Analyzing the diversity of the clonally related sequences in a B-cell repertoire

Sketch of the method

Once we have detected clones for a B-cell repertoire (found by Mixclus or vidjil), we can take a closer look in fewer clones and analyze their diversity. First, we group identical sequences of a given clone into clonotypes, and then we create a list of clonotypes and their respective abundance per clone. Note that each clonotype is represented by a unique sequence. Next, we determine the germline sequence for each clonotype. Then, we calculate the distance between clonotype sequences and the germline(s). The most frequent germline among clonotypes is chosen as first node (or root) of the tree representation. As a next step, we rank the sequences in the clonotype's list based on the increasing distance to the first node. We create a list L with the first ranked sequence, that is, the sequence with the lowest distance to the first node. Then, we add to L the second ranked sequence if the soustraction of their distances is lower than a threshold « t ». We repeat that until no more sequence can be add to L. Elements of L become the new nodes in the tree. We update the clonotype's list by deleting from it, the sequences that have become nodes. We continue this process until all clonotype is placed in the tree. We can then visualize the results in the form of different levels of distance to the germline where the abundance of each clonotype is correlated to its size.

Pseudo Code

START preprocessing
INPUT :
    Description : The distribution of sequences into different clones done by Mixclus or vidjil
    Format : {cl_a : [(seq_a1,prop_a1),(seq_a2,prop_a2),…], cl_b : [(seq_b1,prop_b1),
    (seq_b2,prop_b2),…],…}
    where cl_x is a clone, seq_x1, seq_x2,… are the clonotype sequences in the cl_x and
    prop_x1, prop_x2,… are the percentage of each clonotype in the clone.

n = number of clones to be analyzed
clone(s)_to_analysis = n most abundant clone(s) and their clonotype's information.

FOR *clone* in *clone(s)_to_analysis*
    FOR clontype in clone
        DETERMINE clonotype_germline
    IF clone has multiple associated germline
        SET most abundant clonotype_germline as clone_germline
    Add clone_germline to respective element in clone(s)_to_analysis

OUTPUT :
    Description : The n most abundants clone(s) and their belonging clonotypes, relative fractions and the associated germlines.
    Format : {*cl_n* : [clone_germline, (seq_n1,prop_n1),(seq_n2,prop_n2),…],…}
END preprocessing

START algorithme
INPUT :

clone(s)_to_analysis : n most abundant clone(s) and their clonotype's information. We call the list of clonotypes of clone c, *clonotype_list_c*. It has this format : [clone_germline, (seq_n1,prop_n1), (seq_n2,prop_n2),…]

Y : number of clonotype to visualize

t : neighborhood threshold (find the definition)

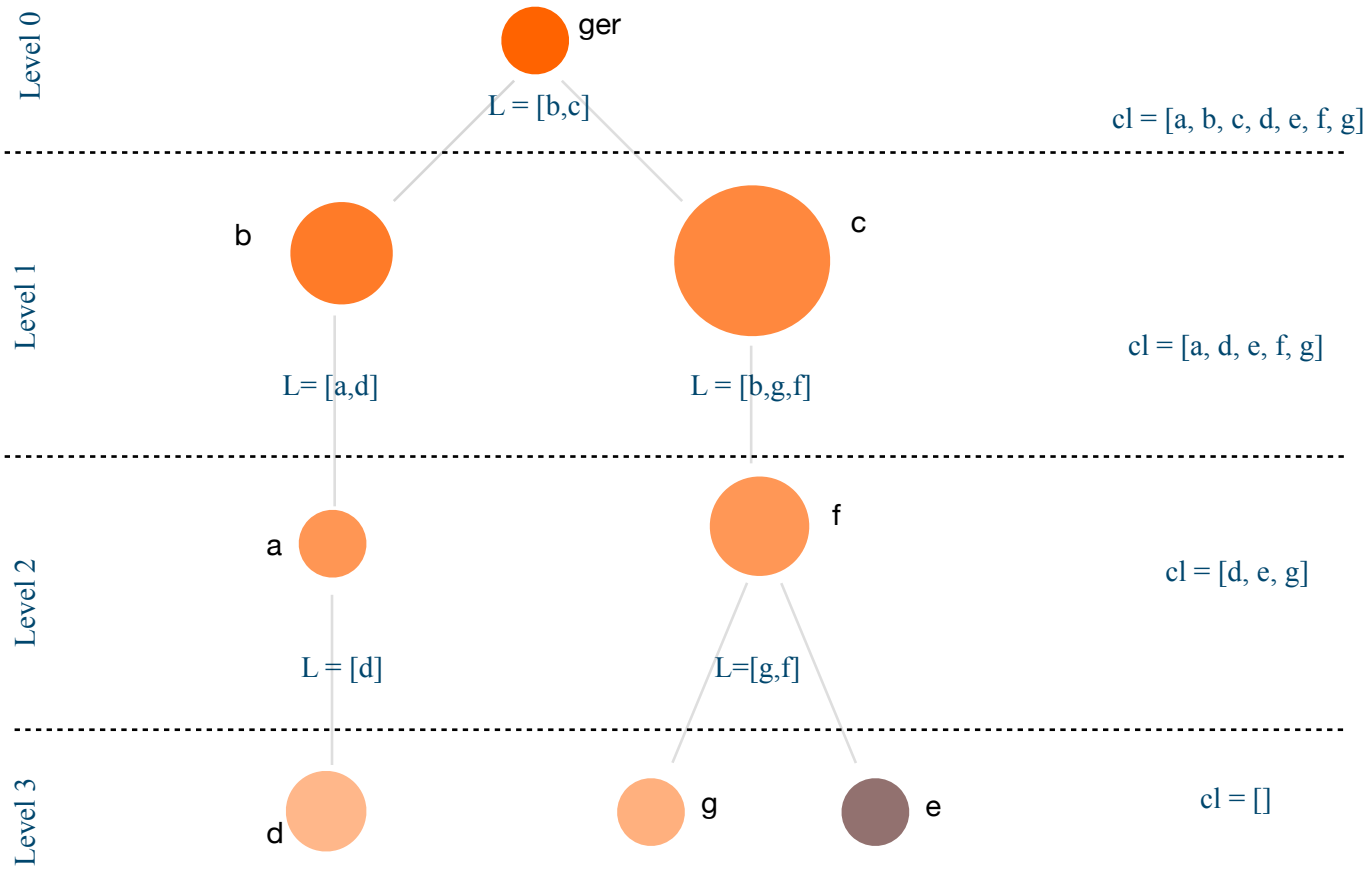
level_node = germline
level = 0
For clone in *clone(s)_to_analysis*
count_node = 0
WHILE *count_node* < Y or *clonotype_list_c* is not empty
IF *level_node* hase one member
L = []
CALCULATE the distance of each sequence in the *clonotype_list_c* to *level_node*
SORT sequences by the increasing distance to *level_node* and keep them in *sorted_clt_list,*
next_node = sorted_clt_list[0]
ADD next_node to L

FOR sequence in *sorted_clt_list*
IF distance (sequence to next_node) < t
ADD sequence to L

LET *upper_node = L*
clonotype_list_c = clonotype_list_c - L
level = level +1
count_node = count_node +1
ELSE
To be completed
Output : the intraclonal diversity of the input clone
END algorithme

Toy example

Clonotype's list. = [(a,a_abundance), (b,b_abundance),(c,c_abundance), (d,d_abundance), (e,e_abundance), (f,f_abundance), (g,g_abundance)]
For simplifying the presentation, we show the clonotype's list as :
cl = [a, b, c, d, e, f, g]
ger is the germline.


Level 0
ger
L = [b,c]
cl = [a, b, c, d, e, f, g]
Level 1
b
c
L= [a,d]
L = [b,g,f]
cl = [a, d, e, f, g]
Level 2
a
f
L = [d]
L=[g,f]
cl = [d, e, g]
Level 3
d
g
e
cl = []