Tutorial

This package is used for estimating viral transmission bottleneck sizes using different methods.

1. Install package and download test dataset

First step, install the package “ViralBottleneck” and download dataset in test_dataset folder

2. Create transmission object

Second step, the transmission object need to be created before bottleneck size estimation. To create transmission object, the working directory need to meet two requirements: transmission pairs table and sample files used for estimation. This package would extract sample files according to the transmission pairs table the users input.

The example of the transmission pairs table is below (in `test_dataset` folder in package)

donor	recipient
donor_3000	50_0_All_r1
donor_3000	50_3_All_r1
donor_3000	50_6_All_r1
donor_3000	50_9_All_r1
donor_3000	50_12_All_r1

Note: Do not put the “-” in name of sample.

After making sure the sample files all exist according to the transmission pairs, start to create transmission object. example code:

```
Sim_trans = read.table("Example_TansmissionPairs.csv",header = TRUE,sep = ",")
Sim_ob = CreateTransmissionObject(Sim_trans)
```

2.1 Subset transmission object

The transmission object could be used as list.

```
#Get first 3 transmission object
Sim_ob_subset = Sim_ob[1:2]
```

3. Summary transmission object

After creating transmission object, the `Summary_ob` function would provide the information of shared sites (the sites belong to shared sites should be sequenced both in donor and recipient.) for users. Example code:

```
Summary_Sim = Summary_ob(Sim_ob)
```

The result:

Donors	Recipients	number.of.shared.sites
donor_3000	50_0_All_r1	13158
donor_3000	50_3_All_r1	13158
donor_3000	50_6_All_r1	13158
donor_3000	50_9_All_r1	13158
donor_3000	50_12_All_r1	13158

4. Transmission bottleneck size estimation

Finally, start to calculate transmission bottleneck size using transmission object.

4.1 Output of `Bottleneck_size_Calculation` function

Take calculation using Beta-binomial method approximate version as an example:

```
BB_App_output =
  Bottleneck_size_Calculation(
           transmission_ob = Sim_ob,
           method ="Beta_binomial_Approximate",
           error_calling = 0,
           variant_calling = 0.03,
           Nbmin = 1,
           Nbmax = 200,
           donor_depth_threshold = 0,
           recipient_depth_threshold = 0
           )
```

Output like:

donor	recipient	transmission_bottleneck_size	CI_low	CI_high
donor_3000	50_0_All_r1	70	64	70
donor_3000	50_3_All_r1	45	30	64
donor_3000	50_6_All_r1	28	20	39
donor_3000	50_9_All_r1	34	23	47
donor_3000	50_12_All_r1	47	31	67

4.2 Specify transmission pairs during estimation

This package provide a chance that if user need to specify some transmission pairs for estimation

```
subset_transmission_pairs = read.table("H1N1_transmission_pairs_specify.csv",header = TRUE,sep = ",")
```

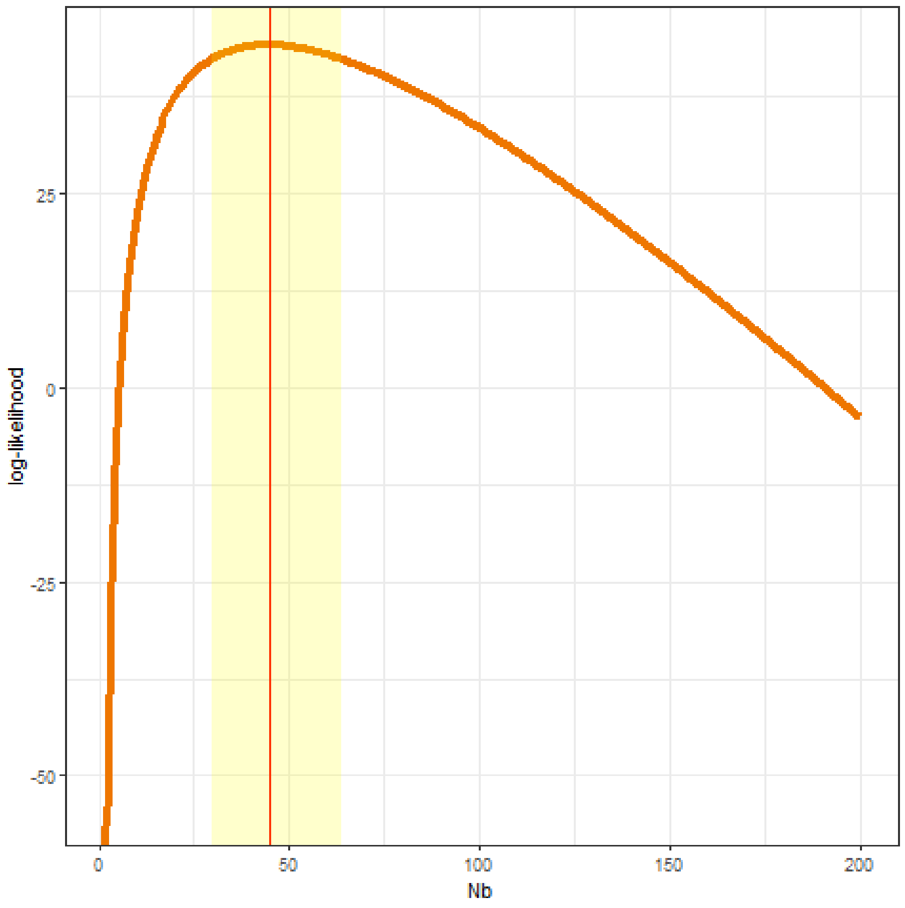
4.3 Calculation

`Bottleneck_size_Calculation` could create plot of likelihood curve for each transmission pairs in working directory. However, this argument just used for the methods using maximum likelihoods estimation, including `KL` method (Emmett et al., 2015), `Presence-Absence` method (Sacristán et al., 2011), `Binomial` method (Leonard et al., 2017), `Beta_binomial_Approximate` method (Leonard et al., 2017) and `Beta_binomial_Exact` method (Leonard et al., 2017). Using `show_table` and `plot` options could help to save output and obtain the plots of likelihood curve for each transmission pairs. (Note: if you want to access the original publication for each methods, you could click the *Publication link* after each methods)

The program would create individual folder for each transmission pair to store the plot. Example code for creating plot:

```
BB_App_output_plot =
           Bottleneck_size_Calculation(
           transmission_ob = Sim_ob,
           method = "Beta_binomial_Approximate",
           error_calling = 0,
           variant_calling = 0.03,
           Nbmin = 1,
           Nbmax = 200,
           donor_depth_threshold = 0,
           recipient_depth_threshold = 0,
           show_table = FALSE,
           plot= TRUE
           )
```

The plot of likelihood curve for one transmission pairs (donor_3000-50_3_All_r1) is below:



4.4 Log file

`Bottleneck_size_Calculation` could create log file containing number of variant used in calculation and number of variant filtered before calculation in working directory.

Example code:

```
BB_App_output_log =
            Bottleneck_size_Calculation(
            transmission_ob = Sim_ob,
            method = "Beta_binomial_Approximate",
            error_calling = 0,
            variant_calling = 0.03,
            Nbmin = 1,
            Nbmax = 200,
            donor_depth_threshold = 0,
            recipient_depth_threshold = 0,
            log= TRUE
            )
```

Output of `log` argument:

donor	recipient	donor_used	donor_unused	recipient_used	recipient_unused
donor_3000	50_0_All_r1	193	12965	193	12965
donor_3000	50_3_All_r1	193	12965	193	12965
donor_3000	50_6_All_r1	193	12965	193	12965
donor_3000	50_9_All_r1	193	12965	193	12965
donor_3000	50_12_All_r1	193	12965	193	12965

4.5 Methods comparison

Given that one major purpose of the package is to compare calculation of bottleneck sizes across methods on the same data set, it would be nice to illustrate this. For example, compare all methods (except Wright-Fisher, see below) on a single pair, Sim_ob[1]:

```
all_methods <-
  c("KL", "Presence-Absence", "Binomial", "Beta_binomial_Approximate", "Beta_binomial_Exact")

compare_methods <-
  t(sapply(all_methods, function(m){
    Bottleneck_size_Calculation(Sim_ob[1], method = m)
  }))

compare_methods
```