sad

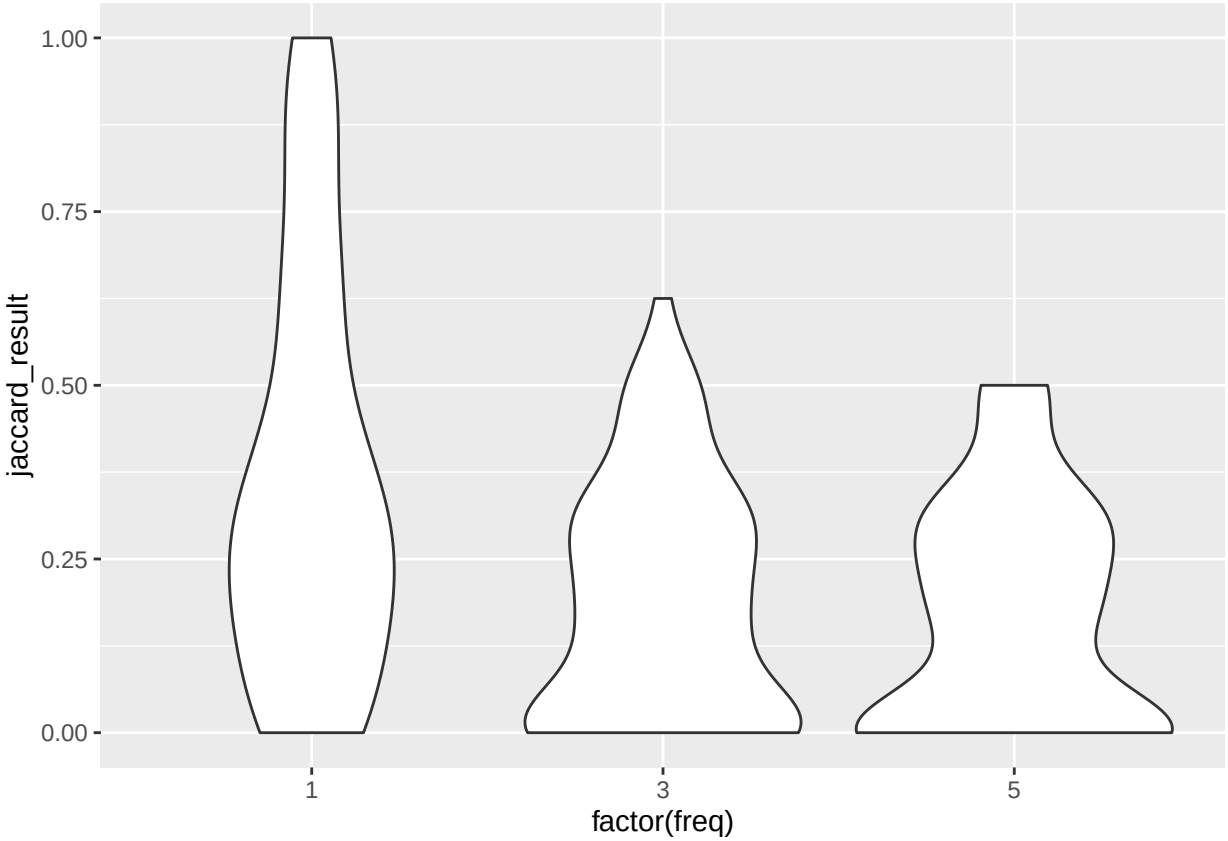
Marcin Stopka, Franciszek Sobota, Grzegorz Dolanowski

2026-01-10

Effect of sampling frequency

We can see on the plots that if time between sampling increases, the results get worse.

```
ggplot(data, aes(x=factor(freq), y=jaccard_result)) +
  geom_violin()
```



We can test if the correlation is negative and we get

```
tst <- cor.test(data$freq, data$jaccard_result, method = "spearman")
```

```
## Warning in cor.test.default(data$freq, data$jaccard_result, method =
## "spearman"): Cannot compute exact p-value with ties
```

```
tst
```

```
##
##  Spearman's rank correlation rho
```

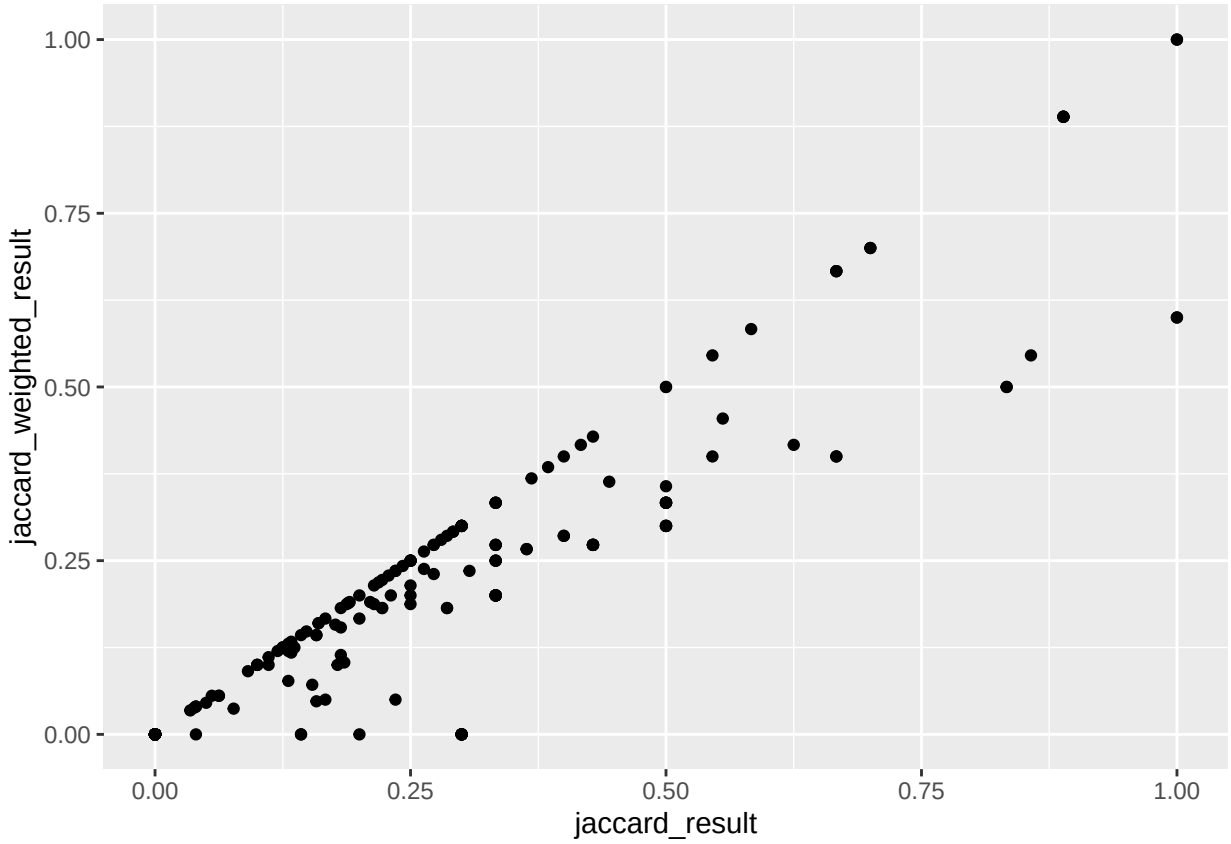
```
##
## data:  data$freq and data$jaccard_result
## S = 3005774, p-value = 1.516e-06
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## -0.304612
```

with low p-value of 1.5155853×10^{-6}‘.

Graph Metrics

We can see that the two graph metrics are very similar on the following plot.

```
ggplot(data, aes(x=jaccard_result, y=jaccard_weighted_result)) + geom_point()
```



they are even exactly the same most of the time

```
mean(data$jaccard_result == data$jaccard_weighted_result)
```

```
## [1] 0.6125
```

Achieved results

The achieved results aren’t very good, the median is low.

```
kable(data.frame(rbind(summary(data$jaccard_result))))
```

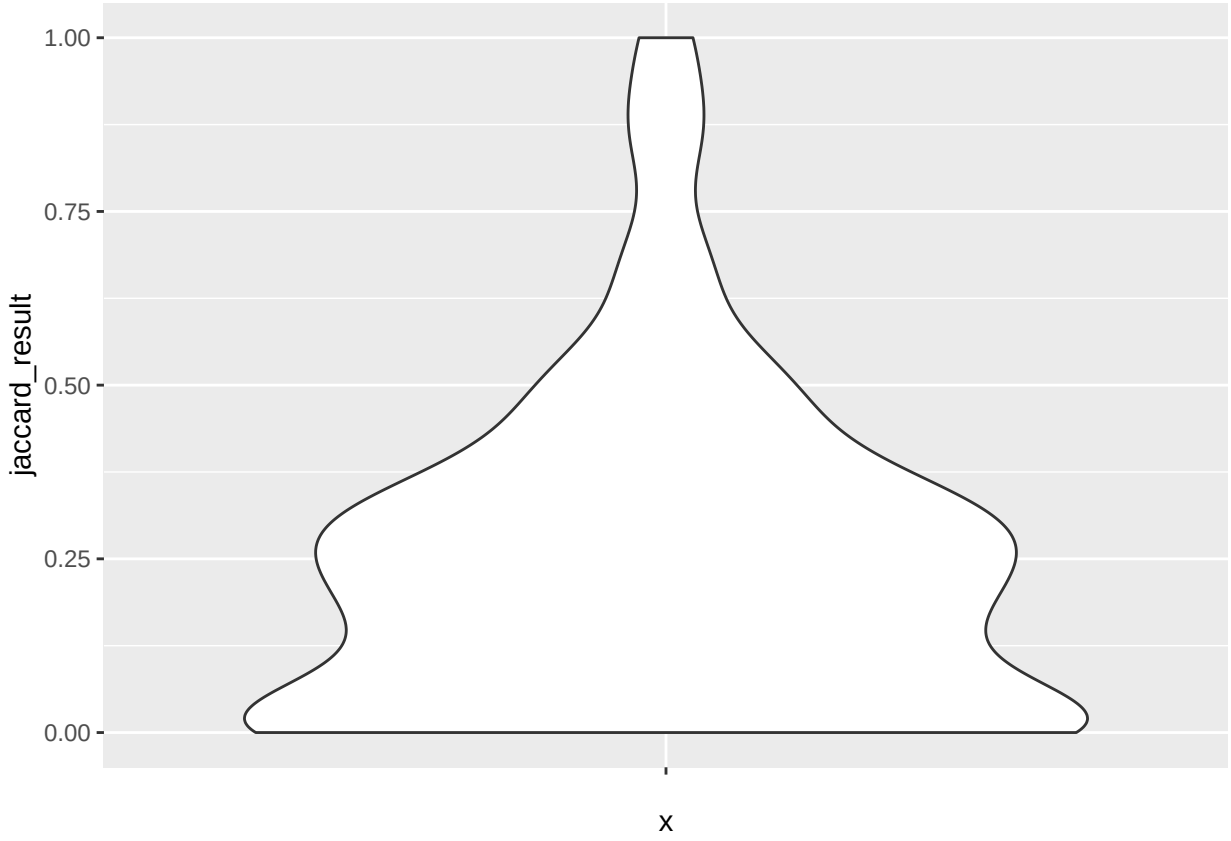
Min.	X1st.Qu.	Median	Mean	X3rd.Qu.	Max.
0	0	0.2	0.2369774	0.3333333	1

We can see that a lot of that is due to the frequency.

```
kable(data %>% group_by(freq) %>% summarise(med=median(jaccard_result)))
```

freq	med
1	0.2828571
3	0.1578947
5	0.1509972

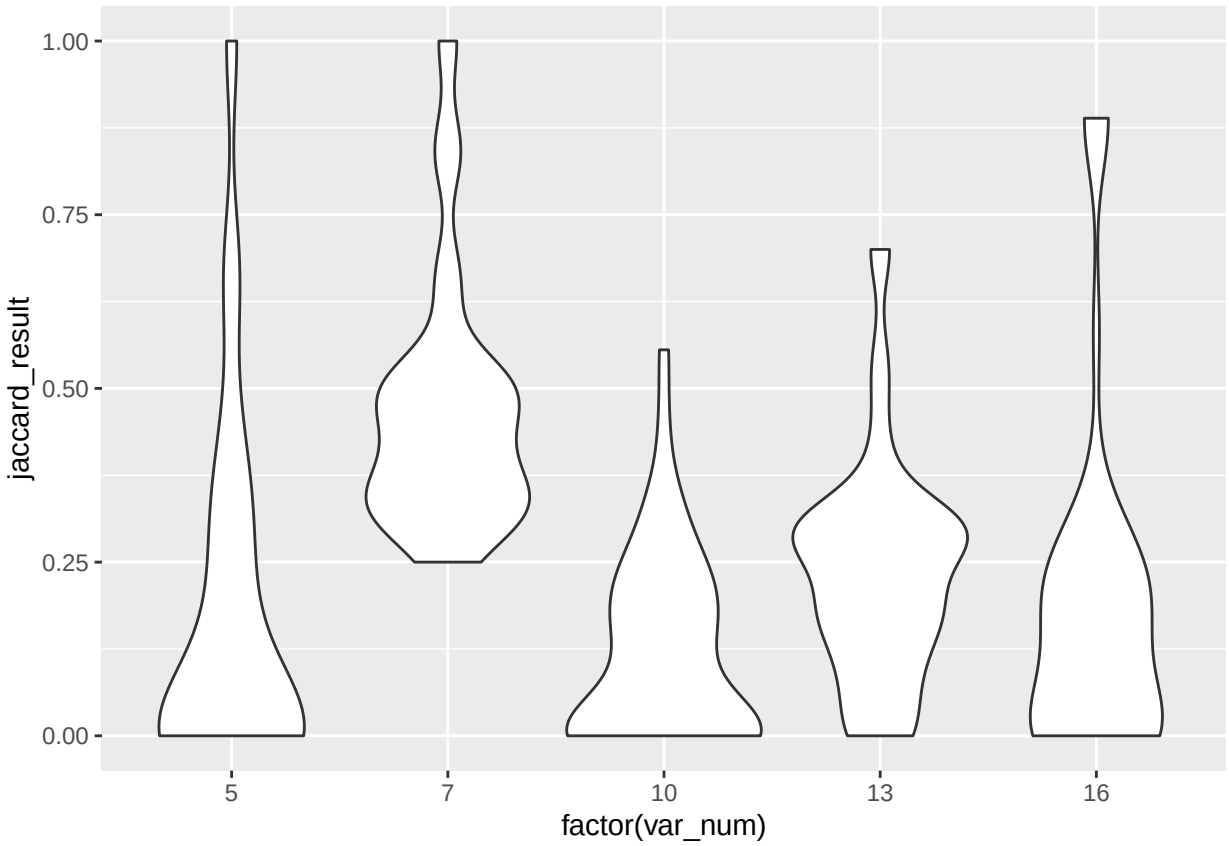
```
ggplot(data, aes(x="", y=jaccard_result)) +
  geom_violin()
```



Effect of number of variables

For some reason the results are best when using 7 variables.

```
ggplot(data, aes(x=factor(var_num), y=jaccard_result)) +
  geom_violin()
```



but the correlation is test does not give anything conclusive

```
cor.test(data$var_num, data$jaccard_result, method = "spearman")
```

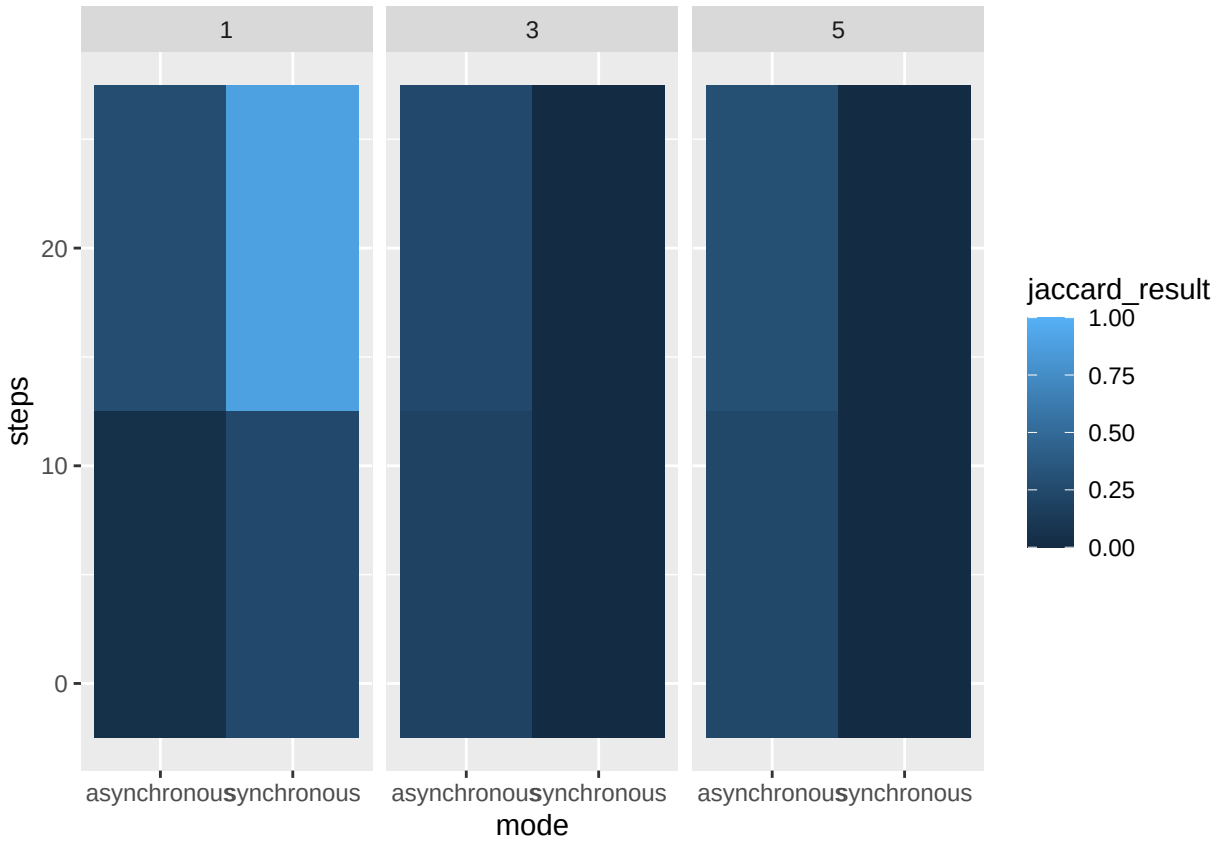
```
## Warning in cor.test.default(data$var_num, data$jaccard_result, method =
## "spearman"): Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  data$var_num and data$jaccard_result
## S = 2551798, p-value = 0.09639
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.1075705
```

Effect of mode

First we can see that the synchronous data is much less resistant to infrequent probes.

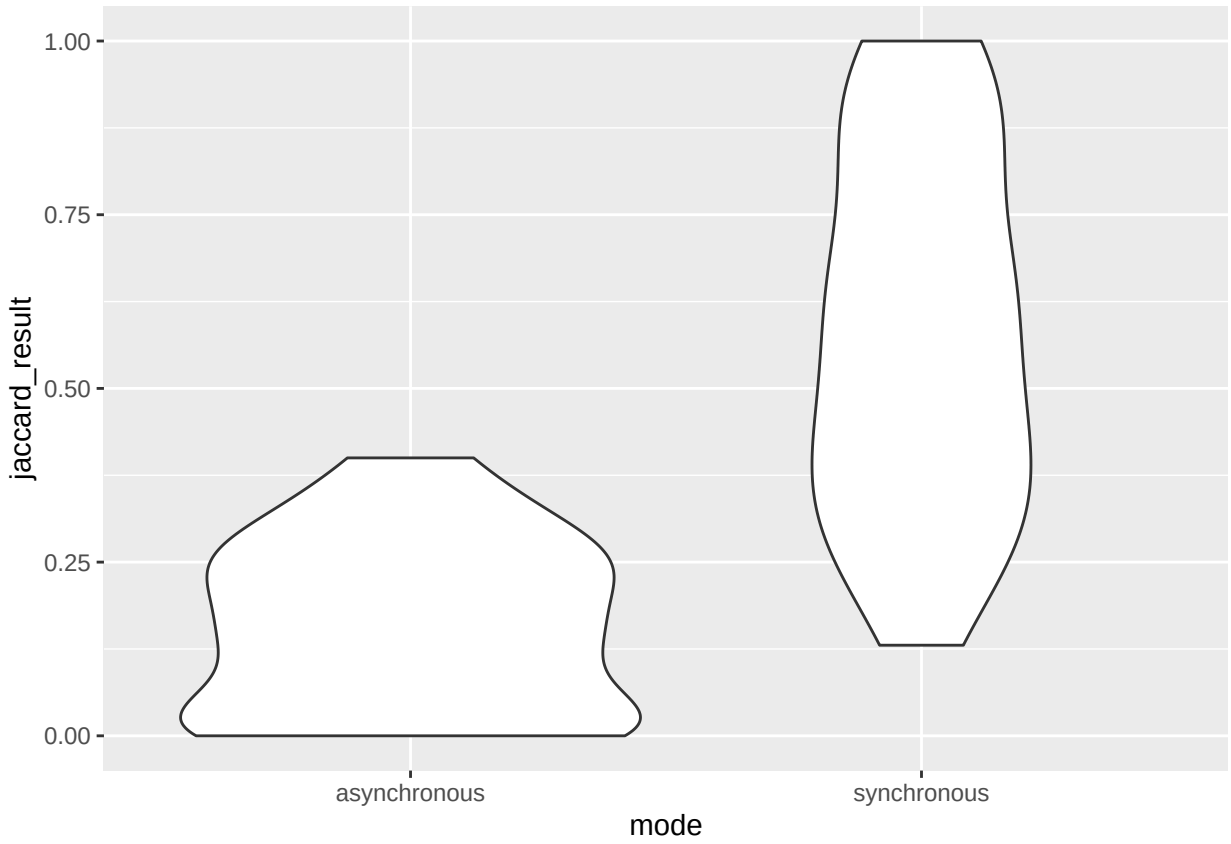
```
ggplot(data, aes(x=mode,y=steps,fill=jaccard_result)) + geom_tile() + facet_wrap(~freq)
```



The asynchronous can deal with it, but synchronous works very bad whenever sampling frequency is not 1.

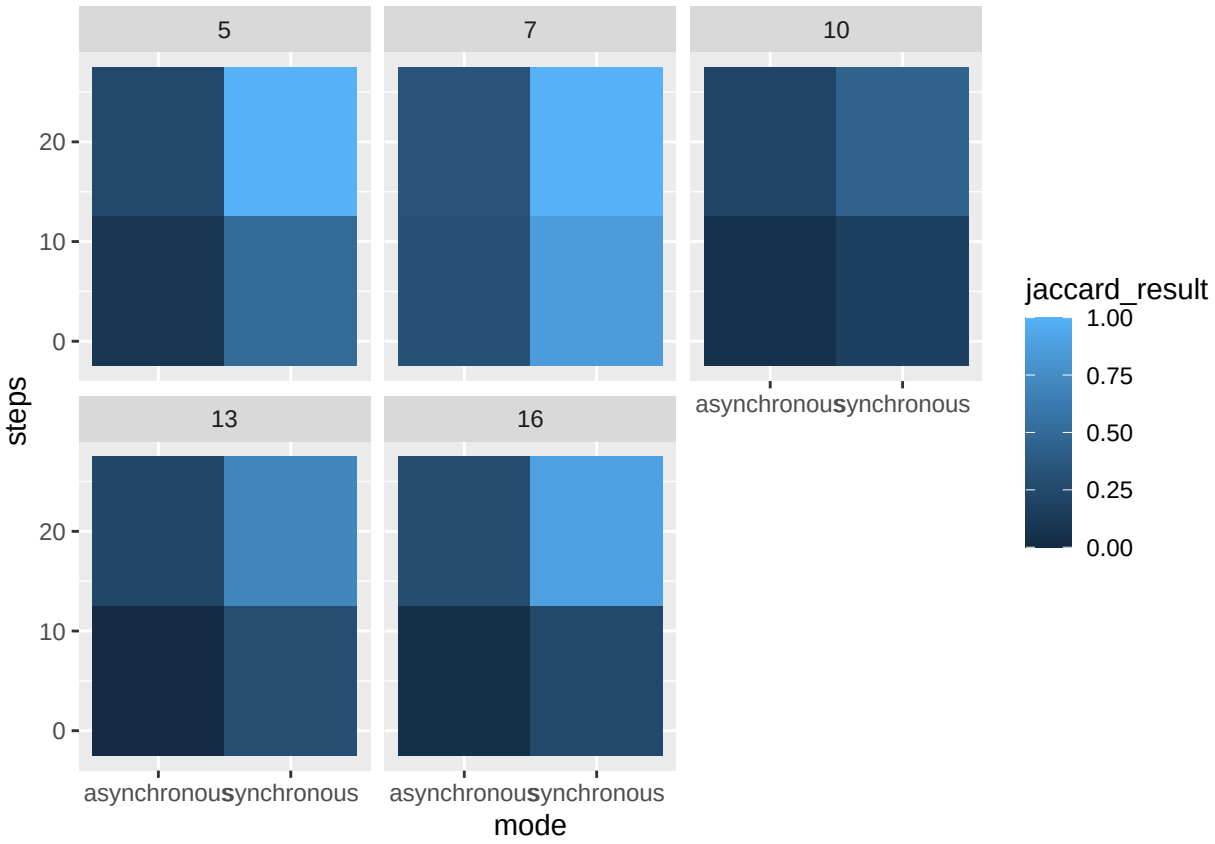
We can see on the following plot that if the frequency is 1, then synchronous data gives generally better results with the given amount of data. It can be because with the same number of transitions more variables get changed.

```
ggplot(data %>% filter(freq==1), aes(x=mode, y=jaccard_result)) +
  geom_violin()
```



If we divide the data by number of variables we can see that with frequency 1, every time the synchronous option is better and the more steps we have, the better.

```
ggplot(data %>% filter(freq==1), aes(x=mode,y=steps,fill=jaccard_result)) + geom_tile() +
    facet_wrap(~var_num)
```



Linear regression

If we fun linear regression with all the data, the results aren't very good. On the other hand, if we filter only by those results that have frequency 1, we get a high R squared.

```
data_with_dummies <- dummy_cols(data) %>% filter(freq==1)
linear_model <- lm(jaccard_result ~
                     var_num + steps + numtraj + attper + mode_asynchronous + score_type_BDE,
                     data_with_dummies)

kable(tidy(linear_model))
```

term	estimate	std.error	statistic	p.value
(Intercept)	0.5433950	0.1530973	3.5493449	0.0006799
var_num	-0.0153794	0.0079192	-1.9420456	0.0559902
steps	0.0148728	0.0056983	2.6100236	0.0109789
numtraj	0.0009427	0.0007741	1.2177312	0.2272477
attper	-0.0643473	0.1869484	-0.3441981	0.7316869
mode_asynchronous	-0.4390521	0.0607926	-7.2221302	0.0000000
score_type_BDE	0.0162628	0.0387058	0.4201649	0.6755992

```
summary(linear_model)$adj.r.squared
```

```
## [1] 0.6525907
```