

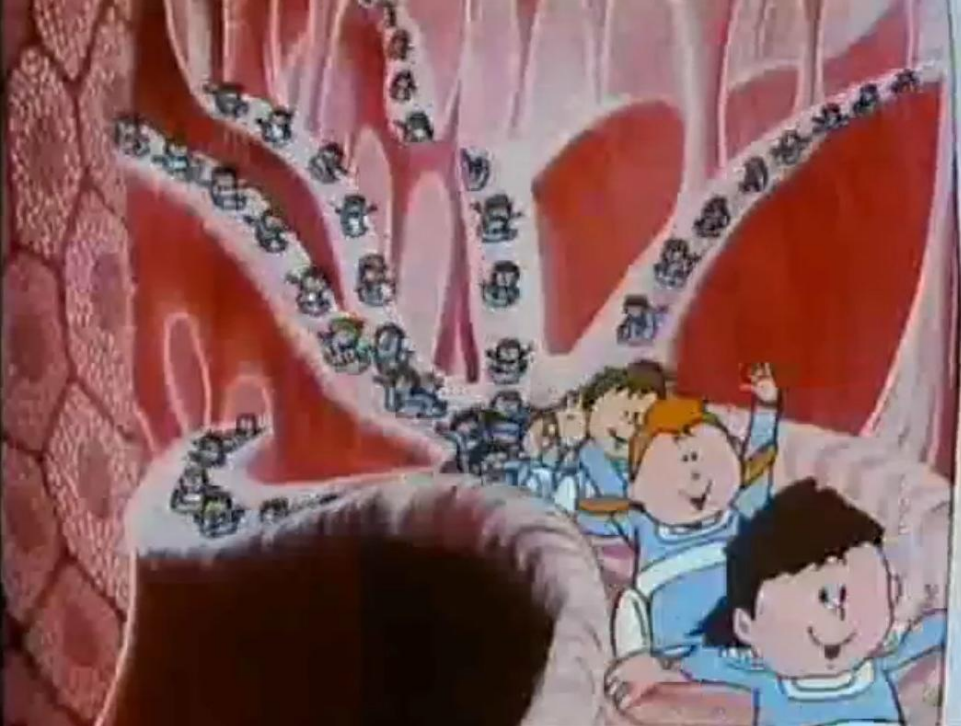
From alignment-free heuristics to an interactive visualization: V(D)J repertoire analysis in the Vidjil platform

Mathieu Giraud, Ryan Herbert,
Mikaël Salson, Florian Thonier

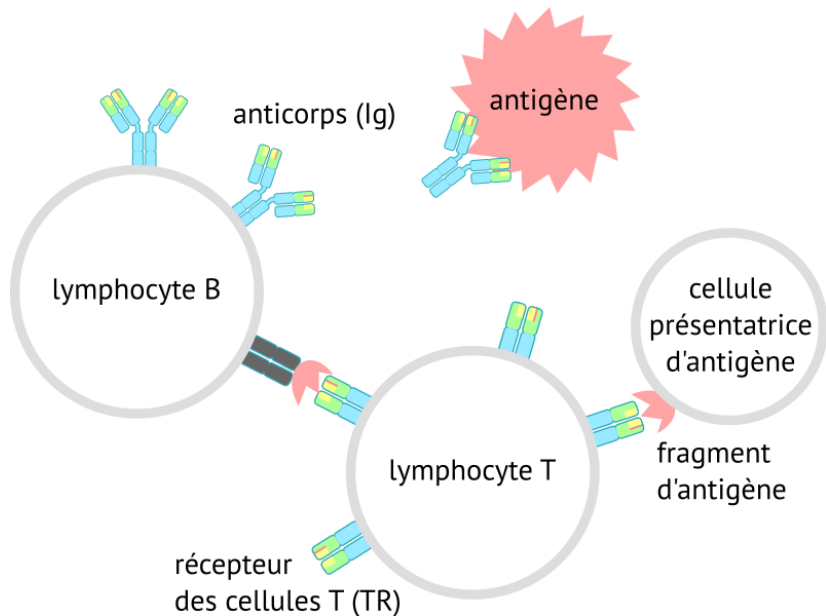


Bonsai bioinformatics, CRISAL (Université Lille, CNRS)
VidjilNet consortium, Inria

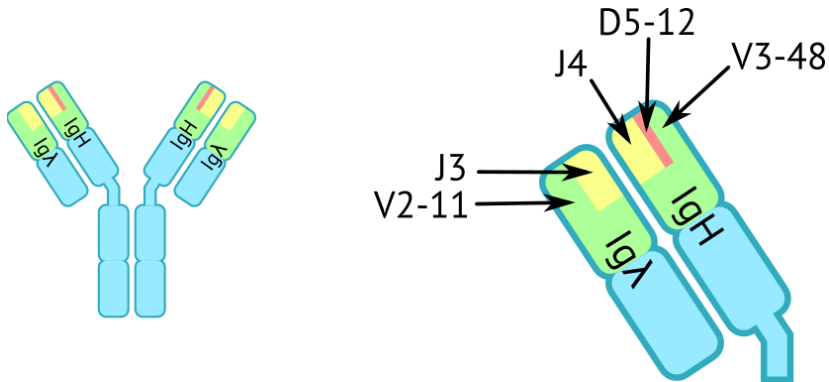
April 25, 2019



The Adaptive Immune System



TCR and Antibody Specificity – V(D)J Recombination

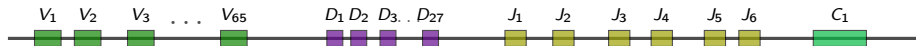


... GGAAGGGCAGAATTA ...
v2-11 GGATGGG GAATTA J3

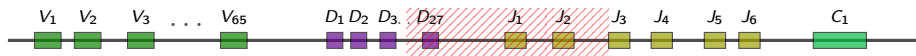
TCR and Antibody Specificity – V(D)J Recombination



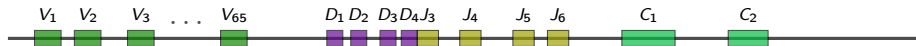
V(D)J recombinations are responsible for receptor diversity



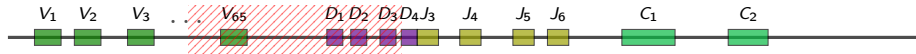
V(D)J recombinations are responsible for receptor diversity



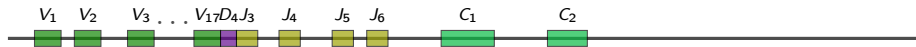
V(D)J recombinations are responsible for receptor diversity



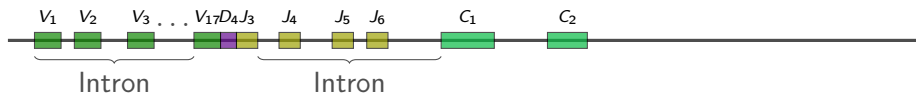
V(D)J recombinations are responsible for receptor diversity



V(D)J recombinations are responsible for receptor diversity



V(D)J recombinations are responsible for receptor diversity



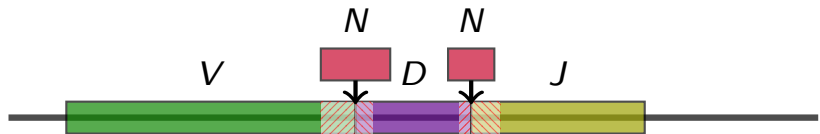
V(D)J recombinations are responsible for receptor diversity



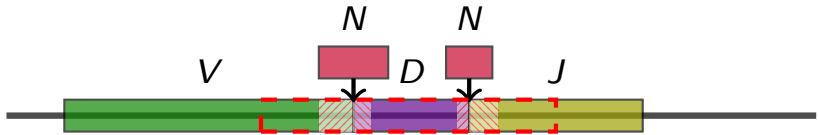
V(D)J recombinations are responsible for receptor diversity



V(D)J recombinations are responsible for receptor diversity

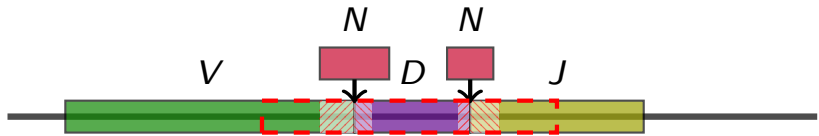


V(D)J recombinations are responsible for receptor diversity

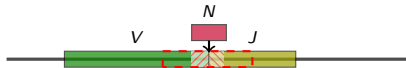


Diversity region

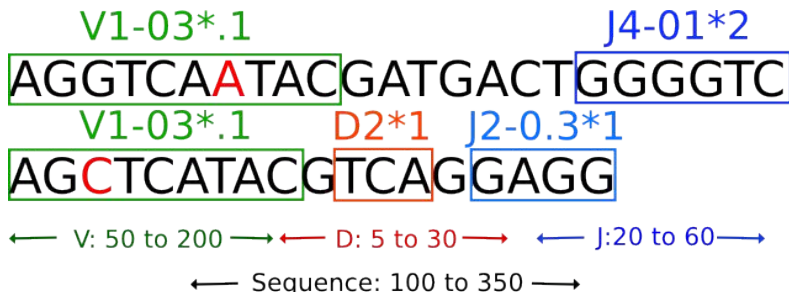
V(D)J recombinations are responsible for receptor diversity



Diversity region

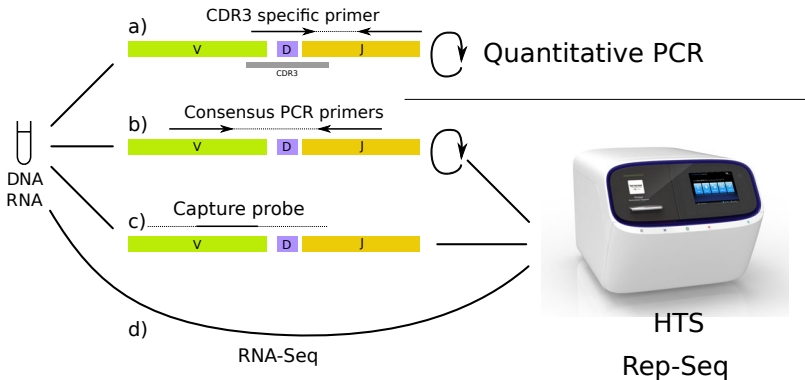


TCR and Antibody Specificity – V(D)J Recombination



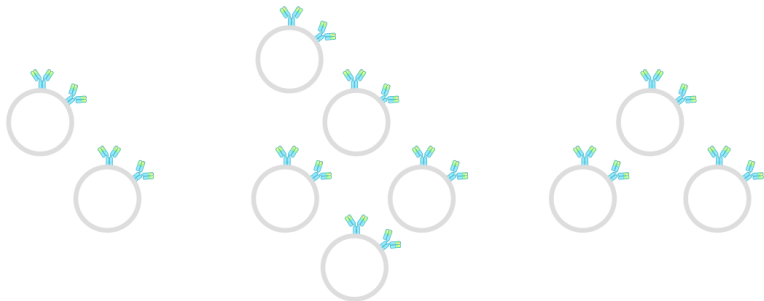
Immune Repertoire Sequencing (RepSeq)

Strategies – Sequencing millions of V(D)J recombinations from T-cells or B-cells



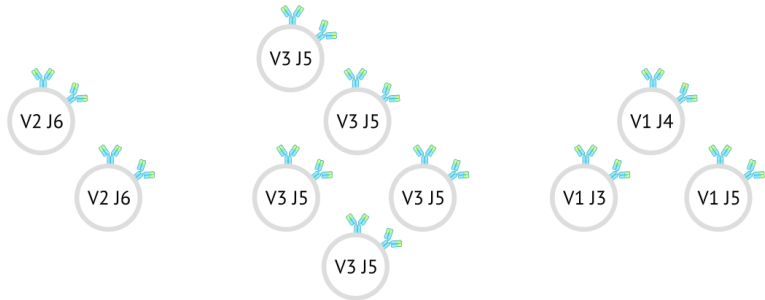
Immune Repertoire Sequencing (RepSeq)

Identification of all VDJ recombinations



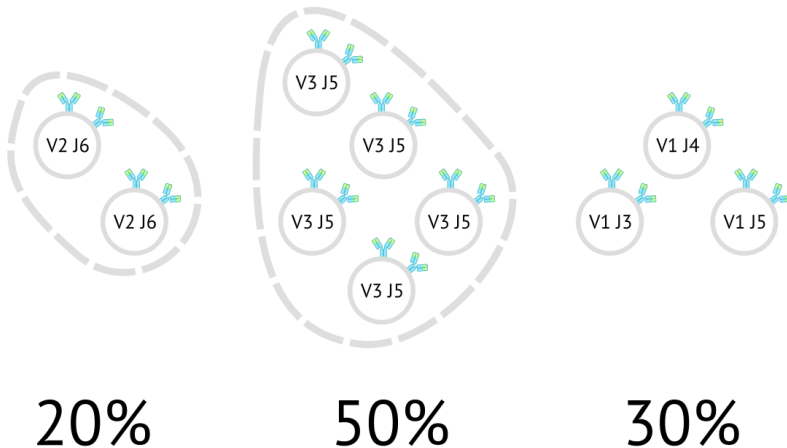
Immune Repertoire Sequencing (RepSeq)

Identification of all VDJ recombinations



Immune Repertoire Sequencing (RepSeq)

Identification of all VDJ recombinations



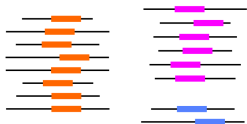
Vidjil

High-throughput Repertoire Sequencing (RepSeq) analysis

Web Application

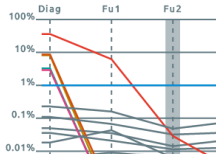
Patient database
Server

Vidjil-algo

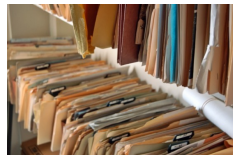


C++

Client



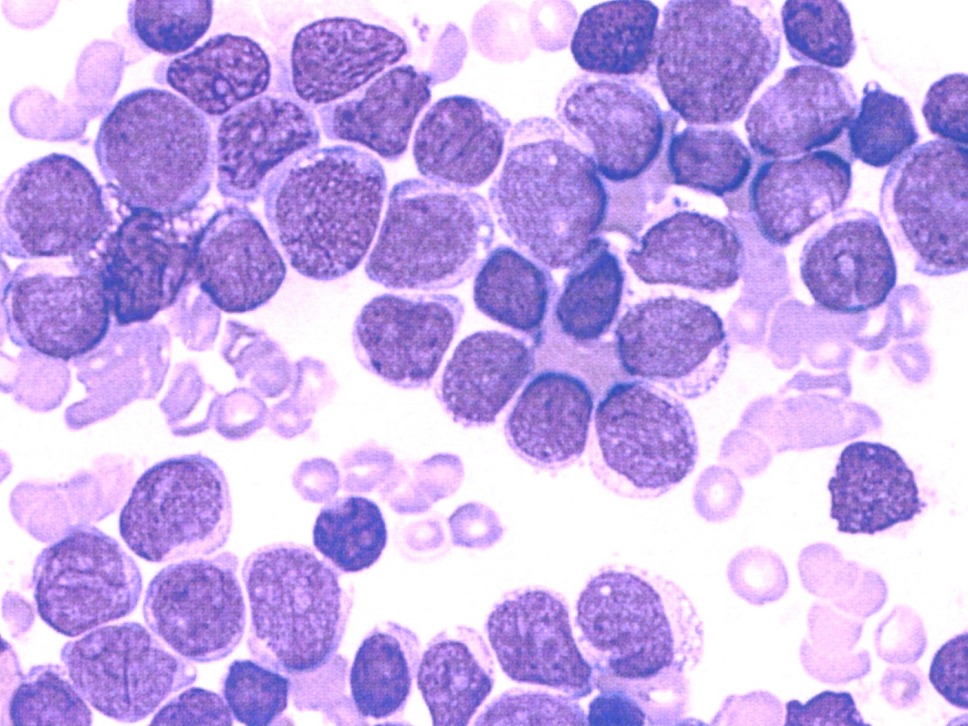
Javascript, d3.js



Python, web2py,
AJAX

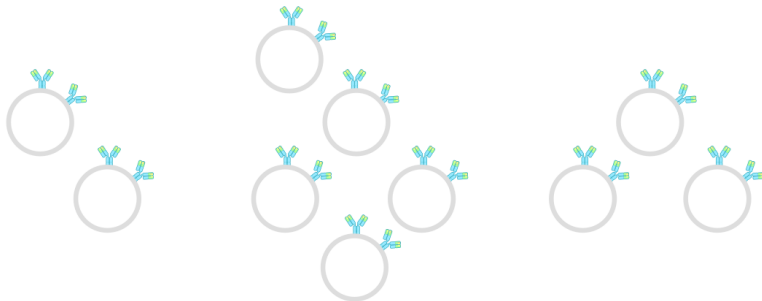
- ▶ code on <http://git.vidjil.org/>
- ▶ open-source (GPL v3), public issue tracker (Gitlab)
- ▶ continuous integration, > 2,000 unit and functional tests

Duez et al., PLOS One, 2016



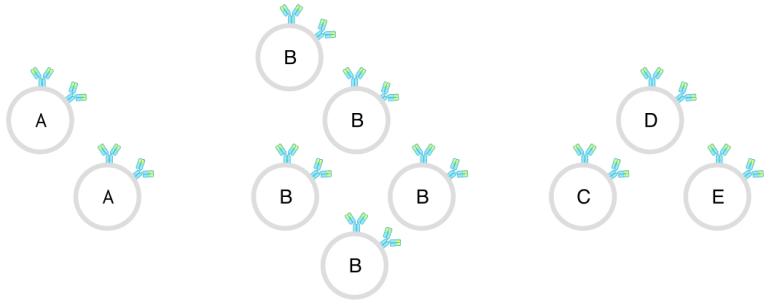
Immune Repertoire Sequencing (RepSeq)

Clone clustering



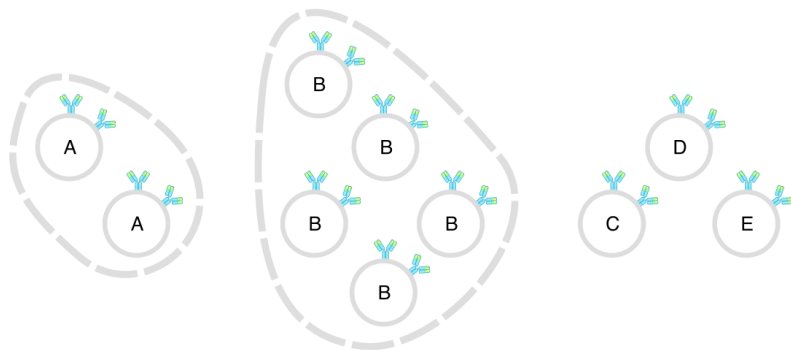
Immune Repertoire Sequencing (RepSeq)

Clone clustering



Immune Repertoire Sequencing (RepSeq)

Clone clustering



20%

50%

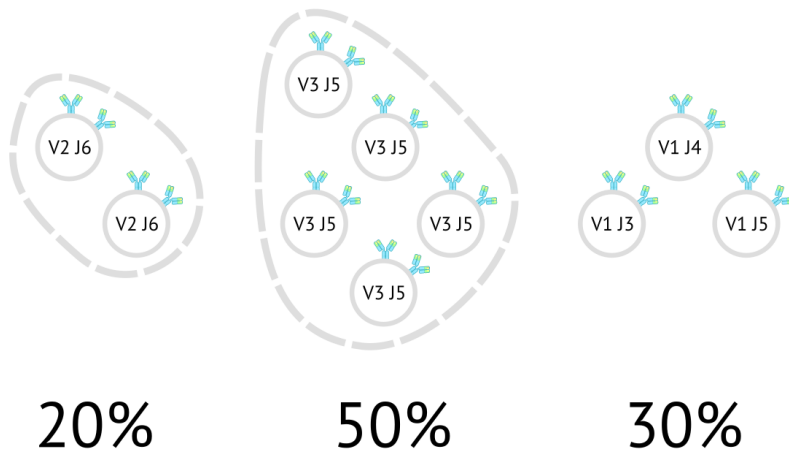
30%

1 000 000 VDJ = 100 s

Giraud, Salson et al., BMC Genomics, 2014

Immune Repertoire Sequencing (RepSeq)

Clone clustering



1 000 000 VDJ = 100 s

Giraud, Salson et al., BMC Genomics, 2014

Fast identification of a window centered on the CDR3

Clone clustering

parts of V genes

ACAC	CACG	ACGG	CGGC	GGCC
GCCG	TCTT	CTTC	TTCC	TCCA
CCAA	CAAC	AACC	ACCT	CCTT
CTTG	TTGG	TGGA	ACTT	...

parts of J genes

ATAC	TACT	ACTT	CCAG	CAGC
AGCA	GCAC	TGGG	GGGC	GGCA
GCAA	CAAG	AAGA	AGAG	GAGT
AGTT	GTTG	TTGG	...	

ACACGGCCGTGTATTACTGTGCGAGAGAGCTGAATACTTCCAGCACTGGGGCC

Fast identification of a window centered on the CDR3

Clone clustering

parts of V genes

ACAC	CACG	ACGG	CGGC	GGCC
GCCG	TCTT	CTTC	TTCC	TCCA
CCAA	CAAC	AACC	ACCT	CCTT
CTTG	TTGG	TGGA	ACTT	...

parts of J genes

ATAC	TACT	ACTT	CCAG	CAGC
AGCA	GCAC	TGGG	GGGC	GGCA
GCAA	CAAG	AAGA	AGAG	GAGT
AGTT	GTTG	TTGG	...	

ACACGGCCGTGTATTACTGTGCGAGAGAGCTGAATACTTCCAGCACTGGGGCC



Fast identification of a window centered on the CDR3

Clone clustering

parts of V genes

ACAC	CACG	ACGG	CGGC	GGCC
GCCG	TCTT	CTTC	TTCC	TCCA
CCAA	CAAC	AACC	ACCT	CCTT
CTTG	TTGG	TGGA	ACTT	...

parts of J genes

ATAC	TACT	ACTT	CCAG	CAGC
AGCA	GCAC	TGGG	GGGC	GGCA
GCAA	CAAG	AAGA	AGAG	GAGT
AGTT	GTTG	TTGG	...	

ACACGGCCGTGTATTACTGTGCGAGAGAGCTGAATACTTCCAGCACTGGGGCC

$O(n)$ alignment-free V(D)J detection algorithm

Fast identification of a window centered on the CDR3

Clone clustering

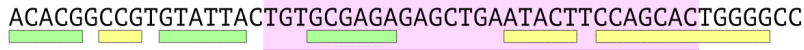
parts of V genes

ACAC	CACG	ACGG	CGGC	GGCC
GCCG	TCTT	CTTC	TTCC	TCCA
CCAA	CAAC	AACC	ACCT	CCTT
CTTG	TTGG	TGGA	ACTT	...

parts of J genes

ATAC	TACT	ACTT	CCAG	CAGC
AGCA	GCAC	TGGG	GGGC	GGCA
GCAA	CAAG	AAGA	AGAG	GAGT
AGTT	GTTG	TTGG	...	

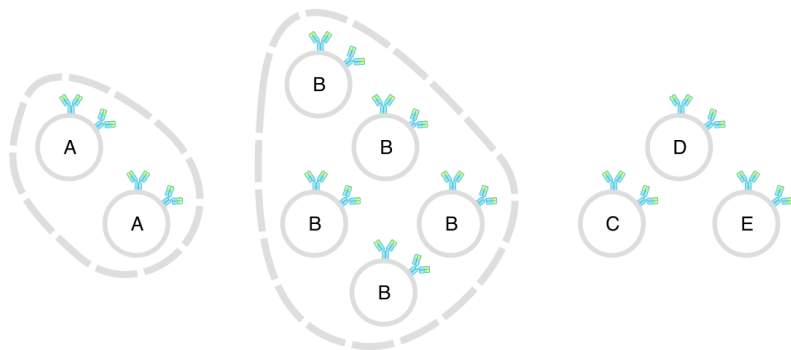
ACACGGCCGTGTATTACTGTGCGAGAGAGCTGAATACTTCCAGCACTGGGGCC



$O(n)$ alignment-free V(D)J detection algorithm

Immune Repertoire Sequencing (RepSeq)

Clone clustering



20%

50%

30%

1 000 000 VDJ = 100 s

Giraud, Salson et al., BMC Genomics, 2014

How the seeds are chosen?

Use sensitive seeds \rightarrow spaced seeds (e.g. ##-##)

How the seeds are chosen?

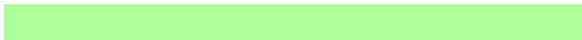
Use sensitive seeds \rightarrow spaced seeds (e.g. ##-##)

Minimize the window shift in case of error

How to prevent the window from being shifted too much?

With a seed $\underbrace{\# \cdots \#}_k$

Actual V



How to prevent the window from being shifted too much?

With a seed $\underbrace{\# \cdots \#}_k$

Actual V



How to prevent the window from being shifted too much?

With a seed $\underbrace{\# \cdots \#}_k$

Actual V



Predicted V



How to prevent the window from being shifted too much?

With a seed $\underbrace{\# \cdots \#}_{\frac{k-1}{2}} - \underbrace{\# \cdots \#}_{\frac{k-1}{2}}$

Actual V



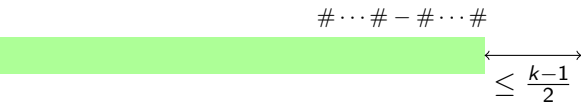
How to prevent the window from being shifted too much?

With a seed $\underbrace{\# \cdots \#}_{\frac{k-1}{2}} - \underbrace{\# \cdots \#}_{\frac{k-1}{2}}$

Actual V



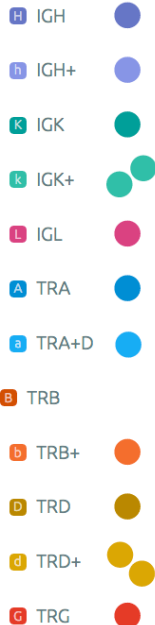
Predicted V



Vidjil-algo

analyses recombinations on all human TR/Ig locus

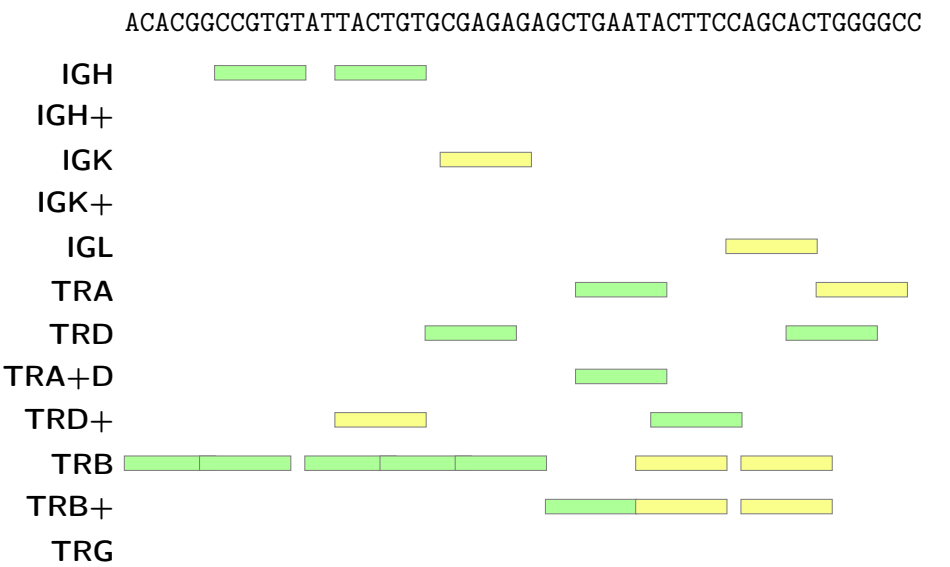
complete recombinations		incomplete/special recombinations	
TRA	Va-Ja		
TRB	Vb-(Db)-Jb	TRB+	Db-Jb
TRD	Vd-(Dd)-Jd	TRD+	Vd-Dd3, Dd2-(Dd)-Jd, Dd2-Dd3
		TRA+D	Vd-(Dd)-Ja, Dd-Ja
TRG	Vg-Jg		
IGH	Vh-(Dh)-Jh	IGH+	Dh-Jh
IGL	Vi-Ji		
IGK	Vk-Jk	IGK+	Vk-KDE, INTRON-KDE



One pass for each recombination system

ACACGGCCGTGTATTACTGTGCGAGAGAGCTGAATACTTCCAGCACTGGGGCC

One pass for each recombination system



How could we find
a V(D)J recombination (if any)
in a single pass?

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

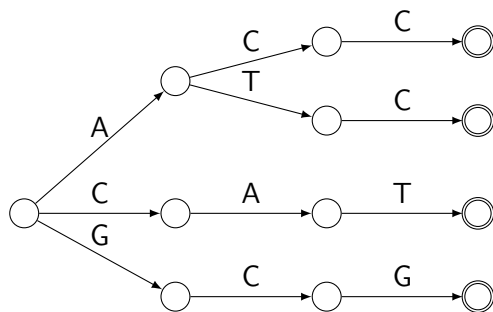
Searches a **set of patterns** P in a text T in time $O(|T|)$

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{\text{ACC}, \text{ATC}, \text{CAT}, \text{GCG}\}$

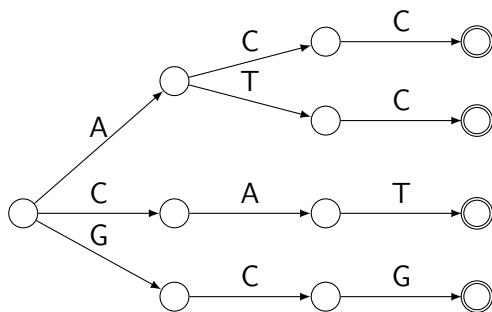


Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{\text{ACC}, \text{ATC}, \text{CAT}, \text{GCG}\}$



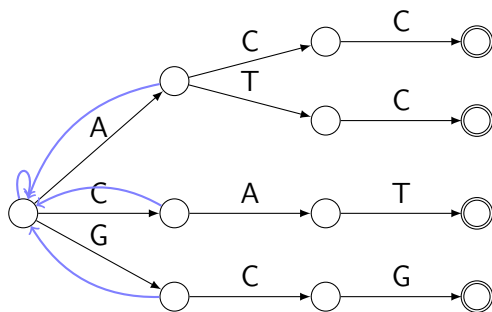
Failure function:
returns the longest
proper suffix accessible
from the initial state

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{ACC, ATC, CAT, GCG\}$



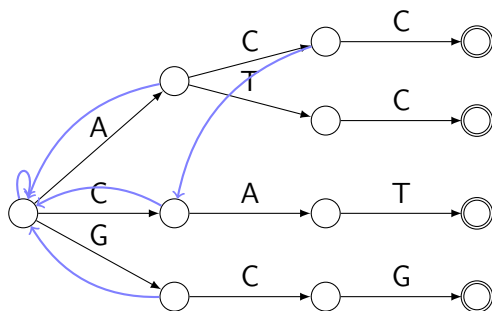
Failure function:
returns the longest
proper suffix accessible
from the initial state

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{ACC, ATC, CAT, GCG\}$



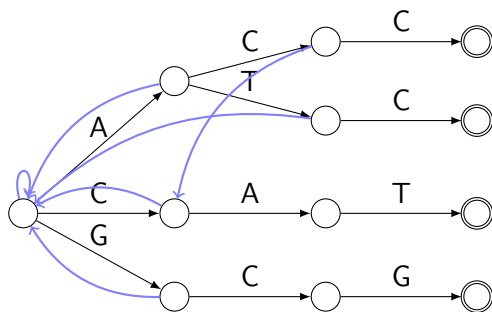
Failure function:
returns the longest
proper suffix accessible
from the initial state

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{ACC, ATC, CAT, GCG\}$



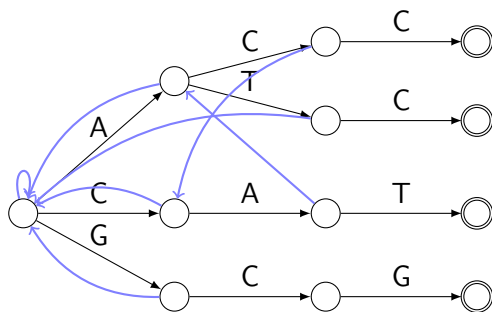
Failure function:
returns the longest
proper suffix accessible
from the initial state

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{\text{ACC}, \text{ATC}, \text{CAT}, \text{GCG}\}$



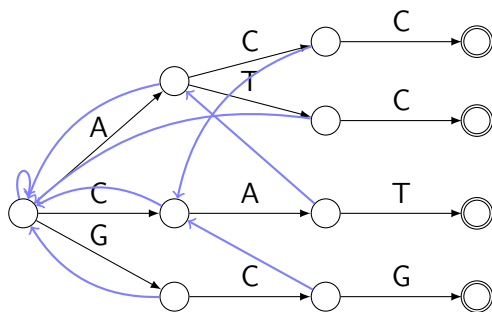
Failure function:
returns the longest
proper suffix accessible
from the initial state

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{\text{ACC}, \text{ATC}, \text{CAT}, \text{GCG}\}$



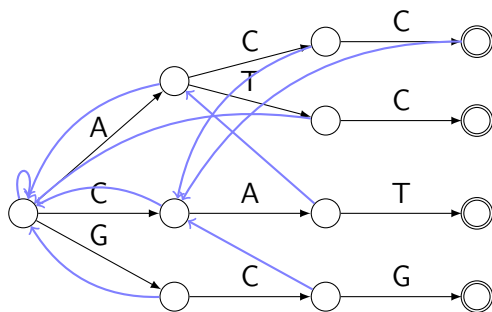
Failure function:
returns the longest
proper suffix accessible
from the initial state

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{ACC, ATC, CAT, GCG\}$



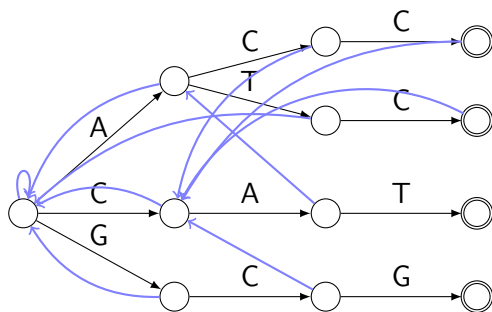
Failure function:
returns the longest
proper suffix accessible
from the initial state

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{ACC, ATC, CAT, GCG\}$



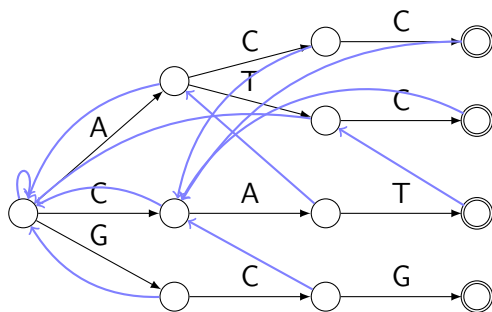
Failure function:
returns the longest
proper suffix accessible
from the initial state

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{ACC, ATC, CAT, GCG\}$



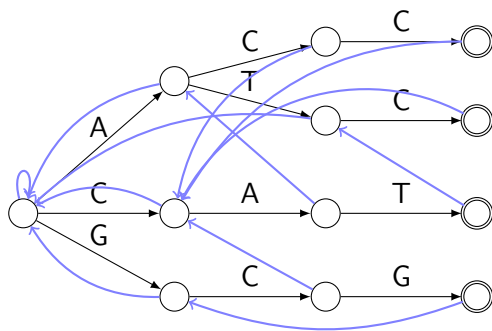
Failure function:
returns the longest
proper suffix accessible
from the initial state

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{\text{ACC}, \text{ATC}, \text{CAT}, \text{GCG}\}$



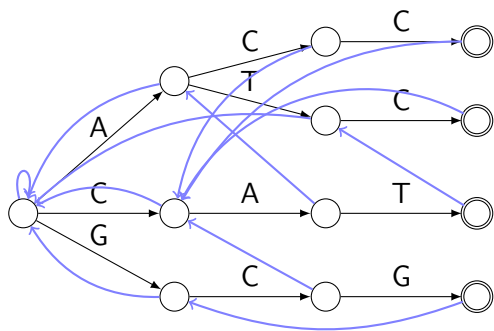
Failure function:
returns the longest
proper suffix accessible
from the initial state

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{ACC, ATC, CAT, GCG\}$



Failure function:
returns the longest
proper suffix accessible
from the initial state

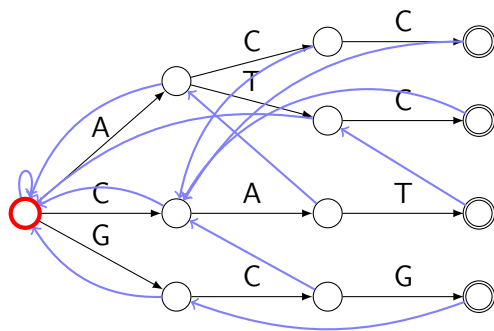
Searching P in $T = \text{ACATCG}$

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{ACC, ATC, CAT, GCG\}$



Failure function:
returns the longest
proper suffix accessible
from the initial state

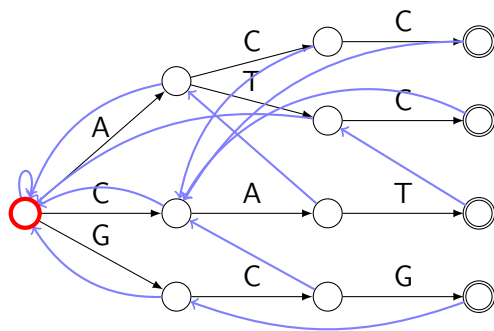
Searching P in $T = ACATCG$

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{ACC, ATC, CAT, GCG\}$



Failure function:
returns the longest
proper suffix accessible
from the initial state

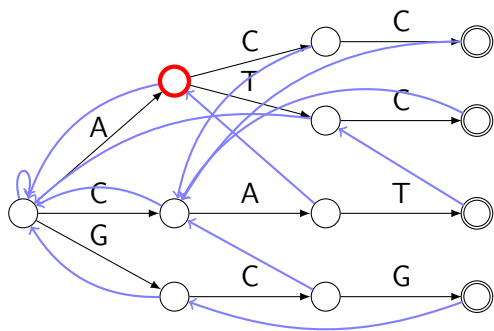
Searching P in $T = \text{ACATCG}$

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{ACC, ATC, CAT, GCG\}$



Failure function:
returns the longest
proper suffix accessible
from the initial state

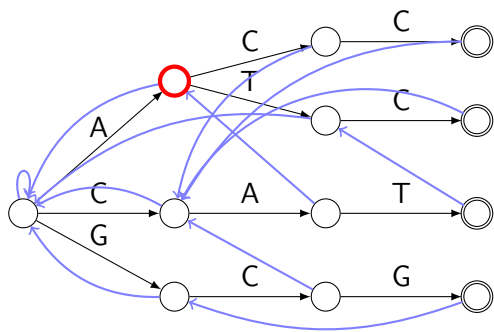
Searching P in $T = \text{ACATCG}$

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{ACC, ATC, CAT, GCG\}$



Failure function:
returns the longest
proper suffix accessible
from the initial state

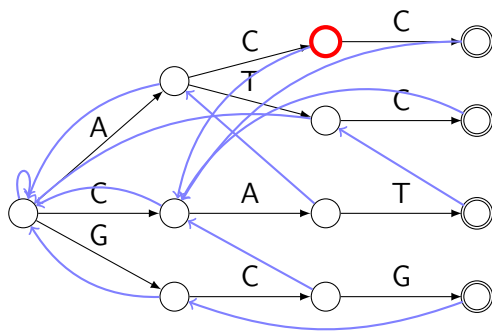
Searching P in $T = ACATCG$

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{\text{ACC}, \text{ATC}, \text{CAT}, \text{GCG}\}$



Failure function:
returns the longest
proper suffix accessible
from the initial state

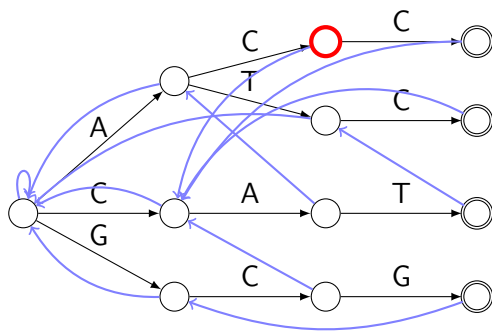
Searching P in $T = \text{ACATCG}$

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{ACC, ATC, CAT, GCG\}$



Failure function:
returns the longest
proper suffix accessible
from the initial state

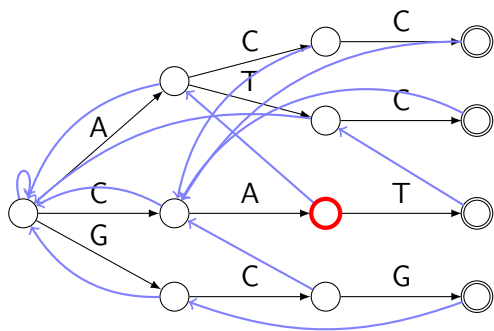
Searching P in $T = AC(AT)CG$

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{ACC, ATC, CAT, GCG\}$



Failure function:
returns the longest
proper suffix accessible
from the initial state

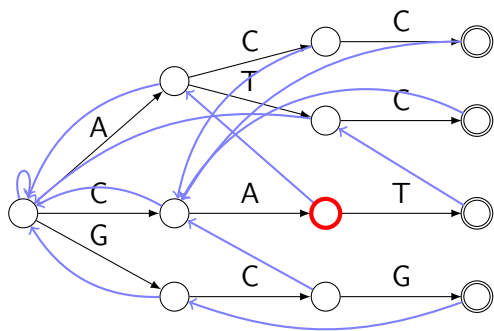
Searching P in $T = AC(AT)CG$

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{ACC, ATC, CAT, GCG\}$



Failure function:
returns the longest
proper suffix accessible
from the initial state

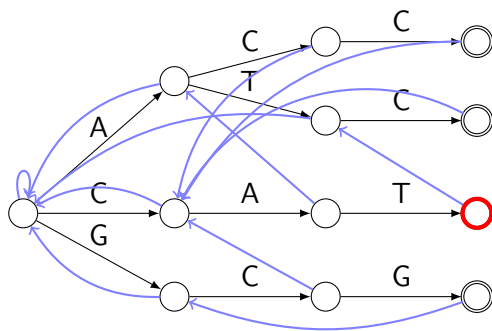
Searching P in $T = ACATCG$

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{\text{ACC}, \text{ATC}, \text{CAT}, \text{GCG}\}$



Failure function:
returns the longest
proper suffix accessible
from the initial state

Searching P in $T = \text{ACATCG}$

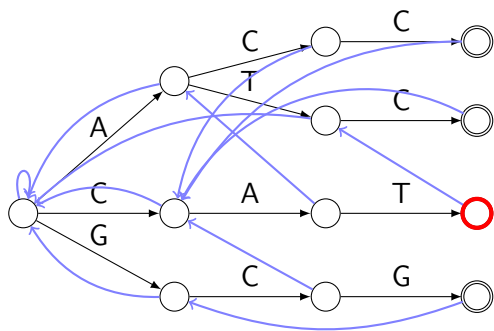
CAT found!

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{ACC, ATC, CAT, GCG\}$



Failure function:
returns the longest
proper suffix accessible
from the initial state

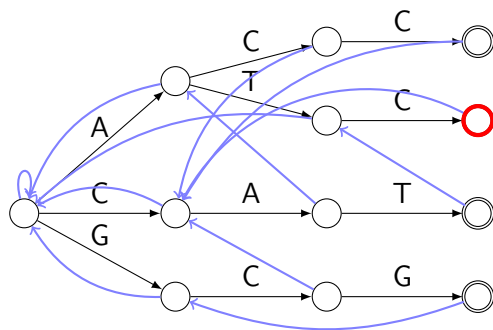
Searching P in $T = ACATCG$

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{ACC, ATC, CAT, GCG\}$



Failure function:
returns the longest
proper suffix accessible
from the initial state

Searching P in $T = ACATCG$

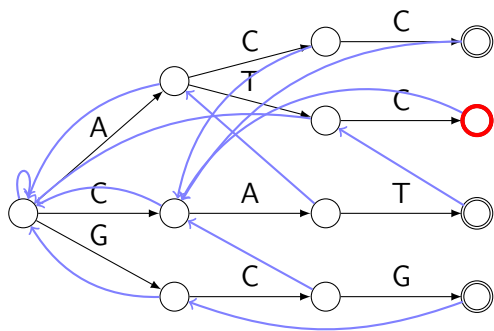
ATC found!

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{ACC, ATC, CAT, GCG\}$



Failure function:
returns the longest
proper suffix accessible
from the initial state

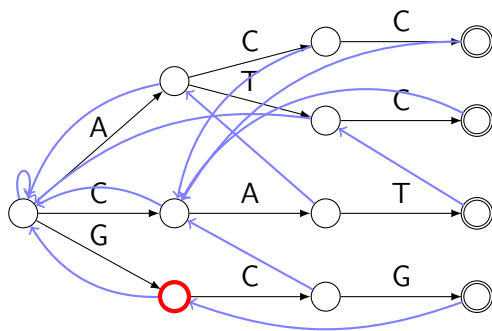
Searching P in $T = ACATCG$

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{ACC, ATC, CAT, GCG\}$



Failure function:
returns the longest
proper suffix accessible
from the initial state

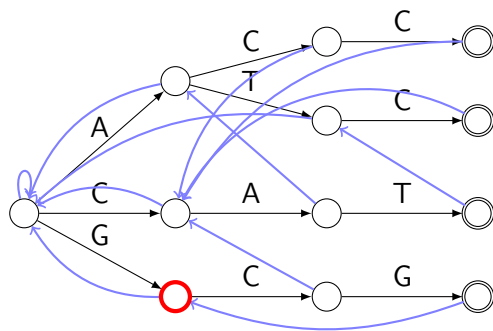
Searching P in $T = ACATCG$

Aho-Corasick automaton: searches patterns in linear time

Introduced by Alfred Aho and Margaret Corasick in 1975

Searches a **set of patterns** P in a text T in time $O(|T|)$

$P = \{ACC, ATC, CAT, GCG\}$



Failure function:
returns the longest
proper suffix accessible
from the initial state

Searching P in $T = ACATCG$

Aho-Corasick automaton for V(D)J detection

Aho-Corasick automaton for V(D)J detection

What are the patterns?

Aho-Corasick automaton for V(D)J detection

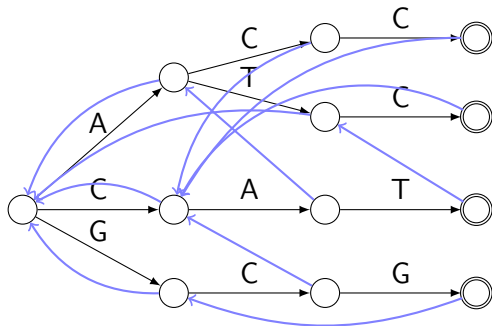
What are the patterns?

(spaced) k-mers from V and J genes

Aho-Corasick automaton for V(D)J detection

What are the patterns?

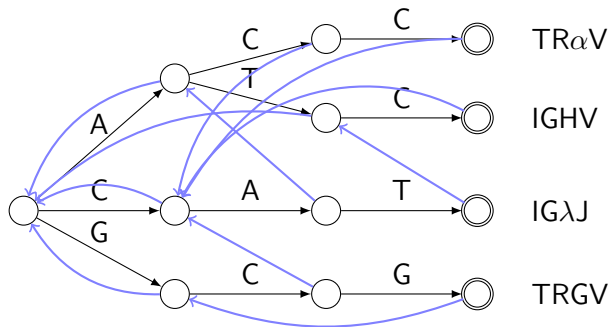
(spaced) k-mers from V and J genes



Aho-Corasick automaton for V(D)J detection

What are the patterns?

(spaced) k-mers from V and J genes



Why an Aho-Corasick automaton?

Querying spaced k-mers could be done with a hash table!

Why an Aho-Corasick automaton?

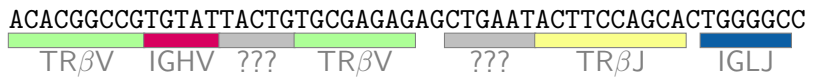
Querying spaced k-mers could be done with a hash table!

With Aho-Corasick automaton: seeds of various lengths and shapes

Analysing all recombinations in a single pass

ACACGGCCGTGTATTACTGTGCGAGAGAGCTGAATACTTCCAGCACTGGGGCC

Analysing all recombinations in a single pass



Analysing all recombinations in a single pass

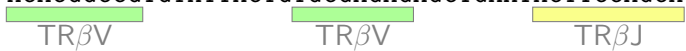


Keep the two most abundant annotations

Here TR β V and TR β J

Analysing all recombinations in a single pass

ACACGGCCGTGTATTACTGTGCGAGAGAGCTGAATACTTCCAGCACTGGGGCC



TR β V TR β V TR β J

Keep the two most abundant annotations

Here TR β V and TR β J

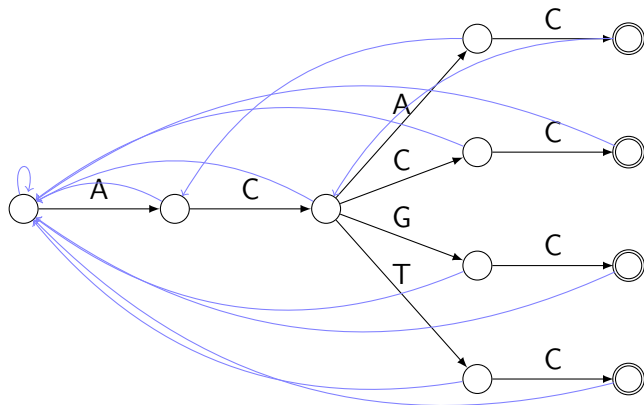
How to include spaced seeds in the AC automaton?

How to include spaced seeds in the AC automaton?

Not in a very smart way: add all possible paths

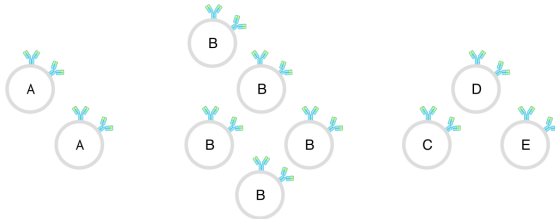
How to include spaced seeds in the AC automaton?

Not in a very smart way: add all possible paths
Indexing AC-C

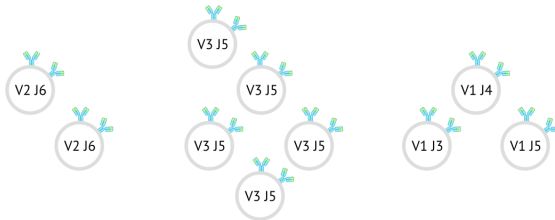


V(D)J detection or V(D)J assignment?

V(D)J detection



V(D)J assignment



Comparison with other software

MiXCR V(D)J-assign all reads (Bolotin *et al*, 2015)

IgReC V(D)J-assign all reads (Shlemov *et al*, 2016)

Vidjil-algo (old) V(D)J-detect all reads
and assign most abundant clusters

Vidjil-algo (new) V(D)J-detect all reads
and assign most abundant clusters

Comparison with other software

MiXCR V(D)J-assign all reads (Bolotin *et al*, 2015)

IgReC V(D)J-assign all reads (Shlemov *et al*, 2016)

Vidjil-algo (old) V(D)J-detect all reads
and assign most abundant clusters

Vidjil-algo (new) V(D)J-detect all reads
and assign most abundant clusters

MiXCR and IgReC do much more things than Vidjil-algo

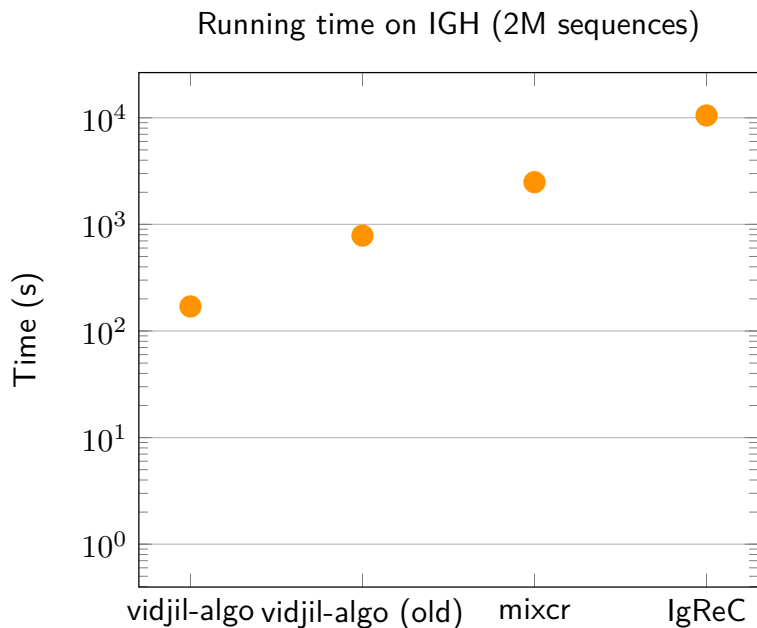
Thus the comparison is unfair
but that's the only one we can do

Benchmark datasets

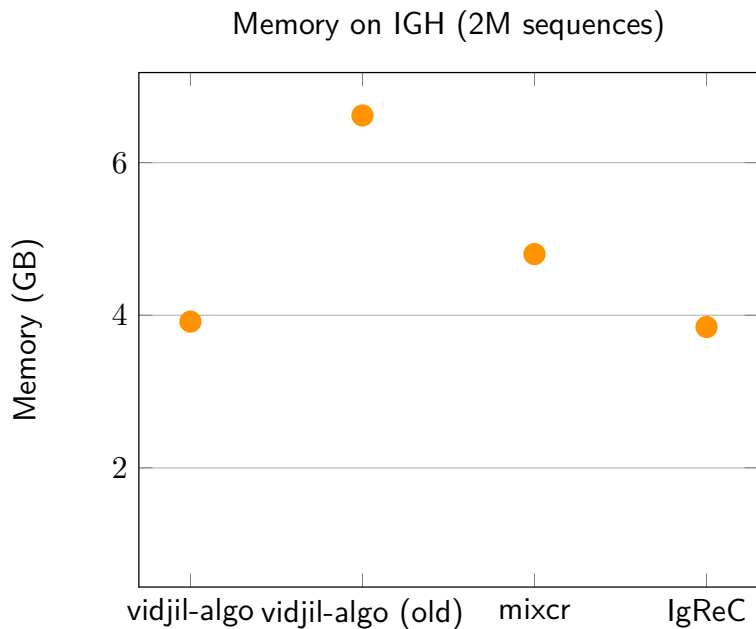
True dataset All V(D)J recombinations, with random indels at junctions and 2% differences

False dataset Random DNA sequences of length 350–450

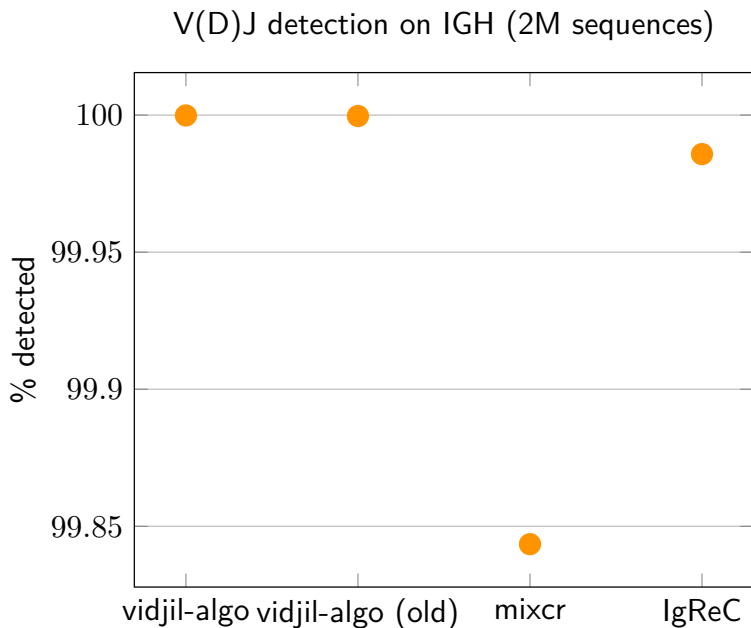
A precise and quicker heuristic



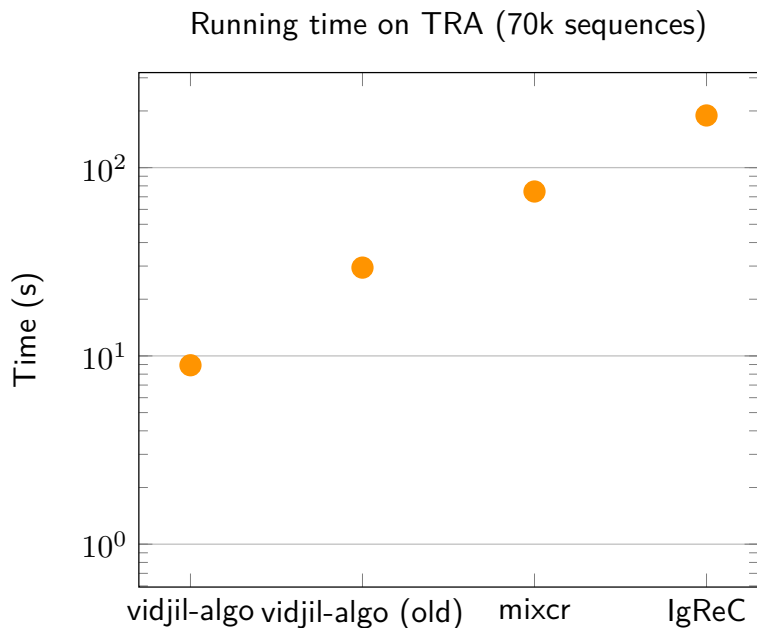
A precise and quicker heuristic



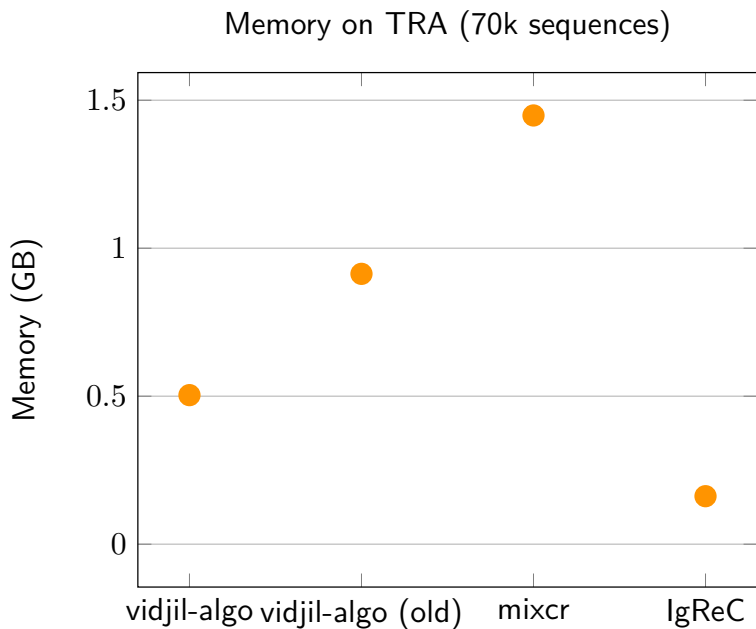
A precise and quicker heuristic



A precise and quicker heuristic

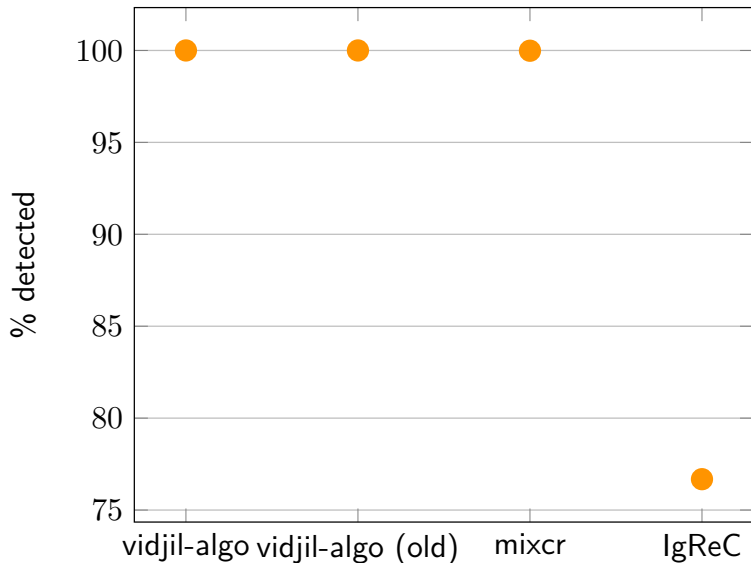


A precise and quicker heuristic



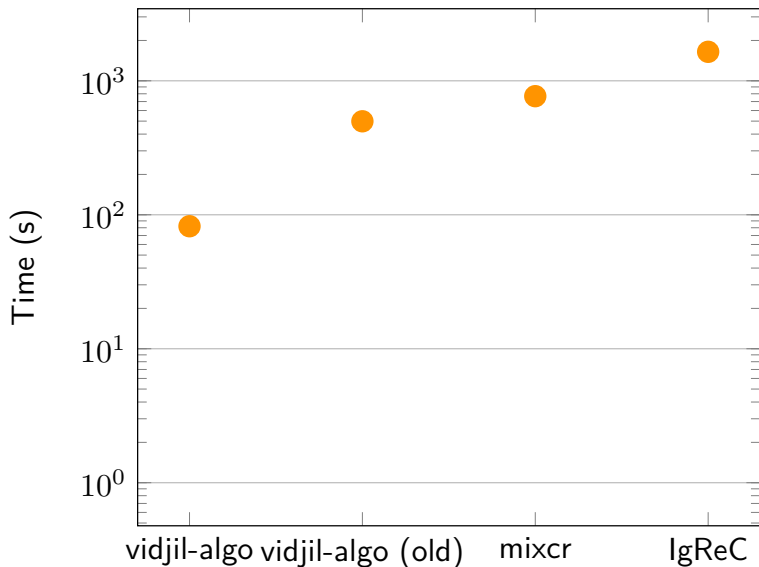
A precise and quicker heuristic

V(D)J detection on TRA (70k sequences)

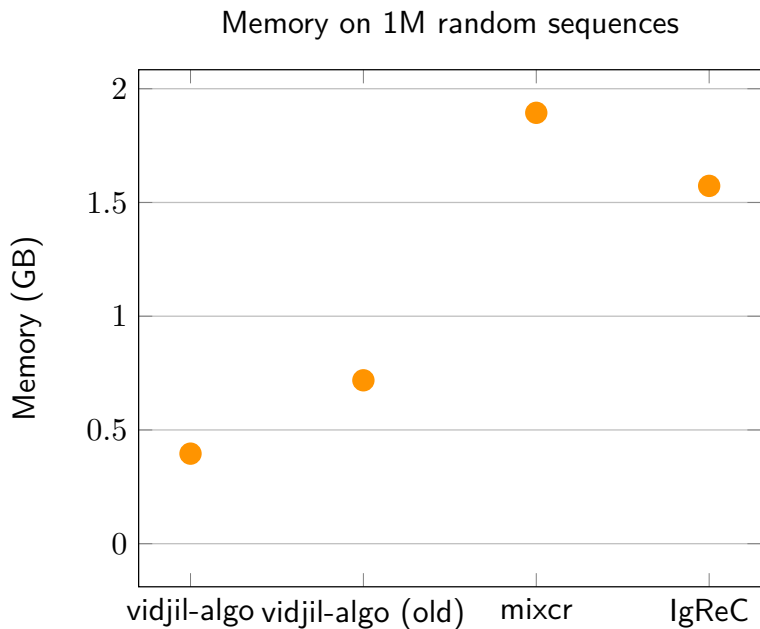


A precise and quicker heuristic

Running time on 1M random sequences

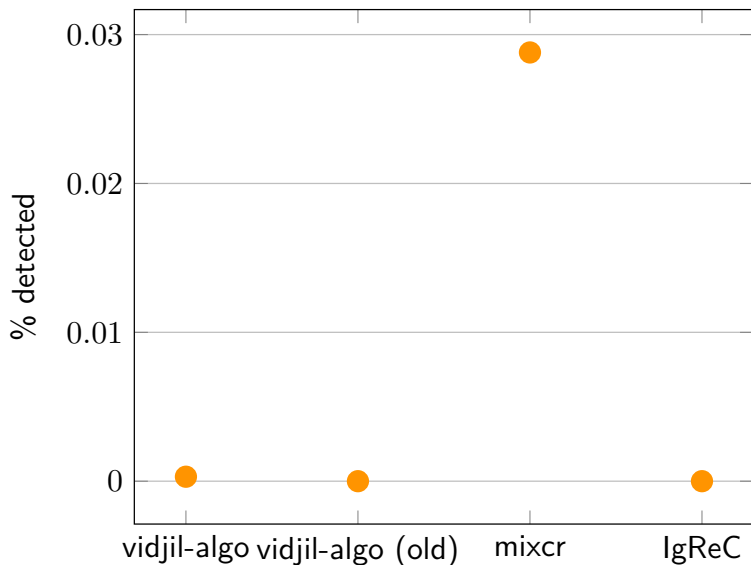


A precise and quicker heuristic



A precise and quicker heuristic

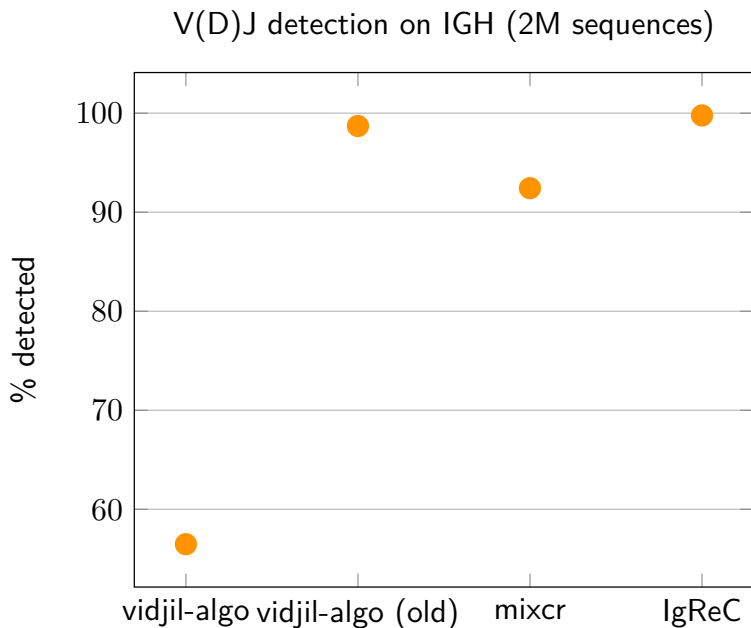
V(D)J detection on 1M random sequences



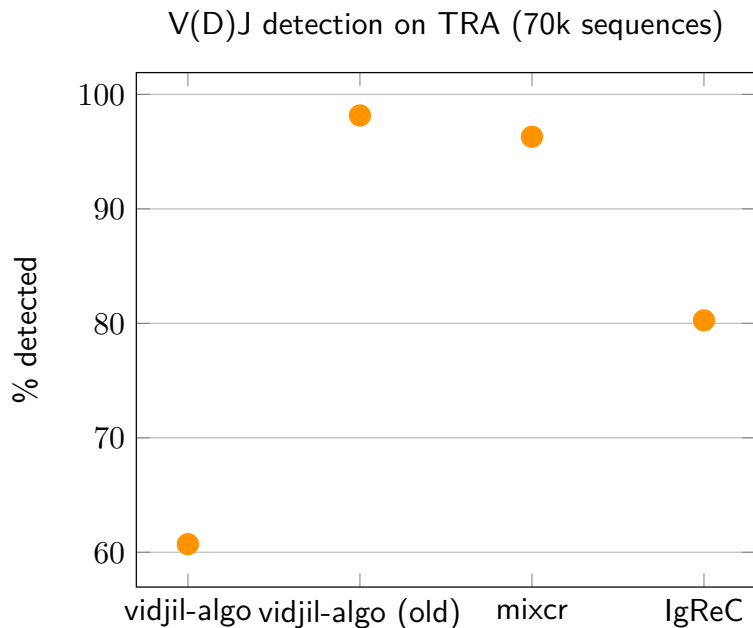
What if we have $V(D)J$ with 10 % errors?

Well...

What if we have V(D)J with 10 % errors?



What if we have V(D)J with 10 % errors?



Faster V(D)J assignment using clone clustering heuristic

parts of V genes

ACAC	CACG	ACGG	CGGC	GGCC
GCCG	TCTT	CTTC	TTCC	TCCA
CCAA	CAAC	AACC	ACCT	CCTT
CTTG	TTGG	TGGA	ACTT	...

parts of J genes

ATAC	TACT	ACTT	CCAG	CAGC
AGCA	GCAC	TGGG	GGGC	GGCA
GCAA	CAAG	AAGA	AGAG	GAGT
AGTT	GTTG	TTGG	...	

ACACGGCCGTGTATTACTGTGCGAGAGAGCTGAATACTTCCAGCACTGGGGCC

Faster V(D)J assignment using clone clustering heuristic

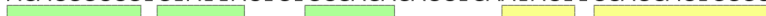
parts of V genes

ACAC	CACG	ACGG	CGGC	GGCC
GCCG	TCTT	CTTC	TTCC	TCCA
CCAA	CAAC	AACC	ACCT	CCTT
CTTG	TTGG	TGGA	ACTT	...

parts of J genes

ATAC	TACT	ACTT	CCAG	CAGC
AGCA	GCAC	TGGG	GGGC	GGCA
GCAA	CAAG	AAGA	AGAG	GAGT
AGTT	GTTG	TTGG	...	

ACACGGCCGTGTATTACTGTGCGAGAGAGCTGAATACTTCCAGCACTGGGGCC



Faster V(D)J assignment using clone clustering heuristic

parts of V genes

ACAC	CACG	ACGG	CGGC	GGCC
GCCG	TCTT	CTTC	TTCC	TCCA
CCAA	CAAC	AACC	ACCT	CCTT
CTTG	TTGG	TGGA	ACTT	...

parts of J genes

ATAC	TACT	ACTT	CCAG	CAGC
AGCA	GCAC	TGGG	GGGC	GGCA
GCAA	CAAG	AAGA	AGAG	GAGT
AGTT	GTTG	TTGG	...	

ACACGGCCGTGTATTACTGTGCGAGAGAGCTGAATACTTCCAGCACTGGGGCC



Faster V(D)J assignment using clone clustering heuristic

parts of V genes

ACAC V2, V4	CACG V2, V4	ACGG V1	CGGC V1, V2, V4	GGCC V2, V4
GCCG V1, V5	TCTT V4	CTTC V6, V7	TTCC V6	TCCA V6, V8
CCAA V6, V8, V9	CAAC V7	AACC V8, V9	ACCT V1, V9	CCTT V4, V5
CTTG V5, V6, V7	TTGG V7	TGGA V7	ACTT V7	...

parts of J genes

ATAC J1, J2	TACT J1, J2, J3	ACTT J4	CCAG J4	CAGC J4
AGCA J1, J2	GCAC J1, J2	TGGG J1, J3	GGGC J1, J3	GGCA J2, J3
GCAA J2, J3	CAAG J2, J3	AAGA J3	AGAG J3	GAGT J3
AGTT J3, J4	GTTG J3, J4	TTGG J3, J4	...	

ACACGGCCGTGTATTACTGTGCGAGAGAGCTGAATACTTCCAGCACTGGGGCC

Faster V(D)J assignment using clone clustering heuristic

parts of V genes

ACAC V2, V4	CACG V2, V4	ACGG V1	CGGC V1, V2, V4	GGCC V2, V4
GCCG V1, V5	TCTT V4	CTTC V6, V7	TTCC V6	TCCA V6, V8
CCAA V6, V8, V9	CAAC V7	AACC V8, V9	ACCT V1, V9	CCTT V4, V5
CTTG V5, V6, V7	TTGG V7	TGGA V7	ACTT V7	...

parts of J genes

ATAC J1, J2	TACT J1, J2, J3	ACTT J4	CCAG J4	CAGC J4
AGCA J1, J2	GCAC J1, J2	TGGG J1, J3	GGGC J1, J3	GGCA J2, J3
GCAA J2, J3	CAAG J2, J3	AAGA J3	AGAG J3	GAGT J3
AGTT J3, J4	GTTG J3, J4	TTGG J3, J4	...	

ACACGGCCGTGTATTACTGTGCGAGAGAGCTGAATACTTCCAGCACTGGGGCC

V1 8
V2 6
V4 3
V5 1

Faster V(D)J assignment using clone clustering heuristic

parts of V genes

ACAC V2, V4	CACG V2, V4	ACGG V1	CGGC V1, V2, V4	GGCC V2, V4
GCCG V1, V5	TCTT V4	CTTC V6, V7	TTCC V6	TCCA V6, V8
CCAA V6, V8, V9	CAAC V7	AACC V8, V9	ACCT V1, V9	CCTT V4, V5
CTTG V5, V6, V7	TTGG V7	TGGA V7	ACTT V7	...

parts of J genes

ATAC J1, J2	TACT J1, J2, J3	ACTT J4	CCAG J4	CAGC J4
AGCA J1, J2	GCAC J1, J2	TGGG J1, J3	GGGC J1, J3	GGCA J2, J3
GCAA J2, J3	CAAG J2, J3	AAGA J3	AGAG J3	GAGT J3
AGTT J3, J4	GTTG J3, J4	TTGG J3, J4	...	

ACACGGCCGTGTATTACTGTGCGAGAGAGCTGAATACTTCCAGCACTGGGGCC

V1 8
V2 6
V4 3
V5 1

Only compare with (most?) detected genes

How does that improve vidjil-algo?

Assigning V(D)J of 10,000 IGH sequences

How does that improve vidjil-algo?

Assigning V(D)J of 10,000 IGH sequences

> 100 **min**
Before

How does that improve vidjil-algo?

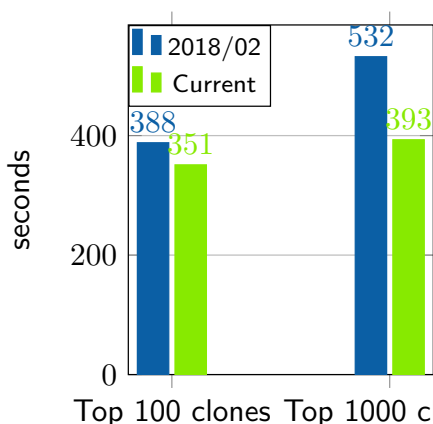
Assigning V(D)J of 10,000 IGH sequences

> 100 **min**
Before

< 5 **min**
With our optimisation

How does that improve vidjil-algo?

Clustering clones from 2.4M reads and assigning V(D)J to the top n clones



Assigning 10 times more sequences in as many time as before

Results quality check: p -value estimation and usual tests

Vidjil-algo – detecting and identifying V(D)J

A linear-time alignment-free V(D)J detection

Much quicker, about as precise as before

In the future:

Consider several results per state

Optimize spaced seeds for each recombination system

Don't just choose the two most abundant gene types

Integrate to the Vidjil platform

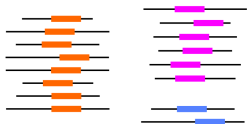
Vidjil

High-throughput Repertoire Sequencing (RepSeq) analysis

Web Application

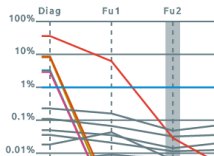
Patient database
Server

Vidjil-algo

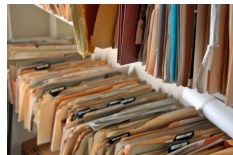


C++

Client



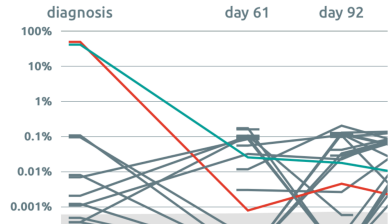
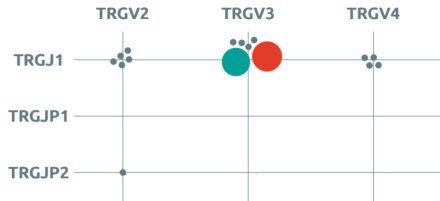
Javascript, d3.js



Python, web2py,
AJAX

- ▶ code on <http://git.vidjil.org/>
- ▶ open-source (GPL v3), public issue tracker (Gitlab)
- ▶ continuous integration, > 2,000 unit and functional tests

Duez et al., PLOS One, 2016

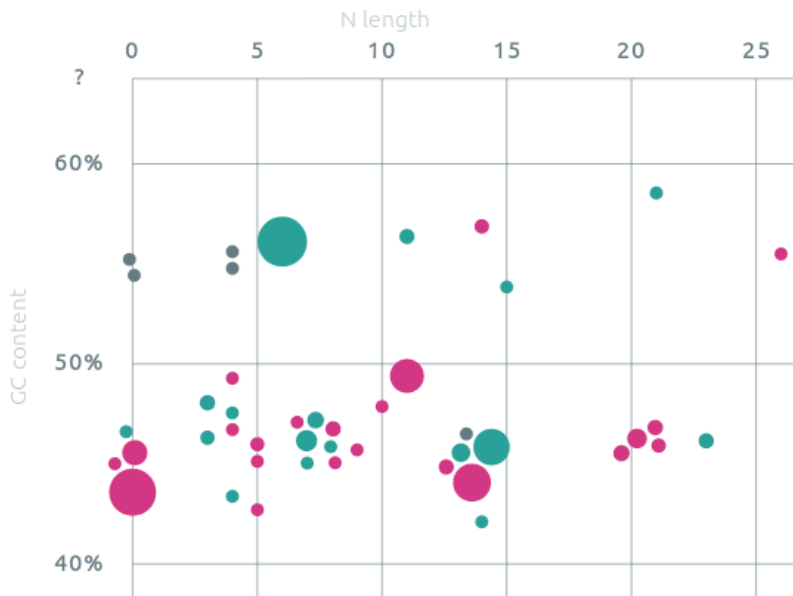


merge align > to IMGT/V-QUEST > to IgBlast > to Blast

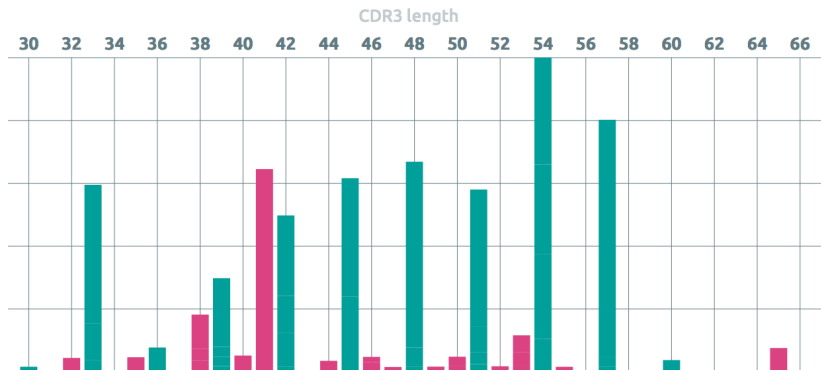
4 clones, 738 494 reads (90.53%)

✗ TRGV3 1/11/2 J1	49.30%	★ i	GCCACCTGGGACAGCTCCC-TT-GTTC--ATTATAAGAAACTCTTTGGCAGTG
✗ TRGV3 4/1/2 J1	41.23%	★ i	GCCACCTGGG--A--T--A--T--T--ATTATAAGAAACTCTTTGGCAGTG
✗ TRGV3 3/16/3 J1	0.0021%	★ i	GCCG-CTTGGGA-ACC CAATTTGGTACGGGTTATAAGAAACTCTTTGGCAGTG
✗ TRGV3 5/4/2 J1	+	★ i	GCCACCTGGG---GC--CA-A-T--T--A-TA--AGAAACTCTTTGGCAGTG

Plot clones – Grid view

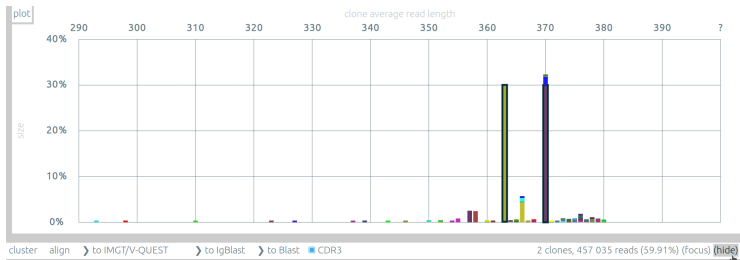


Plot clones – Bar view



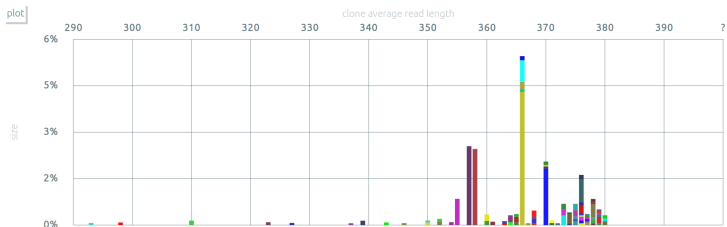
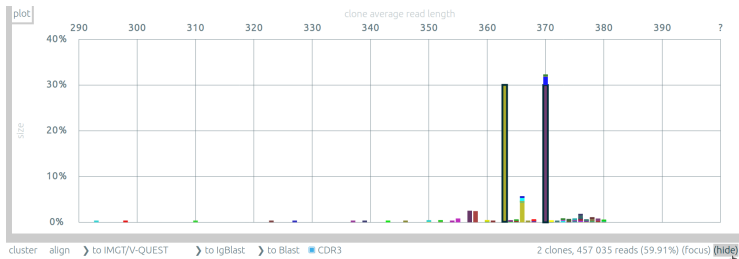
Browsing and filtering clones

Hide dominant clones to study other clones



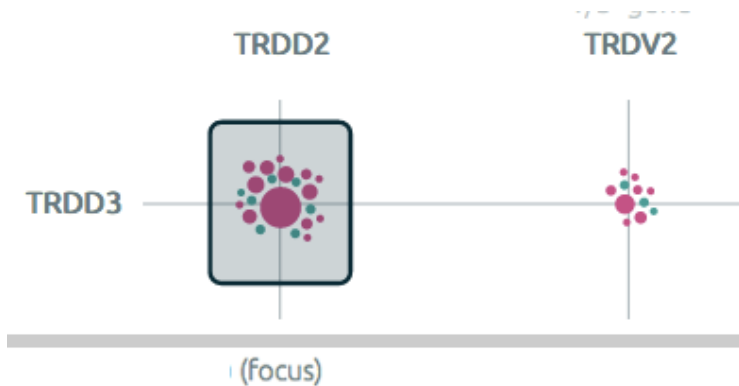
Browsing and filtering clones

Hide dominant clones to study other clones



Browsing and filtering clones

Focus on any subset of clones



Clustering options and undo

cluster align > to IMGT/V-QUEST ▼ > to IgBlast > to Blast

×	IGHV1-24 0/CGACCC/3 D6-13 1/TAAC/1	1.409%	★	i
×	IGHV1-24 0//3 D6-6 0/GAGG/3 J6*03	1.208%	★	i
×	IGHV1-24 0//3 D3-10 6/11/7 J6*02	1.170%	★	i
×	IGHV1-24 0/GCCCCG/0 D3-3 6/CGGA/6	1.069%	★	i

import/export cluster color by tag ▼ filter

IGH

IGH

analyze

select

revert to previous clusters

cluster selected clones

cluster by V/5'

cluster by J/3'

cluster by locus

cluster by similarity

break selected clusters

break all clusters

Further inspect the sequences with other software

- ▶ Blast, igBlast
- ▶ IMGT/V-QUEST
- ▶ V mutation status from IMGT/V-QUEST

ster align > to IMGT/V-QUEST ▼ > to IgBlast > to Blast ☒ CDR3-IMG ☒ V/D/J-IMG

IGHV4-39 3/7/0 D6-13 1/9/1 J5*02 81.53% 99.64% + ★ i .CCATATCCGTAGAC



Patient database and server

Autonomous RepSeq analysis pipeline in a clinical/research lab



Patient database and server

Autonomous RepSeq analysis pipeline in a clinical/research lab



Upload

upload list

L1413893.fasta



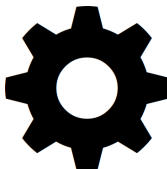


Patient database and server

Autonomous RepSeq analysis pipeline in a clinical/research lab



Upload



Process

upload list
L1413893.fasta

last processing	status
2015-02-09	RUNNING

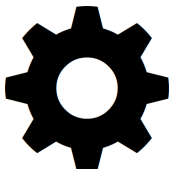


Patient database and server

Autonomous RepSeq analysis pipeline in a clinical/research lab



Upload



Process



Analyze

upload list
L1413893.fasta

last processing	status
2015-02-09	RUNNING

see the result:

[multi](#)

Automatically testing the software as much as possible

Vidjil-algo \simeq 2,000 tests (including “curated sequences”)

Web app \simeq 1,000 tests on internal behaviour,
 \simeq 100 tests on practical behaviour (“when I click
here, I should see the clone sequence”)

numericalAxis: use nice_number_digits in FloatAxis

Closes [#2731](#).

Edited 37 minutes ago by Mathieu Giraud

Request to merge [feature-c/2731-nice_number_dig...](#)  into dev



Pipeline [#11003](#)      passed for [658ff73a](#).



Deployed to [review/feature-c/2731-nice_number_digits](#) on

[feature-c-2731-nice-number-digits.ci.vidjil.org/?data=analysis-example.vidjil](#) 32 minutes ago

Stop environment

Developing end-to-end testing for more realistic tests

Every step in the Vidjil platform is tested. . .

Developing end-to-end testing for more realistic tests

Every step in the Vidjil platform is tested. . . but independently

Developing end-to-end testing for more realistic tests

Every step in the Vidjil platform is tested. . . but independently

Need to develop end-to-end testing

Developing end-to-end testing for more realistic tests

Every step in the Vidjil platform is tested... but independently

Need to develop end-to-end testing

1. Create patient
2. Upload file
3. Launch process
4. View results and analyze them
5. Compare results
6. ...

30,000

samples, since 2016

6,000

ALL/CLL patients at diagnosis in 7 hospitals, since 2016

> 50 regular users in > 30 hospital or research labs

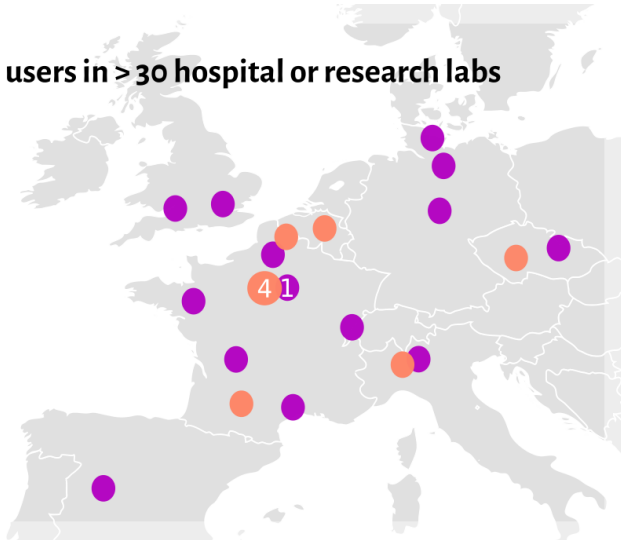
Canada ●

US ● ●

Brasil ●

Japan ● ●

South Korea ●



Developing and maintaining a web application
Offering support to users

This is not the job of a research group

VidjilNet consortium in 2018/19

Inria Foundation → Inria

January 2018

- ▶ VidjilNet started at Inria Foundation

Summer 2018

- ▶ new Inria direction
- ▶ reorganization of Inria and its foundation

December 2018

- ▶ VidjilNet reintegrated to Inria

VidjilNet: Why?

Any member of the not-for-profit VidjilNet consortium

- ▶ **participate to a community of members**, showing interest in the Vidjil platform and potentially other tools, collectively deciding development priorities
- ▶ **benefit from contracted services** for clinical or research work, including remote maintenance of in-hospital servers or data hosting through accredited partner

no biological or sequencing service, no bioinformatics services outside hematological/immunological studies

Vidjil – a platform for the analysis of V(D)J recombinations

Vidjil-algo

A quick and sensitive algorithm

Web platform

A user-friendly interface

VidjilNet

Used in routine hospital practice

Open source

vidjil.org