Tracegram analysis

Stefano Coretta

February 21, 2017

1 Data import

First, we import all degg-tracing files contained in the results folder into the degg data frame, and we merge with languages and words (the first data frame list the languages of each participant, the latter the variables for each word). Two separate data frames for Italian and Polish are then created.

```
degg <- list.files(path = "pilot/results",
                   pattern = "*-degg-tracing.csv",
                   full.names = TRUE) %>%
    map_df(~read_csv(.))
```

```
## Parsed with column specification:
## cols(
##   speaker = col_character(),
##   file = col_character(),
##   word = col_character(),
##   abs.time = col_double(),
##   time = col_double(),
##   maximum = col_double(),
##   minimum = col_double()
## )
## Parsed with column specification:
## cols(
##   speaker = col_character(),
##   file = col_character(),
##   word = col_character(),
##   abs.time = col_double(),
##   time = col_double(),
##   maximum = col_double(),
##   minimum = col_double()
## )
## Parsed with column specification:
## cols(
##   speaker = col_character(),
##   file = col_character(),
##   word = col_character(),
##   abs.time = col_double(),
##   time = col_double(),
##   maximum = col_double(),
##   minimum = col_double()
## )
```

```
languages <- read_csv("./pilot/stimuli/languages.csv")
```

```
## Parsed with column specification:
## cols(
##   speaker = col_character(),
##   language = col_character()
## )
```

```
words <- read_csv("./pilot/stimuli/nonce.csv")
```

```
## Parsed with column specification:
## cols(
##   item = col_integer(),
##   word = col_character(),
##   ipa = col_character(),
##   c1 = col_character(),
##   c1phonation = col_character(),
##   vowel = col_character(),
##   anteropost = col_character(),
##   height = col_character(),
##   c2 = col_character(),
##   c2phonation = col_character(),
##   c2place = col_character(),
##   language = col_character()
## )
```

```
degg <- left_join(degg, languages) %>%
    left_join(y = words) %>%
    mutate_if(is.character, as.factor)
```

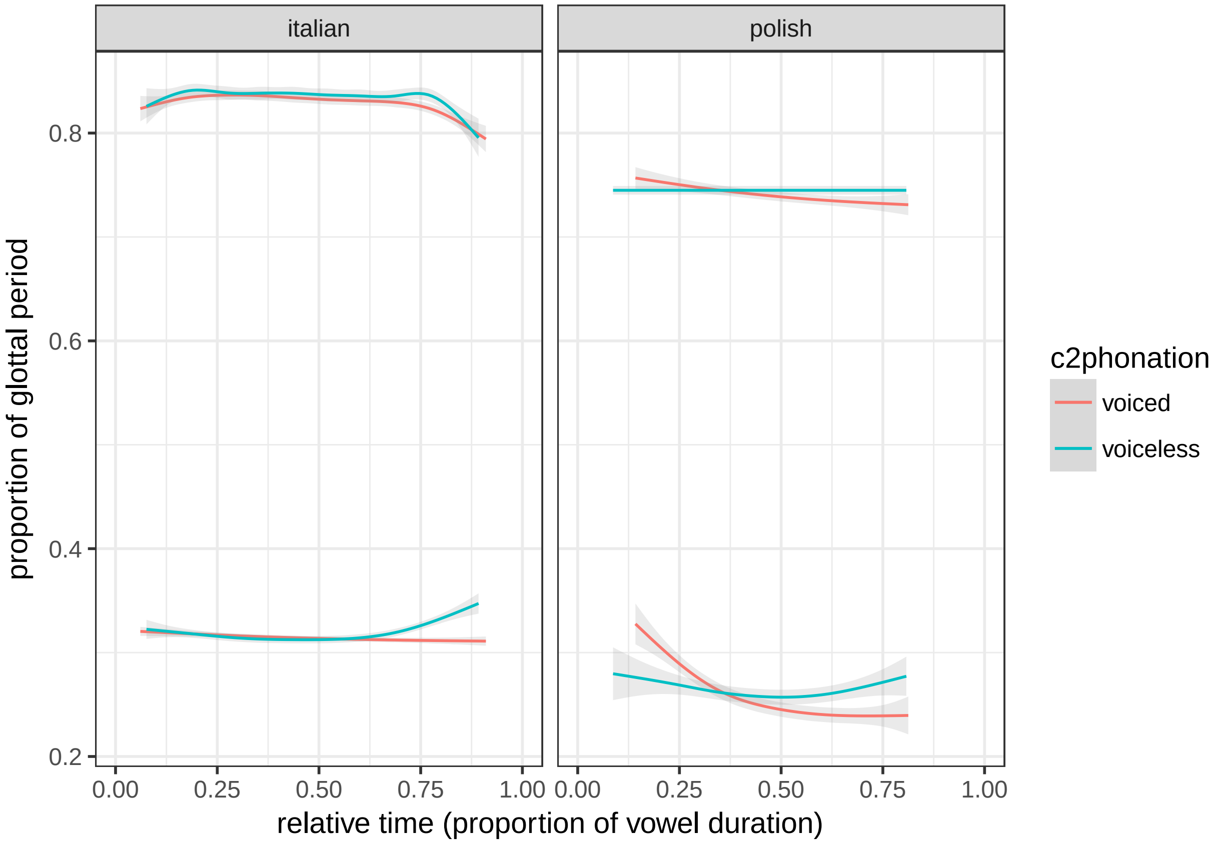
```
## Joining, by = "speaker"

## Joining, by = c("word", "language")
```

```
degg.it <- filter(degg, language == "italian")
degg.pl <- filter(degg, language == "polish")
```

We can now plot smoothers for DEGG maxima and minima for both languages. The x-axis shows the normalised time as a proportion of the vowel duration, while values on the y-axix indicates the proportion of the glottal period at x time. The lines in the lower half of the graphs are the DEGG maxima, while the lines in the upper half are the minima. The second half of the vowel proportion contains information important for the effects of the phonation of the consonant following the vowel.

```
ggplot(degg, aes(x = time, colour = c2phonation)) +
    facet_grid(. ~ language) +
    geom_smooth(aes(y = maximum), size = 0.5, alpha = 0.2) +
    geom_smooth(aes(y = minimum), size = 0.5, alpha = 0.2) +
    xlab("relative time (proportion of vowel duration)") +
    ylab("proportion of glottal period") +
    xlim(0, 1)
```

```
## `geom_smooth()` using method = 'gam'
## `geom_smooth()` using method = 'gam'
```



As it can be seen from the graph, both Italian and Polish show a raising profile of the DEGG maxima in the voiceless condition in the second half of the vowel (Polish data are a bit more messy, but can still be interpreted this way).

2 SSANOVA for Italian

We can now calculate the SSANOVA model for dEGG maxima and minima.

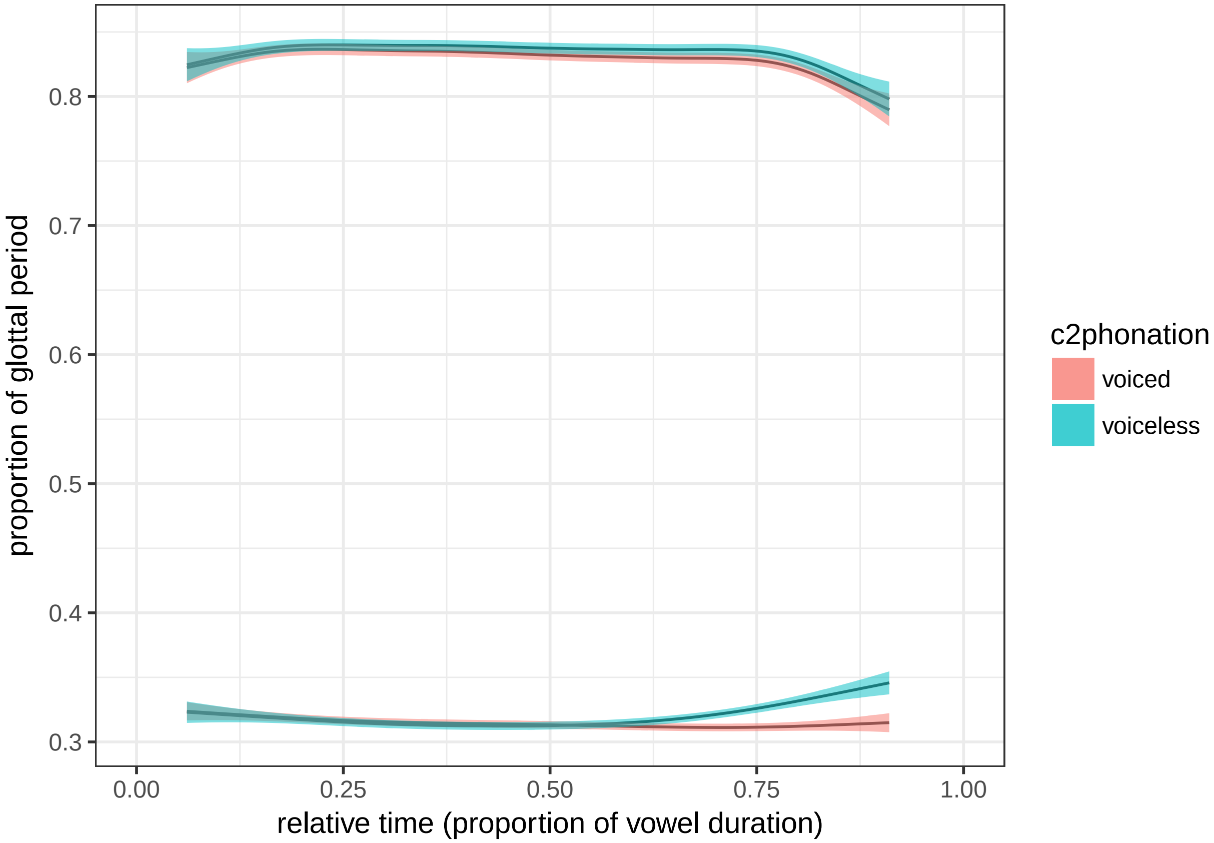
```
maximum.model <- ssanova(maximum ~ c2phonation + time + c2phonation:time,
data = degg.it)

minimum.model <- ssanova(minimum ~ c2phonation + time + c2phonation:time,
data = degg.it)
```

And calculate the predicted splines based on the SSANOVA models.

```
predicted.degg <- expand.grid(time = seq(min(degg.it$time), max(degg.it$time), length = 100),
                    c2phonation = c("voiced", "voiceless"))

predicted.degg$maximum.fit <- predict(maximum.model, predicted.degg, se = TRUE)$fit
predicted.degg$maximum.SE <- predict(maximum.model, predicted.degg, se = TRUE)$se.fit
predicted.degg$minimum.fit <- predict(minimum.model, predicted.degg, se = TRUE)$fit
predicted.degg$minimum.SE <- predict(minimum.model, predicted.degg, se = TRUE)$se.fit
```

Finally, we can plot the predicted splines for Italian with 95% confidence levels for significance evaluation.

```
ggplot(predicted.degg, aes(x = time, colour = c2phonation, group = c2phonation)) +
    geom_line(aes(y = maximum.fit), alpha = 1, colour = "grey20") +
    geom_ribbon(aes(ymin = maximum.fit-(1.96*maximum.SE),
                ymax = maximum.fit+(1.96*maximum.SE),
                fill = c2phonation ),alpha = 0.5,colour = "NA") +
    geom_line(aes(y = minimum.fit), alpha = 1, colour = "grey20") +
    geom_ribbon(aes(ymin = minimum.fit-(1.96*minimum.SE),
                    ymax = minimum.fit+(1.96*minimum.SE),
                    fill = c2phonation ),alpha = 0.5,colour = "NA") +
    xlab("relative time (proportion of vowel duration)") +
    ylab("proportion of glottal period") +
    xlim(0, 1)
```



3 SSANOVA for Polish

```
maximum.model <- ssanova(maximum ~ c2phonation + time + c2phonation:time,
data = degg.pl)

minimum.model <- ssanova(minimum ~ c2phonation + time + c2phonation:time,
data = degg.pl)
```

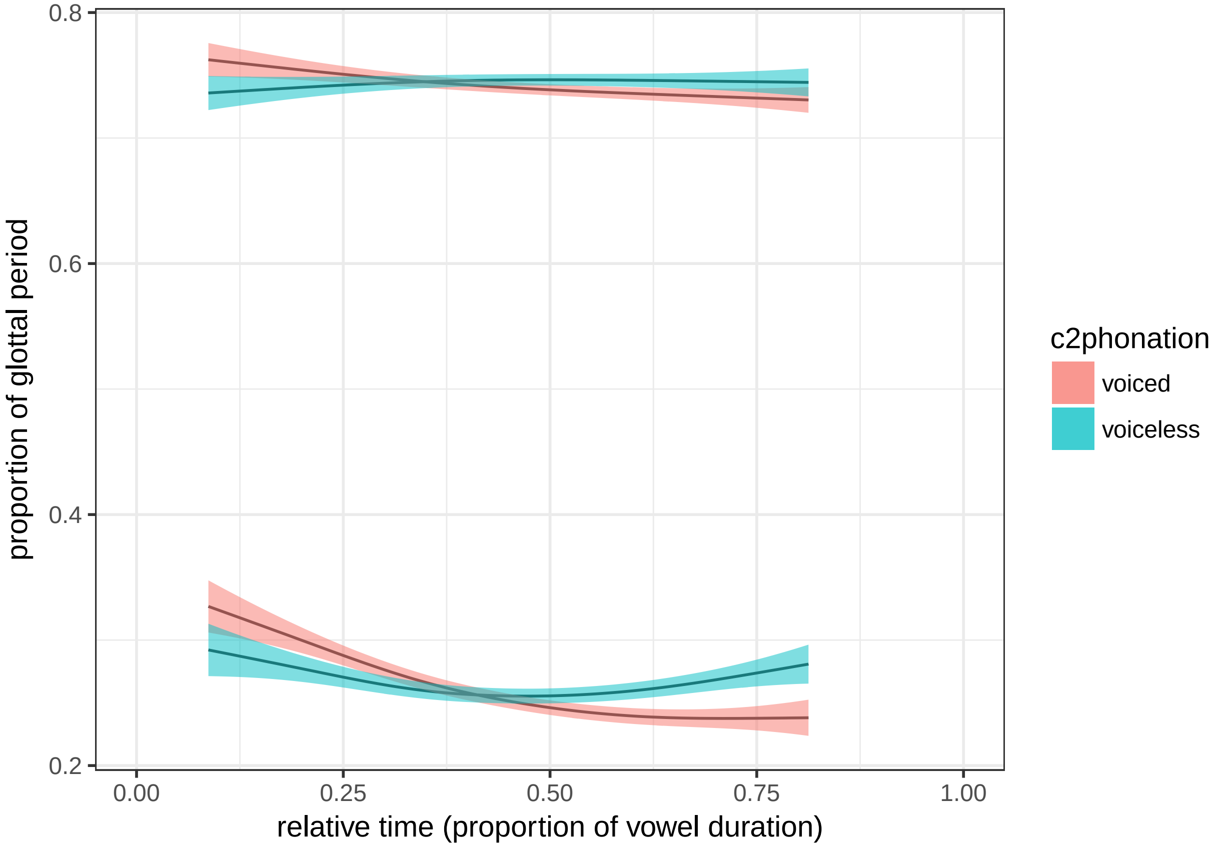
And calculate the predicted splines based on the SSANOVA models.

```
predicted.degg <- expand.grid(time = seq(min(degg.pl$time), max(degg.pl$time), length = 100),
                    c2phonation = c("voiced", "voiceless"))
```

```
predicted.degg$maximum.fit <- predict(maximum.model, predicted.degg, se = TRUE)$fit
predicted.degg$maximum.SE <- predict(maximum.model, predicted.degg, se = TRUE)$se.fit
predicted.degg$minimum.fit <- predict(minimum.model, predicted.degg, se = TRUE)$fit
predicted.degg$minimum.SE <- predict(minimum.model, predicted.degg, se = TRUE)$se.fit
```

Finally, we can plot the predicted splines for Polish with 95% confidence levels for significance evaluation.

```
ggplot(predicted.degg, aes(x = time, colour = c2phonation, group = c2phonation)) +
    geom_line(aes(y = maximum.fit), alpha = 1, colour = "grey20") +
    geom_ribbon(aes(ymin = maximum.fit-(1.96*maximum.SE),
                ymax = maximum.fit+(1.96*maximum.SE),
                fill = c2phonation ),alpha = 0.5,colour = "NA") +
    geom_line(aes(y = minimum.fit), alpha = 1, colour = "grey20") +
    geom_ribbon(aes(ymin = minimum.fit-(1.96*minimum.SE),
                    ymax = minimum.fit+(1.96*minimum.SE),
                    fill = c2phonation ),alpha = 0.5,colour = "NA") +
    xlab("relative time (proportion of vowel duration)") +
    ylab("proportion of glottal period") +
    xlim(0, 1)
```


proportion of glottal period
relative time (proportion of vowel duration)
c2phonation
voiced
voiceless