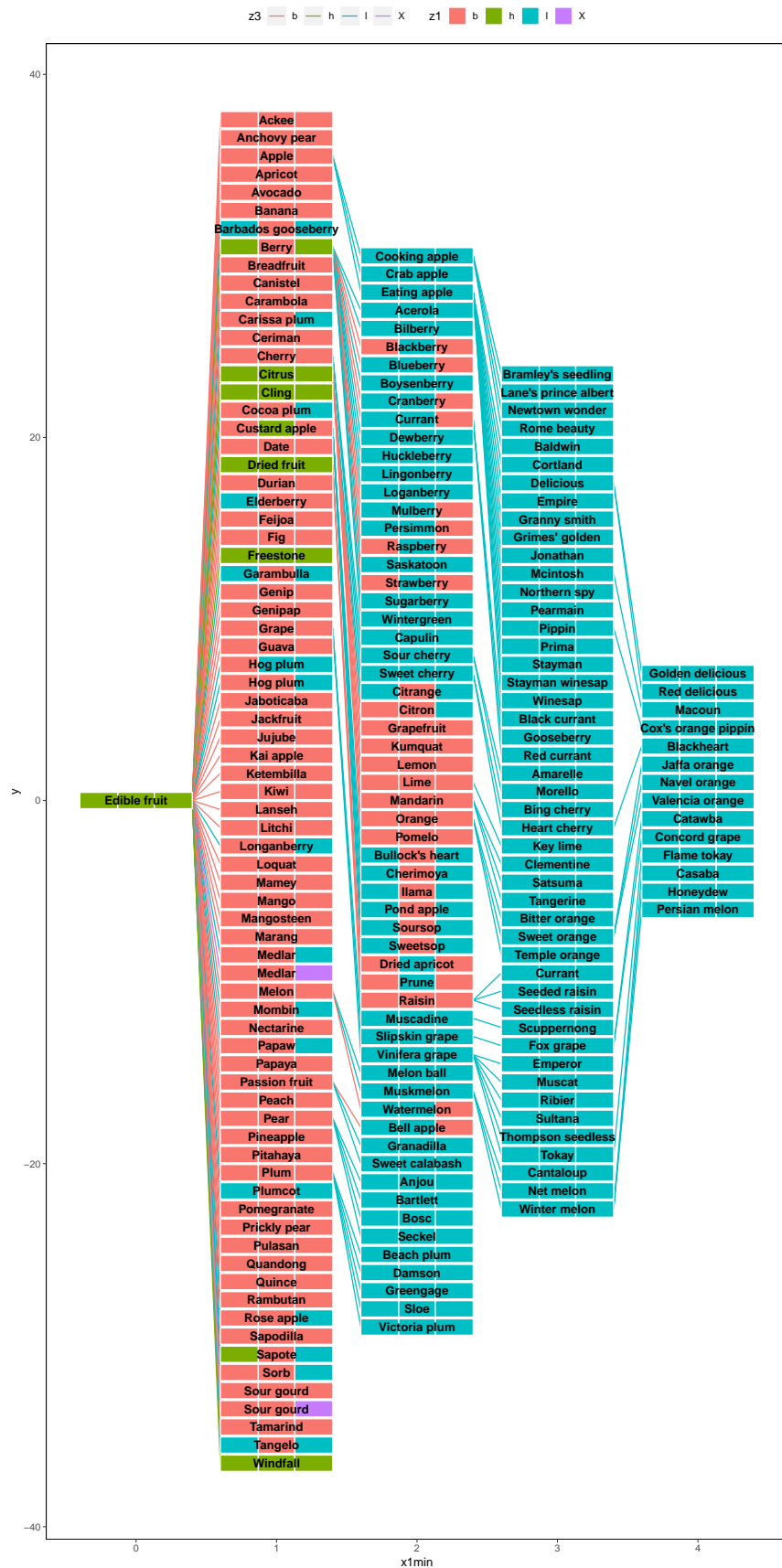
```

```

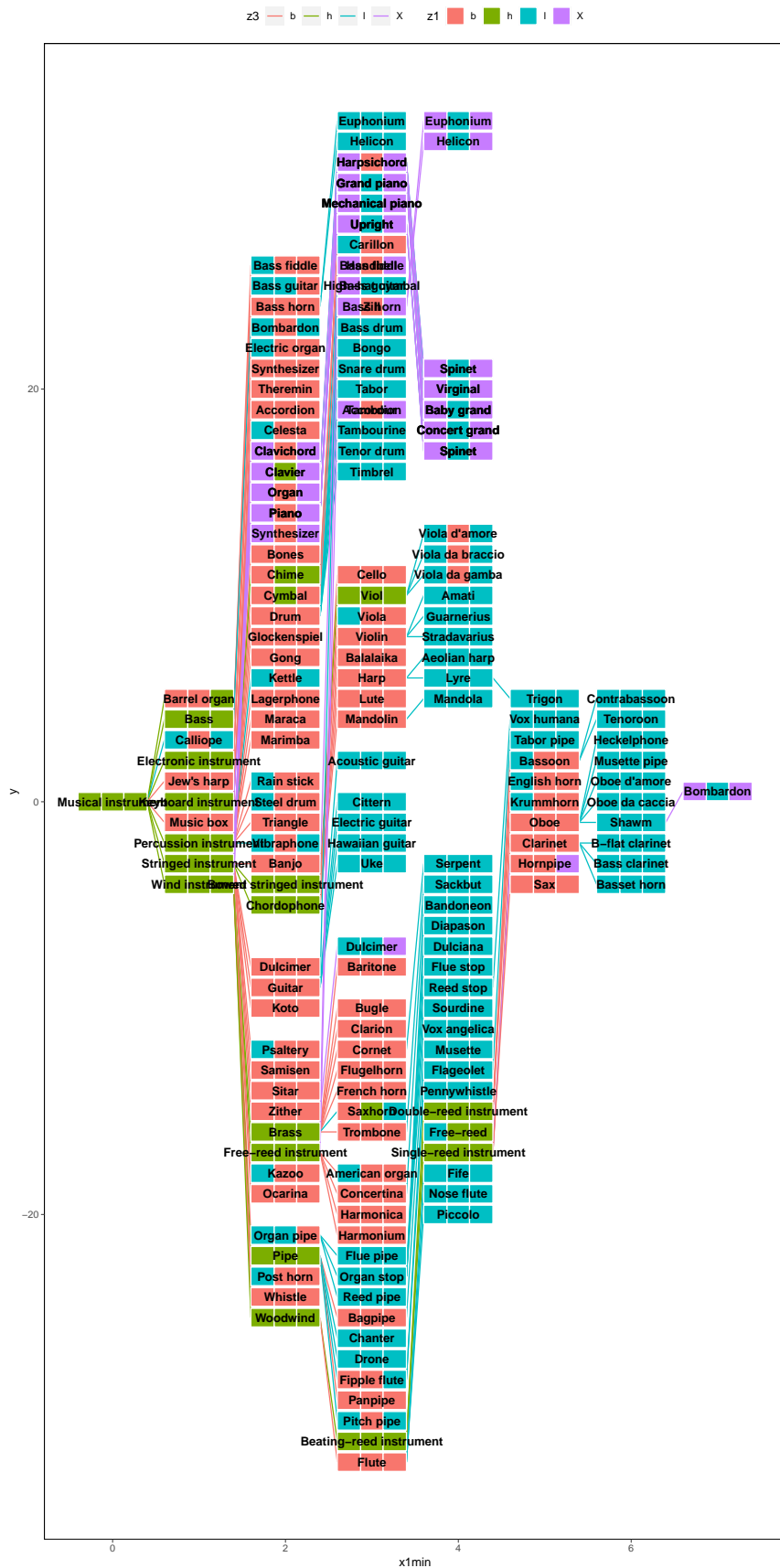
fruit_tree_data <- tree_viz_df(
  bl_fruits$synsets,
  bl_fruits$synonyms,
  bl_fruits$ancestors,
  bl_fruits$judgmt_a1,
  bl_fruits$judgmt_a2,
  bl_fruits$judgmt_a3,
  bl_fruits$tree_depth,
  5)

fruit_tree <- make_tree_viz(fruit_tree_data)
fruit_tree

```



```
instrument_tree_data <- tree_viz_df(  
  bl_instruments$synsets,  
  bl_instruments$synonyms,  
  bl_instruments$ancestors,  
  bl_instruments$judgmt_a1,  
  bl_instruments$judgmt_a2,  
  bl_instruments$judgmt_a3,  
  bl_instruments$tree_depth,  
  9)  
  
instrument_tree <- make_tree_viz(instrument_tree_data)  
instrument_tree
```





```
tool_tree_data <- tree_viz_df(  
  bl_tools$synsets,  
  bl_tools$synonyms,  
  bl_tools$ancestors,  
  bl_tools$judgmt_a1,  
  bl_tools$judgmt_a2,  
  bl_tools$judgmt_a3,  
  bl_tools$tree_depth,  
  7)  
  
tool_tree <- make_tree_viz(tool_tree_data)  
tool_tree
```

