Shazam: Tuning clonal assignment thresholds by calculating distance to nearest neighbor

Namita Gupta

2016-02-20

Contents

Estimating the optimal distance threshold for partitioning clonally related sequences is accomplished by calculating the distance from each sequence in the data set to its nearest neighbor and finding the break point in the resulting bi-modal distribution that separates clonally related from unrelated sequences. This is done via the following steps:

1. Load a Change-O tab-delimited database file.
2. Calculate the nearest neighbor distances for each sequence.
3. Generate a histogram of the nearest neighbor distances and inspect for the threshold separating the two modes.

Load Change-O data

A small example Change-O tab-delimited database file is included in the `shazam` package. Calculating the nearest neighbor distances requires the following fields (columns) to be present in the Change-O file: `V_CALL`, `J_CALL`, `JUNCTION_LENGTH`, and `JUNCTION`.

```
# Subset example data to one sample
library(shazam)
db <- subset(InfluenzaDb, BARCODE == "RL013")
```

Calculate nearest neighbor distances

The function for calculating distance between every sequence and its nearest neighbor takes a few parameters to adjust how the distance is measured. If a genotype has been inferred using the methods in the `tigger` package, and a `V_CALL_GENOTYPED` field has been added to the database, then this column may be used instead of the default `V_CALL` column by specifying the `vCallColumn` argument. This will allows the more accurate V call from `tigger` to be used for grouping of the sequences. Furthermore, for more leniency toward ambiguous V(D)J segment calls, the parameter `first` can be set to `FALSE`. Setting `first=FALSE` will use the union of all possible genes to group sequences, rather than the first gene in the field. The `model` parameter determines which underlying SHM model is used to calculate the distance. The default model is `hs1f`, a human Ig-specific single

nucleotide model similar to a transition/transversion model (Yaari et al, 2013). Other options include nucleotide Hamming distance (`ham`), amino acid Hamming distance (`aa`), single nucleotide (`m1n`) and 3-mer (`m3n`) mouse models (Smith et al, 1996), and a 5-mer model inferred from human data (`hs5f`) (Yaari et al, 2013). For models that are not symmetric (e.g., distance from A to B is not equal to the distance from B to A), there is a `symmetry` parameter that allows the user to specify taking the average or the minimum of the two distances to determine the overall distance.

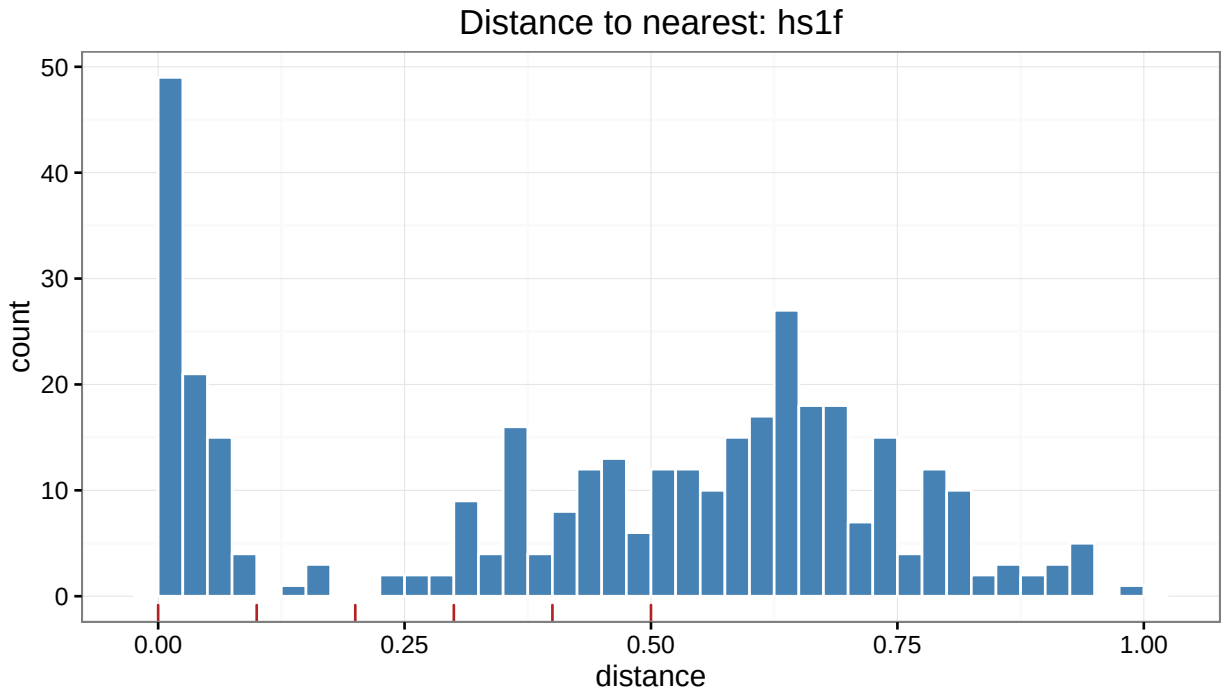
```
# Use hs1f model and normalize by junction length
dist_hs1f <- distToNearest(db, model="hs1f", first=FALSE, normalize="length",
                           nproc=1)

# Use genotyped V assignments and 5-mer model
dist_hs5f <- distToNearest(db, vCallColumn="V_CALL_GENOTYPED", model="hs5f",
                           first=FALSE, normalize="none", nproc=1)
```

Generate histogram

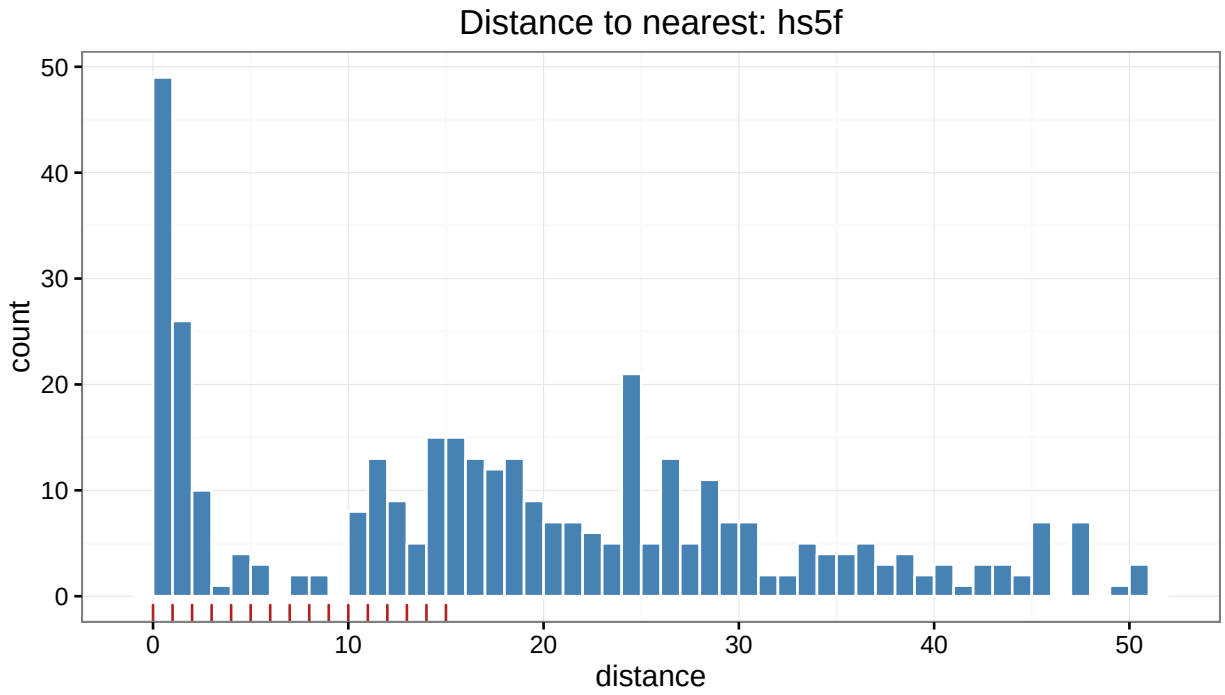
The primary use of the distance to nearest calculation in the Change-O pipeline is to determine the optimal threshold for separating clonally related sequences (represented by sequences with "near"" neighbors) from singletons (sequences without "near"" neighbor), which show up as two modes in a histogram.

```
# Generate m1n histogram
library(ggplot2)
p1 <- ggplot() + theme_bw() +
    ggtitle("Distance to nearest: hs1f") + xlab("distance") +
    geom_histogram(data=dist_hs1f, aes(x=DIST_NEAREST), binwidth=0.025,
                   fill="steelblue", color="white") +
    geom_rug(aes(x=seq(0.0, 0.5, 0.1)), color="firebrick")
plot(p1)
```

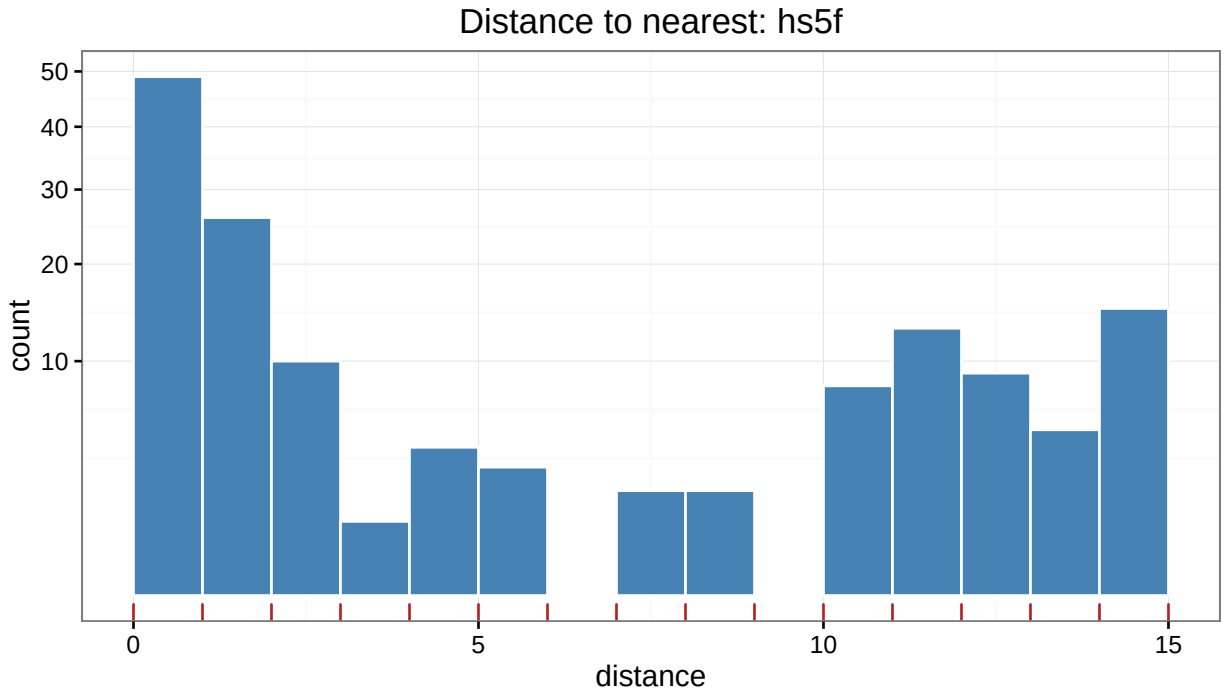


In this example, the length normalized `hs1f` model distance threshold would be set to a value near 0.1.

```
# Generate hs5f histogram
p2 <- ggplot() + theme_bw() +
    ggtitle("Distance to nearest: hs5f") + xlab("distance") +
    geom_histogram(data=dist_hs5f, aes(x=DIST_NEAREST), binwidth=1,
                   fill="steelblue", color="white") +
    geom_rug(aes(x=seq(0, 15, 1)), color="firebrick")
plot(p2)
```




```
# Zoom in to find threshold
p3 <- p2 + xlim(c(0, 15)) + scale_y_sqrt()
plot(p3)
```



In this example, the unnormalized `hs5f` model distance threshold would be set to a value near 4.