

## Test mclapply assess\_reference\_loo() on cluster

This is just a quick thing to see how much faster it is with 20 cores.

### Get on the cluster and get a node with 20 cores

```
srunch --pty -c 20 /bin/bash
```

### Start R and run a test on the non-parallel version of rubias

```
module load R
R

library(tidyverse)
library(rubias)

set.seed(10)
hundy_colls <- sample(unique(chinook$collection), 40)
hundy_colls

hundy_coll_list <- lapply(hundy_colls, function(x) tibble(collection = x, ppn = 1.0)) %>%
  setNames(paste("100%", hundy_colls, sep = "_"))

system.time({one_core <- assess_reference_loo(
  reference = chinook,
  gen_start_col = 5,
  reps = 80,
  mixsize = 50,
  alpha_collection = hundy_coll_list
)})

# This is how long that took:
  user  system elapsed
294.587   1.555 296.976
```

### Now, start a clean session of R and run it with 20 cores

```
# install the parallel version from GitHub
devtools::install_github("eriqande/rubias", ref = "mclapply-assess-reference-loo")

# now run the same test (and time it)
library(tidyverse)
library(rubias)
```

```

set.seed(10)
hundy_colls <- sample(unique(chinook$collection), 40)
hundy_colls

hundy_coll_list <- lapply(hundy_colls, function(x) tibble(collection = x, ppn = 1.0)) %>%
  setNames(paste("100%", hundy_colls, sep = "_"))

# ask for 20 cores (mc.cores = 20)
system.time({twenty_core <- assess_reference_loo(
  reference = chinook,
  gen_start_col = 5,
  reps = 80,
  mixsize = 50,
  alpha_collection = hundy_coll_list,
  mc.cores = 20
)})

# This is how long that took:
  user system elapsed
357.887  14.507  21.782

```

So, in terms of elapsed time that is about a 13.6X speedup.

So, if Ben has 20 cores on his Linux box and sees the same improvements, that will bring his jobs from 35 to 2.5 hours. Cool.