Correlations Between Complexity Measures

Chris Bentz

January 7, 2021

Load Libraries

If the libraries are not installed yet, you need to install them using, for example, the command: install.packages("ggplot2").

```
library(readr)
library(ggplot2)
library(gridExtra)
library(GGally)
library(Hmisc)
library(ggrepel)
```

Load the Data

The participants' results are loaded as csv files directly from the github repository into separate data frames. We only use the name of the first author (lower case) to name the data frame.

```
#Track A (Parallel Bible Corpus, PBC)
gutierrez.results <- read_csv("https://raw.githubusercontent.com/IWMLC/language-complexity-metrics/maste
# remove the parentheses in column names
colnames(gutierrez.results) <- sub("\\(", "", colnames(gutierrez.results))
colnames(gutierrez.results) <- sub("\\)", "", colnames(gutierrez.results))
# replace "+" by "."
colnames(gutierrez.results) <- gsub("\\+", ".", colnames(gutierrez.results))
oh.results <- read_csv("https://raw.githubusercontent.com/IWMLC/language-complexity-metrics/master/PBCt

#TRACK B (Universal Dependencies, UD)
brunato.results  <- read_csv("https://raw.githubusercontent.com/IWMLC/language-complexity-metrics/maste
coltekin.results  <- read_csv("https://raw.githubusercontent.com/IWMLC/language-complexity-metrics/maste
semenuks.results  <- read_csv("https://raw.githubusercontent.com/IWMLC/language-complexity-metrics/maste
sinnemaki.results  <- read_csv("https://raw.githubusercontent.com/IWMLC/language-complexity-metrics/mas
sozinova.results  <- read_csv("https://raw.githubusercontent.com/IWMLC/language-complexity-metrics/maste
```

Sanity check, look at the number of rows and columns of the data frames.

```
#Track A (should be 49 rows)
track.a.rows <- c(nrow(gutierrez.results), nrow(oh.results))
print(track.a.rows) # this corresponds to the number of languages
```

```
## [1] 49 49
```

```
track.a.cols <- c(ncol(gutierrez.results), ncol(oh.results))
print(track.a.cols) # this is the number of measures per team
```

```
## [1] 14  5
```

```
#Track B (should be 63 rows)
track.b.rows <- c(nrow(brunato.results), nrow(coltekin.results), nrow(semenuks.results), nrow(sinnemaki
print(track.b.rows) # this corresponds to the number of languages
```

```
## [1] 63 63 63 63 63
```

```
track.b.cols <- c(ncol(brunato.results), ncol(coltekin.results), ncol(semenuks.results), ncol(sinnemaki
print(track.b.cols) # this is the number of measures per team
```

```
## [1] 13  8  4  8  4
```

Preprocessing

Put data into a single data frame.

```
track.a <- cbind(gutierrez.results, oh.results[, 3:ncol(oh.results)])
track.b <- cbind(brunato.results, coltekin.results[, 3:ncol(coltekin.results)],
                 semenuks.results[, 3:ncol(semenuks.results)],
                 sinnemaki.results[, 3:ncol(sinnemaki.results)],
                 sozinova.results[, 3:ncol(sozinova.results)])
```

Check data frames by looking at the first six rows.

```
head(track.a)
```

```
##    id        language GM_H1gram GM_H3gram     GM_TTR  GM_TTR.H1  GM_TTR.H3
## 1 aey           Amele 0.6119331  0.780252 0.06065136 0.04000000 0.03947368
## 2 amp        Alamblak 0.7057048  0.782454 0.10978104 0.05000000 0.06382979
## 3 ape         Bukiyip 0.6311849  0.763648 0.05561431 0.03333333 0.03614458
## 4 apu         Apurinã 0.5809332  0.567297 0.09235873 0.02985075 0.02857143
## 5 arn      Mapudungun 0.5744377  0.780808 0.08595837 0.04347826 0.03488372
## 6 arz Egyptian Arabic 0.8083907  0.887509 0.12832227 0.18181818 0.25000000
##   GM_TTR.H1.H3 GM_H1gram_fullyparallelised GM_H3gram_fullyparallelised
## 1   0.04081633                   0.5684752                   0.5908496
## 2   0.09090909                   0.6732652                   0.6431306
## 3   0.04081633                   0.6519671                   0.5913951
## 4   0.03571429                   0.5924908                   0.5236397
## 5   0.03333333                   0.5983614                   0.5964545
## 6   0.22222222                   0.7258967                   0.7482857
##   GM_TTR_fullyparallelised GM_TTR.H1_fullyparallelised
## 1                0.1342350                  0.03125000
## 2                0.2033711                  0.07692308
## 3                0.1193411                  0.04166667
## 4                0.2053489                  0.04651163
## 5                0.1457532                  0.04166667
## 6                0.3107240                  0.22222222
##   GM_TTR.H3_fullyparallelised GM_TTR.H1.H3_fullyparallelised       O_MC
## 1                  0.03636364                     0.03225806 -0.5125000
## 2                  0.06060606                     0.06818182 -0.4288462
## 3                  0.03333333                     0.03947368 -0.5021739
## 4                 0.03508772                        0.03488372 -0.5660714
## 5                 0.04255319                        0.04000000 -0.5351852
## 6                 0.25000000                        0.23076923 -0.4086207
##      O_WID O_SID
## 1 1.208351    NA
## 2 1.161317    NA
## 3 0.927140    NA
## 4 1.656120    NA
## 5 1.226694    NA
## 6 1.972695    NA
```

```
head(track.b)
```

```
##           id  language BV_n_tokens BV_char_per_tok BV_verbal_head_per_sent
## 1         afr Afrikaans   0.6915035       0.7406356               0.5973389
## 2         ara    Arabic   1.0000000       0.5700036               0.7373122
## 3         bul Bulgarian   0.3804936       0.7071177               0.4128931
## 4         cat   Catalan   0.8656840       0.6355104               0.6404372
## 5     ces_cac     Czech   0.5430299       0.7629205               0.4396123
## 6 ces_fictree     Czech   0.3553259       0.6460943               0.5242218
##   BV_verbal_root_perc BV_avg_token_per_clause BV_avg_links_len BV_avg_max_depth
## 1           0.7934703               0.8065034        0.9435208        0.7908597
## 2           0.2567031               0.9005952        0.7894060        1.0000000
## 3           0.8396519               0.5572851        0.6627371        0.4848930
## 4           0.8565512               1.0000000        0.8081980        0.7408015
## 5           0.7177322               0.7180483        0.7227964        0.6323912
## 6           0.8266025               0.4074251        0.6584553        0.4199076
##   BV_avg_subordinate_chain_len BV_subordinate_pre BV_subordinate_post
## 1                    0.6690569        0.195677426           0.7325159
## 2                    0.8753356        0.006573444           1.0000000
## 3                    0.4185603        0.140466212           0.4342773
## 4                    0.6893540        0.147016174           0.7545315
## 5                    0.4598093        0.092272362           0.5474183
## 6                    0.4772712        0.130639721           0.5117906
##   BV_avg_verb_edges CR_inflection_accuracy    CR_ttr   CR_msp    CR_mfe
## 1         0.8840035              0.6670443 0.1853184 1.219609  6.427348
## 2         0.6585259              0.1000941 0.2988734 1.475221  6.976586
## 3         0.6340592              0.3490676 0.3770606 1.514225  9.336138
## 4         0.8522001              0.5170170 0.2757109 1.266932  6.275705
## 5         0.6145715              0.3340556 0.4492450 1.588910 11.145461
## 6         0.6190423              0.3130136 0.3712954 1.633426 11.673832
##   CR_cfe_form_feat CR_cfe_feat_form S_idMean    S_idSD     SI_dm SI_hm
## 1      7.53990e-06      8.95090e-06 1.442039 1.0039941 1.0000000     0
## 2      2.19027e-05      6.96940e-06 1.407543 1.0389387 1.0000000     0
## 3      4.78590e-06      1.11903e-05 1.554631 1.0983221 1.0000000     0
## 4      1.17581e-05      1.05572e-05 1.567901 1.1048304 1.0000000     0
## 5      1.37917e-05      6.86860e-06 1.754058 1.0347654 0.8430774     0
## 6      1.55421e-05      1.07678e-05 1.821586 0.9678328 0.9334056     0
##   SI_dep_dl SI_double_dl SI_head_dl SI_zero_dl SBS_INF SBS_DER
## 1  2.346654           NA         NA         NA   0.199   0.093
## 2  1.830569           NA         NA         NA   0.955   0.032
## 3  2.275108           NA         NA         NA   1.020   0.209
## 4  2.641555           NA         NA         NA   0.845   0.059
## 5  1.745720           NA         NA   1.837112   1.439   0.117
## 6  1.425285           NA         NA   1.609121   1.461   0.148
```

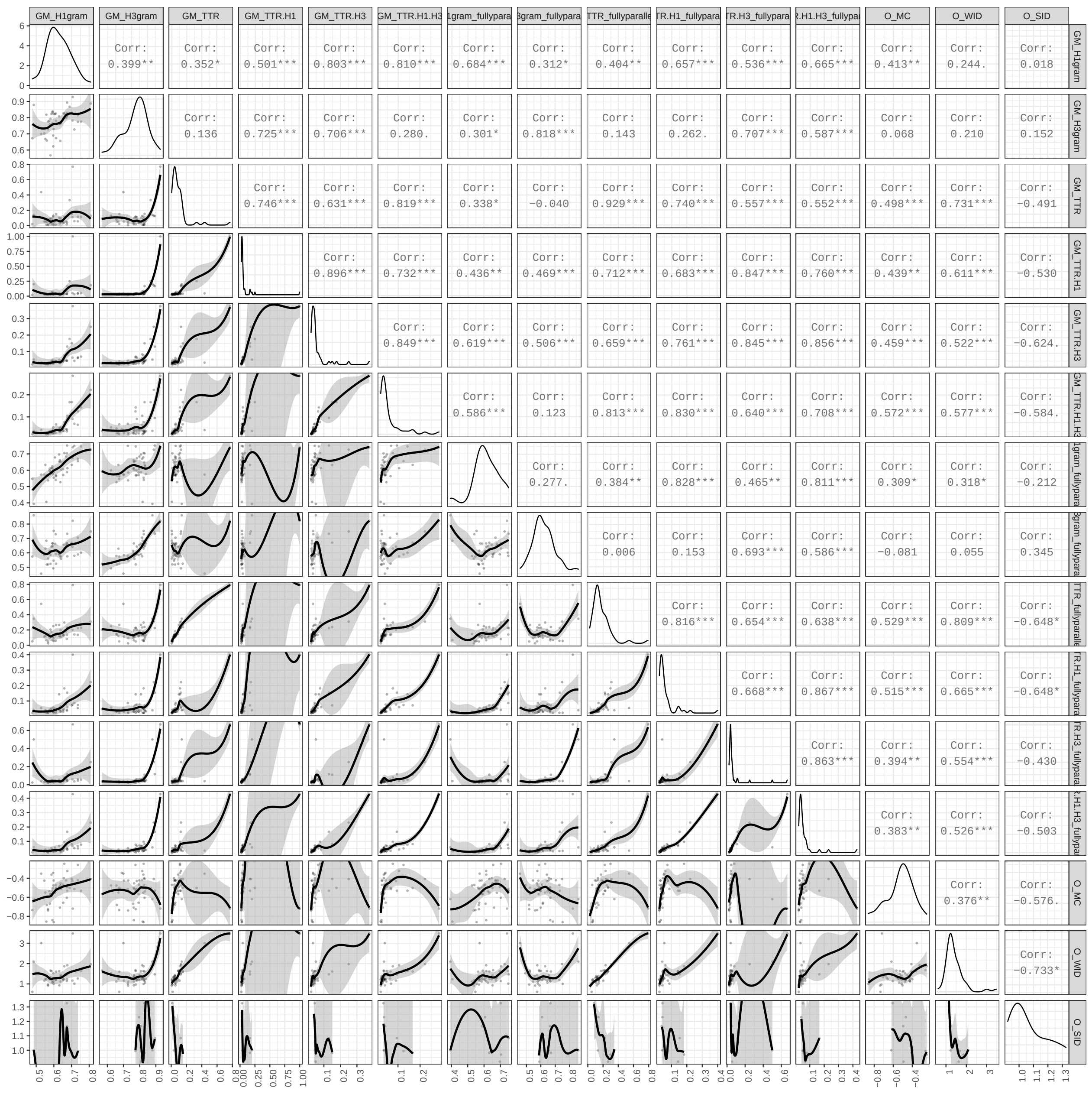
Plot Correlations by Track

TRACK A

We here plot correlations between selected measures of the respective track (trying to exclude the ones which are somewhat redundant). The Spearman correlation coefficient is reported instead of the Pearson correlation coefficient. This is because we are only interested whether there is a correlation between the rankings of complexities, regardless of whether this is a linear relationship. We therefore also use the local regression smoothers in the plots (loess) rather than linear models (lm). Note: warning messages are disabled here as there are datasets with NAs, and for each plot this throws a warning message using the ggpairs() plotting function. NAs are delt with by removing the entire row, containing an NA value.

```r
#remove the first two columns for plotting
track.a.short <- track.a[, 3:ncol(track.a)]

track.a.plot <- ggpairs(track.a.short,
                        lower = list(continuous = wrap("smooth_loess", alpha = 0.3,
                                                        lwd = 0.5, size = 2)),
                        upper = list(continuous = wrap('cor', method = "spearman"))) +
                        theme_bw() +
                        theme(axis.text.x = element_text(angle = 90, hjust = 1))
print(track.a.plot)
```



Safe plot to file.

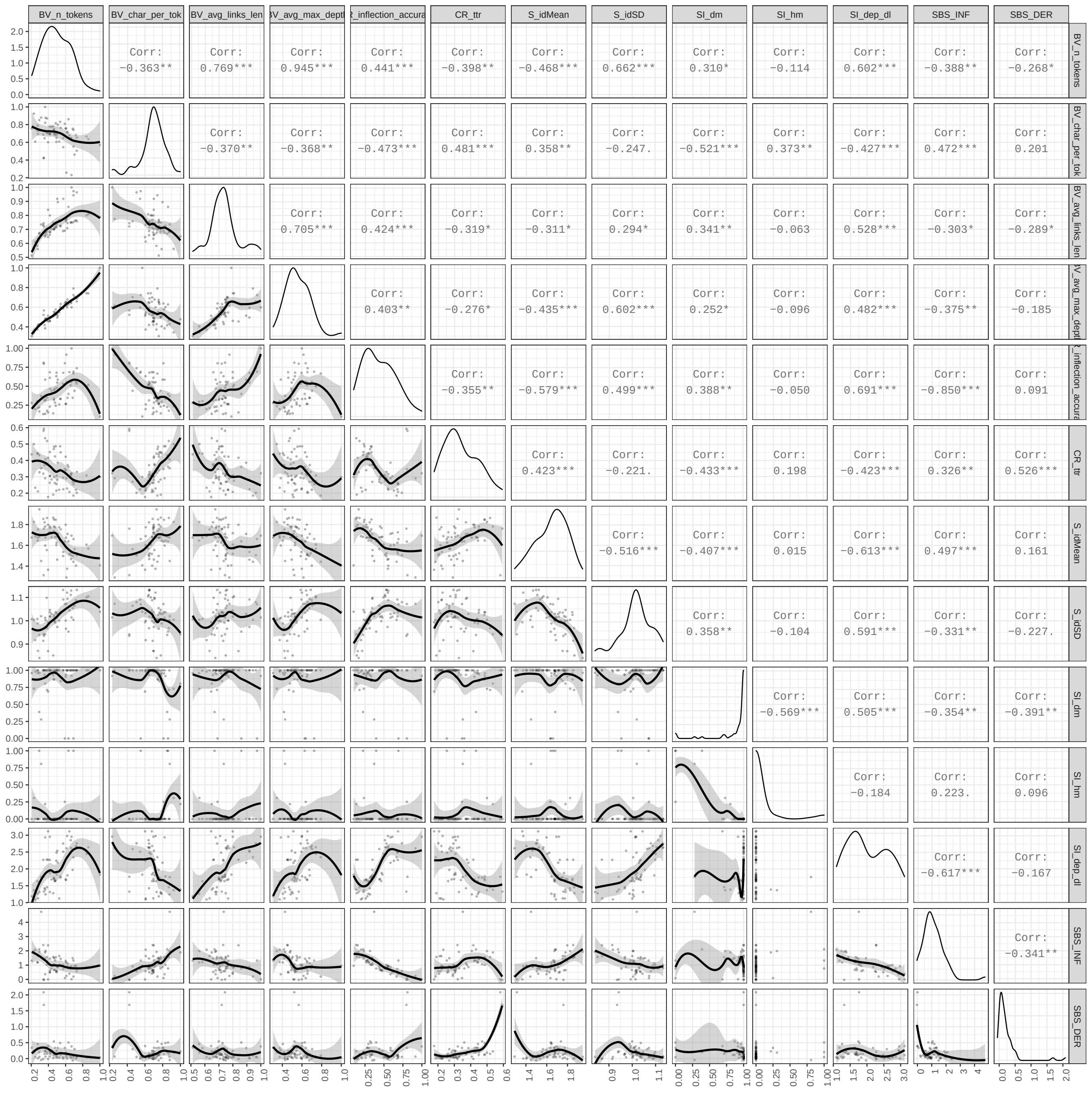
```
ggsave("~/Github/ComplexityMetaAnalyses/Figures/Corrs/TrackA/track_a_plot.pdf", track.a.plot,
       dpi = 300, scale = 1, width = 15, height = 15, device = cairo_pdf)
```

TRACK B

Same for the Track B data. Not all measures are included here (there would be 27). To include them all, the "columns" argument in the code below might be removed.

```
#remove the first to columns for plotting
track.b.short <- track.b[, 3:ncol(track.b)]

track.b.plot <- ggpairs(track.b.short, progress = TRUE,
                        lower = list(continuous = wrap("smooth_loess", alpha = 0.3,
                                                       lwd = 0.5, size = 2)),
                        upper = list(continuous = wrap('cor', method = "spearman")),
                        columns = c("BV_n_tokens", "BV_char_per_tok", "BV_avg_links_len",
                              "BV_avg_max_depth", "CR_inflection_accuracy", "CR_ttr",
                              "S_idMean", "S_idSD", "SI_dm", "SI_hm", "SI_dep_dl",
                              "SBS_INF", "SBS_DER")) +
                        theme_bw() +
                        theme(axis.text.x = element_text(angle = 90, hjust = 1))
print(track.b.plot)
```



Safe plot to file.

```
ggsave("~/Github/ComplexityMetaAnalyses/Figures/Corrs/TrackB/track_b_plot.pdf", track.b.plot,
       dpi = 300, scale = 1, width = 15, height = 15, device = cairo_pdf)
```

Significant Correlations after Bonferroni Correction

Not all of the correlations displayed above are going to be significant. We only select the ones still significant after correcting for multiple testing. Therefore, first calculate Spearman correlations and uncorrected p-values using the function rcorr(). These are then stored in a data frame where the first two columns give the names of the correlated measures.

TRACK A

```
#transform the data frame to a matrix
track.a.matrix <- as.matrix(track.a.short)
#apply the rcorr function to this matrix to get matrices of Spearman correlations and uncorrected p-val
track.a.cor <- rcorr(track.a.matrix, type = "spearman")$r
track.a.pvalues <- rcorr(track.a.matrix, type = "spearman")$P
track.a.n <- rcorr(track.a.matrix, type = "spearman")$n
#convert these matrices to a data frame again
track.a.df <- data.frame(row = rownames(track.a.pvalues)[row(track.a.pvalues)[upper.tri(track.a.pvalues)
           col = colnames(track.a.pvalues)[col(track.a.pvalues)[upper.tri(track.a.pvalues)]],
           pvalue = track.a.pvalues[upper.tri(track.a.pvalues)],
           corr = track.a.cor[upper.tri(track.a.cor)],
           num = track.a.n[upper.tri(track.a.n)])
head(track.a.df)
```

```
##         row       col       pvalue      corr num
## 1 GM_H1gram GM_H3gram 4.549026e-03 0.3986735  49
## 2 GM_H1gram    GM_TTR 1.317108e-02 0.3518367  49
## 3 GM_H3gram    GM_TTR 3.502930e-01 0.1363265  49
## 4 GM_H1gram GM_TTR.H1 2.477667e-04 0.5007655  49
## 5 GM_H3gram GM_TTR.H1 3.771262e-09 0.7252501  49
## 6    GM_TTR GM_TTR.H1 7.641920e-10 0.7460706  49
```

TRACK B

Same as above for Track A.

```
track.b.matrix <- as.matrix(track.b.short)
track.b.cor <- rcorr(track.b.matrix, type = "spearman")$r
```

```
## Warning in sqrt(npair - 2): NaNs produced
```

```
track.b.pvalues <- rcorr(track.b.matrix, type = "spearman")$P
```

```
## Warning in sqrt(npair - 2): NaNs produced
```

```
track.b.n <- rcorr(track.b.matrix, type = "spearman")$n
```

```
## Warning in sqrt(npair - 2): NaNs produced
```

```
track.b.df <- data.frame(row = rownames(track.b.pvalues)[row(track.b.pvalues)[upper.tri(track.b.pvalues)
           col = colnames(track.b.pvalues)[col(track.b.pvalues)[upper.tri(track.b.pvalues)]],
           pvalue = track.b.pvalues[upper.tri(track.b.pvalues)],
           corr = track.b.cor[upper.tri(track.b.cor)],
           num = track.b.n[upper.tri(track.b.n)])
#head(track.b.df)
```

Apply Bonferroni Correction

Apply the so-called Bonferroni correction, namely, multiply the p-values by the overall number of tests done. Arguably, this is the simplest, and also most conservative method for correcting the p-values. There are less-conservative alternatives such as the Holm-Bonferroni correction. Since the approach here is purely exploratory, and we have many measures and hence pairwise correlations anyways, we decided to go for the most conservative method.

```
# compute the overall number of tests, i.e. multiply the number of measures in each track with the same
n.test <- ncol(track.a.short)*(ncol(track.a.short) - 1) + ncol(track.b.short)*(ncol(track.b.short) - 1)
# add corrected pvalues to data frames
track.a.df$pvalue.correct <- track.a.df$pvalue*n.test
track.b.df$pvalue.correct <- track.b.df$pvalue*n.test
```

Remove all correlations which are not significant anymore. And then order them from highest to lowest coefficient.

```
# Track A
track.a.df <- track.a.df[track.a.df$pvalue.correct < 0.05, ]
track.a.df <- track.a.df[order(-track.a.df$corr), ]

# Track B
track.b.df <- track.b.df[track.b.df$pvalue.correct < 0.05, ]
track.b.df <- track.b.df[order(-track.b.df$corr), ]
```

Correlations still significant after Bonferroni correction for Track A:

```
print(track.a.df)
```

```
##                             row                             col       pvalue
## 31                       GM_TTR       GM_TTR_fullyparallelised 0.000000e+00
## 10                    GM_TTR.H1                      GM_TTR.H3 0.000000e+00
## 65 GM_TTR.H1_fullyparallelised GM_TTR.H1.H3_fullyparallelised 3.552714e-15
## 66 GM_TTR.H3_fullyparallelised GM_TTR.H1.H3_fullyparallelised 6.217249e-15
## 60                    GM_TTR.H3 GM_TTR.H1.H3_fullyparallelised 1.731948e-14
## 15                    GM_TTR.H3                   GM_TTR.H1.H3 1.243450e-14
## 49                    GM_TTR.H1    GM_TTR.H3_fullyparallelised 6.483702e-14
## 50                    GM_TTR.H3    GM_TTR.H3_fullyparallelised 7.926992e-14
## 42                 GM_TTR.H1.H3    GM_TTR.H1_fullyparallelised 5.682121e-13
## 43 GM_H1gram_fullyparallelised    GM_TTR.H1_fullyparallelised 6.787904e-13
## 13                       GM_TTR                   GM_TTR.H1.H3 6.155076e-13
## 23                    GM_H3gram    GM_H3gram_fullyparallelised 2.202238e-12
## 45    GM_TTR_fullyparallelised    GM_TTR.H1_fullyparallelised 2.940537e-12
## 34                 GM_TTR.H1.H3       GM_TTR_fullyparallelised 3.945289e-12
## 62 GM_H1gram_fullyparallelised GM_TTR.H1.H3_fullyparallelised 5.050182e-12
## 11                    GM_H1gram                   GM_TTR.H1.H3 1.898037e-12
## 87    GM_TTR_fullyparallelised                          O_WID 5.985212e-12
## 7                     GM_H1gram                      GM_TTR.H3 3.868461e-12
## 41                    GM_TTR.H3    GM_TTR.H1_fullyparallelised 5.331886e-10
## 59                    GM_TTR.H1 GM_TTR.H1.H3_fullyparallelised 5.860898e-10
## 6                        GM_TTR                      GM_TTR.H1 7.641920e-10
## 39                       GM_TTR    GM_TTR.H1_fullyparallelised 2.823907e-09
## 14                    GM_TTR.H1                   GM_TTR.H1.H3 2.268288e-09
## 81                       GM_TTR                          O_WID 5.209562e-09
## 5                     GM_H3gram                      GM_TTR.H1 3.771262e-09
## 32                    GM_TTR.H1       GM_TTR_fullyparallelised 1.988542e-08
## 61                 GM_TTR.H1.H3 GM_TTR.H1.H3_fullyparallelised 2.673641e-08
## 47                    GM_H3gram    GM_TTR.H3_fullyparallelised 2.842440e-08
## 8                     GM_H3gram                      GM_TTR.H3 1.430519e-08
## 53 GM_H3gram_fullyparallelised    GM_TTR.H3_fullyparallelised 6.724529e-08
## 16                    GM_H1gram    GM_H1gram_fullyparallelised 1.151531e-07
## 40                    GM_TTR.H1    GM_TTR.H1_fullyparallelised 1.242348e-07
## 55 GM_TTR.H1_fullyparallelised    GM_TTR.H3_fullyparallelised 2.824460e-07
## 56                    GM_H1gram GM_TTR.H1.H3_fullyparallelised 3.339067e-07
## 88 GM_TTR.H1_fullyparallelised                          O_WID 3.374804e-07
## 33                    GM_TTR.H3       GM_TTR_fullyparallelised 4.658630e-07
## 37                    GM_H1gram    GM_TTR.H1_fullyparallelised 5.418223e-07
## 54    GM_TTR_fullyparallelised    GM_TTR.H3_fullyparallelised 6.189858e-07
## 51                 GM_TTR.H1.H3    GM_TTR.H3_fullyparallelised 1.258620e-06
## 64    GM_TTR_fullyparallelised GM_TTR.H1.H3_fullyparallelised 1.425442e-06
## 9                        GM_TTR                      GM_TTR.H3 1.147100e-06
## 20                    GM_TTR.H3    GM_H1gram_fullyparallelised 3.541727e-06
## 82                    GM_TTR.H1                          O_WID 5.154513e-06
## 57                    GM_H3gram GM_TTR.H1.H3_fullyparallelised 1.419533e-05
## 63 GM_H3gram_fullyparallelised GM_TTR.H1.H3_fullyparallelised 1.515231e-05
## 21                 GM_TTR.H1.H3    GM_H1gram_fullyparallelised 1.538187e-05
## 84                 GM_TTR.H1.H3                          O_WID 2.145134e-05
## 72                 GM_TTR.H1.H3                           O_MC 1.734576e-05
## 48                       GM_TTR    GM_TTR.H3_fullyparallelised 4.770488e-05
## 89 GM_TTR.H3_fullyparallelised                          O_WID 5.298099e-05
##         corr num pvalue.correct
## 31 0.9293478  47   0.000000e+00
## 10 0.8964778  49   0.000000e+00
## 65 0.8667671  47   3.240075e-12
## 66 0.8629274  47   5.670131e-12
## 60 0.8560560  47   1.579537e-11
## 15 0.8493235  49   1.134026e-11
## 49 0.8466749  47   5.913137e-11
## 50 0.8451810  47   7.229417e-11
## 42 0.8297251  47   5.182095e-10
## 43 0.8282468  47   6.190568e-10
## 13 0.8193559  49   5.613430e-10
## 23 0.8181082  47   2.008441e-09
## 45 0.8155179  47   2.681769e-09
## 34 0.8128417  47   3.598103e-09
## 62 0.8105608  47   4.605766e-09
## 11 0.8095555  49   1.731010e-09
## 87 0.8089732  47   5.458514e-09
## 7  0.8030520  49   3.528037e-09
## 41 0.7610477  47   4.862680e-07
## 59 0.7598993  47   5.345139e-07
## 6  0.7460706  49   6.969431e-07
## 39 0.7398388  47   2.575403e-06
## 14 0.7320927  49   2.068679e-06
## 81 0.7314986  47   4.751120e-06
## 5  0.7252501  49   3.439391e-06
## 32 0.7121212  47   1.813550e-05
## 61 0.7076149  47   2.438361e-05
## 47 0.7066723  47   2.592306e-05
## 8  0.7062877  49   1.304633e-05
## 53 0.6930157  47   6.132771e-05
## 16 0.6840888  47   1.050196e-04
## 40 0.6828034  47   1.133021e-04
## 55 0.6684715  47   2.575907e-04
## 56 0.6654522  47   3.045229e-04
## 88 0.6652589  47   3.077821e-04
## 33 0.6593407  47   4.248671e-04
## 37 0.6565223  47   4.941420e-04
## 54 0.6540133  47   5.645151e-04
## 51 0.6402374  47   1.147862e-03
## 64 0.6377491  47   1.300003e-03
## 9  0.6314178  49   1.046155e-03
## 20 0.6188549  47   3.230055e-03
## 82 0.6106870  47   4.700916e-03
## 57 0.5874324  47   1.294615e-02
## 63 0.5858709  47   1.381891e-02
## 21 0.5855098  47   1.402826e-02
## 84 0.5774115  47   1.956362e-02
## 72 0.5723256  49   1.581934e-02
## 48 0.5570282  47   4.350685e-02
## 89 0.5542505  47   4.831866e-02
```

Correlations still significant after Bonferroni correction for Track B:

```
print(track.b.df)
```

```
##                         row                          col       pvalue
## 253                SI_dep_dl                 SI_double_dl 0.000000e+00
## 276             SI_double_dl                   SI_head_dl 0.000000e+00
## 300               SI_head_dl                   SI_zero_dl 0.000000e+00
## 314                   CR_msp                      SBS_INF 0.000000e+00
## 16               BV_n_tokens             BV_avg_max_depth 0.000000e+00
## 7                BV_n_tokens      BV_avg_token_per_clause 0.000000e+00
## 20       BV_avg_token_per_clause             BV_avg_max_depth 0.000000e+00
## 24       BV_verbal_head_per_sent BV_avg_subordinate_chain_len 0.000000e+00
## 105                   CR_msp                       CR_mfe 1.418865e-13
## 11               BV_n_tokens             BV_avg_links_len 1.880718e-13
## 315                   CR_mfe                      SBS_INF 3.888001e-13
## 21          BV_avg_links_len             BV_avg_max_depth 1.114653e-10
## 222     CR_inflection_accuracy                   SI_dep_dl 1.905962e-09
## 15       BV_avg_token_per_clause             BV_avg_links_len 7.305712e-10
## 158      BV_avg_token_per_clause                      S_idSD 9.133534e-10
## 50       BV_avg_token_per_clause            BV_avg_verb_edges 2.596476e-09
## 154              BV_n_tokens                       S_idSD 3.423637e-09
## 37               BV_n_tokens          BV_subordinate_post 9.996910e-09
## 51          BV_avg_links_len            BV_avg_verb_edges 1.625949e-08
## 31       BV_verbal_head_per_sent           BV_subordinate_pre 1.992449e-08
## 36  BV_avg_subordinate_chain_len          BV_subordinate_pre 2.628625e-08
## 43             BV_avg_max_depth          BV_subordinate_post 3.838314e-08
## 151                      CR_mfe                     S_idMean 1.586801e-07
## 160            BV_avg_max_depth                       S_idSD 1.748367e-07
## 211                 BV_n_tokens                    SI_dep_dl 4.647820e-07
## 46                  BV_n_tokens            BV_avg_verb_edges 1.952684e-07
## 229                      S_idSD                    SI_dep_dl 8.127537e-07
## 215      BV_avg_token_per_clause                    SI_dep_dl 1.004935e-05
## 41       BV_avg_token_per_clause          BV_subordinate_post 7.131573e-06
## 27              BV_avg_links_len BV_avg_subordinate_chain_len 7.646974e-06
## 52              BV_avg_max_depth            BV_avg_verb_edges 7.820502e-06
## 216             BV_avg_links_len                    SI_dep_dl 1.752547e-05
## 338                       CR_ttr                      SBS_DER 9.330432e-06
## 230                        SI_dm                    SI_dep_dl 4.464129e-05
## 165      CR_inflection_accuracy                       S_idSD 3.690263e-05
## 318                     S_idMean                      SBS_INF 3.418259e-05
## 150                       CR_msp                     S_idMean 3.653193e-05
## 171                     S_idMean                       S_idSD 1.532149e-05
## 173              BV_char_per_tok                        SI_dm 1.200458e-05
## 23               BV_char_per_tok BV_avg_subordinate_chain_len 2.468442e-06
## 210                        SI_dm                        SI_hm 1.141080e-06
## 148      CR_inflection_accuracy                     S_idMean 8.270105e-07
## 228                     S_idMean                    SI_dep_dl 2.411887e-07
## 322                    SI_dep_dl                      SBS_INF 1.948739e-07
## 224                       CR_msp                    SI_dep_dl 4.198780e-09
## 225                       CR_mfe                    SI_dep_dl 1.409184e-11
## 103      CR_inflection_accuracy                       CR_mfe 2.220446e-16
## 312      CR_inflection_accuracy                      SBS_INF 0.000000e+00
## 90       CR_inflection_accuracy                       CR_msp 0.000000e+00
## NA                          <NA>                         <NA>           NA
##          corr num pvalue.correct
## 253  1.0000000   3   0.000000e+00
## 276  1.0000000   2   0.000000e+00
## 300  1.0000000   2   0.000000e+00
## 314  0.9475806  63   0.000000e+00
## 16   0.9452765  63   0.000000e+00
## 7    0.9151786  63   0.000000e+00
## 20   0.8714478  63   0.000000e+00
## 24   0.8623752  63   0.000000e+00
## 105  0.7711899  63   1.294005e-10
## 11   0.7687692  63   1.715215e-10
## 315  0.7624043  63   3.545857e-10
## 21   0.7050691  63   1.016563e-07
## 222  0.6911624  58   1.738237e-06
## 15   0.6824117  63   6.662809e-07
## 158  0.6795795  63   8.329783e-07
## 50   0.6658986  63   2.367986e-06
## 154  0.6621544  63   3.122357e-06
## 37   0.6471294  63   9.117181e-06
## 51   0.6400250  63   1.482865e-05
## 31   0.6370008  63   1.817114e-05
## 36   0.6328245  63   2.397306e-05
## 43   0.6270161  63   3.500542e-05
## 151  0.6041192  63   1.447163e-04
## 160  0.6024866  63   1.594511e-04
## 211  0.6016949  59   4.238812e-04
## 46   0.6006144  63   1.780847e-04
## 229  0.5914085  59   7.412314e-04
## 215  0.5402104  59   9.165006e-03
## 41   0.5322581  63   6.503995e-03
## 27   0.5307700  63   6.974040e-03
## 52   0.5302899  63   7.132298e-03
## 216  0.5276447  59   1.598323e-02
## 338  0.5264876  63   8.509354e-03
## 230  0.5053256  59   4.071285e-02
## 165  0.4987787  62   3.365520e-02
## 318  0.4970238  63   3.117452e-02
## 150  0.4954397  63   3.331712e-02
## 171 -0.5155530  63   1.397320e-02
## 173 -0.5209789  63   1.094818e-02
## 23  -0.5540515  63   2.251219e-03
## 210 -0.5689882  63   1.040665e-03
## 148 -0.5789076  62   7.542336e-04
## 228 -0.6133255  59   2.199641e-04
## 322 -0.6170076  59   1.777250e-04
## 224 -0.6760959  59   3.829287e-06
## 225 -0.7443519  59   1.285176e-08
## 103 -0.8238686  62   2.025047e-13
## 312 -0.8500667  62   0.000000e+00
## 90  -0.8714714  62   0.000000e+00
## NA          NA  NA             NA
```

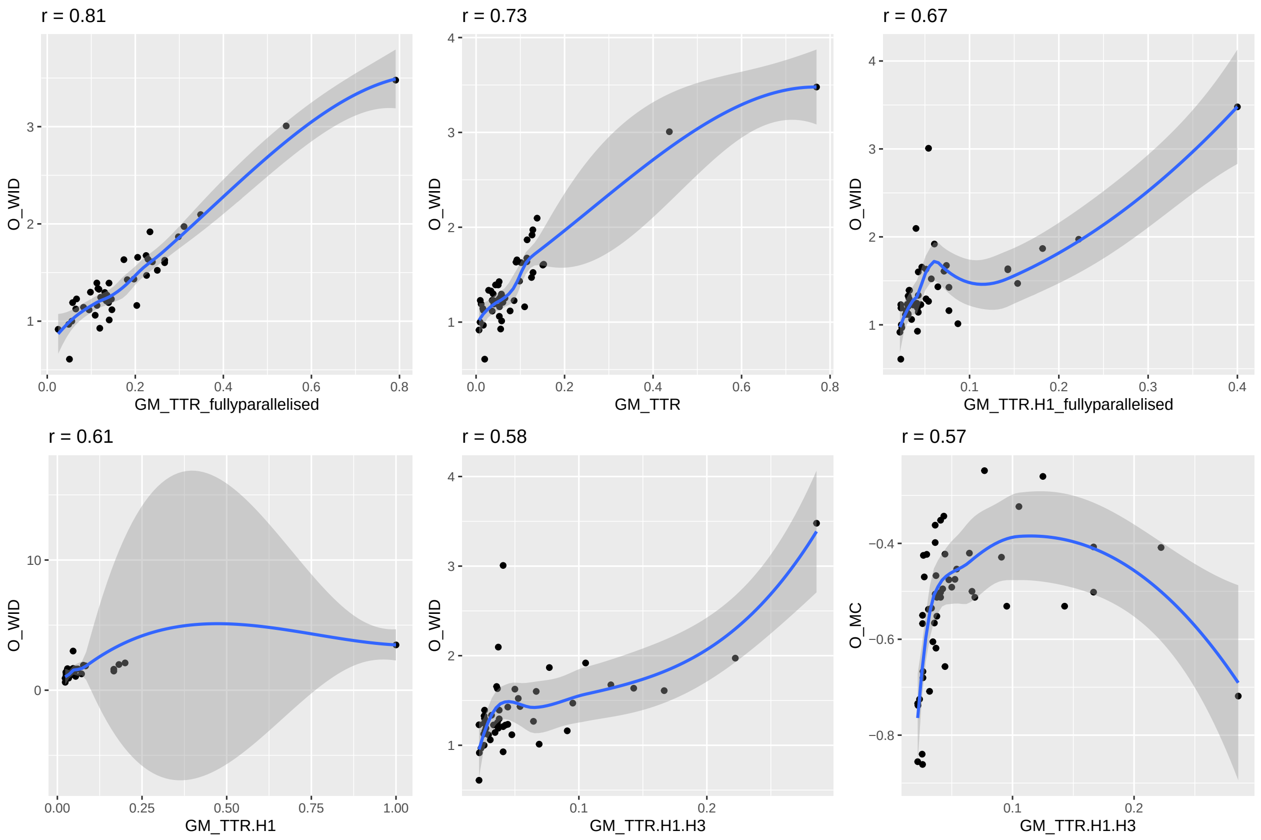
Positive Correlations

We here plot the six highest *positive* correlations (in terms of Spearman coefficients) which are still significant after the Bonferroni correction *and* which are found between measures proposed by *different participants* (there are many measures by the same participants that highly correlate). These are hand-picked from the lists above.

TRACK A

Plot the six significant correlations with highest Spearman coefficients for Track A. Warning messages are disabled since there are several NAs that throw errors.

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```

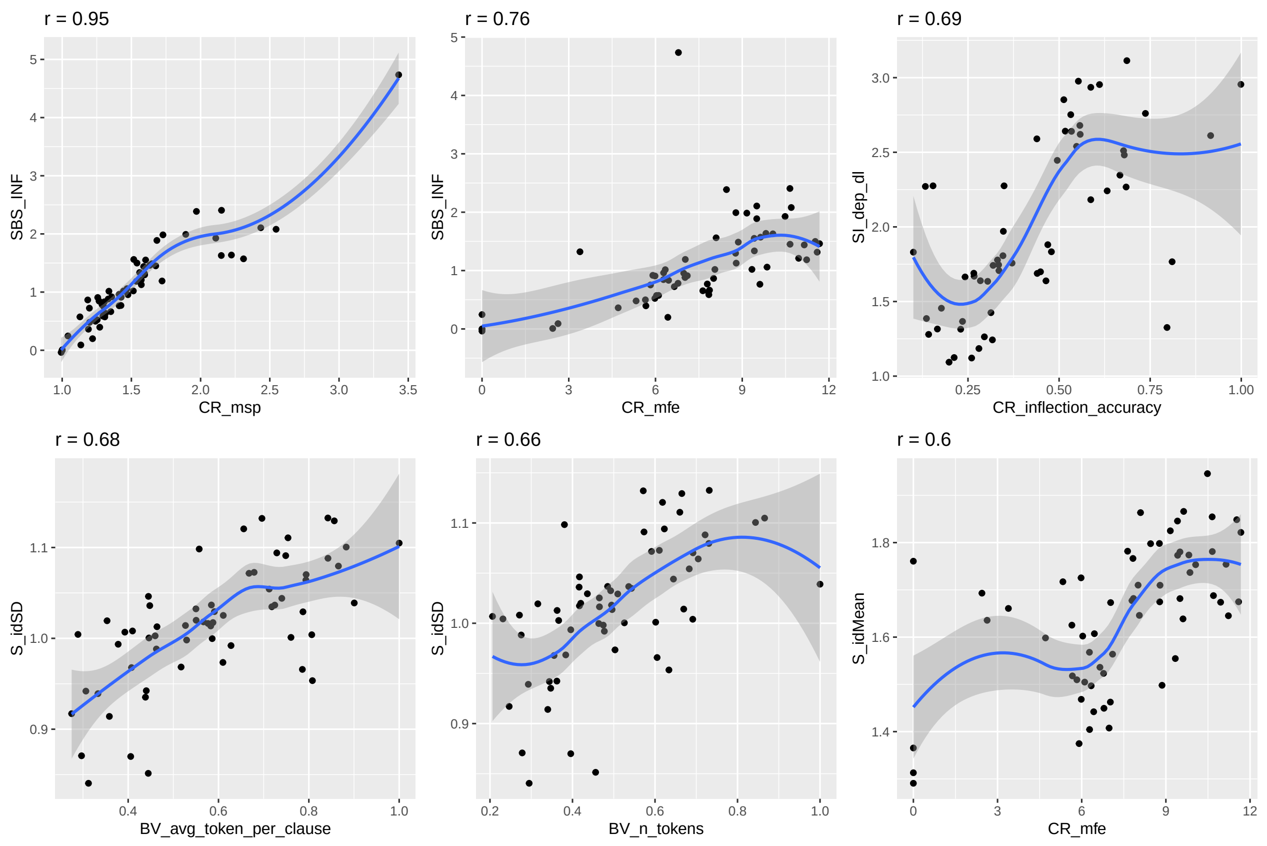


```
ggsave("~/Github/ComplexityMetaAnalyses/Figures/Corrs/TrackA/track_a_plot_corrected.pdf", track.a.plot.
```

TRACK B

Plot the six significant correlations with highest Spearman coefficients for Track B. Warning messages are disabled since there are several NAs that throw errors.

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```



```
ggsave("~/Github/ComplexityMetaAnalyses/Figures/Corrs/TrackB/track_b_plot_corrected.pdf", track.b.plot.
```

Detailed Plots

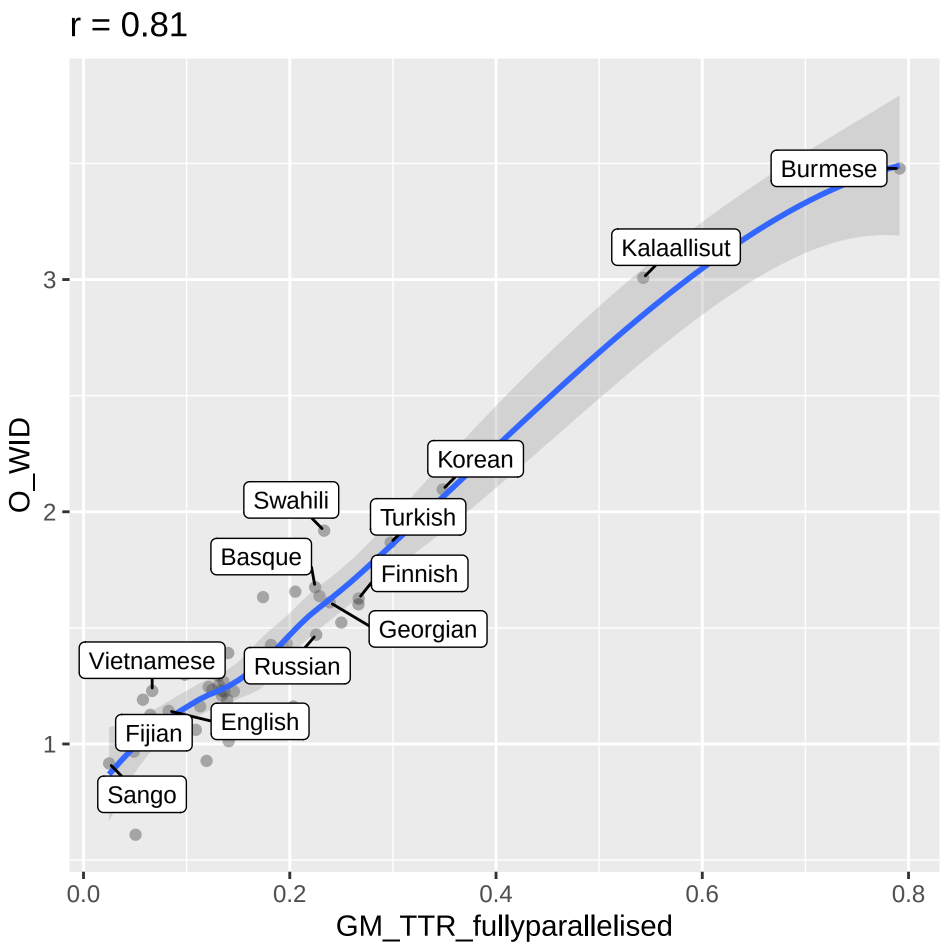
The code below adds labels to the points of plots, which helps with the intepretation of results. We here choose the two plots of Track A and Track B with the highest positive Spearman correlations.

TRACK A

```
#track.a <- track.a[track.a$id != "mya", ] # remove the outlier Burmese (mya)

track.a.plot1.detailed <- ggplot(track.a, aes(x = GM_TTR_fullyparallelised, y = O_WID)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = loess, alpha = 0.3) +
  geom_label_repel(data = track.a[track.a$language == "Fijian" | track.a$language == "Sango" | track.a$
                  min.segment.length = 0,
                  #nudge_x = 0.1,
                  aes(label = language),
                      size = 3) +
  labs(title = paste("r = ",
                    round(track.a.df[track.a.df$row == "GM_TTR_fullyparallelised" & track.a.df$col ==
                    sep = "")) +
  theme(legend.position = "none")
track.a.plot1.detailed
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
ggsave("~/Github/ComplexityMetaAnalyses/Figures/Corrs/TrackA/track_a_plot1_detailed.pdf", track.a.plot1
```

```
## `geom_smooth()` using formula 'y ~ x'
```

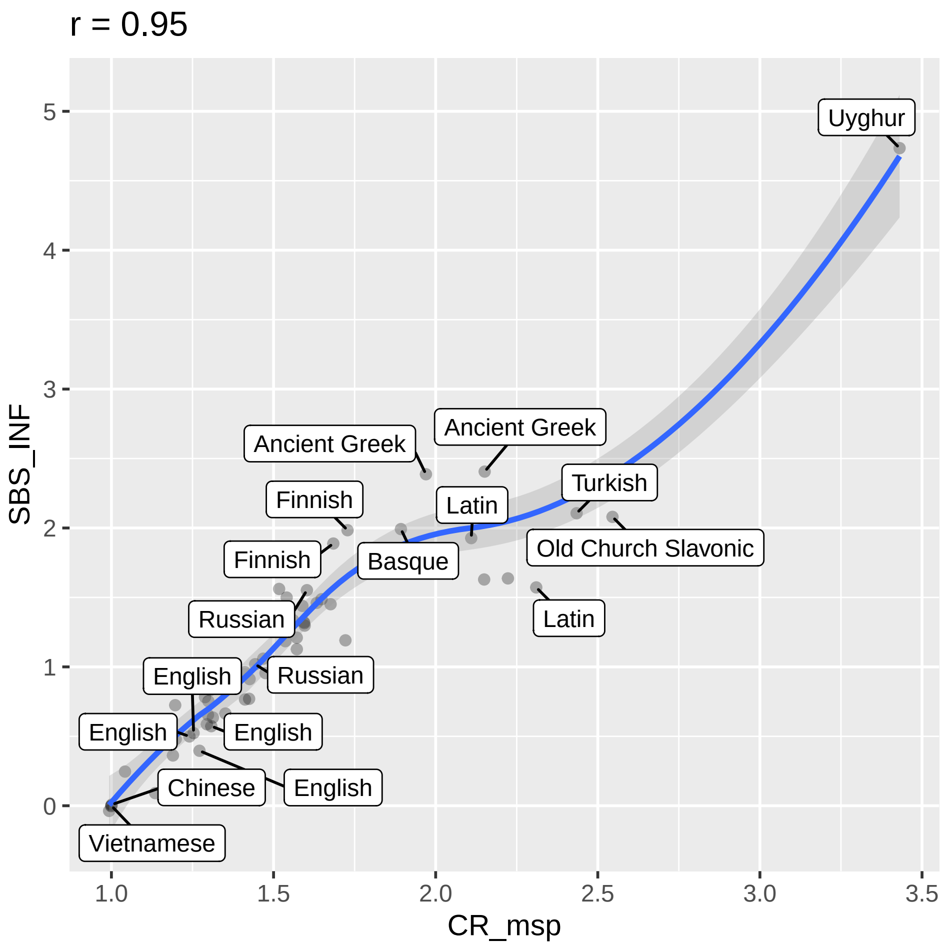
Some comments: This plot shows that the Type-Token Ratio (TTR) and the Word Information Density (WID) are highly correlated across the languages of the Parallel Bible Corpus sample. Burmese (mya) is an outlier here with very high TTR and WID. This is an artifact of the writing system, since it does not delimit orthographic words by white spaces, but rather phrases. For Kalaallisut, on the other hand, the result makes sense (if we accept the latinized writing proposed for this language). Some of the low TTR languages include Sango (sag), Fijian (fij), Thai (tha), and Yoruba (yor).

TRACK B

```
#track.b <- track.b[track.b$id != "uig", ] # remove the outlier Uyghur (uig)

track.b.plot1.detailed <- ggplot(track.b, aes(x = CR_msp, y = SBS_INF)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = loess, alpha = 0.3) +
  geom_label_repel(data = track.b[track.b$language == "Chinese" | track.b$language == "Vietnamese" | tr
                   min.segment.length = 0,
                   #nudge_x = 0.1,
                    aes(label = language),
                    size = 3) +
  labs(title = paste("r = ",
                      round(track.b.df[track.b.df$row == "CR_msp" & track.b.df$col == "SBS_INF", ]$corr,
                      sep = "")) +
  theme(legend.position = "none")
track.b.plot1.detailed
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
ggsave("~/Github/ComplexityMetaAnalyses/Figures/Corrs/TrackB/track_b_plot1_detailed.pdf", track.b.plot1
```

```
## `geom_smooth()` using formula 'y ~ x'
```

Some comments: This plot shows the correlation between the so-called Mean Size of Morphological Paradigms (MSP), which is defined by CR as "simply the number of word-form types divided by the number of lemma types", and the difference in unigram entropy of word tokens in the original texts and the lemmatized texts (INF) as defined by SBS. It is certainly not unexpected, but reassuring, to see these measure highly correlated. The outlier to the high end Uyghur (uig) is likely *not* an artifact, as this language indeed has many productive morphological paradigms. Other languages to the high end of morphological complexity include Ancient Greek (grc), Classical Latin (lat), Turkish (tur), and Old Church Slavonic (chu). Languages to the low end are Vietnamese (vie), Indonesian (ind), Mandarin Chinese (cmn), and Afrikaans (afr). Note that the very low morphological complexity scores of Korean (kor) are an artifact of the way the Korean data is presented in the UD. Namely, the "lemmas" given for Korean are actually merely morphologically segmented forms rather than inflectionally neutralized forms as for the other languages. Thus, it makes sense that the MSP is exactly 1 and the INF is 0.

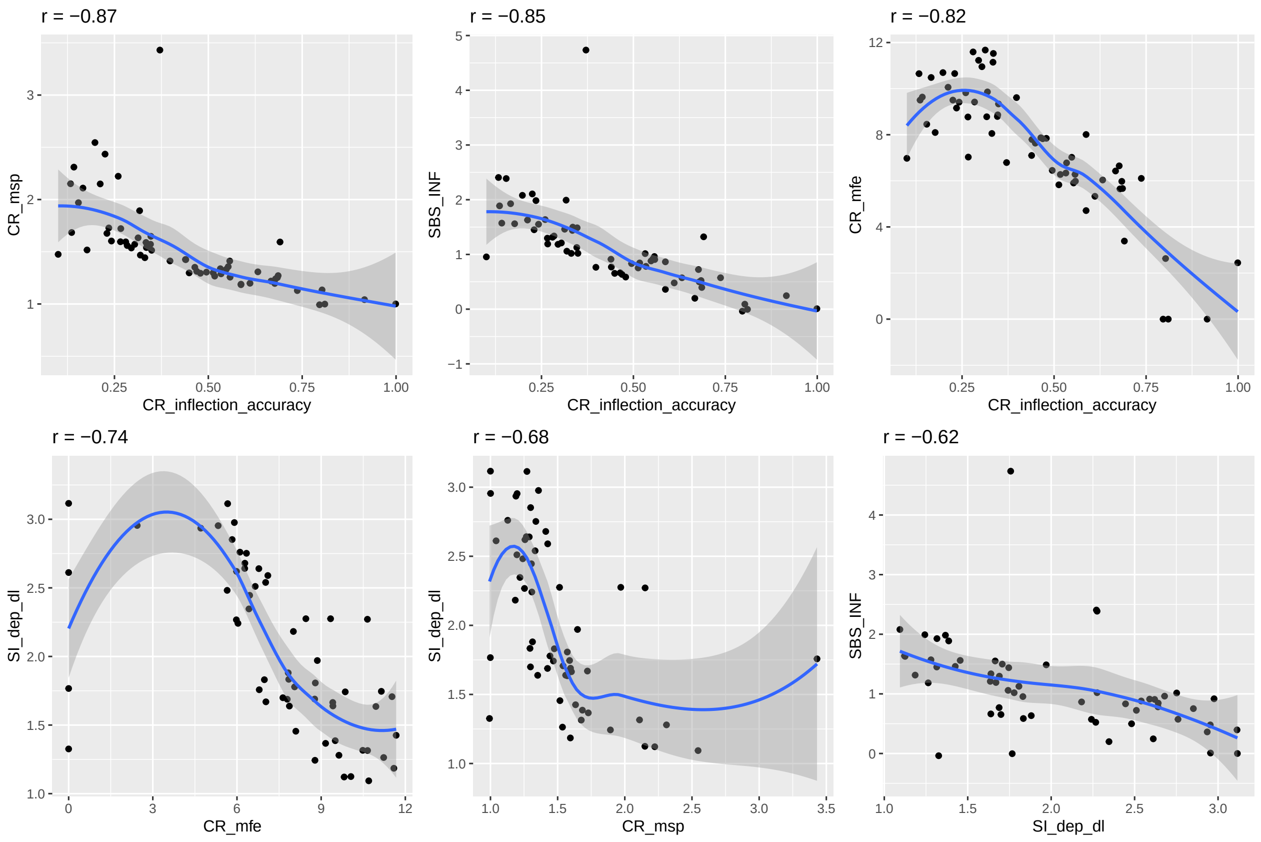
Negative Correlations

I here plot the six highest *negative* correlations (in terms of Spearman coefficients) which are still significant after the Bonferroni correction *and* which are found between measures proposed by *different participants* (there are many measures by the same participants that highly correlate). These are hand-picked from the lists above (could also be implemented more elegantly).

TRACK B (TRACK A does not yield negative correlations)

Plot the six significant correlations with lowest (i.e. negative) Spearman coefficients for Track B. Warning messages are disabled since there are several NAs that throw errors.

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```



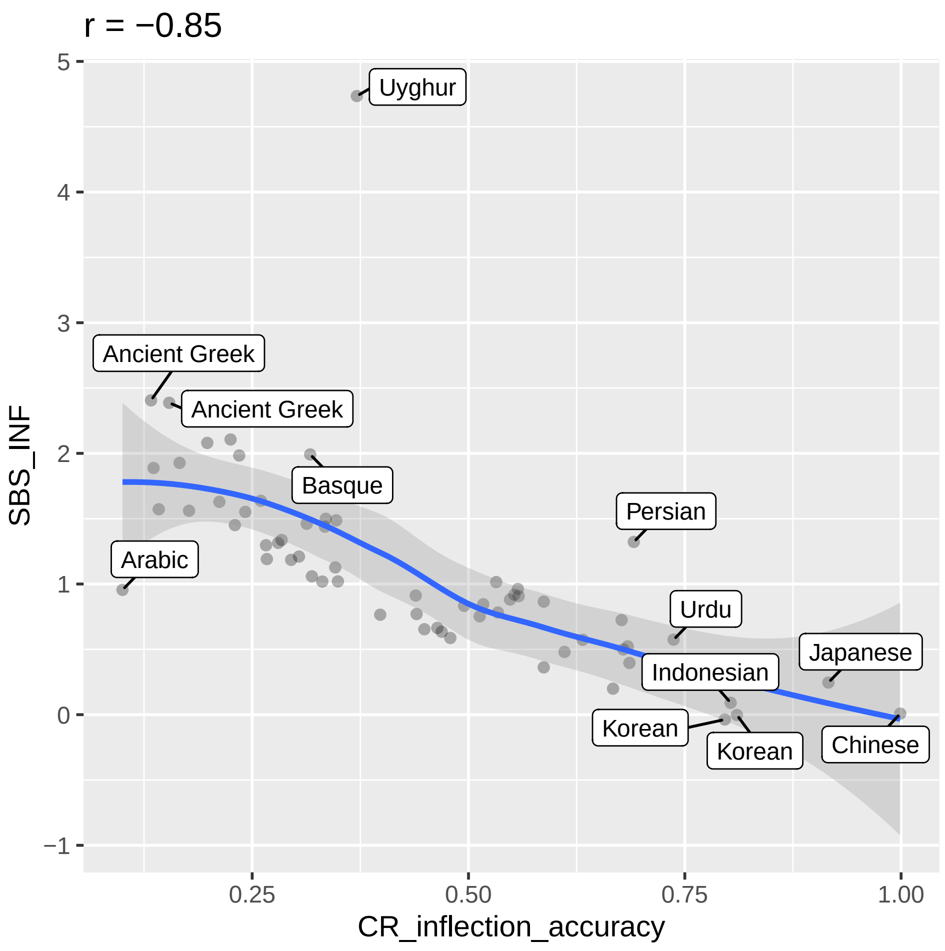
```
ggsave("~/Github/ComplexityMetaAnalyses/Figures/Corrs/TrackB/track_b_plot_negative_corrected.pdf", track
```

Detailed Plots

```
#track.b <- track.b[track.b$id != "uig", ] # remove the outlier Uyghur (uig)

track.b.plot1.negative.detailed <- ggplot(track.b, aes(x = CR_inflection_accuracy, y = SBS_INF)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = loess, alpha = 0.3) +
  geom_label_repel(min.segment.length = 0,
                   #nudge_x = 0.1,
                   aes(label = language),
                   size = 3) +
  labs(title = paste("r = ",
                     round(track.b.df[track.b.df$row == "CR_inflection_accuracy" & track.b.df$col == "SI
                     sep = "")) +
  theme(legend.position = "none")
track.b.plot1.negative.detailed
```

```
## `geom_smooth()` using formula 'y ~ x'
```



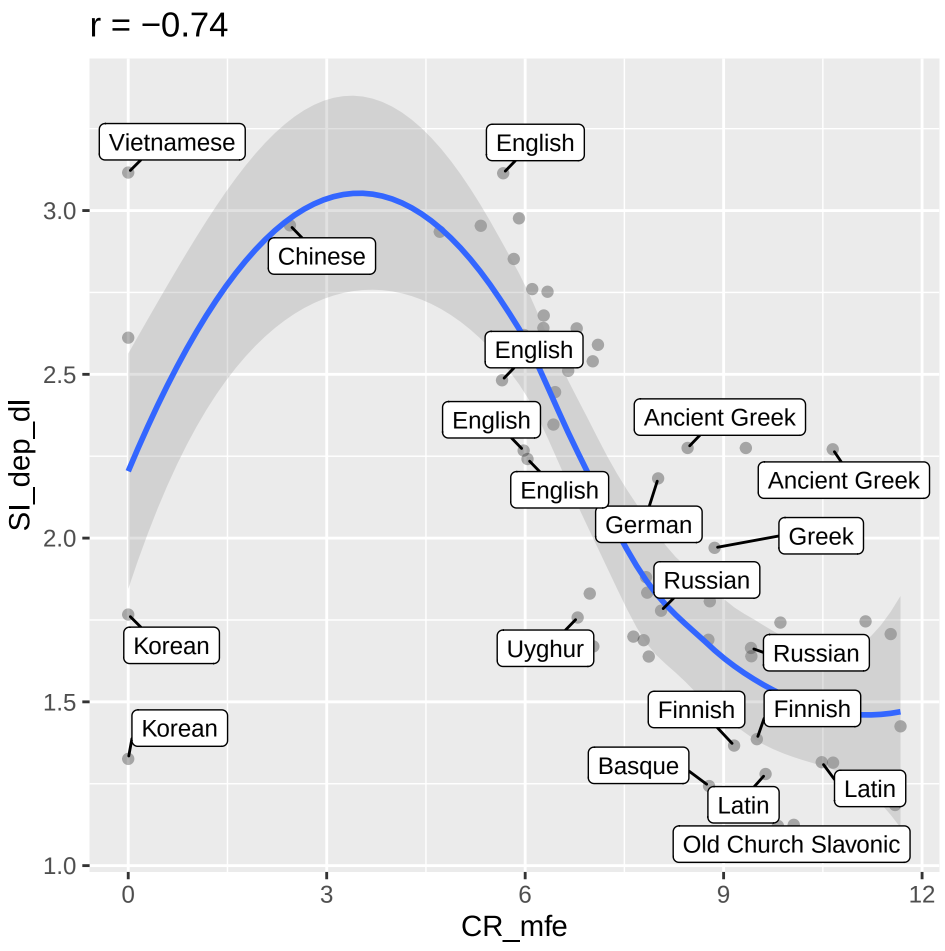
```
ggsave("~/Github/ComplexityMetaAnalyses/Figures/Corrs/TrackB/track_b_plot1_negative_detailed.pdf", track
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
#track.b <- track.b[track.b$language != "Korean", ] # remove the outlier Korean
```

```
track.b.plot2.negative.detailed <- ggplot(track.b, aes(x = CR_mfe, y = SI_dep_dl)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = loess, alpha = 0.3) +
  geom_label_repel(data = track.b[track.b$language == "Chinese" | track.b$language == "Vietnamese" | tra
                  min.segment.length = 0,
                  #nudge_x = 0.1,
                  aes(label = language),
                  size = 3) +
  labs(title = paste("r = ",
                   round(track.b.df[track.b.df$row == "CR_mfe" & track.b.df$col == "SI_dep_dl", ]$cor
                   sep = "")) +
  theme(legend.position = "none")
track.b.plot2.negative.detailed
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
ggsave("~/Github/ComplexityMetaAnalyses/Figures/Corrs/TrackB/track_b_plot2_negative_detailed.pdf", track
```

```
## `geom_smooth()` using formula 'y ~ x'
```

Conclusions

Some more general observations based on these analyses include:

- Many of the measures proposed by the same participants highly correlate. This is the case, for instance, for the measures proposed by GM in Track A, but also measures of BV and SI in Track B. In the case of GM, this is because many of the measures are virtually the same, but with minor shades of modification. In the case of BV, while at first sight the measures seem to conceptually differ, they essentially boil down to the same underlying causes. For example, the number of tokens in a sentence highly predicts the average maximal depth of a tree over the sentence. In the case of SI, the highly correlating measures in fact only have very few data points (only two or three in some cases). So, arguably all of these intra-participant correlations are somewhat artificial in the sense that they are either redundant or driven by inappropriate sample size.
- There are several strong positive correlations between simple measures relating to the number of types and tokens (GM_TTR_fullyparallelised, BV_n_tokens, etc.), and measures of information density (O_WID, S_idSD). Interestingly, this is the case for both tracks, since Oh used the Bible texts, and Semenuks used the UD. Information density is generally assumed to be a measure of "syntax" that has psycholinguistic relevance in terms of language processing. However, the fact that it is highly predictable by some of the simplest word frequency measures potentially goes to show that the underlying reasons for complexity are ironically quite simple.
- We have mainly discussed positive correlations here, meaning that certain measures are essentially (better or worse) replacements of other measures. In fact, the majority of correlations still significant after the Bonferroni correction are positive (in Track A all of them are positive, in Track B 37 out of 49 are positive). Some of the negative correlations we do find are between CR's "inflection accuracy" and different measures of inflectional complexity (CR_mfe, SBS_INF, CR_msp). This makes perfect sense given that inflection accuracy is a measure that reflects the difficulty of NLP tools to automatically deal with inflectional morphology. The more complex the morphology, the lower the accuracy of the automated tool. A negative correlation that seems both robust and potentially interesting is that the dependency lengths in noun phrases with marked possessives (SI_dep_dl) apparently are in a clear trade-off with different measures of inflectional complexity. However, the fact that there are few such instances of robust negative correlations between measures of different domains suggests that there are relatively few clear complexity trade-offs. This is sth. to further think about.