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
```

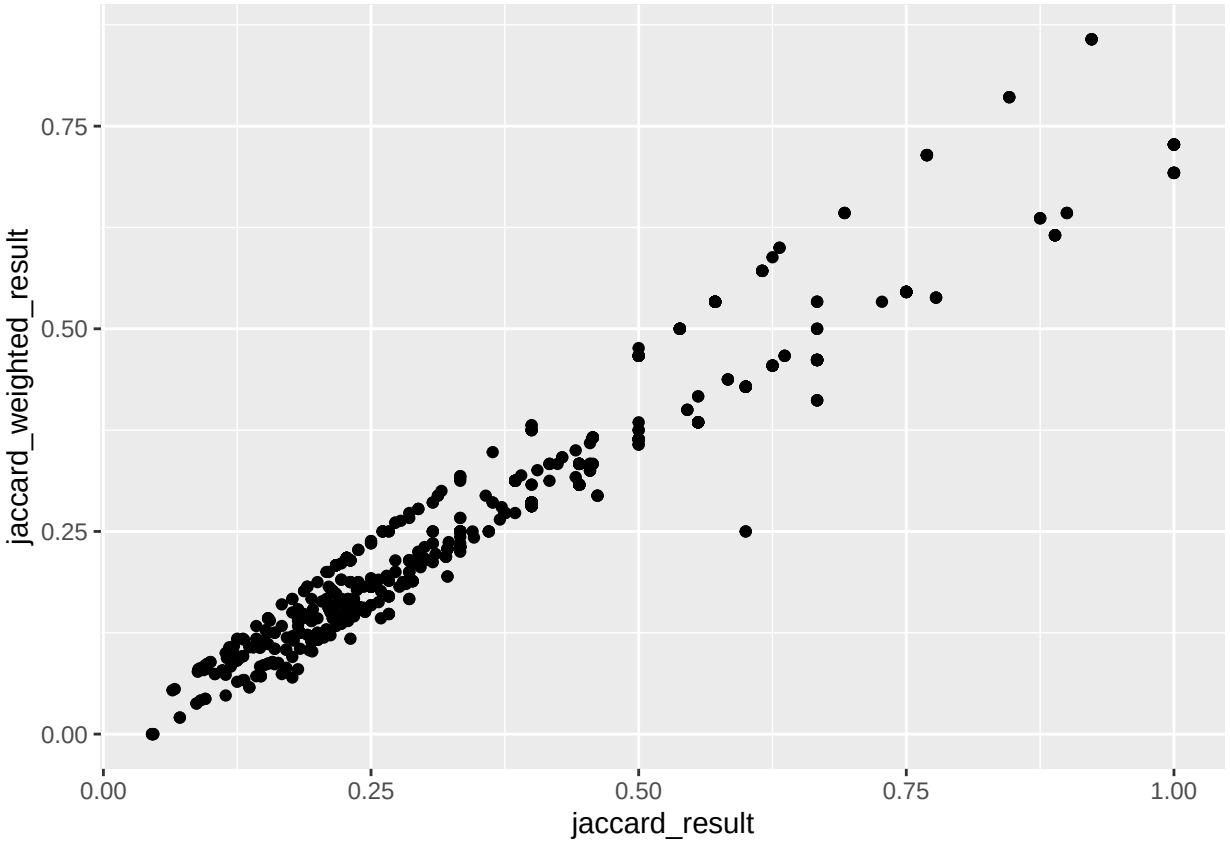
```
##
## data:  data$freq and data$jaccard_result
## S = 72771285, p-value = 4.609e-06
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.1698082
```

with low p-value of 4.6092639×10^{-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()
```



Achieved results

The summary of the achieved results can be seen in the following table

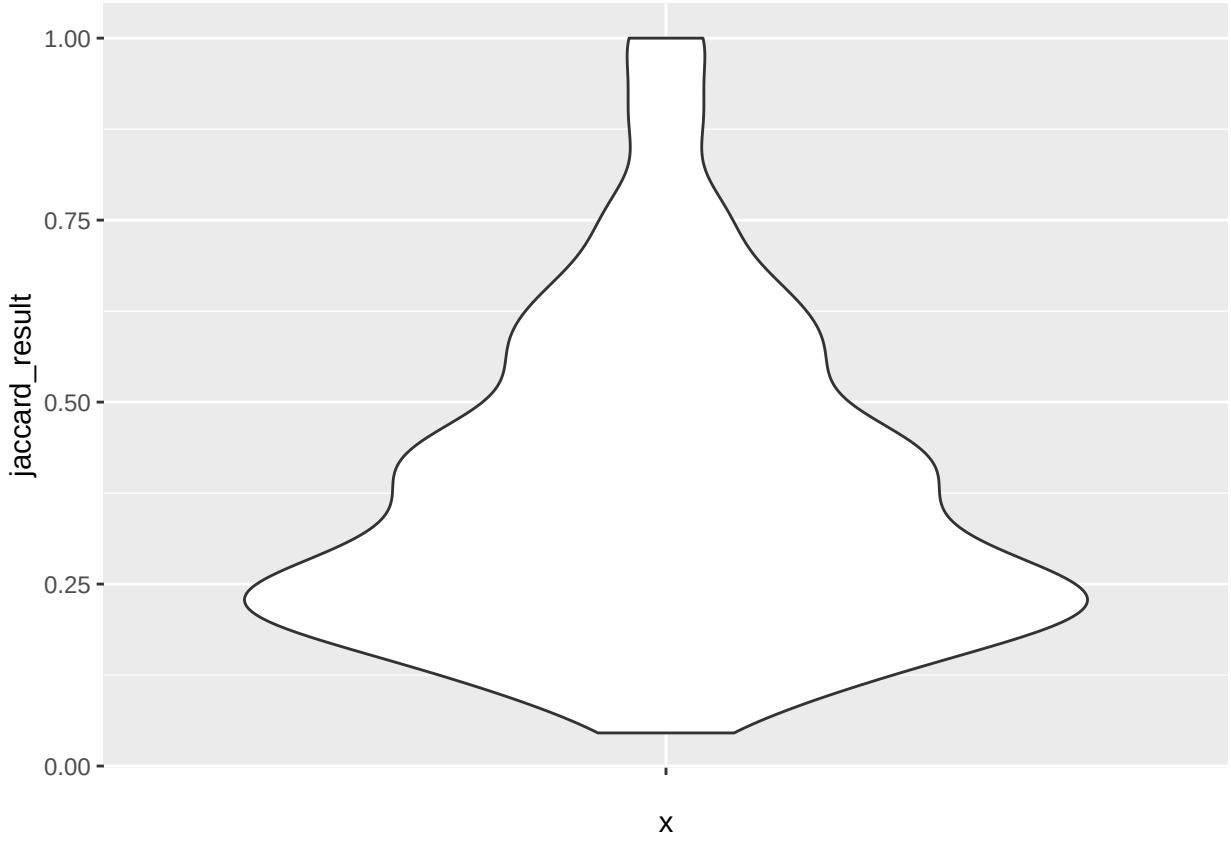
```
kable(data.frame(rbind(summary(data$jaccard_result))))
```

Min.	X1st.Qu.	Median	Mean	X3rd.Qu.	Max.
0.0454545	0.218982	0.3333333	0.3804797	0.5	1

and a violin plot of all results looks as follows. We can see that it is rather bottom heavy, but there are also

high results present.

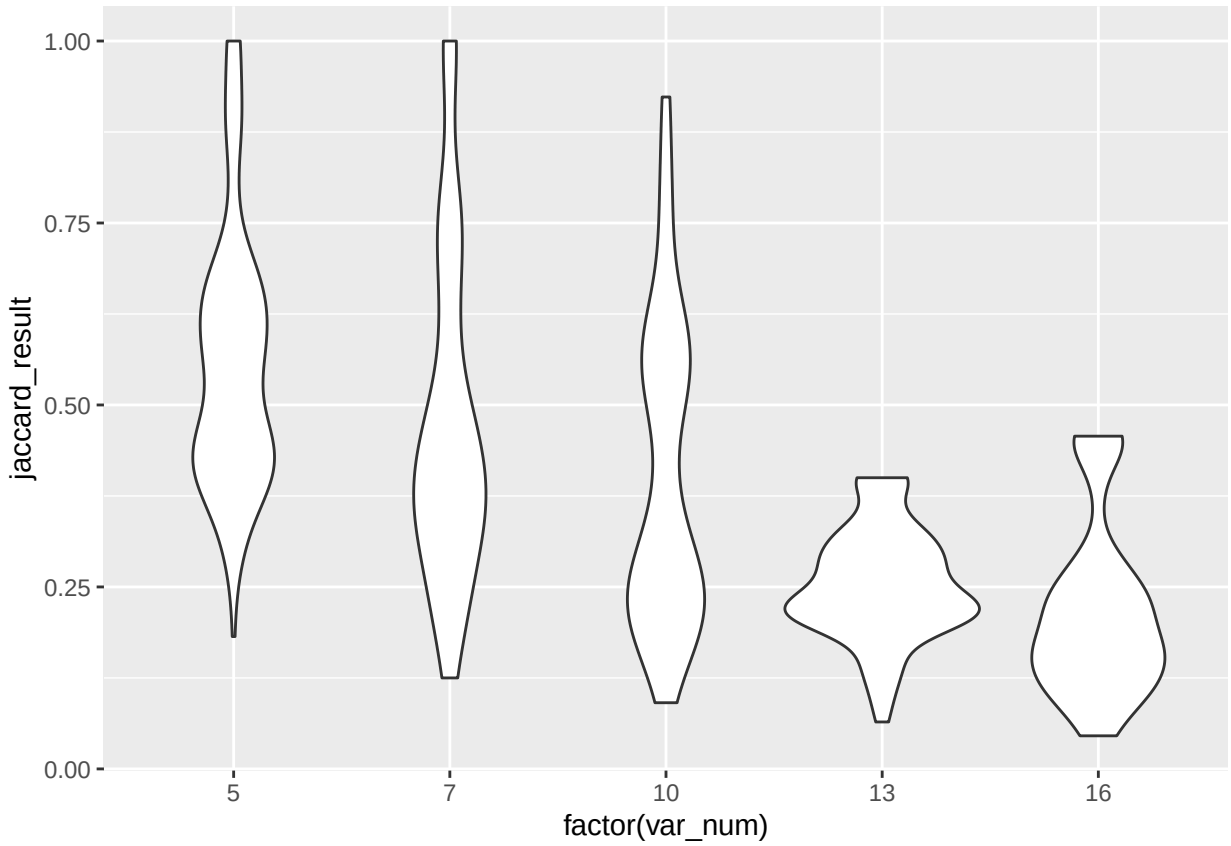
```
ggplot(data, aes(x="", y=jaccard_result)) +
  geom_violin()
```



Effect of number of variables

As we add more variables, the results become worse.

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



as can be seen in the correlation test

```
tst <- 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
```

```
tst
```

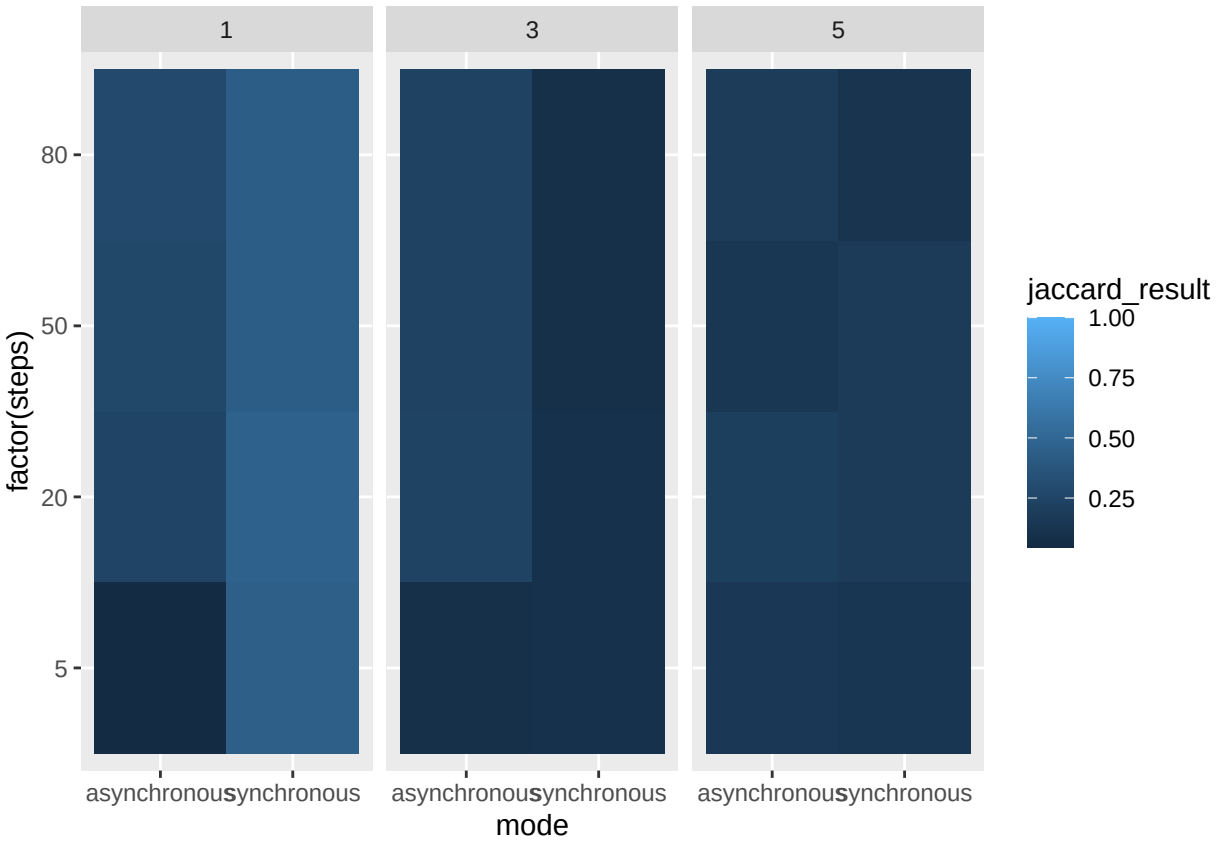
```
##
##  Spearman's rank correlation rho
##
## data:  data$var_num and data$jaccard_result
## S = 102450245, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.6469014
```

with low p-value of $1.3815967 \times 10^{-86}$.

Effect of mode

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

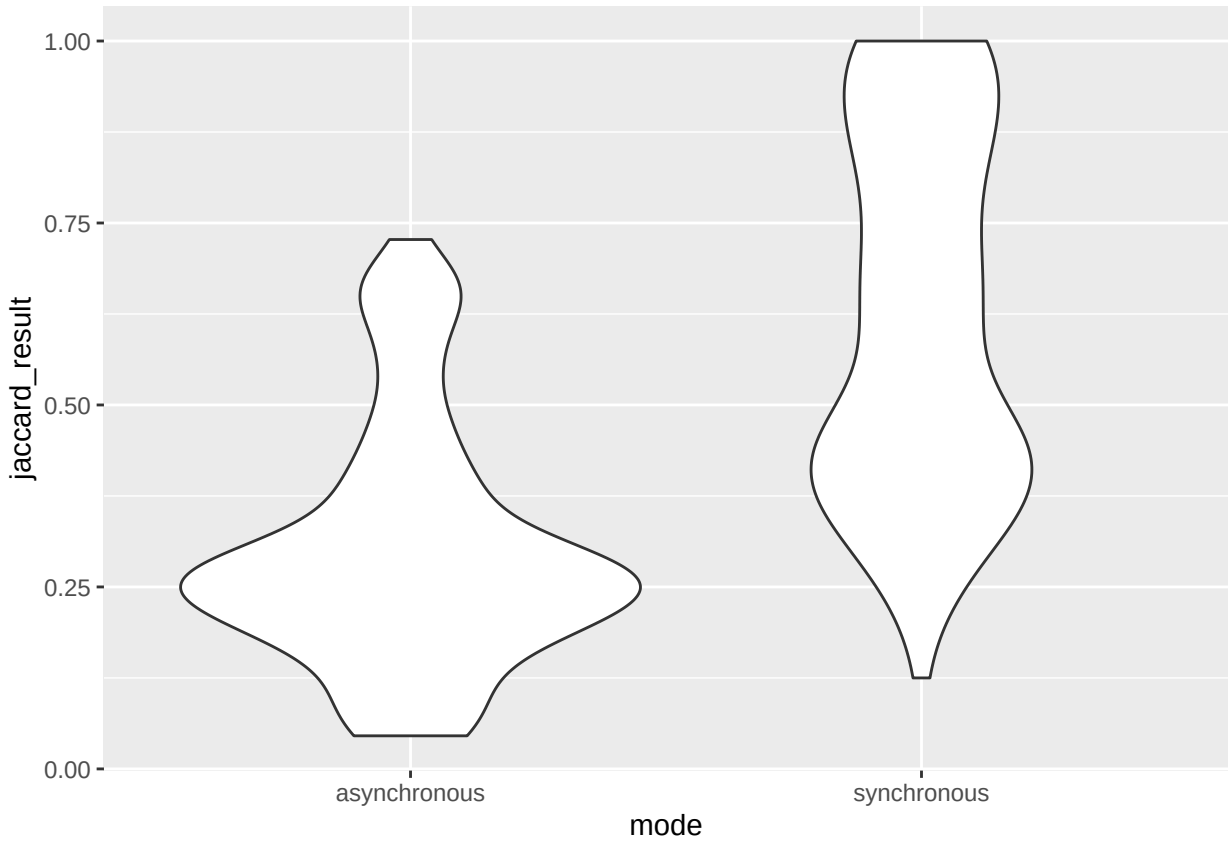
```
ggplot(data, aes(x=mode,y=factor(steps),fill=jaccard_result)) + geom_tile() + facet_wrap(~freq)
```



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

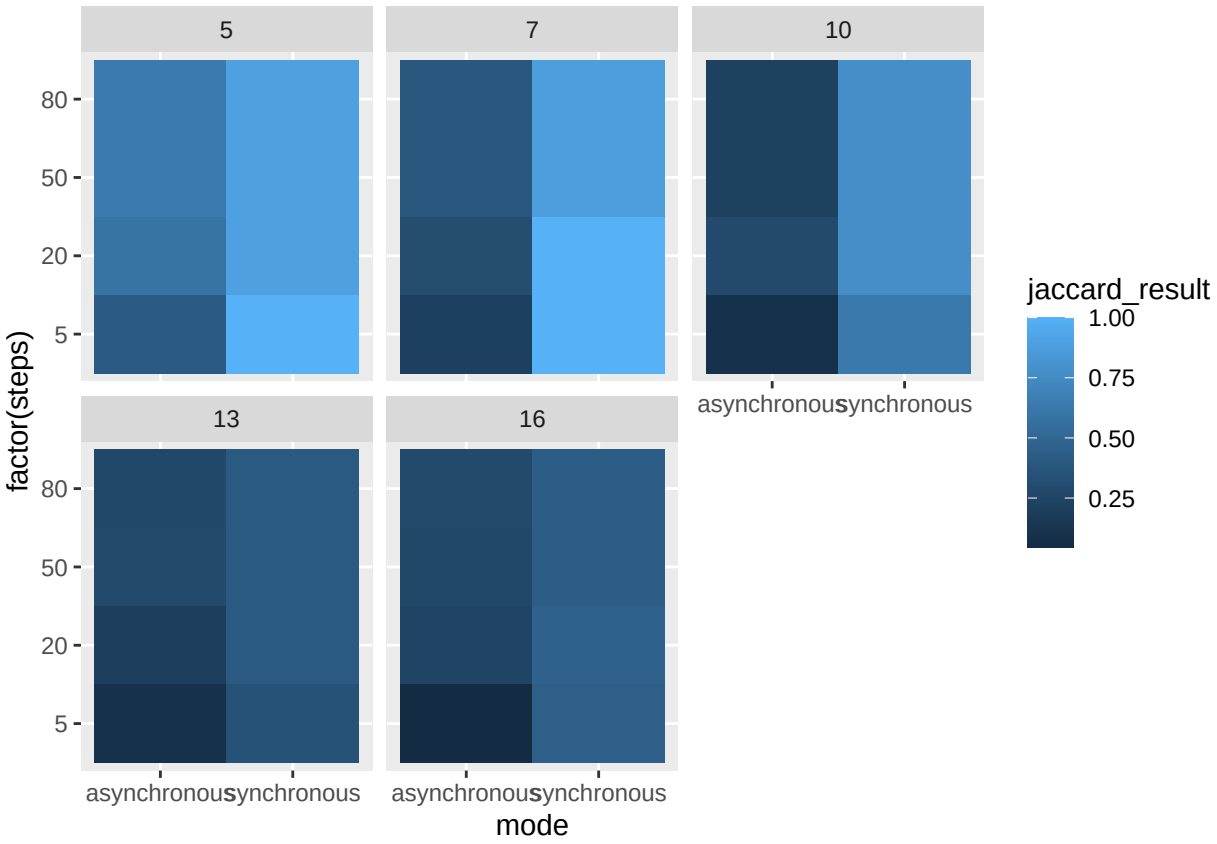
We can see on the following plot that if the frequency is 1, then synchronous data gives generally better results with the given amount of data. It can be because with the same number of transitions more variables get changed.

```
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.

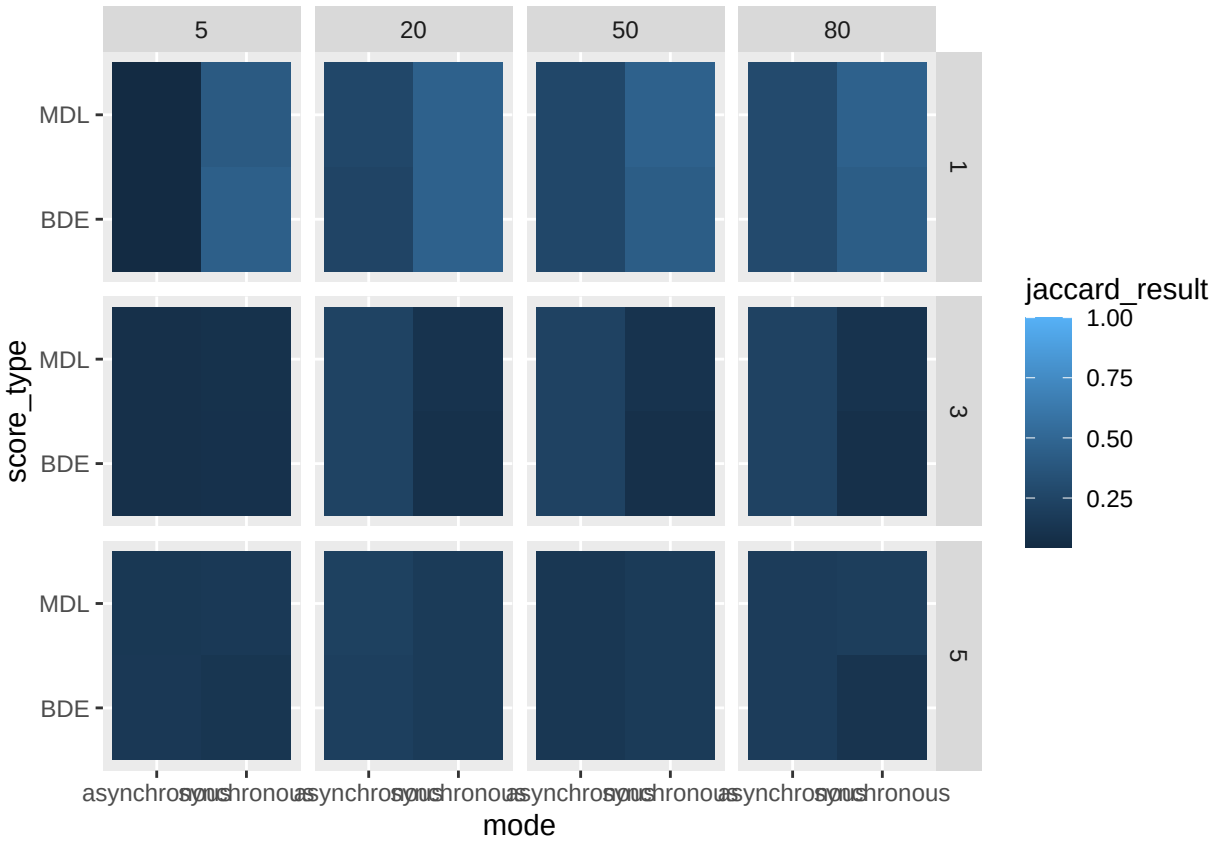
```
ggplot(data %>% filter(freq==1), aes(x=mode,y=factor(steps),fill=jaccard_result)) + geom_tile() +
    facet_wrap(~var_num)
```



Scoring mode

As we can see here, the scoring mode has little to no effect on the result given other parameters.

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



We can also use Kolmogorov-Smirnov test to check if they give the same distributions and it gives us that we cannot rule that out with any sensible confidence.

```
ks.test((data %>% filter(score_type=='MDL'))$jaccard_result,
        (data %>% filter(score_type=='BDE'))$jaccard_result)
```

```
## Warning in ks.test.default((data %>% filter(score_type ==
## "MDL"))$jaccard_result, : p-value will be approximate in the presence of ties
```

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  (data %>% filter(score_type == "MDL"))$jaccard_result and (data %>% filter(score_type == "BDE
## D = 0.033333, p-value = 0.9883
## alternative hypothesis: two-sided
```

Linear regression

If we fun linear regression with all the data, it doesn't work the best, gives adjusted R squared of 0.5322852 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)	0.6743135	0.0543629	12.4039194	0.0000000
var_num	-0.0324585	0.0023826	-13.6231772	0.0000000
steps	0.0006077	0.0004548	1.3363419	0.1827411
numtraj	0.0019080	0.0002546	7.4943826	0.0000000
attper	0.1843695	0.0556043	3.3157421	0.0010600
mode_asynchronous	-0.2998369	0.0186085	-16.1128680	0.0000000
score_type_BDE	-0.0070976	0.0168021	-0.4224244	0.6731048

And with data filtered by frequency equal to 1, we get adjusted R squared of 0.7621167.