tca_split

1 Synopsis

- This vignette shows the behavior of `tca()` and `tca_split()`. First, we show that the modded version of `TCA::tca()` produces the same result using Hannum et. al. 2013 Chr22 data as the fir by the 1.2.1 version.
- For `tca_split()`, if tau is estimated from the data, either set `vars.mle = TRUE` or the chunk size must be big enough.
- Used appropriately, `tca_split()` and `tca()` returns highly correlated results `(cor > 0.99)`.

```
library(TCA)
library(furrr)
#> Loading required package: future
```

2 Replicating Hannum et. al. 2013 fit

```
load("./vignettes/hannum.chr22.RData")
```

- First we fit the data following `tca-vignette.Rmd` using `tca()` version 1.2.1.

```
set.seed(1234)
tca_fit_fns <- purrr::partial(
  tca,
  X = hannum$X,
  W = hannum$W,
  C1 = hannum$cov[, c("gender", "age")],
  C2 = hannum$cov[, 3:ncol(hannum$cov)]
)

tca.mdl.hannum <- tca_fit_fns()
tca.mdl.hannum.mle <- tca_fit_fns(vars.mle = TRUE)
```

- Then we install the modded version and re-fit the models.

```
detach("package:TCA", unload = TRUE)
remove.packages("TCA")
devtools::install_github("hhp94/TCA@profiling")
```

```
library(TCA) # Modded fit

set.seed(1234)
tca.mdl.hannum.mod <- tca_fit_fns()
tca.mdl.hannum.mle.mod <- tca_fit_fns(vars.mle = TRUE)
```

- `compare_fit_corr` compares the correlation between the estimates. We see that the mod did not affect the fit results.

```
compare_fit_corr(tca.mdl.hannum, tca.mdl.hannum.mod)
#> $mus_hat
```

```
#> [1] 1 1 1 1 1 1
#>
#> $sigmas_hat
#> [1] 1 1 1 1 1 1
#>
#> $deltas_hat
#>  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
#>
#> $gammas_hat
#>  [1] 1 1 1 1 1 1 1 1 1 1 1 1
#>
#> $deltas_hat_pvals
#>  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
#>
#> $gammas_hat_pvals
#>  [1] 1 1 1 1 1 1 1 1 1 1 1 1
#>
#> $gammas_hat_pvals.joint
#> [1] 1 1
compare_fit_corr(tca.mdl.hannum.mle, tca.mdl.hannum.mle.mod)
#> $mus_hat
#> [1] 1 1 1 1 1 1
#>
#> $sigmas_hat
#> [1] 1 1 1 1 1 1
#>
#> $deltas_hat
#>  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
#>
#> $gammas_hat
#>  [1] 1 1 1 1 1 1 1 1 1 1 1 1
#>
#> $deltas_hat_pvals
#>  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
#>
#> $gammas_hat_pvals
#>  [1] 1 1 1 1 1 1 1 1 1 1 1 1
#>
#> $gammas_hat_pvals.joint
#> [1] 1 1
```

3 tca_split()

- First we run a sequential `TCA::tca()` and one with `vars.mle = TRUE`

```
set.seed(1234)
n_features <- 150
data <- test_data(40, n_features, 3, 2, 2, 0.03, verbose = FALSE)
```

```
tca_seq <- tca(
  X = data$X, W = data$W, C1 = data$C1, C2 = data$C2,
  log_file = NULL, verbose = FALSE
)
```

```
tca_seq_mle <- tca(
  X = data$X, W = data$W, C1 = data$C1, C2 = data$C2,
  log_file = NULL, verbose = FALSE, vars.mle = TRUE,
  max_iters = 20
)
```

- Then we do the same for `tca_split()` under the following scenarios:
  1. There are as many chunks as there are parallel workers, in this case 4.
  2. Extreme case where there are as many chunks as there are features, in this case 150.
  3. Same as 2., but `vars.mle = TRUE`

```
split_X_4 <- split_input(data$X, 4, shuffle = TRUE) # Split X into 4 chunks
split_X <- split_input(data$X, n_features) # Split X into as many chunks as feat
```

- Fit with `TCA::tca_split()`

```
# Not actually ran to save time
plan(multisession, workers = 4)
# There are as many chunks as there are parallel workers
tca_par_4 <- tca_split(
  split_X_4, W = data$W, C1 = data$C1, C2 = data$C2,
  log_file_prefix = NULL, verbose = FALSE
)
# There are as many chunks as there are features
tca_par <- tca_split(
  split_X, W = data$W, C1 = data$C1, C2 = data$C2,
  log_file_prefix = NULL, verbose = FALSE
)
# There are as many chunks as there are features, `vars.mle = TRUE`
tca_par_mle <- tca_split(
  split_X, W = data$W, C1 = data$C1, C2 = data$C2,
  log_file_prefix = NULL, verbose = FALSE,
  vars.mle = TRUE, max_iters = 20
)
plan(sequential)
```

3.1 Results

- We see that for 4 chunks, the `tca_split()` and `tca()` fits are very correlated.

```
compare_fit_corr(tca_seq, tca_par_4) # tca_seq vs tca_split with 4 Chunks of X
#> $mus_hat
#> [1] 1 1 1
#>
#> $sigmas_hat
#> [1] 0.9636 0.9725 0.9720
#>
#> $deltas_hat
#> [1] 1 1
#>
#> $gammas_hat
#> [1] 0.9999 0.9998 0.9998 0.9998 0.9998 0.9999
#>
#> $deltas_hat_pvals
#> [1] 0.9973 0.9977
```

```
#>
#> $gammas_hat_pvals
#> [1] 0.9881 0.9957 0.9945 0.9942 0.9943 0.9970
#>
#> $gammas_hat_pvals.joint
#> [1] 0.9915 0.9957
```

- However, for as many chunks as there are features, the correlation expectedly drops significantly. Especially for `sigmas_hat` and `gammas_hat_pvals`.

```
# tca_seq vs tca_split with as many chunks of X as there is features
compare_fit_corr(tca_seq, tca_par)
#> $mus_hat
#> [1] 0.9998 0.9995 0.9996
#>
#> $sigmas_hat
#> [1] 0.8801 0.8856 0.9231
#>
#> $deltas_hat
#> [1] 0.9998 0.9998
#>
#> $gammas_hat
#> [1] 0.9975 0.9978 0.9976 0.9968 0.9990 0.9988
#>
#> $deltas_hat_pvals
#> [1] 0.9859 0.9904
#>
#> $gammas_hat_pvals
#> [1] 0.9733 0.9411 0.9714 0.9593 0.9825 0.9712
#>
#> $gammas_hat_pvals.joint
#> [1] 0.9824 0.9640
```

- `tau_hat` are close enough and is close to true `tau`

```
unname(tca_seq$tau_hat)
#> [1] 0.0344326
mean(tca_par_4$tau_hat)
#> [1] 0.03199654
mean(tca_par$tau_hat)
#> [1] 0.028251
```

- For as many chunks as there are features `vars.mle = TRUE`, the correlation stays high.

```
# tca_seq vs tca_split with as many chunks of X as there is features, vars.mle = TRUE
compare_fit_corr(tca_seq_mle, tca_par_mle)
#> $mus_hat
#> [1] 1.0000 0.9999 0.9999
#>
#> $sigmas_hat
#> [1] 0.9968 0.9881 0.9977
#>
#> $deltas_hat
#> [1] 0.9999 0.9996
#>
#> $gammas_hat
```

```
#> [1] 0.9995 0.9998 0.9987 0.9993 0.9993 0.9994
#>
#> $deltas_hat_pvals
#> [1] 0.9973 0.9456
#>
#> $gammas_hat_pvals
#> [1] 0.9947 0.9982 0.9855 0.9938 0.9958 0.9936
#>
#> $gammas_hat_pvals.joint
#> [1] 0.9990 0.9994
```