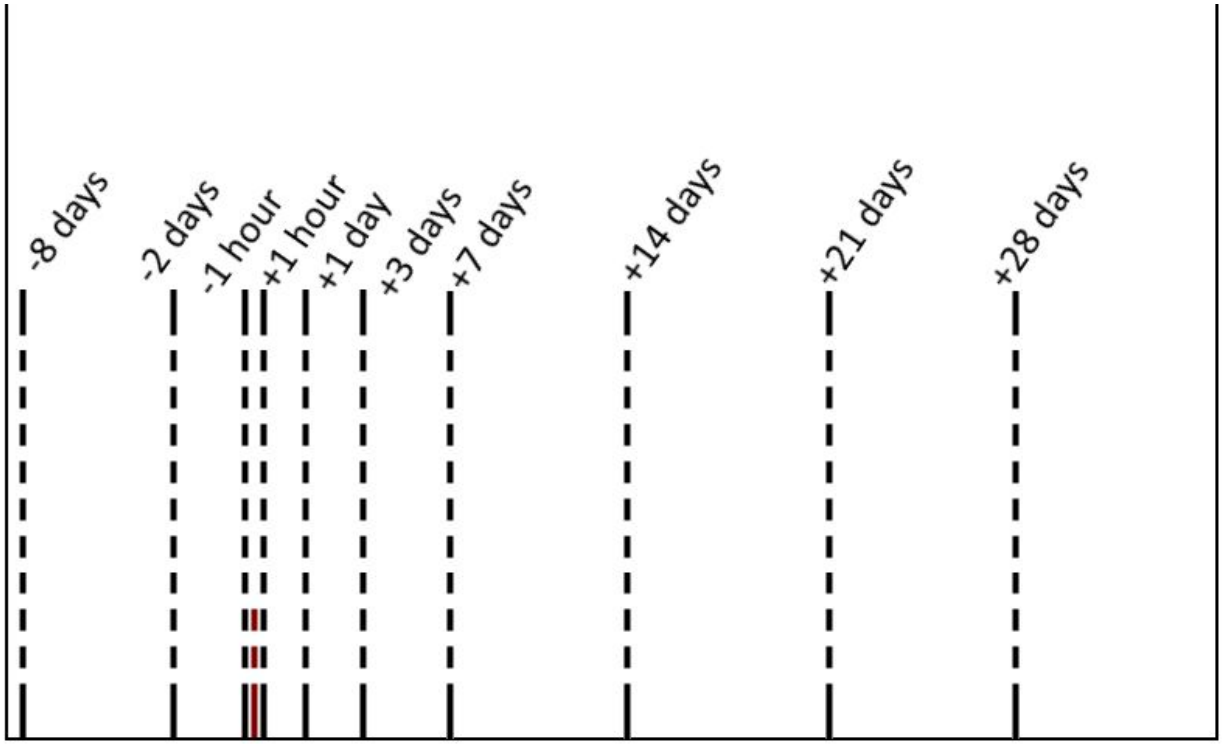
Analysis of immune response dynamics using Rep-seq data

Andrey Slabodkin

Available data:

3 flu vaccinated people with 10 Rep-seq datasets per individual



Approaches

Possible approaches to immune response dynamics analysis:

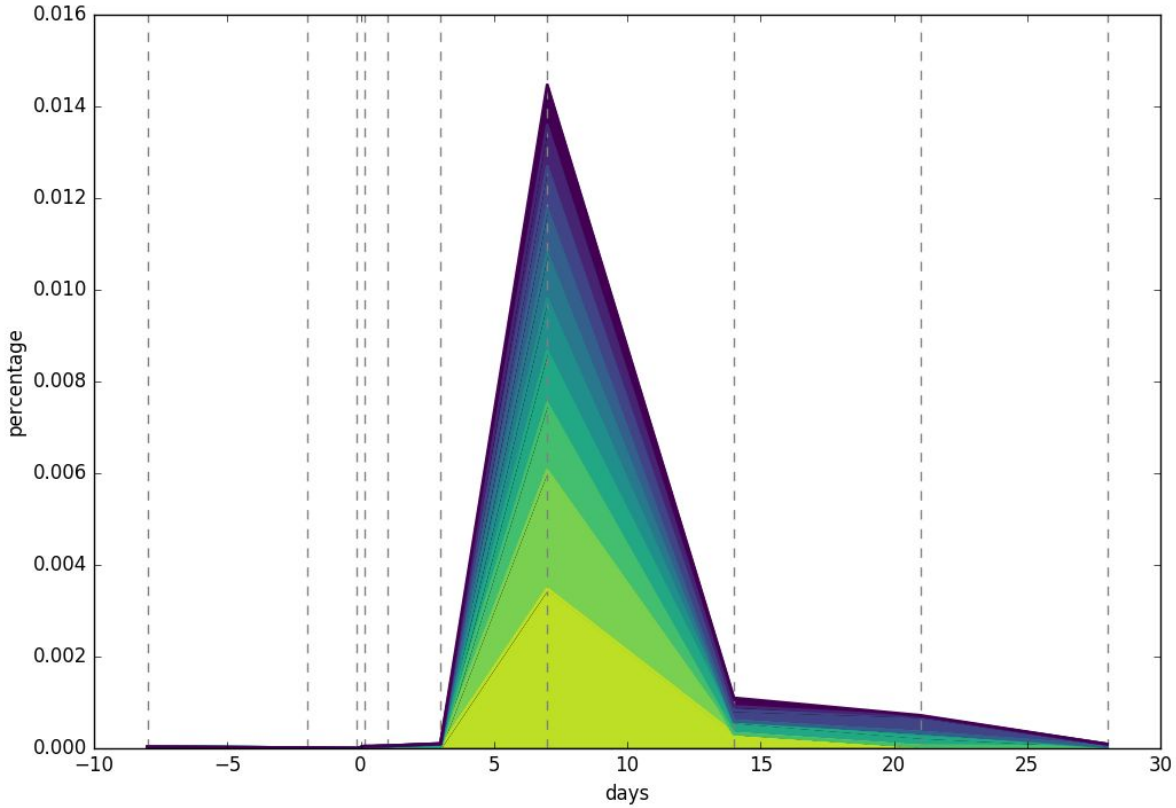
- Simple quantitative analysis
- Studying of clonal trees growth

Approaches

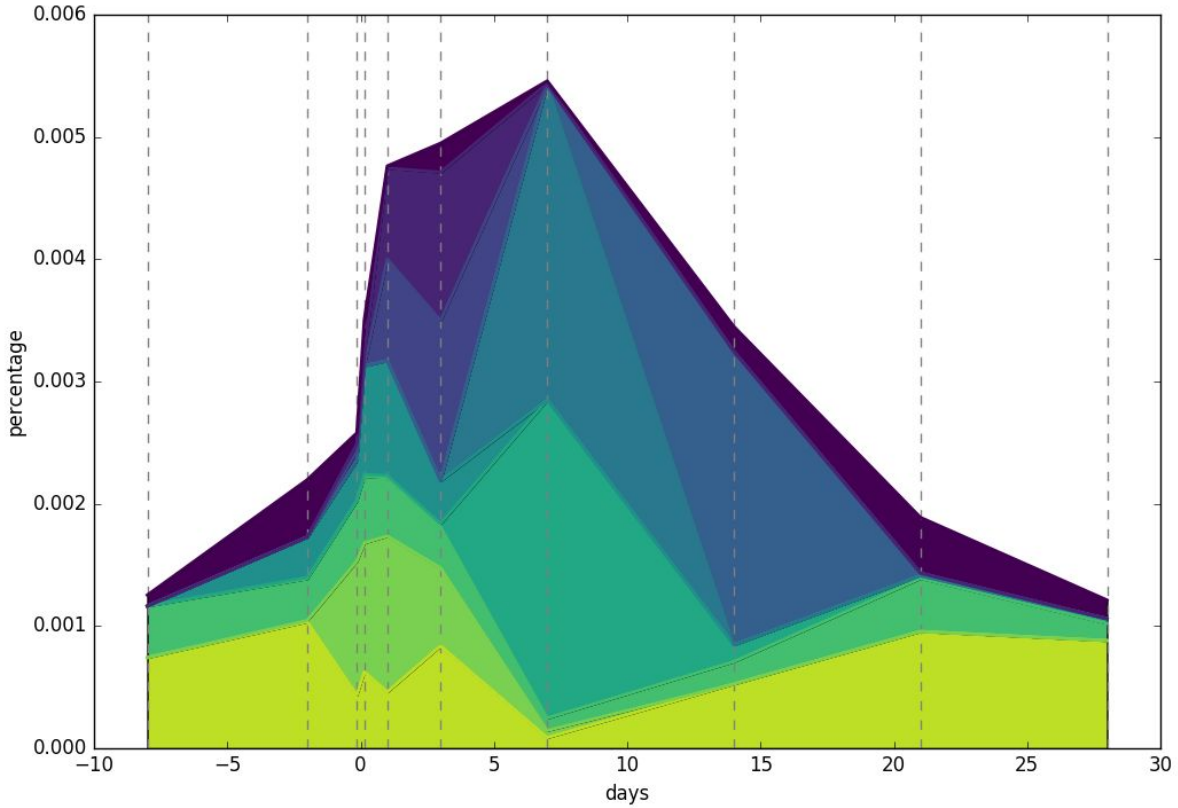
Possible approaches to immune response dynamics analysis:

- Simple quantitative analysis
- Studying of clonal trees growth

10 clones, most abundant clones on the 7th day



10 most abundant clones during the whole period



Approaches

Possible approaches to immune response dynamics analysis:

- Simple quantitative analysis
- Studying of clonal trees growth

To analyze tree dynamics we need trees, so we need to learn how to reconstruct clonal trees.

To analyze tree dynamics we need trees, so we need to learn how to reconstruct clonal trees.

Antevolo

Antevolo algorithm

- Given a dataset of reads of the same isotype
- Split the reads by the VJ-type
- In each component compute the Hamming graph on the CDR3 sequences
- If there is an edge between reads *A* and *B* in the CDR3 graph, we say that *A* and *B* are connected with an edge of one of the following types:

Antevolo algorithm

- Given a dataset of reads of the same isotype
- Split the reads by the VJ-type
- In each component compute the Hamming graph on the CDR3 sequences
- If there is an edge between reads *A* and *B* in the CDR3 graph, we say that *A* and *B* are connected with an edge of one of the following types:

Antevolo algorithm

- Given a dataset of reads of the same isotype
- Split the reads by the VJ-type
- In each component compute the Hamming graph on the CDR3 sequences
- If there is an edge between reads *A* and *B* in the CDR3 graph, we say that *A* and *B* are connected with an edge of one of the following types:

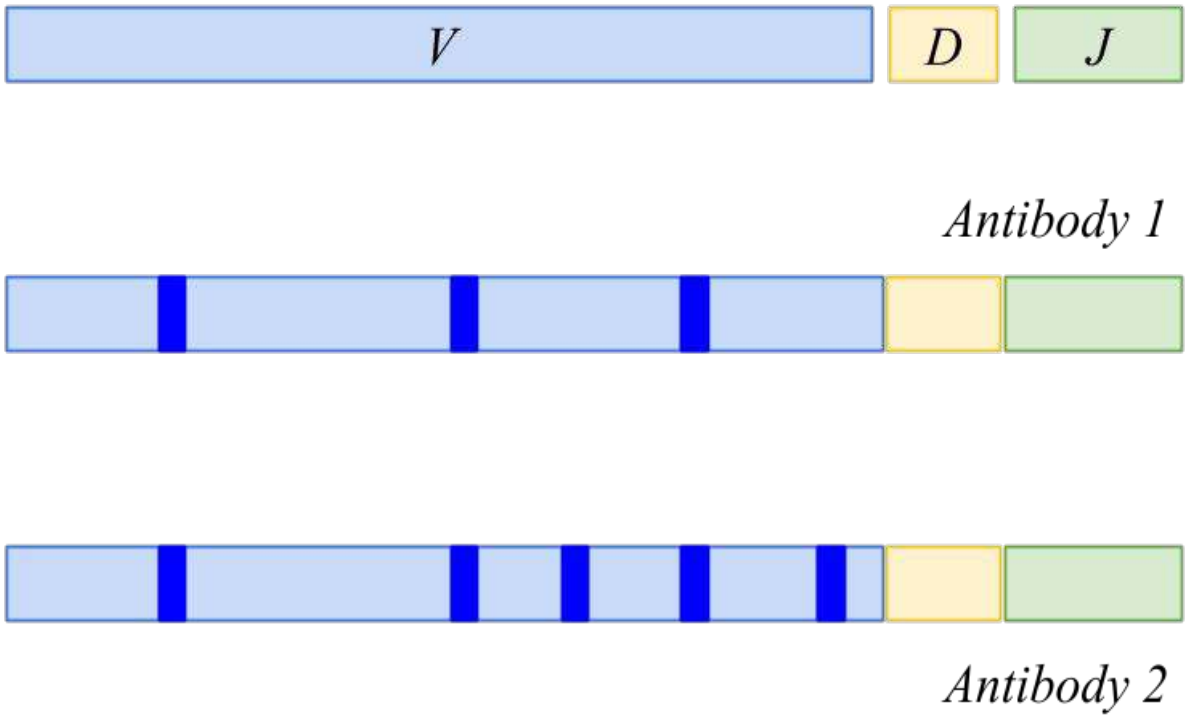
Antevolo algorithm

- If there is an edge between reads *A* and *B* in the CDR3 graph, we say that *A* and *B* are connected with an edge of one of the following types:
  - We call the edge ***directed***, if all the VSHMs of B are presented in A.
  - We call the edge ***undirected***, if the VSHMs sets of *B* and *A* are equal (if two reads differ only in CDR3)
  - Otherwise, we call it ***intersected***.

Antevolo algorithm

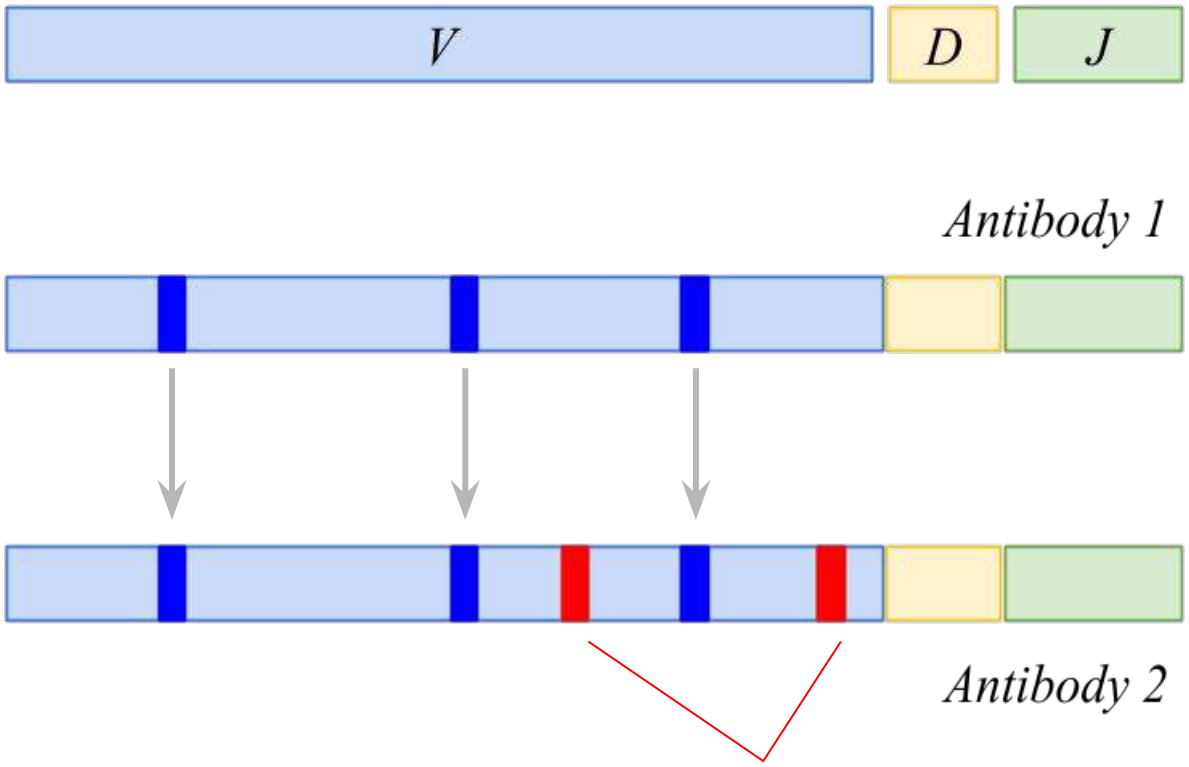
- If there is an edge between reads *A* and *B* in the CDR3 graph, we say that *A* and *B* are connected with an edge of one of the following types:
  - We call the edge ***directed***, if all the VSHMs of B are presented in A.
  - We call the edge ***undirected***, if the VSHMs sets of *B* and *A* are equal (if two reads differ only in CDR3)
  - Otherwise, we call it ***intersected***.

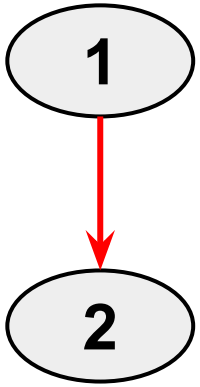
Directed edge case



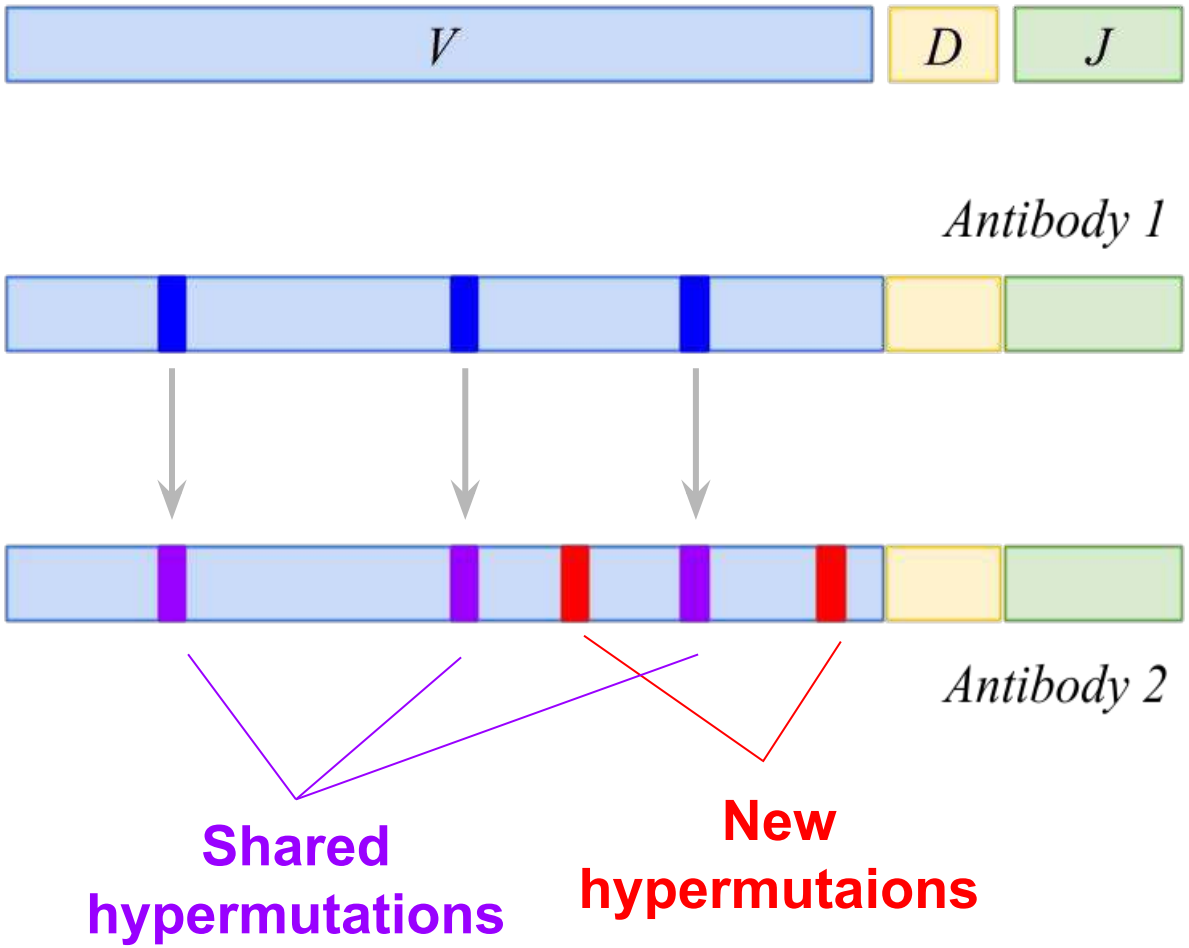
germline segments

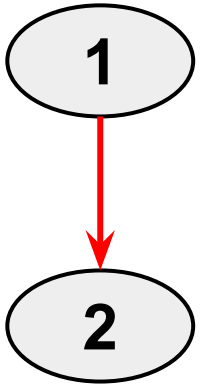
Directed edge case





Directed edge case

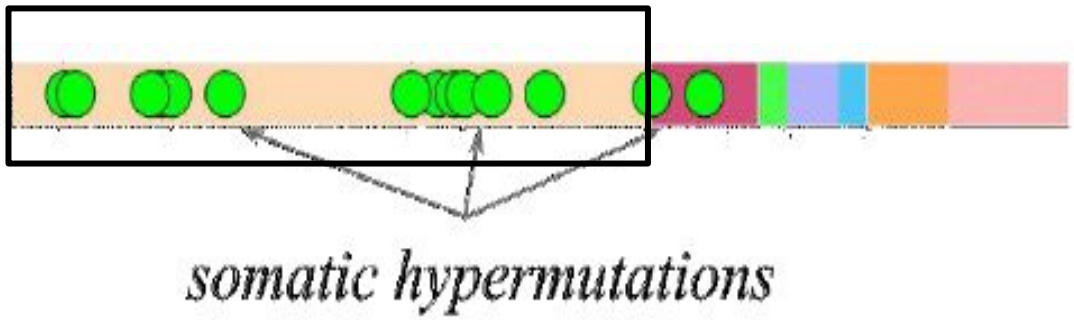




Antevolo algorithm

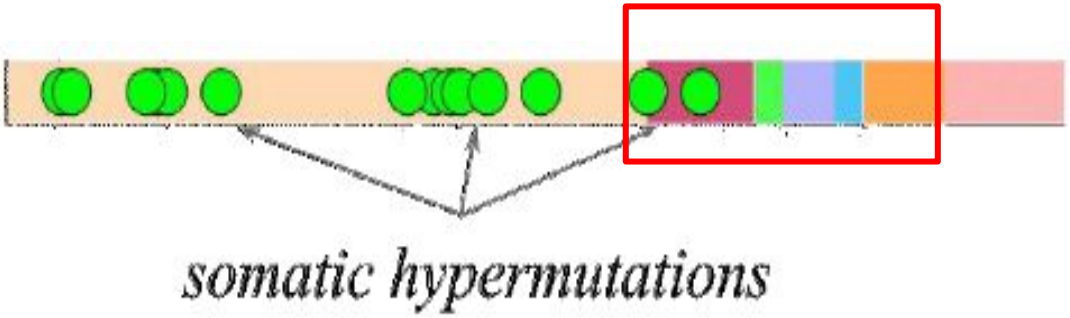
- If there is an edge between reads *A* and *B* in the CDR3 graph, we say that *A* and *B* are connected with an edge of one of the following types:
  - We call the edge ***directed***, if all the VSHMs of B are presented in A.
  - We call the edge ***undirected***, if the VSHMs sets of *B* and *A* are equal (if two reads differ only in CDR3)
  - Otherwise, we call it ***intersected***.

SHMs in V segments are easy to find



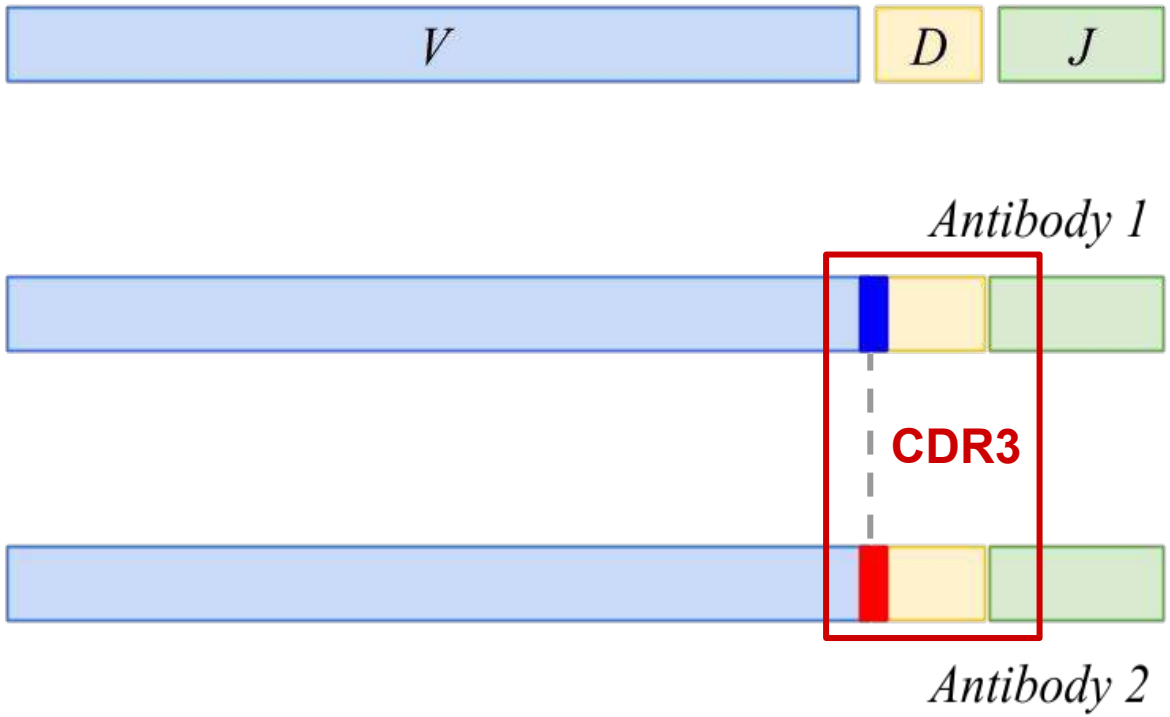
- One can easily identify mutations in the V segment using alignment against the **template** (germline V segment)

SHMs in CDR3 are difficult to identify

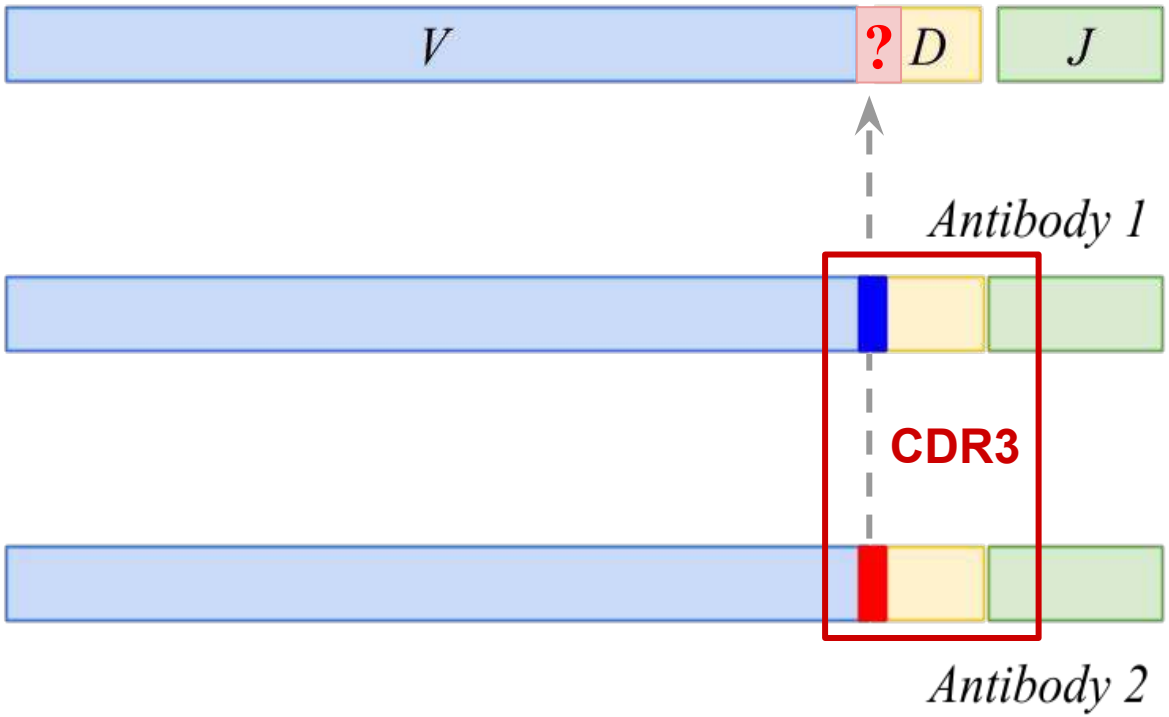


- One can easily identify mutations in the V segment using alignment against the **template** (germline V segment)
- **But there is no template for CDR3!**

Undirected edge case



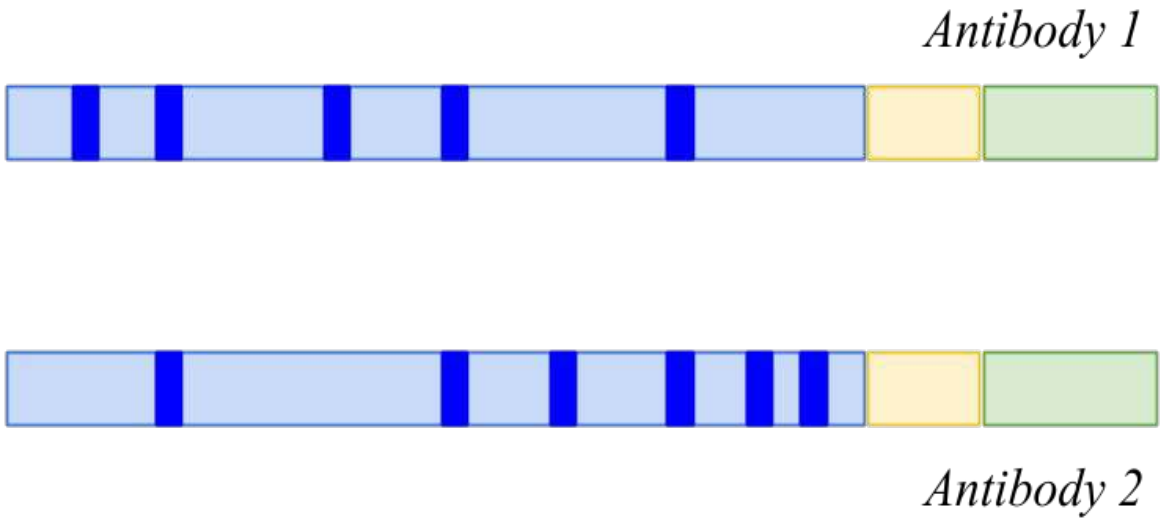
Undirected edge case



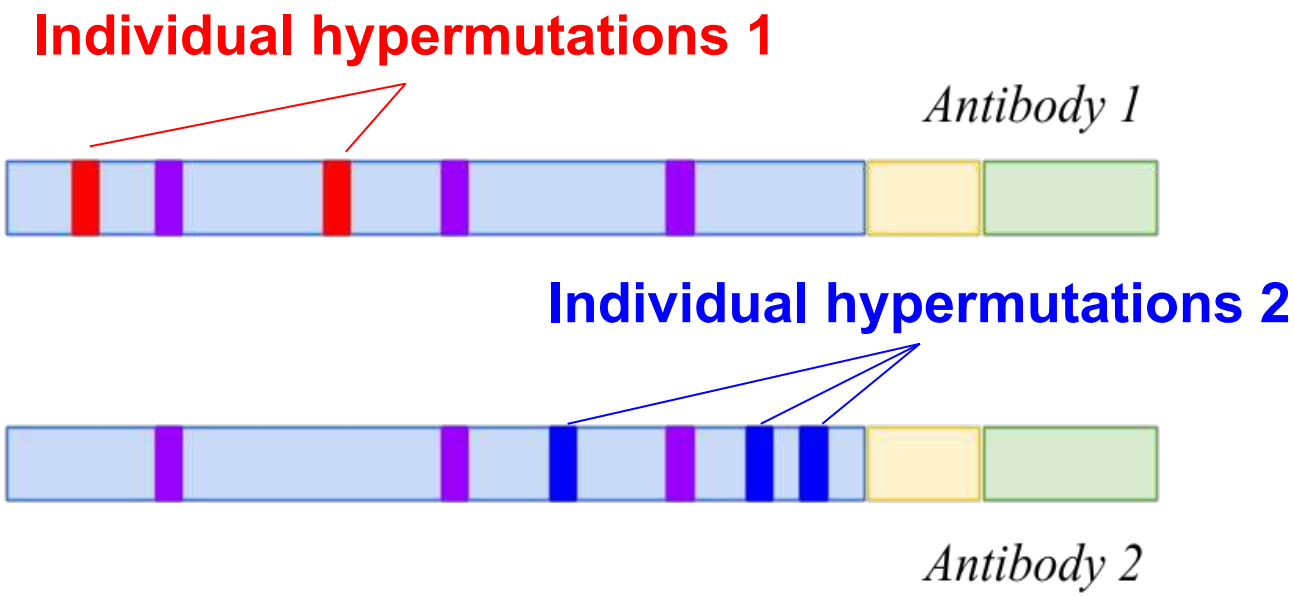
Antevolo algorithm

- If there is an edge between reads *A* and *B* in the CDR3 graph, we say that *A* and *B* are connected with an edge of one of the following types:
  - We call the edge ***directed***, if all the VSHMs of B are presented in A.
  - We call the edge ***undirected***, if the VSHMs sets of *B* and *A* are equal (if two reads differ only in CDR3)
  - Otherwise, we call it ***intersected***.

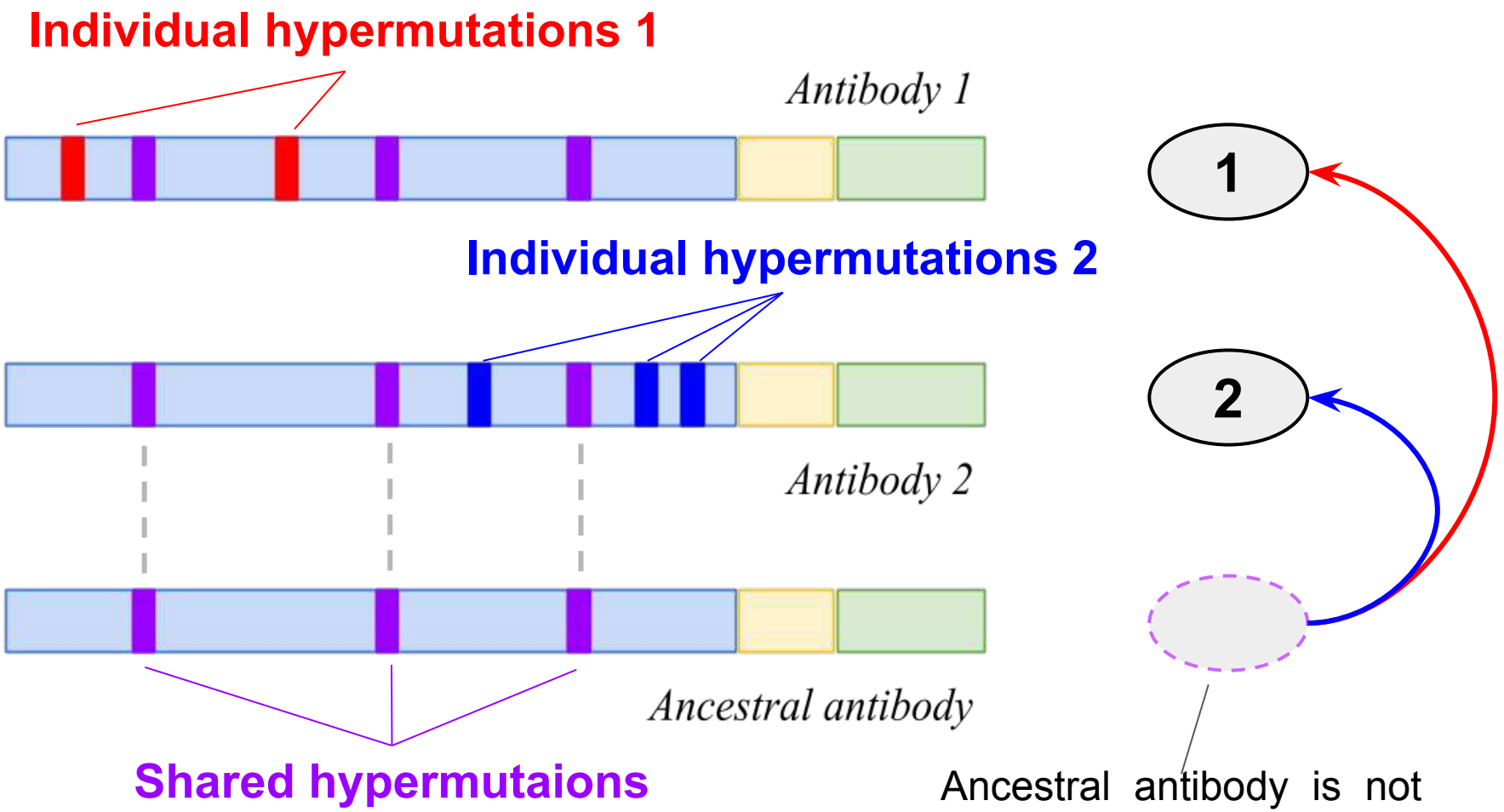
Intersected edge case



Intersected edge case



Intersected edge case

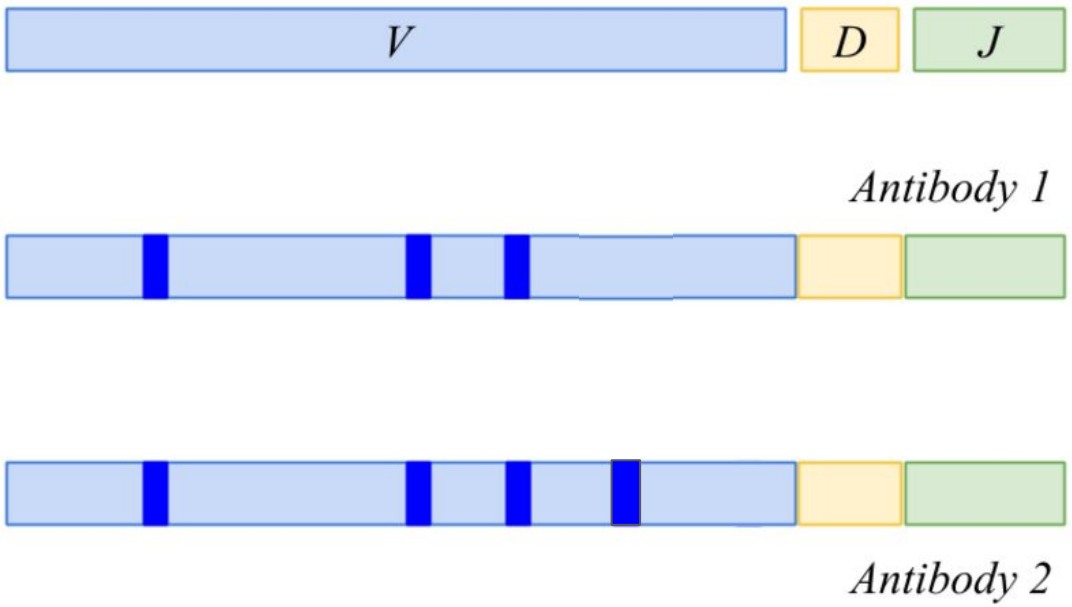


We want to construct a forest using only *directed* and *undirected* edges and then reconstruct some missing vertices with *intersected* edges.

New edge type

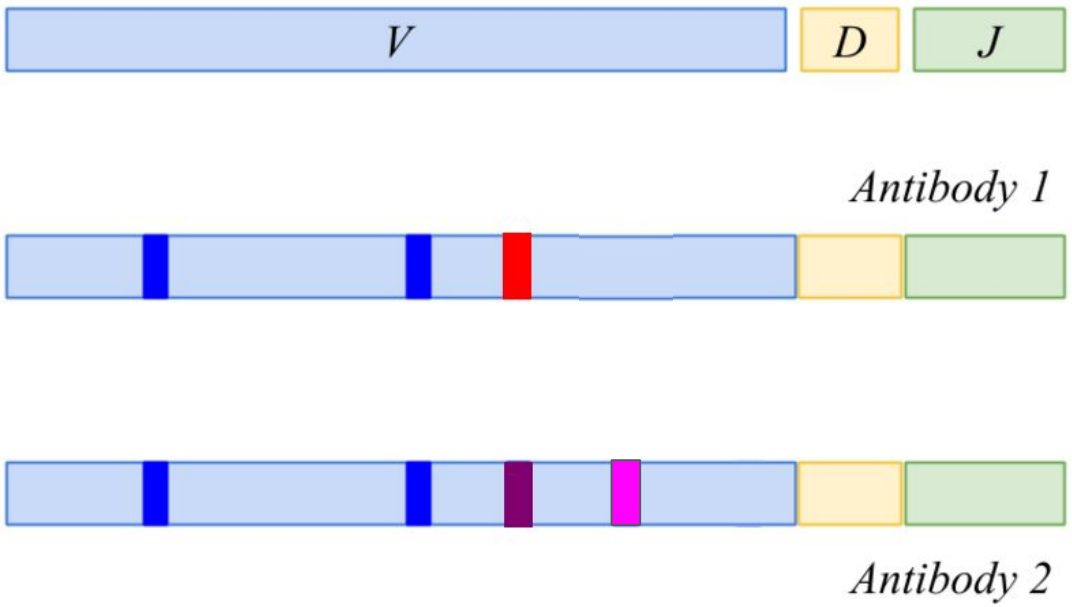
- Sometimes not all the VSHMs of B are presented in A, but the individual hypermutations of A and B have the same positions. We call such an edge ***double mutated***

Double mutated edge case



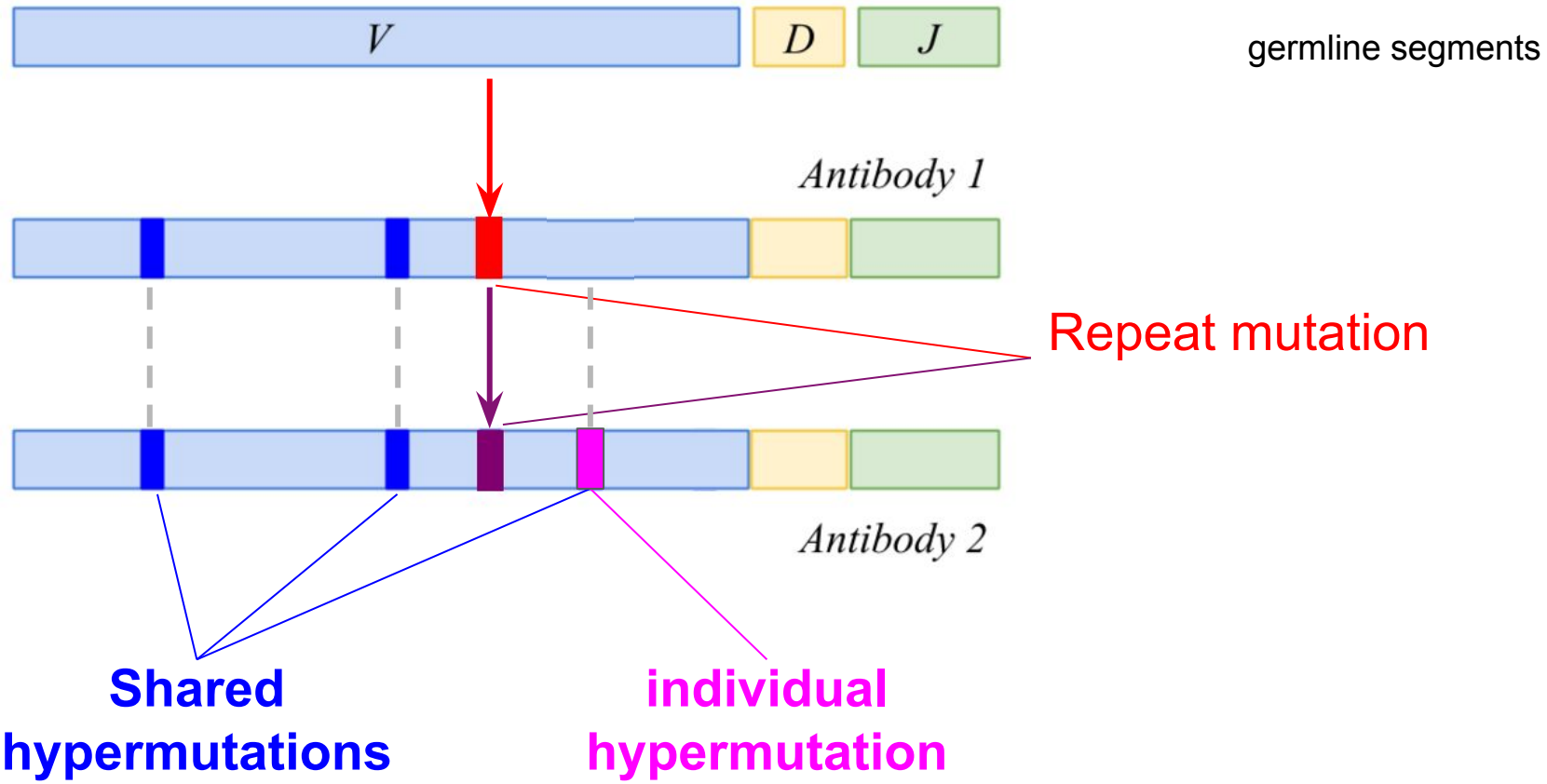
germline segments

Double mutated edge case



germline segments

Double mutated edge case



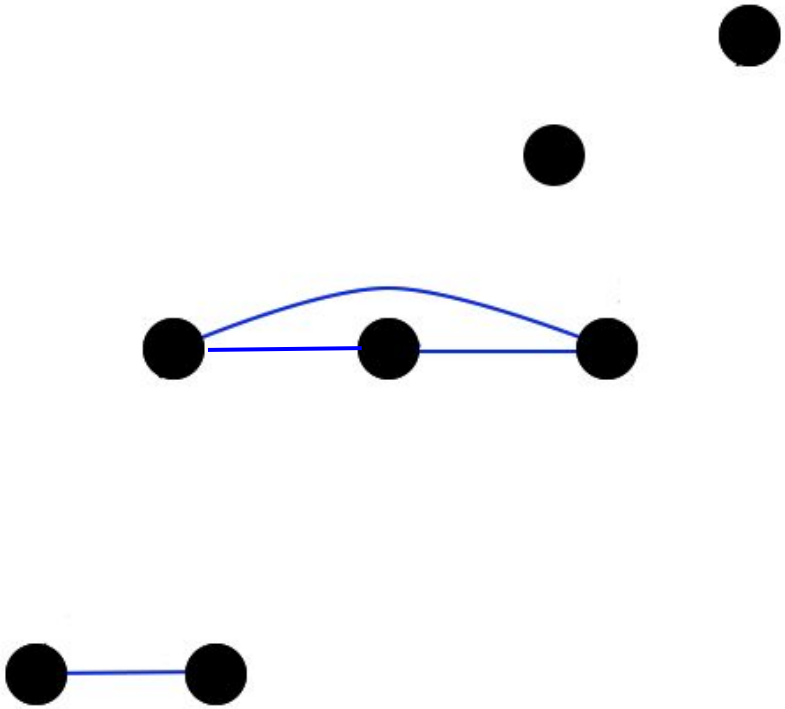
Tree (forest) construction

- Consider a graph containing only ***undirected*** edges. First, we compute connected components of this graph
  - Mutations in the CDR3 are more probable, so we want to draw as many undirected edges as it is possible
  - We will set the directions in the components later
- For each component we then choose the shortest directed edge with a component's vertex as the endpoint
- And then we set the inner-components edges directions using probabilistic SHM model

Tree (forest) construction

- Consider a graph containing only ***undirected*** edges. First, we compute connected components of this graph
  - Mutations in the CDR3 are more probable, so we want to draw as many undirected edges as it is possible
  - We will set the directions in the components later
- For each component we then choose the shortest directed edge with a component's vertex as the endpoint
- And then we set the inner-components edges directions using probabilistic SHM model

Tree (forest) construction



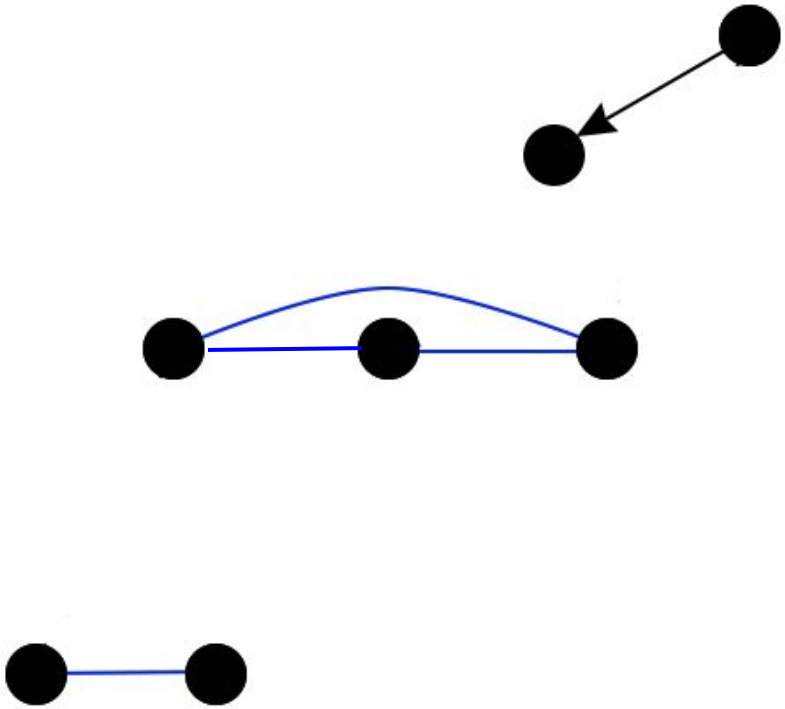
Blue edges are **undirected**.
Black edges are **directed.**

Vertices are ranked by height due to their SHMs number

Tree (forest) construction

- Consider a graph containing only ***undirected*** edges. First, we compute connected components of this graph
  - Mutations in the CDR3 are more probable, so we want to draw as many undirected edges as it is possible
  - We will set the directions in the components later
- For each component we then choose the shortest directed edge with a component's vertex as the endpoint
- And then we set the inner-components edges directions using probabilistic SHM model

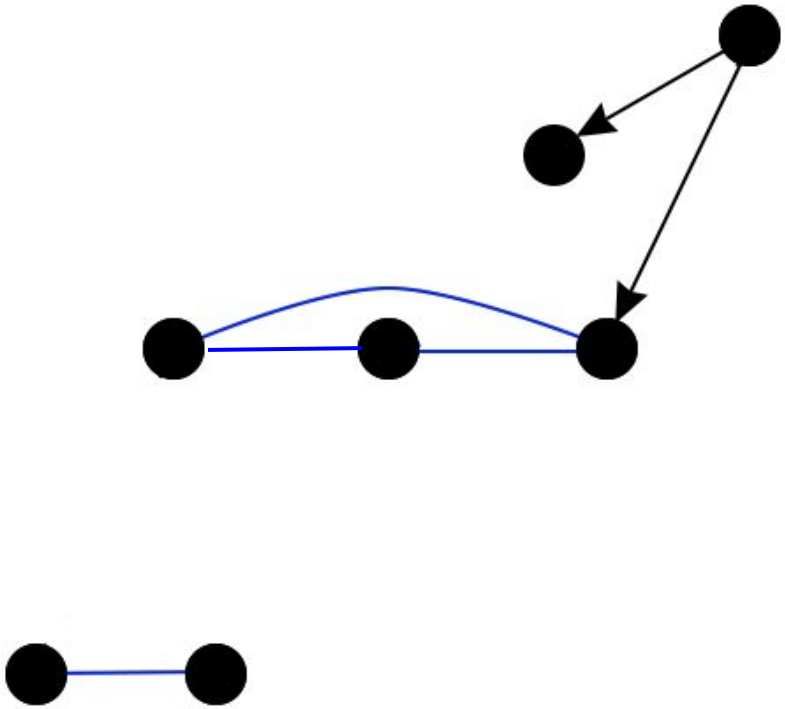
Tree (forest) construction



Blue edges are **undirected**.
Black edges are **directed.**

Vertices are ranked by height due to their SHMs number

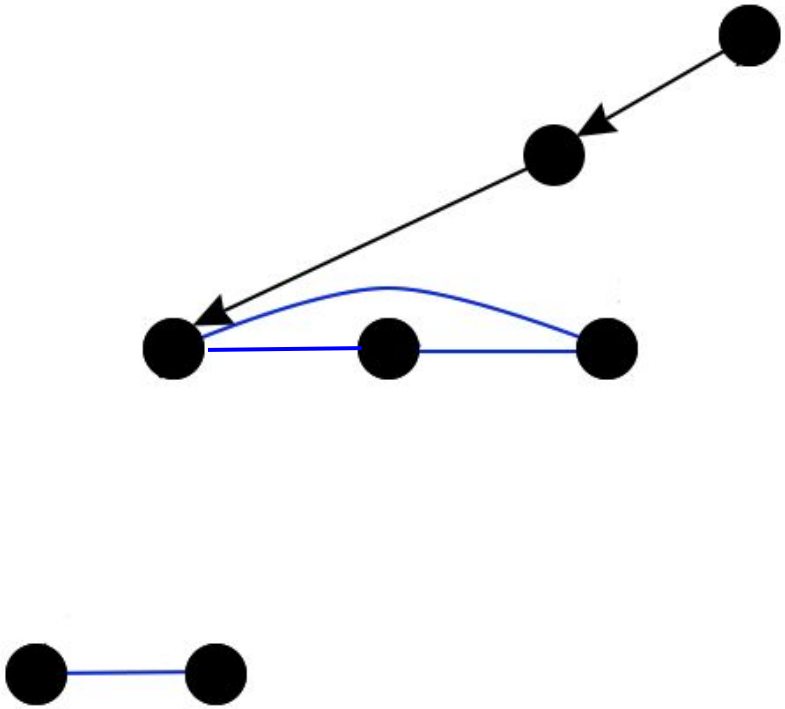
Tree (forest) construction



Blue edges are **undirected**.
Black edges are **directed.**

Vertices are ranked by height due to their SHMs number

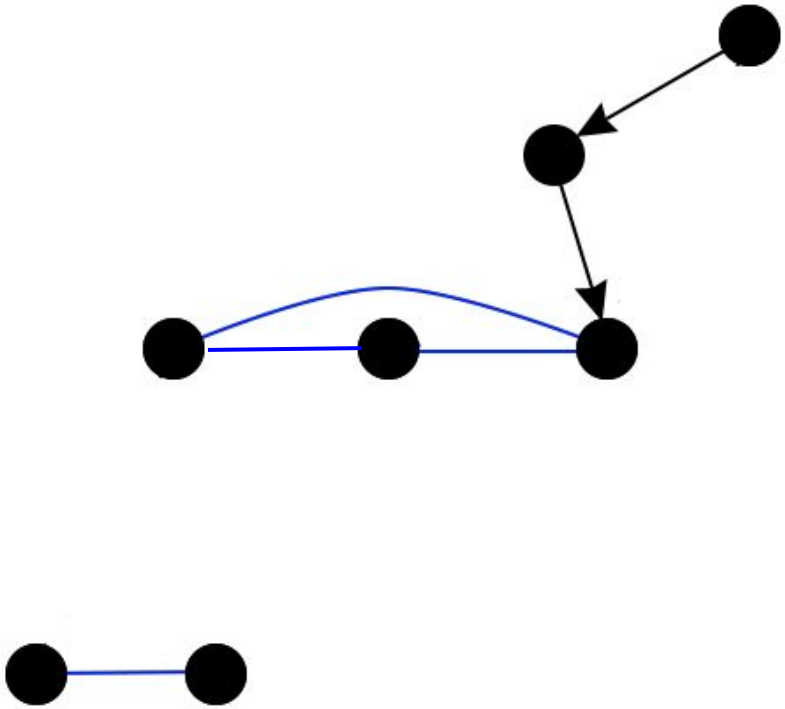
Tree (forest) construction



Blue edges are **undirected**.
Black edges are **directed.**

Vertices are ranked by height due to their SHMs number

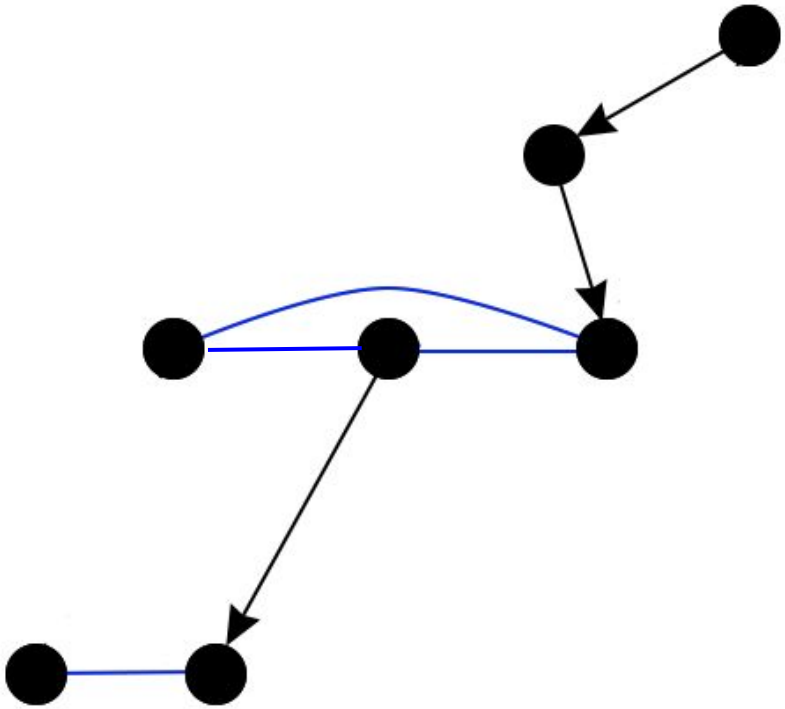
Tree (forest) construction



Blue edges are **undirected**.
Black edges are **directed.**

Vertices are ranked by height due to their SHMs number

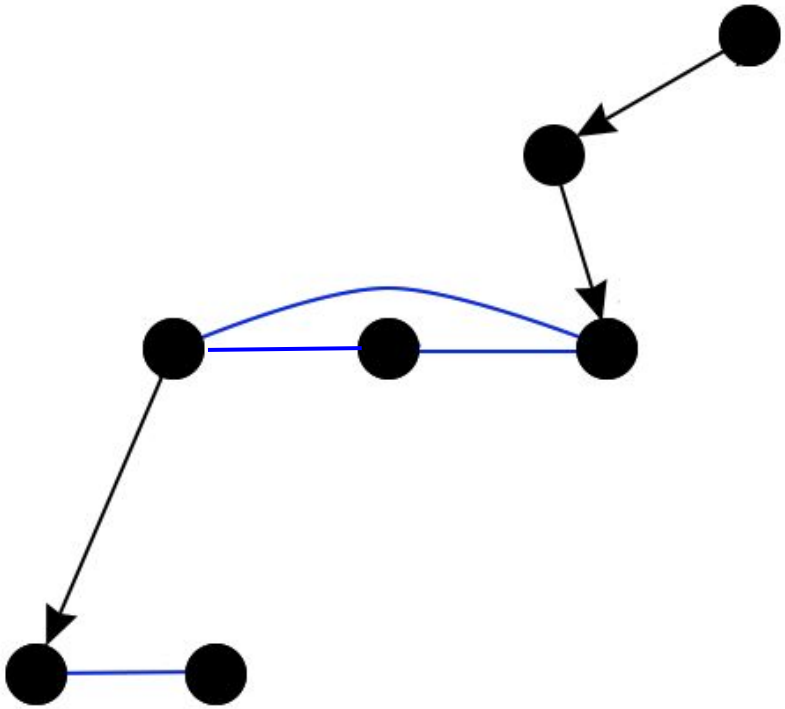
Tree (forest) construction



Blue edges are **undirected**.
Black edges are **directed.**

Vertices are ranked by height due to their SHMs number

Tree (forest) construction



Blue edges are **undirected**.
Black edges are **directed.**

Vertices are ranked by height due to their SHMs number

Tree (forest) construction

- Consider a graph containing only ***undirected*** edges. First, we compute connected components of this graph
  - Mutations in the CDR3 are more probable, so we want to draw as many undirected edges as it is possible
  - We will set the directions in the components later
- For each component we then choose the shortest directed edge with a component's vertex as the endpoint
- And then we set the inner-components edges directions using probabilistic SHM model

Probabilistic SHM model

5-mer	Freq	A	C	G	T
ACAAC	**83**	–	0.24	***0.48***	0.28
GGCGT	**1742**	0.22	–	0.12	***0.66***
CCGTC	**12**	0.35	***0.52***	–	0.13
TCTCC	**516**	0.32	***0.54***	0.14	–

- The SHM model takes into account both the mutated nucleotide and its neighbours
- Detect new hot spots and compares SHMs in IG chains

Yaari et al., *Front Immunol,* 2013

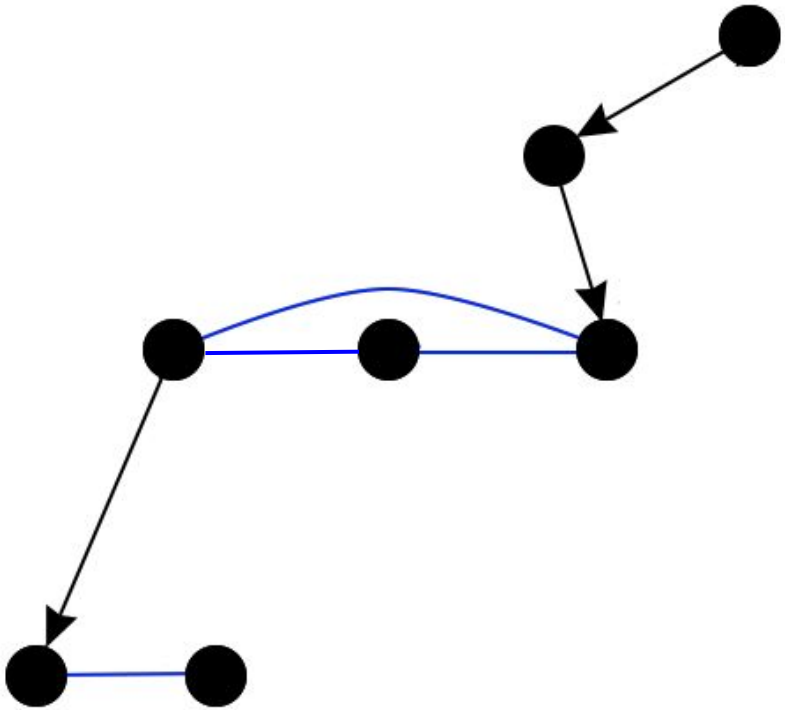
Tree (forest) construction

- Consider a graph containing only ***undirected*** edges. First, we compute connected components of this graph
  - Mutations in the CDR3 are more probable, so we want to draw as many undirected edges as we can
  - We will set the directions in the components later
- For each component we then choose the shortest directed edge with a component's vertex as the endpoint
- And then we set the inner-components edges directions using probabilistic SHM model

Tree (forest) construction

- We set the inner-components edges directions
  - Consider a connected component on undirected edges.
  - Split each edge (a, b) into a pair of directed edges (a, b) and (b, a)
  - The model gives us the weights of the edges
  - So we simply find the minimum directed spanning tree (also called arborescence) rooted in the endpoint of the parent directed edge on which our component is hanged

Tree (forest) construction



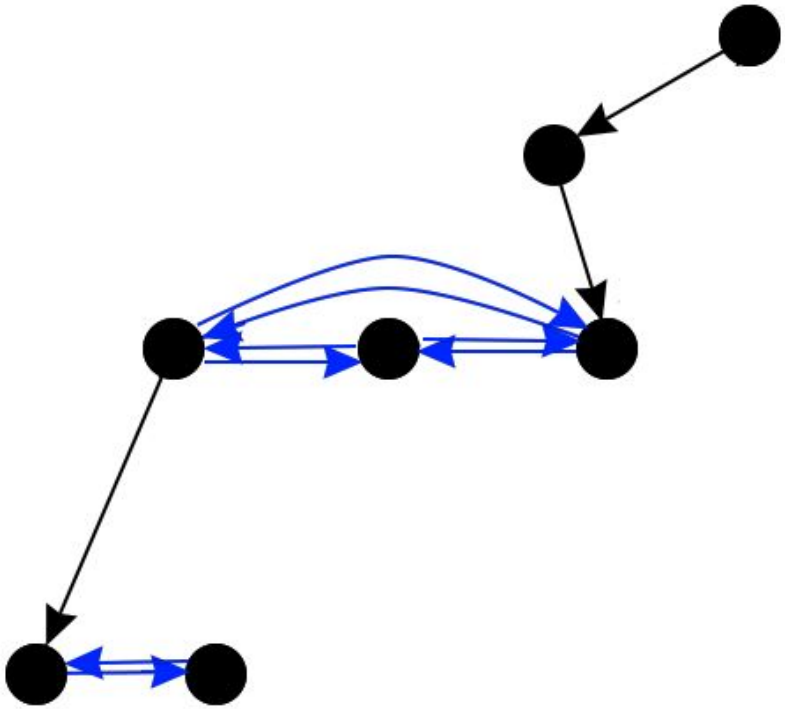
Blue edges are **undirected**.
Black edges are **directed.**

Vertices are ranked by height due to their SHMs number

Tree (forest) construction

- We set the inner-components edges directions
  - Consider a connected component on undirected edges.
  - Split each edge into a pair
  - The model gives us the weights of the edges
  - So we simply find the minimum directed spanning tree (also called arborescence) rooted in the endpoint of the parent edge on which our component is hanged

Tree (forest) construction



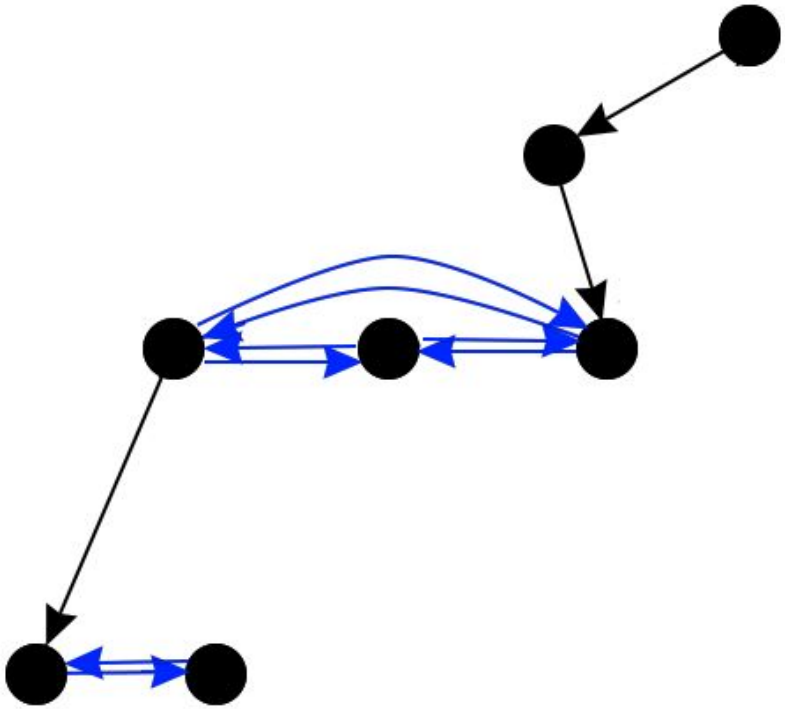
Blue edges are **undirected**.
Black edges are **directed.**

Vertices are ranked by height due to their SHMs number

Tree (forest) construction

- We set the inner-components edges directions
  - Consider a connected component on undirected edges.
  - Split each edge into a pair
  - The model gives us the weights of the edges
  - So we simply find the minimum directed spanning tree (also called arborescence) rooted in the endpoint of the parent edge on which our component is hanged

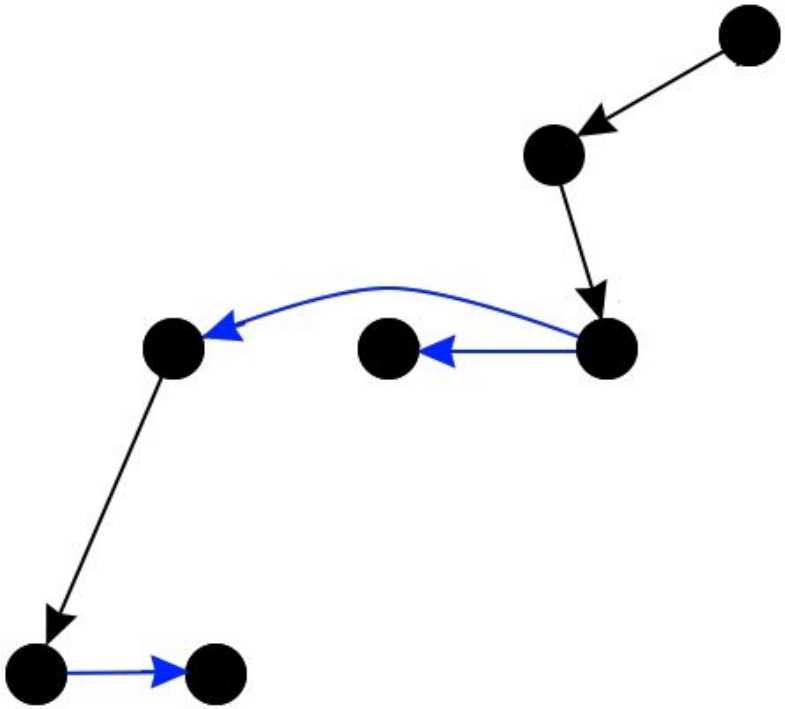
Tree (forest) construction



Blue edges are **undirected**.
Black edges are **directed.**

Vertices are ranked by height due to their SHMs number

Tree (forest) construction



Blue edges are **undirected**.
Black edges are **directed.**

Vertices are ranked by height due to their SHMs number

Repertoire analysis

Nonproductive clones

- A clone is nonproductive if
  - it has a SHM into stop-codon or
  - is out-of-frame.
- It is reasonable to assume that nonproductive clones do not evolve
- But they do!

Nonproductive clones

- A clone is nonproductive if
  - it has a SHM into stop-codon or
  - is out-of-frame.
- It is reasonable to assume that nonproductive clones do not evolve
- But they do!

Nonproductive clones

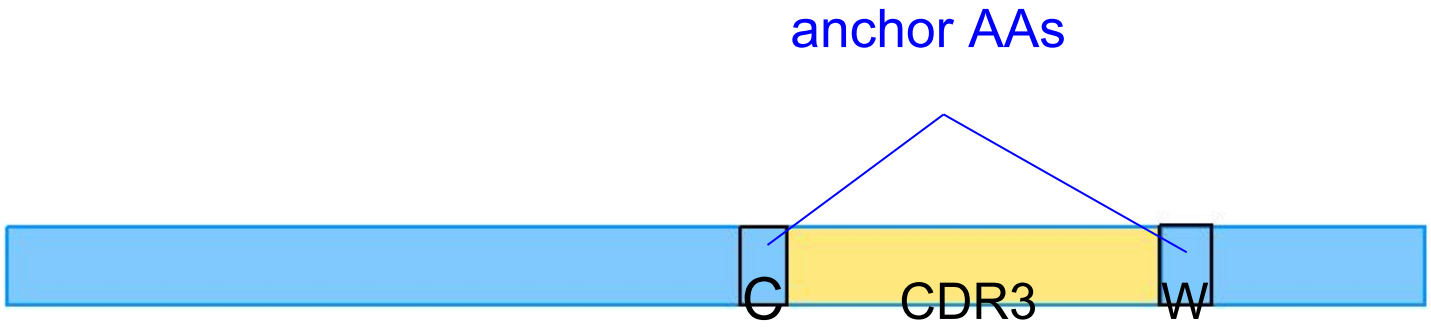
- A clone is nonproductive if
  - it has a SHM into stop-codon or
  - is out-of-frame.
- It is reasonable to assume that nonproductive clones do not evolve
- But they do!

Nonproductive clones

- Mutation probability (non-leaf vertices percentage)
  - for an arbitrary clone: *30±5%* around the datasets
  - for nonproductives:
    - heavy chain: *< 4%* (could be explained with algorithm errors)
    - light chains: *16%* (way too much, may be allelic inclusions)

Anchor amino acids

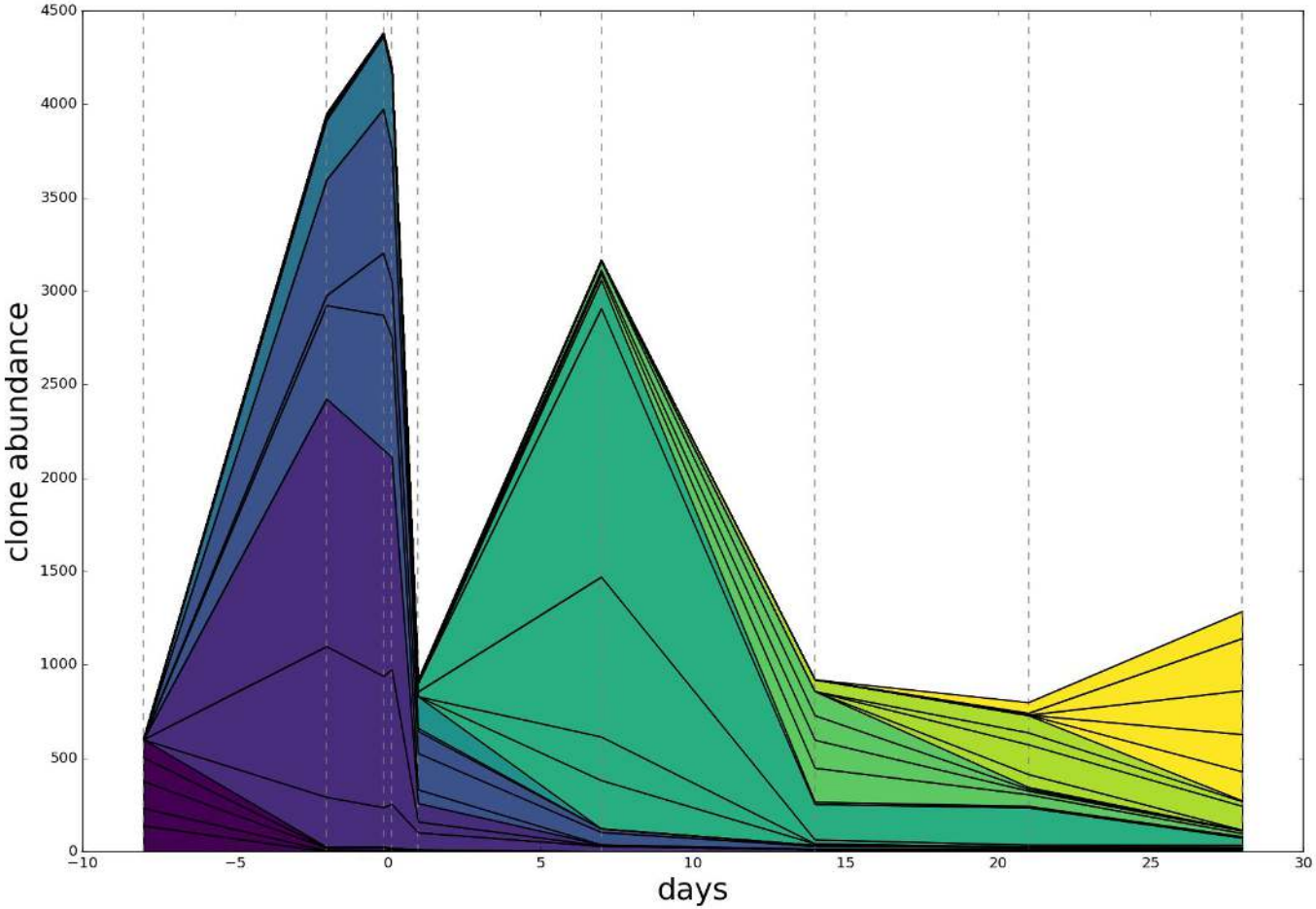
- Impotrant AA positions around the CDR3
- Change of anchor AAs can totally disrupt the CDR3 structure



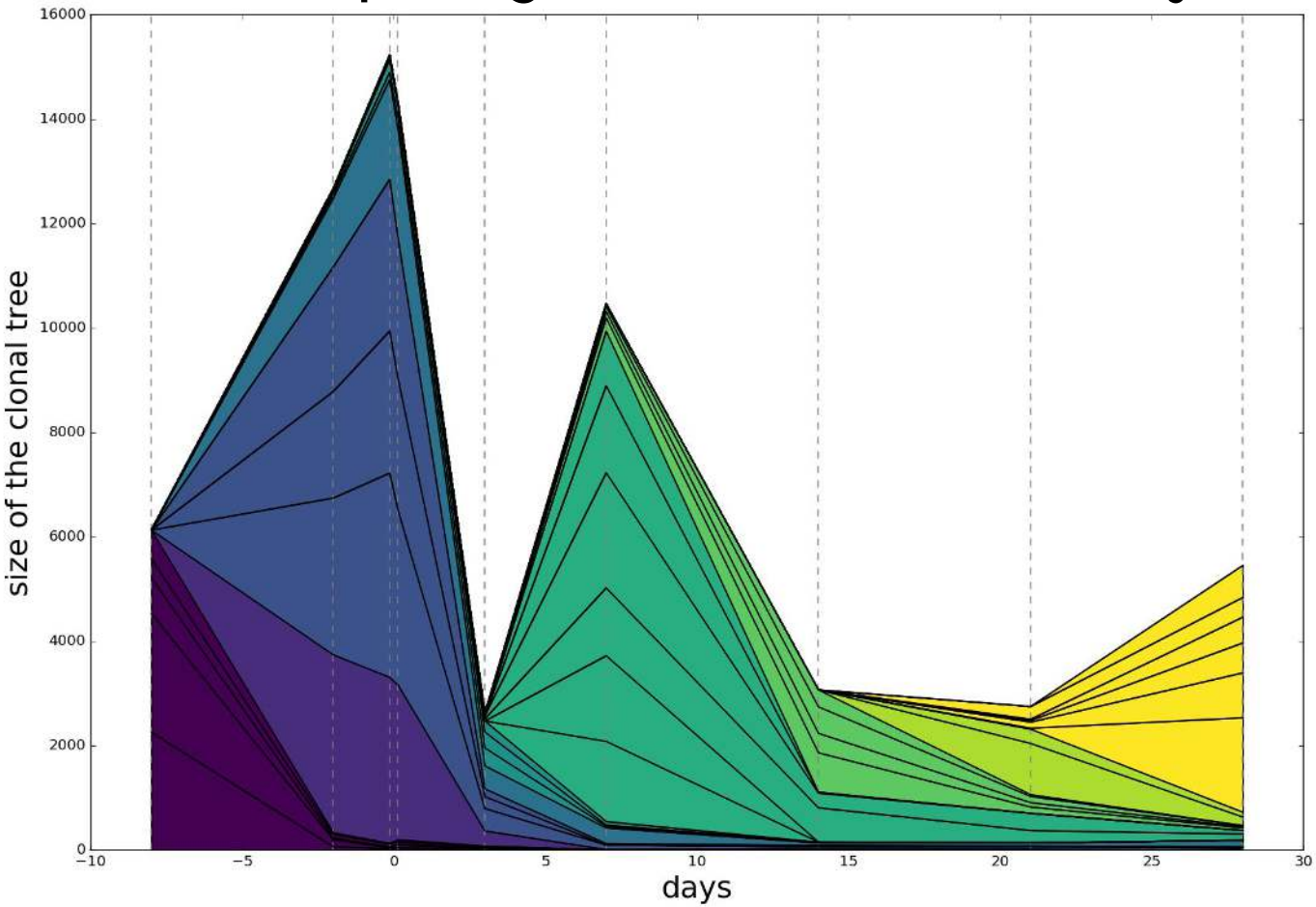
Mutation probability

- Mutation probability (non-leaf vertices percentage)
  - for an arbitrary clone: *30±5%* around the datasets
  - for vertices that differ from their parents on at least one anchor AA: *5-6%*

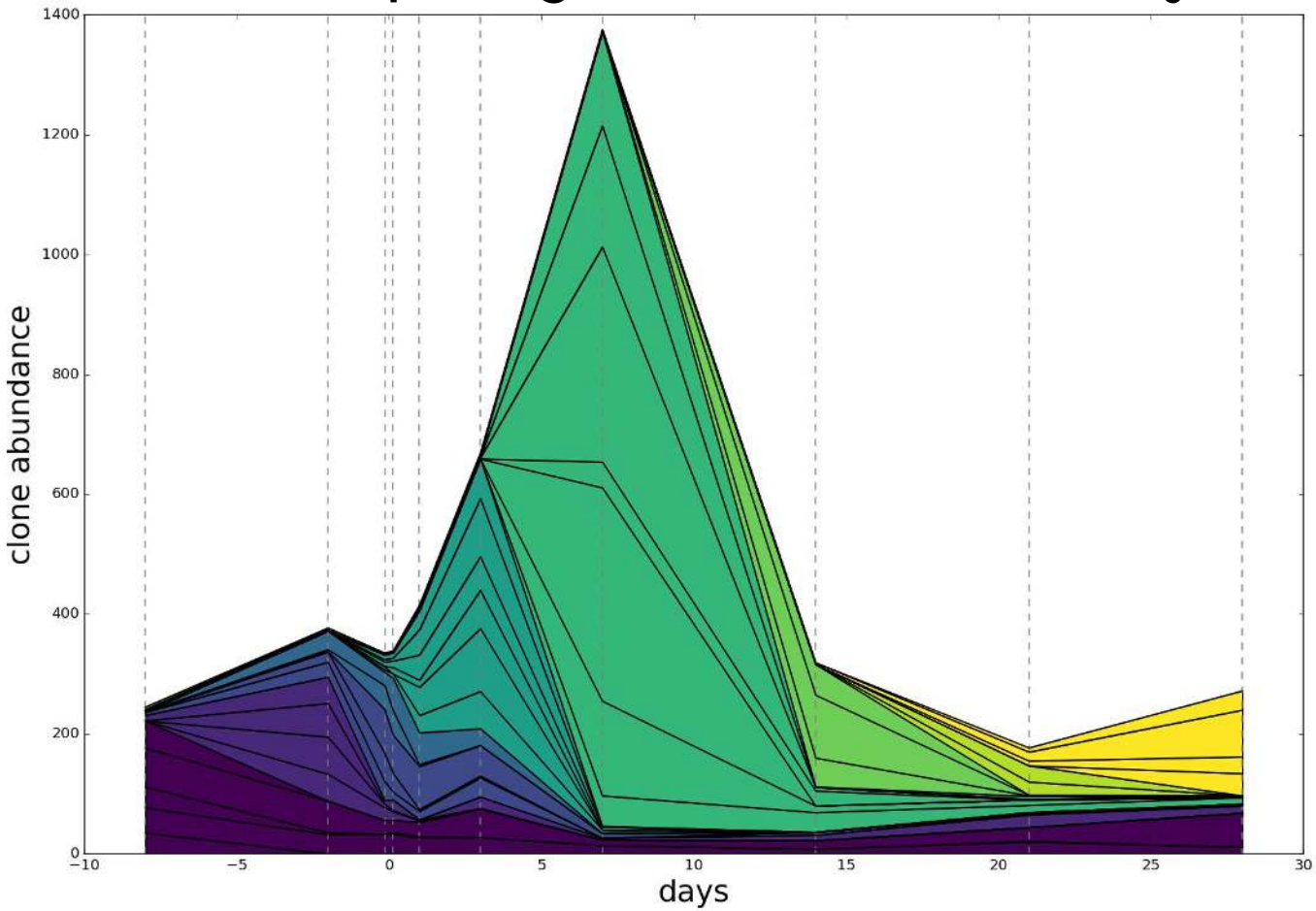
FV, top5 largest clones on each day



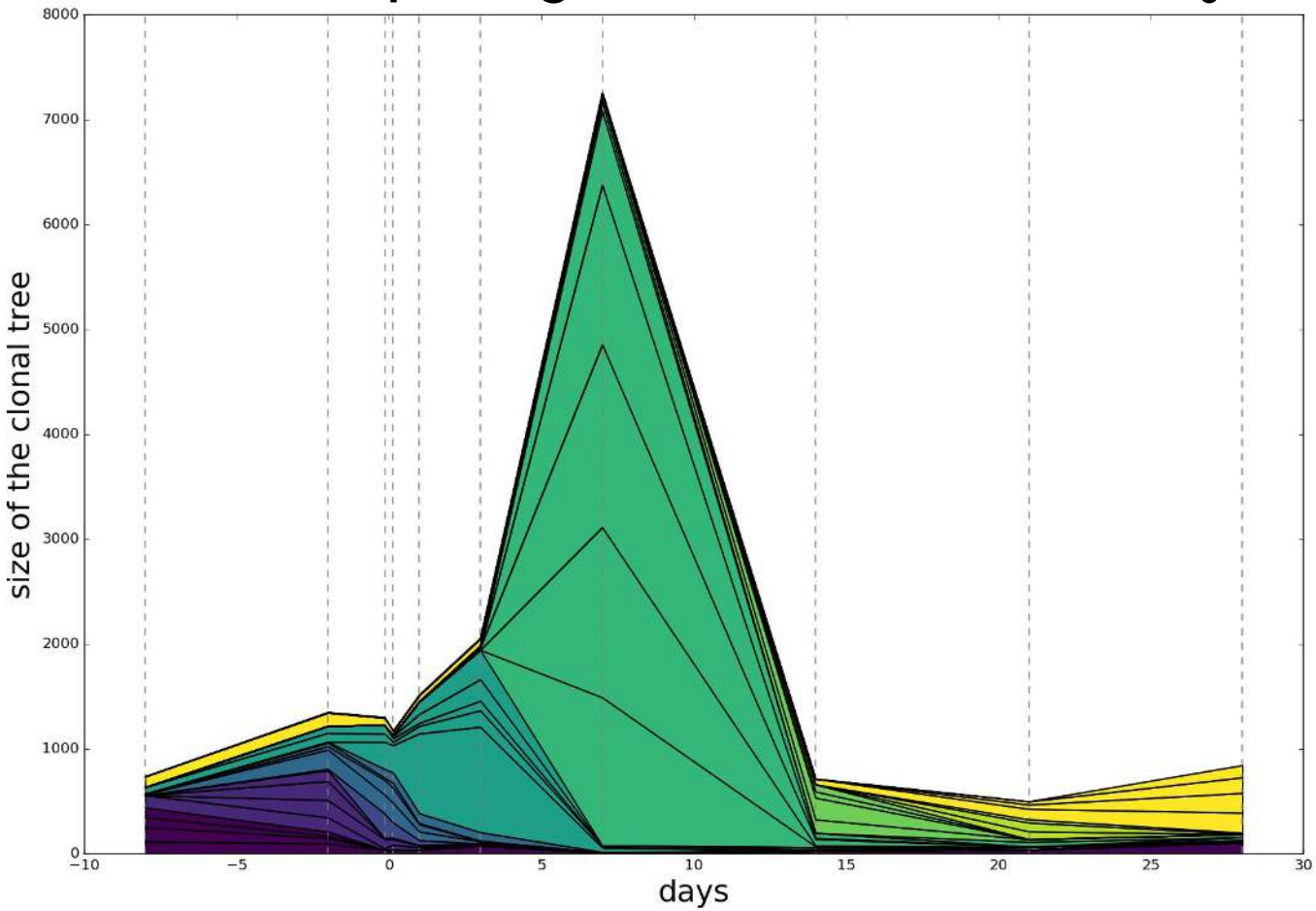
FV, top5 largest clusters on each day



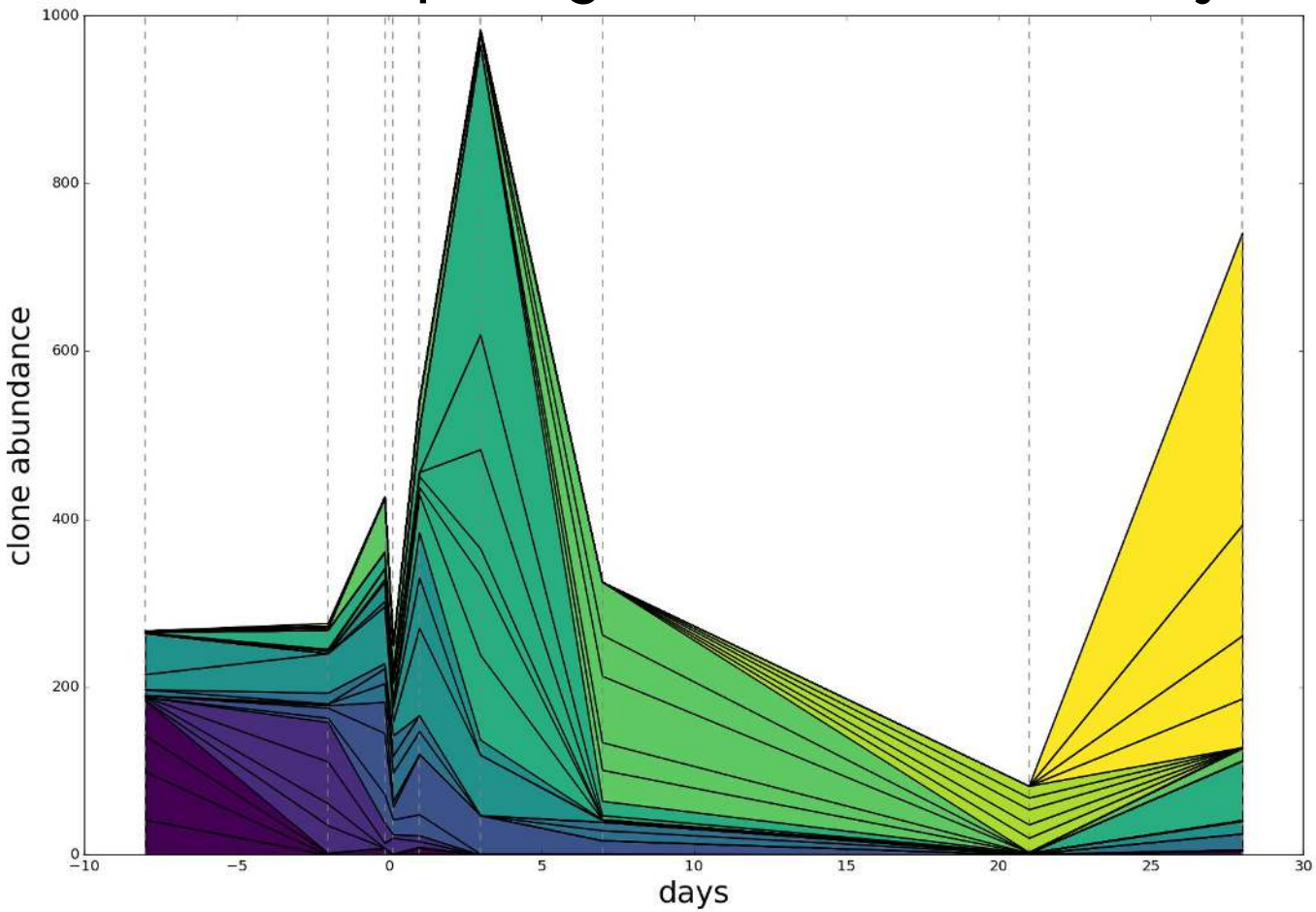
IDO, top5 largest clones on each day



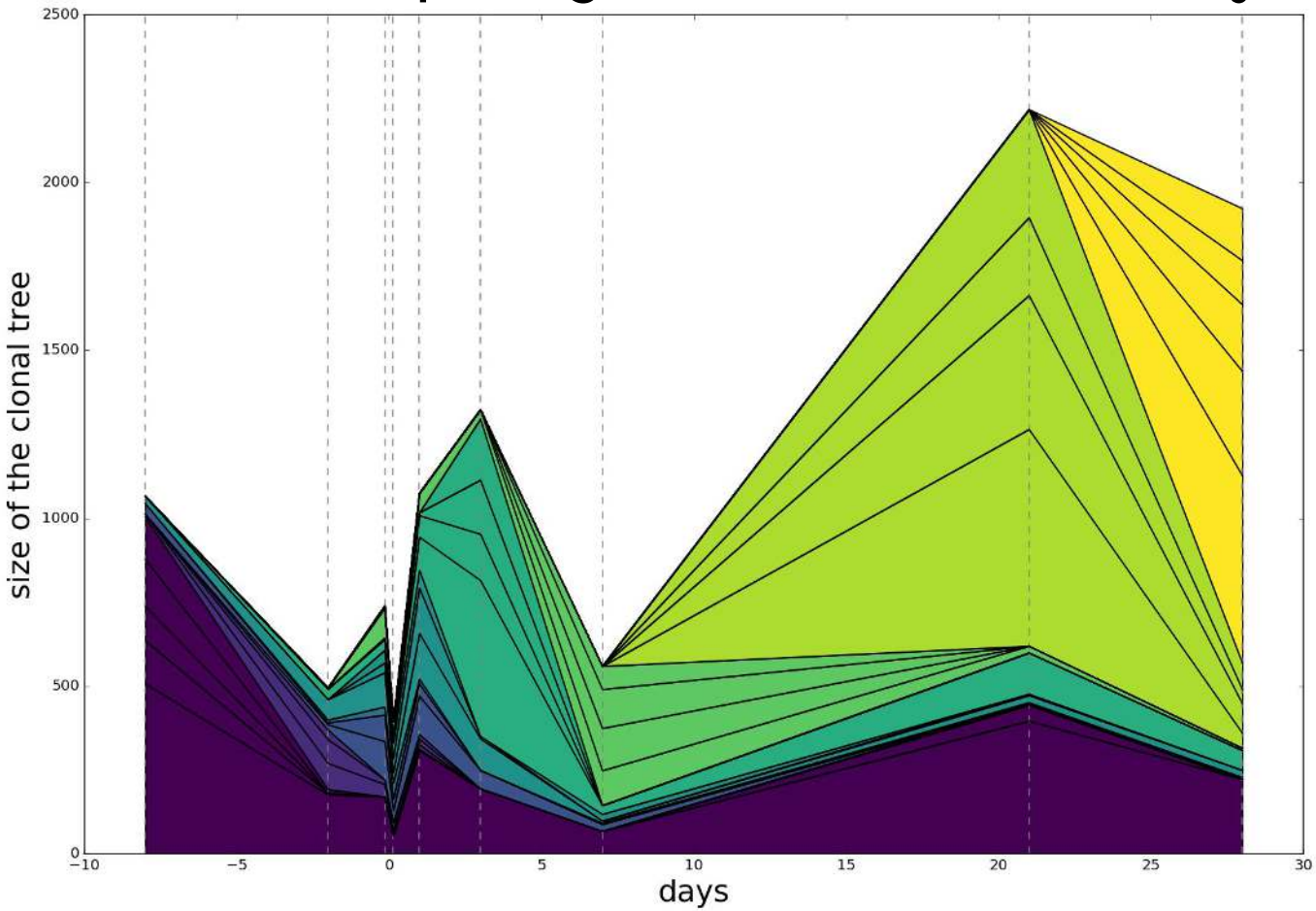
IDO, top5 largest clusters on each day



GMC, top5 largest clones on each day



GMC, top5 largest clusters on each day



Thank you for your attention!