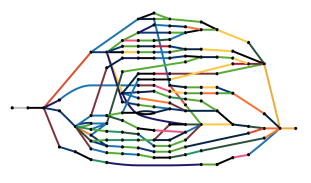


Self-Nonself Discrimination

The Negative Selection Algorithm

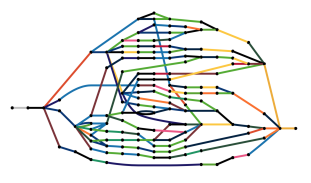
Lecture
Artificial Immune Systems for Anomaly Detection

Course Natural Computing, March 3rd, 2020

Johannes Textor
Institute for Computing and Information Sciences

Learning Objectives

1. Understand how the immune system approaches anomaly detection.
2. Know what negative selection algorithms are.
3. Assignment: understand how one would implement a negative selection algorithm in practice.



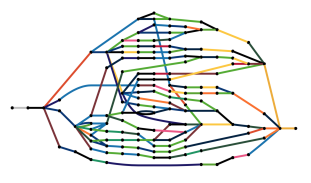
Self-Nonself Discrimination

The Negative Selection Algorithm

Lecture Outline

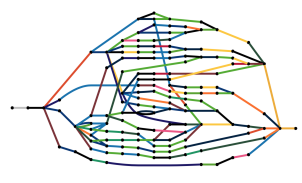
1. Self-Nonself Discrimination

2. The Negative Selection Algorithm



Self-Nonself Discrimination

The Negative Selection Algorithm



Self-Nonself Discrimination

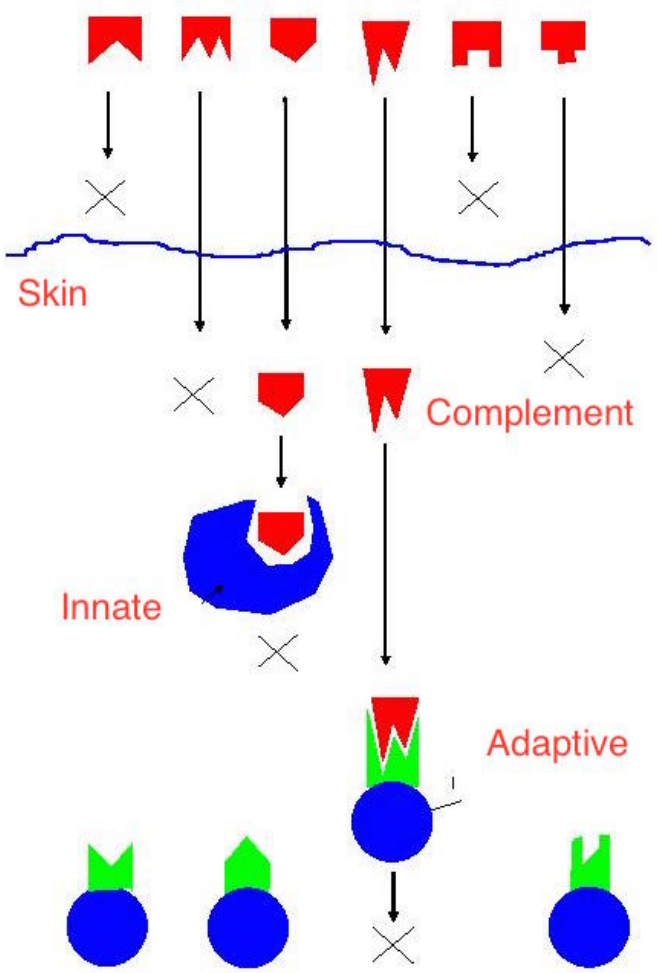
The Negative Selection Algorithm

The Immune System

The immune system (IS) consists of various different subsystems:

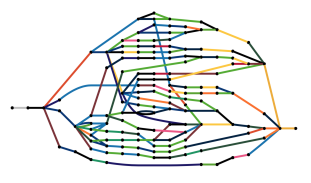
1. The skin.
2. The complement system, soluble proteins that “make life difficult” for pathogens.
3. The innate IS, cells that respond to pathogens in a generic way, and do not form memory.
4. The adaptive IS, cells that are diverse and specialized, and do form memory (immunity).

These components act together to provide protection against pathogens.



(Illustration by Steven Hofmeyr)

The Adaptive Immune System



Self-Nonself Discrimination

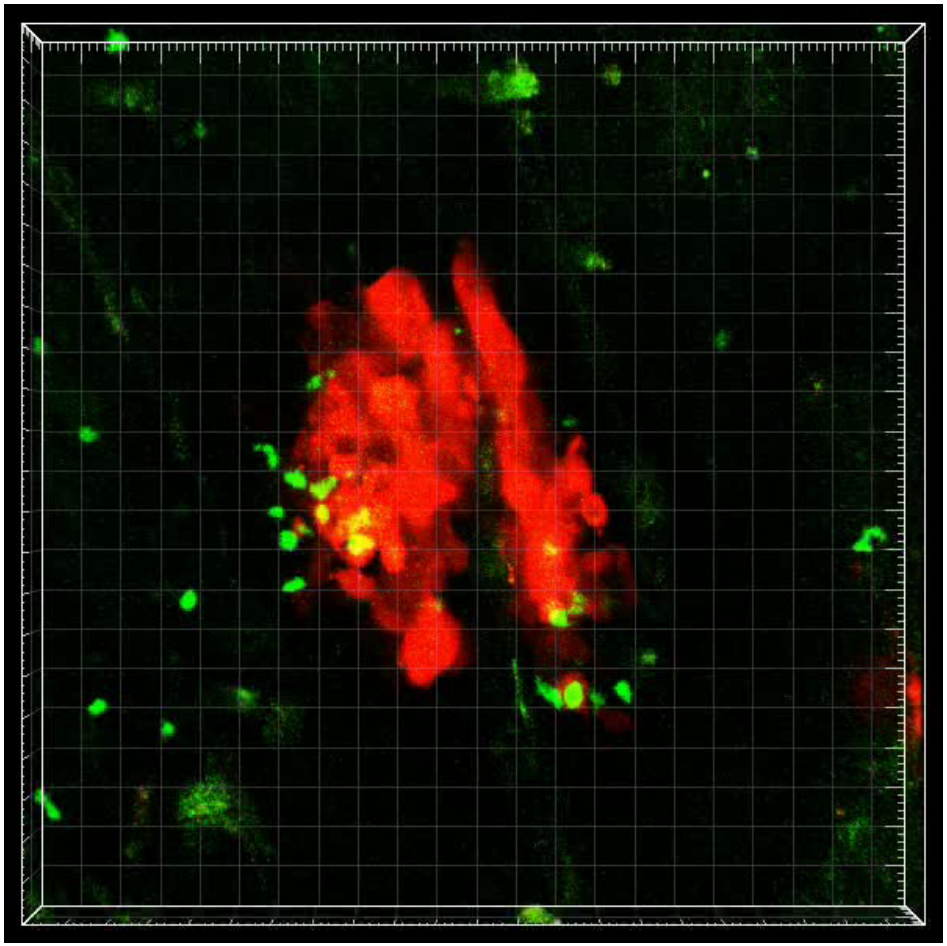
The Negative Selection Algorithm

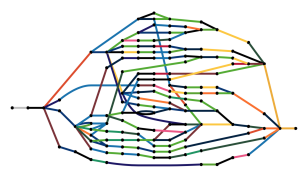
The adaptive immune system consists of two major subsystems: the B cell system and the T cell system.

- B cells are cells that are born in the bone marrow. They attack pathogens by secreting soluble molecules called antibodies.
- T cells are also born in the bone marrow, but they are "educated" in a special organ called the thymus. Some T cells can attack and kill virus-infected cells.

T Cells in Action

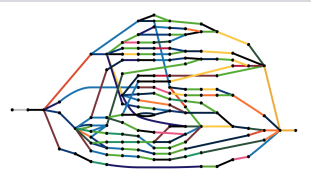
T cells (green) and virus-infected skin cells (red).
(Ariotti et al., PNAS 2012, DOI: 10.1073/pnas.1208927109)





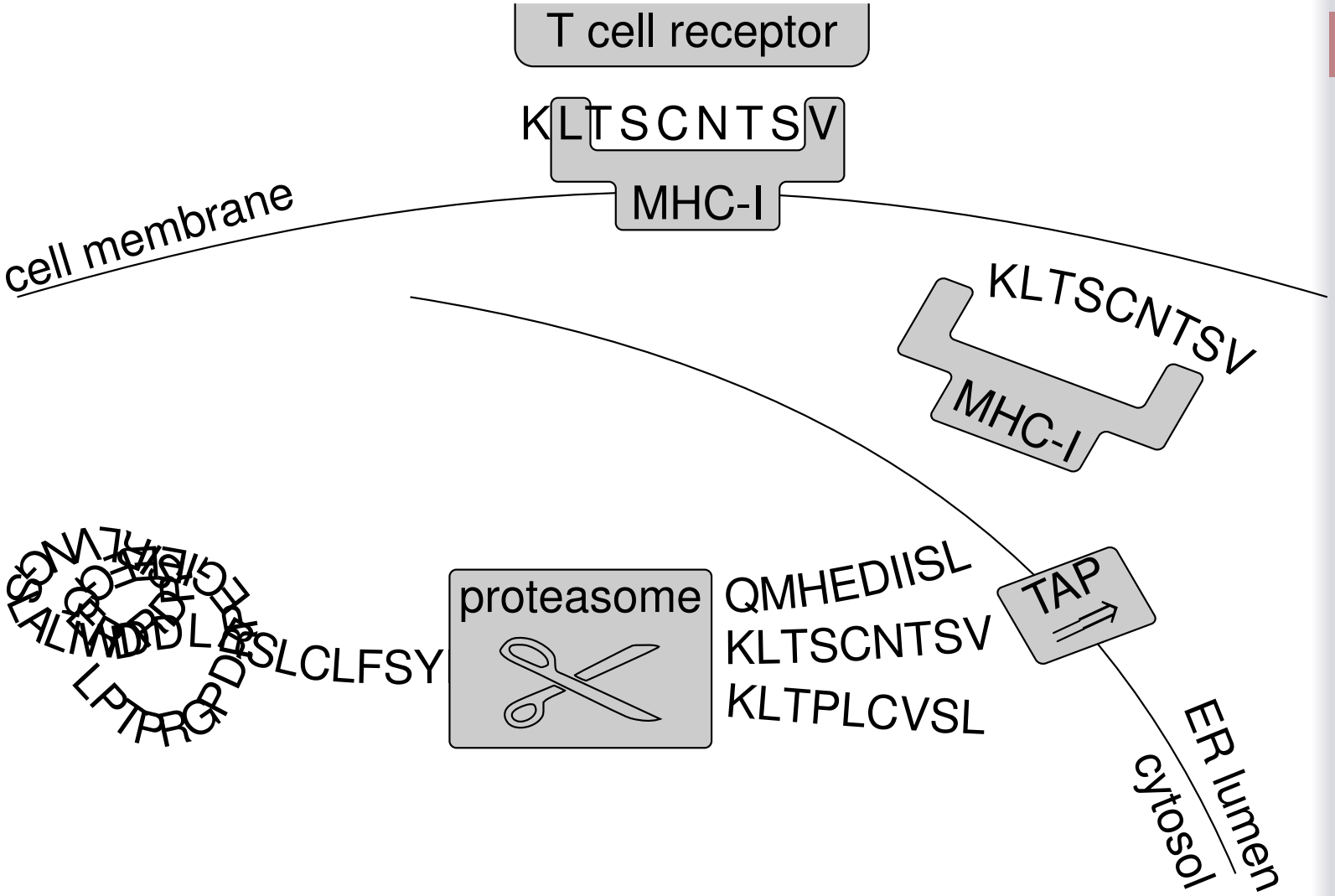
Self-Nonself Discrimination

The Negative Selection Algorithm

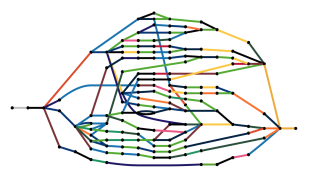


What T Cells Do

All human cells constantly produce and degrade proteins. Small pieces of degraded proteins (peptides) are “presented” on membrane molecules (MHC). T cells examine MHC-presented peptides.



Self-Nonself-Discrimination



Self-Nonself Discrimination

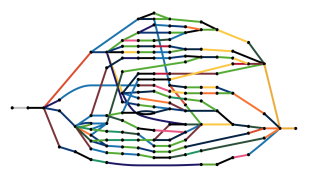
The Negative Selection Algorithm

Healthy cells present only “normal” (self) peptides, coming from the regular host proteins. Cells than can present “anomalous” (nonself) peptides include:

- Virus-infected cells.
- Cancer cells.
- Dying cells.

The task of (CD8) T cells is to kill cells that present anomalous peptides, while they must tolerate cells that present only normal peptides. How is that achieved?

Specificity and Diversity



Self-Nonself Discrimination

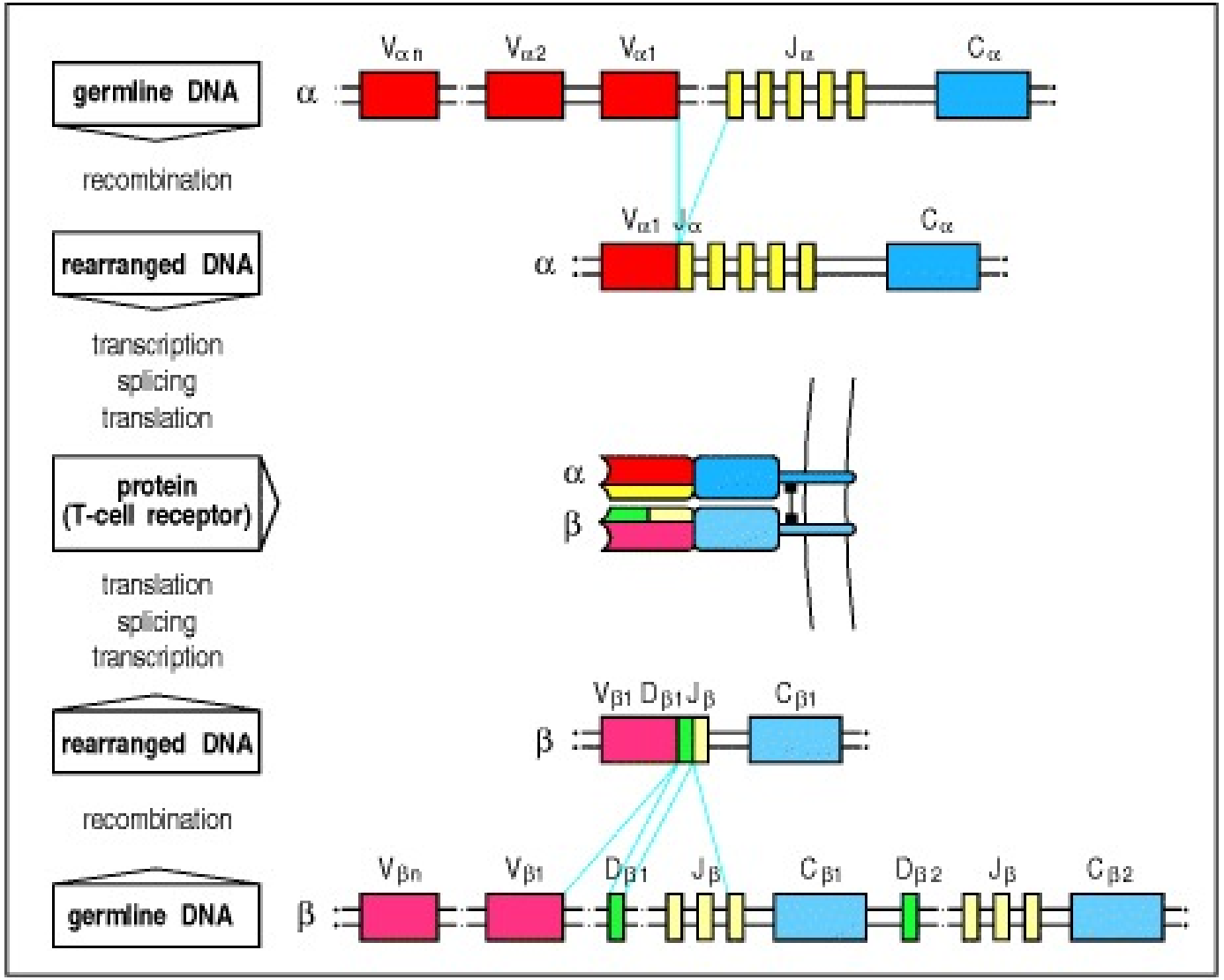
The Negative Selection Algorithm

A T Cell's receptor determines whether or not it binds to a given peptide.

Each T cell is extremely specific and binds to a very small set of peptides (probability for random peptide: $\sim 1/100,000$).

T cells are extremely diverse. Each T cell carries a T cell receptor that is essentially unique.

How T Cell Receptors Are Made

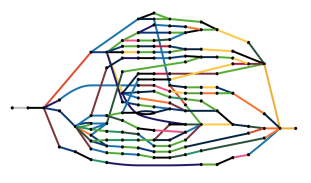


Immunobiology: The Immune System in Health and Disease. 5th edition. Janeway CA Jr, Travers P, Walport M, et al. New York: Garland Science; 2001.

How T Cell Receptors Are Made

Element	Immunoglobulin		α:β receptors	
	H	κ+λ	β	α
Variable segments (V)	65	70	52	~70
Diversity segments (D)	27	0	2	0
D segments read in 3 frames	rarely	–	often	–
Joining segments (J)	6	5(κ) 4(λ)	13	61
Joints with N- and P-nucleotides	2	50% of joints	2	1
Number of V gene pairs	3.4×10^6		5.8×10^6	
Junctional diversity	$\sim 3 \times 10^7$		$\sim 2 \times 10^{11}$	
Total diversity	$\sim 10^{14}$		$\sim 10^{18}$	

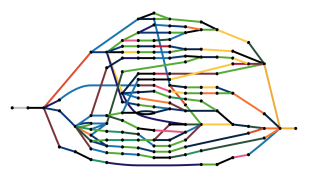
Immunobiology: The Immune System in Health and Disease. 5th edition. Janeway CA Jr, Travers P, Walport M, et al. New York: Garland Science; 2001.



Self-Nonself Discrimination

The Negative Selection Algorithm

Negative Selection



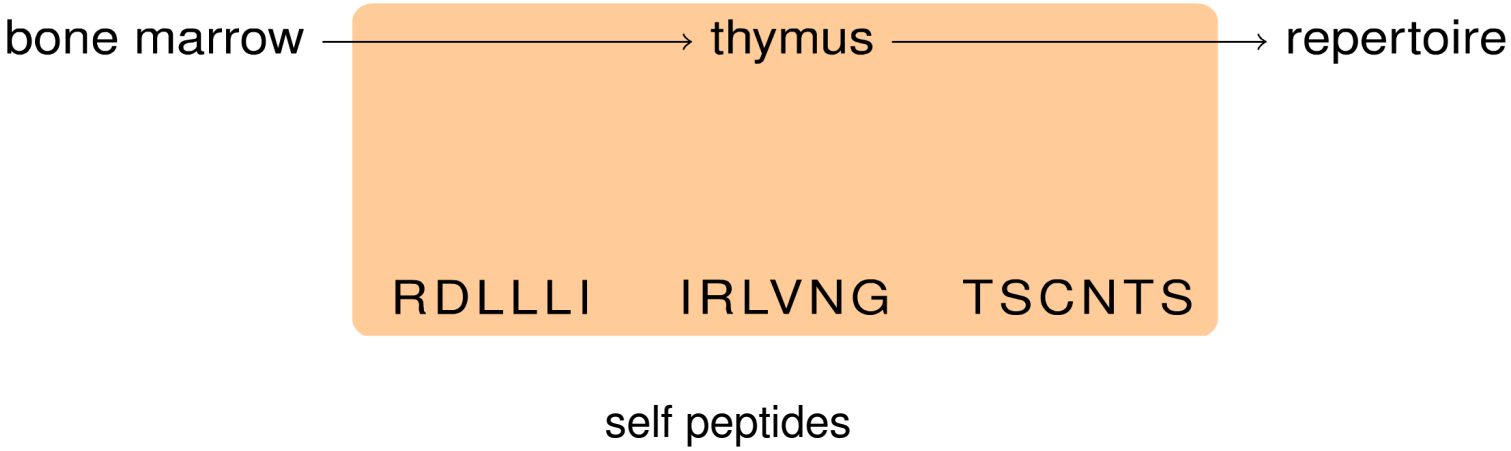
Self-Nonself Discrimination

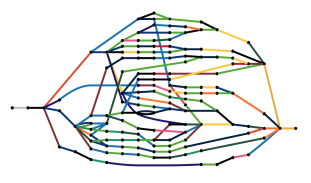
The Negative Selection Algorithm

Since each T cell receptor is randomly made, it might react to anything, including “normal” (self) peptides. Such reactions can be dangerous.

Negative selection is a process that identifies, and deletes, cells that react to self.

T cells that survive negative selection react to anything except self. The set of surviving T cells is called the T cell repertoire.



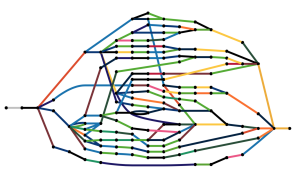


Why Do Computer Scientists Care?

The immune system has a number of attractive features that computer scientists would like to replicate.

- Multi-layered protection.
- Distributed detection.
- Biodiversity.
- Detection of previously unseen foreign material.

Stephanie Forrest, Steven A. Hofmeyr, and Anil Somayaji.
Computer Immunology.
Communications of the ACM, 40(10):88–96, 1997.



Computer Immunology

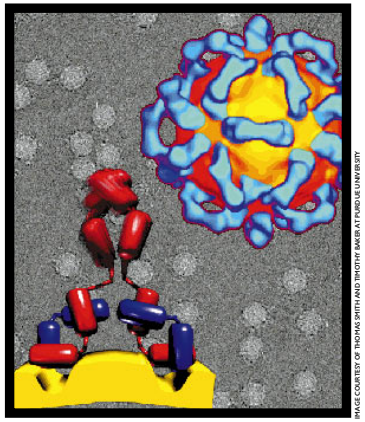


IMAGE COURTESY OF THOMAS SMITH AND TIMOTHY MAVER AT PURDUE UNIVERSITY

Analogies with immunology represent an important step toward the vision of robust, distributed protection for computers.

Computer Immunology

Stephanie Forrest, Steven A. Hofmeyr, and Anil Somayaji

NATURAL IMMUNE SYSTEMS PROTECT ANIMALS from dangerous foreign pathogens, including bacteria, viruses, parasites, and toxins. Their role in the body is analogous to that of computer security systems in computing. Although there are many differences between living organisms and computers, the similarities are compelling and could point the way to improved computer security.[1] Improvements can be achieved by designing computer immune systems with some of the important properties of natural immune systems, including multilayered protection; highly distributed detector, effector, and memory systems; diversity of detection ability across individuals; inexact matching strategies; and sensitivity to most new foreign patterns. Some of these properties are well known but seldom implemented successfully; other properties are less well known. The immune system provides a persuasive example of how they might be implemented in a coherent system.

The immune system comprises cells and molecules.[2] Recognition of foreign protein, called *antigen*, occurs when immune system detectors, including T cells, B cells, and antibodies, bind to antigen. Binding between detector and antigen is determined by the physical and chemical properties of their binding regions. Binding is highly specific, so each detector recognizes only a limited set of structurally related antigen. When a detector and antigen bind, a com-

[1]The authors gratefully acknowledge support from the National Science Foundation (IRI-9157644), the Office of Naval Research (N00014-95-1-0364), Defense Advanced Research Projects Agency (N00014-96-1-0680), Interval Research Corp., the MIT Artificial Intelligence Laboratory, and the Santa Fe Institute.

[2]A good source for basic immunology is [6]; a computer scientist's overview of immunology is given at http://www.cs.unm.edu/~stevesh/imm-html/immune-system.html.

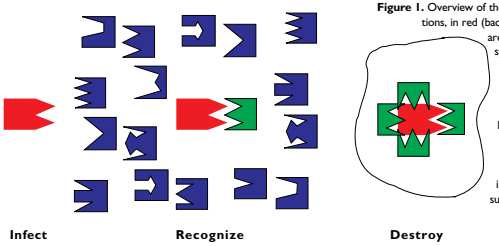
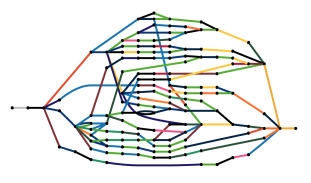


Figure 1. Overview of the immune system. Infections, in red (bacteria, viruses, parasites), are recognized by immune system detectors, in blue and green (T cells, B cells, and antibodies), when molecular bonds form between them. Infections are eliminated by general-purpose scavenger cells (macrophages), indicated by the thin line surrounding the detector/antigen complex.

Source: Figure prepared by R. Hightower.

(Forrest et al., Comm. of the ACM, 1997; 10.1145/262793.262811)

Artificial Immune Systems



Self-Nonself Discrimination

The Negative Selection Algorithm

Broadly speaking, an artificial immune system (AIS) is any algorithm or engineered system that is inspired by the real immune system.
Since the immune system is so complex and diverse, so are AISs. Examples of more sucessful AISes include:

- The dendritic cell algorithm, inspired by a certain cell type of the innate IS.
- The immune network algorithm, inspired by a long-disproved theory of how the immune system works.
- The B cell algorithm, inspired by how antibodies are made.

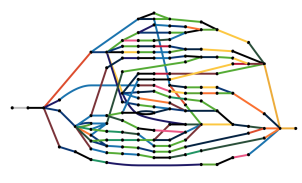
This lecture focuses on one particular AIS: the negative selection algorithm.
The negative selection algorithm is a learning algorithm. To explain what it does, we first take a small detour, and define what a learning algorithm does.

Learning theory

- In the 1960s, researchers wanted to build computers that perform cognitive tasks (artificial intelligence).
- A first approach was to defiend these tasks in a strict formal language.
- This gave rise to what is now known as (computational) learning theory.

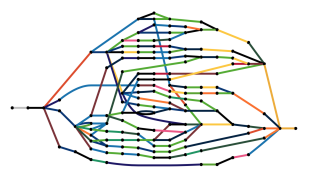
Leslie Valiant, Recipient of the A.M. Turing Award 2010

We will now introduce one formal model of learning.



Self-Nonself Discrimination

The Negative Selection Algorithm



Inductive inference

An inductive inference learning scenario is defined by:

- A learning domain $\mathcal{X}$ – any set.
- A concept class $\mathcal{C}$ – any set of subsets of $\mathcal{X}$.
- A hypothesis space $\mathcal{H}$ – an approximation of $\mathcal{C}$.
- The learner is given a sequence of examples from a target concept $c \in \mathcal{C}$ and outputs a sequence of hypotheses from $\mathcal{H}$
- These hypotheses should converge to the concept c.

(Gold, Information and Control, 1967)

Slow learning *(McClelland et al, Psych Rev 1995)*

Any process that

- learns from a sequence of examples;
- stores only a characterization of these examples; and
- improves the characterization as new examples are seen.

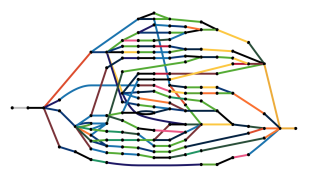
Inductive inference: Example

- Learning domain $\mathcal{X}$: images
- Concept class $\mathcal{C}$: Latvian words
- Hypothesis space $\mathcal{H}$: English words

Let's try to learn the concept *balts*.

input:

output:



Self-Nonself Discrimination

The Negative Selection Algorithm

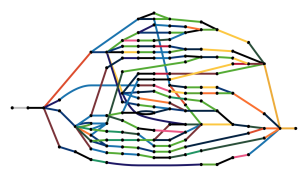
Inductive inference: Example

- Learning domain $\mathcal{X}$: images
- Concept class $\mathcal{C}$: Latvian words
- Hypothesis space $\mathcal{H}$: English words

Let's try to learn the concept *balts*.

input:

output: egg



Self-Nonself Discrimination

The Negative Selection Algorithm

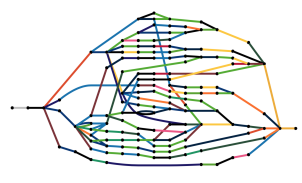
Inductive inference: Example

- Learning domain $\mathcal{X}$: images
- Concept class $\mathcal{C}$: Latvian words
- Hypothesis space $\mathcal{H}$: English words

Let's try to learn the concept *balts*.

input:

output: egg food



Self-Nonself Discrimination

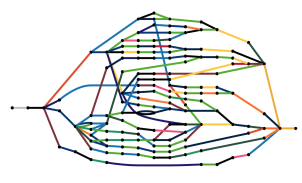
The Negative Selection Algorithm

Inductive inference: Example

- Learning domain $\mathcal{X}$: images
- Concept class $\mathcal{C}$: Latvian words
- Hypothesis space $\mathcal{H}$: English words

Let's try to learn the concept *balts*.

input:			
output:	egg	food	white



Self-Nonself Discrimination

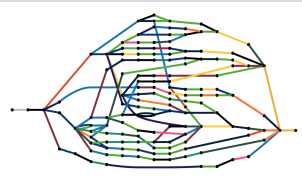
The Negative Selection Algorithm

Inductive inference: Example

- Learning domain $\mathcal{X}$: images
- Concept class $\mathcal{C}$: Latvian words
- Hypothesis space $\mathcal{H}$: English words

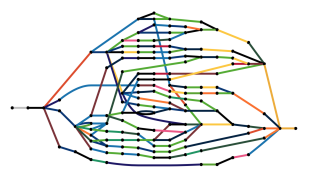
Let's try to learn the concept *balts*.

input:				
output:	egg	food	white	white



Self-Nonself Discrimination

The Negative Selection Algorithm



Inductive inference: Example

- Learning domain $\mathcal{X}$: images
- Concept class $\mathcal{C}$: Latvian words
- Hypothesis space $\mathcal{H}$: English words

Let's try to learn the concept *balts*.

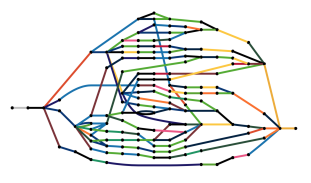
input:				
output:	egg	food	white	white

Generalization

Which of these is not *balts*?

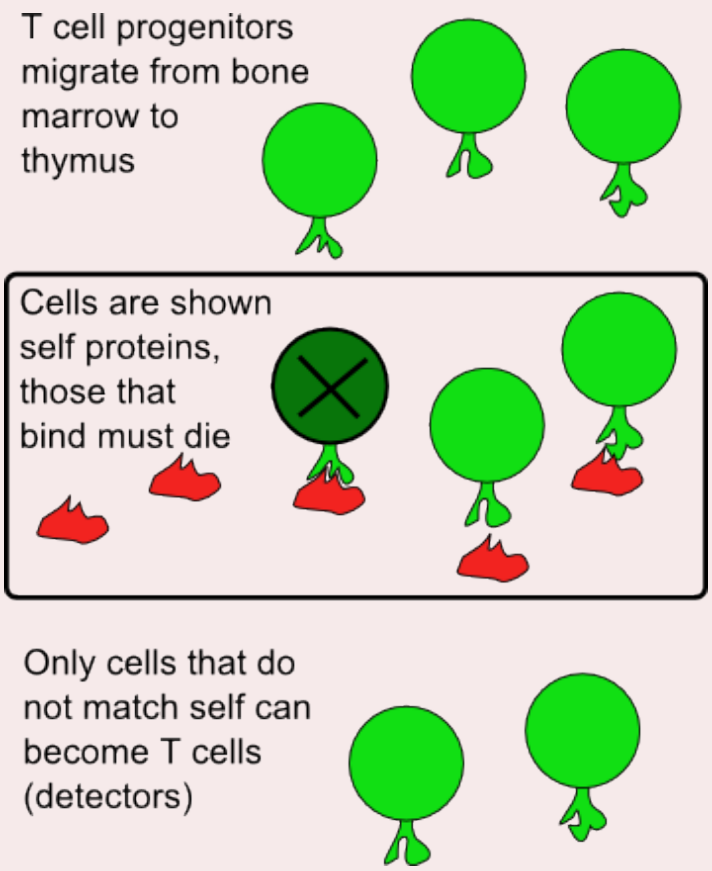
To answer, we do not need to ...

- have seen everything *balts*
- have seen anything not *balts*



General Structure of a Negative Selection Algorithm

Negative Selection in the IS



The negative selection algorithm

input: self-set $S \subseteq$ SELF,
monitoring set $M \subseteq \mathcal{X}$

output: For each $m \in M$:
either SELF or NON-SELF

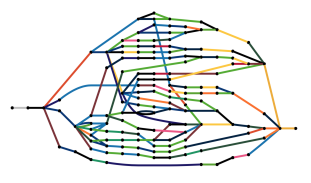
$D \leftarrow$ Set of detectors that do not match any $s \in S$

for each $m \in M$ **do**
  if m matches any $d \in D$ **then**
    output "m IS NON-SELF"
  else
    output "m IS SELF"

(Forrest, 1997)

concept class	subsets of $\mathcal{X}$
target concept	SELF
hypothesis	set of detectors D

Different Kinds of Negative Selection Algorithms



Negative selection is a one-class classification problem: we need to solve a binary classification problem (self-nonself), but are given only examples from one class (self).
Negative selection is a generic algorithm scheme; for concrete applications, we need to define the detectors and the matching rule.

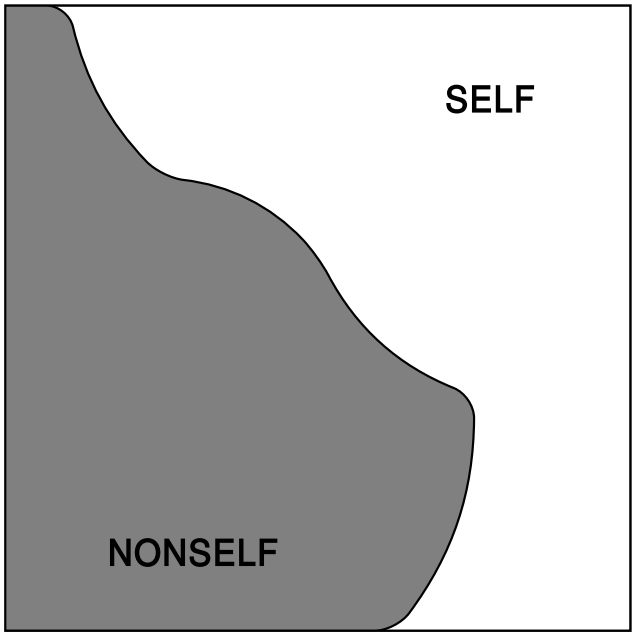
Real-valued negative selection

Learning domain: $\mathbb{R}^d$ (d-dimensional vectors)
Detectors: hyperspheres, hypercubes, ...

String-based negative selection

Learning domain: Σ^ℓ (Strings of length ℓ)
Detectors: patterns

Real-Valued Negative Selection in 2D

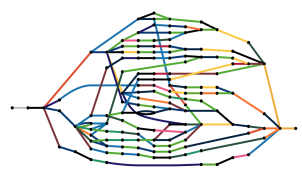


Negative selection

input: self-set $S \subseteq$ SELF
output: detector set D
$D \leftarrow$ random set of circles

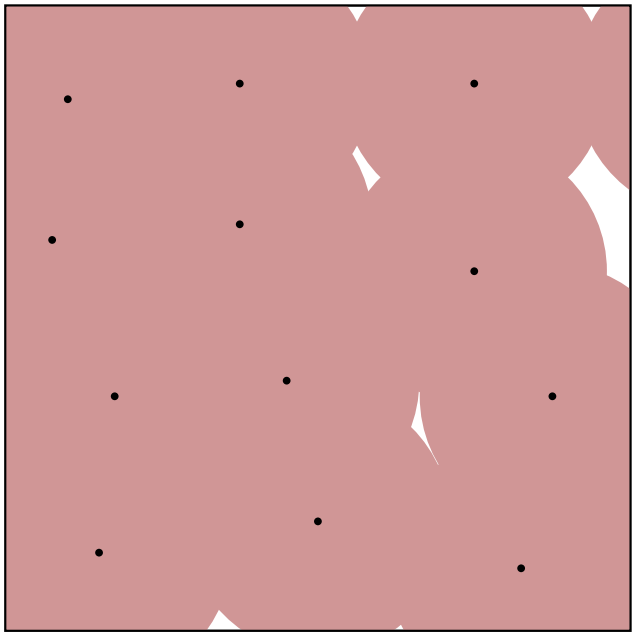
for each $s \in S$
    delete from D all circles
        that contain s
output h

- Learning domain $\mathcal{X}$: $\mathbb{R}^2$
- Concept class $\mathcal{C}$: subregions of $\mathbb{R}^2$
- Hypotheses $h \in \mathcal{H}$: sets of circles with fixed radius r.
- Generalization occurs, and is linked is linked to r.



Self-Nonself Discrimination
The Negative Selection Algorithm

Real-Valued Negative Selection in 2D

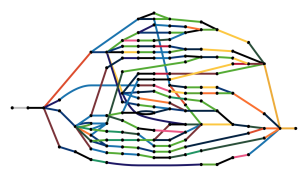


Negative selection

input: self-set $S \subseteq$ SELF
output: detector set D
$D \leftarrow$ random set of circles

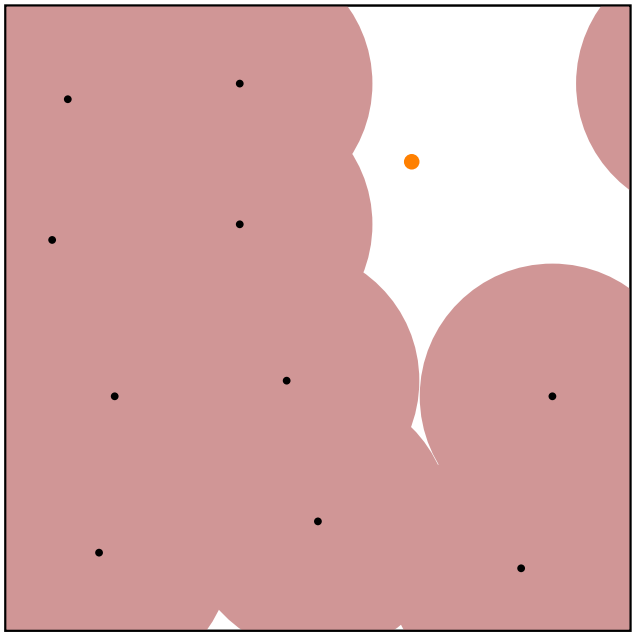
for each $s \in S$
    delete from D all circles
        that contain s
output h

- Learning domain $\mathcal{X}$: $\mathbb{R}^2$
- Concept class $\mathcal{C}$: subregions of $\mathbb{R}^2$
- Hypotheses $h \in \mathcal{H}$: sets of circles with fixed radius r.
- Generalization occurs, and is linked is linked to r.



Self-Nonself Discrimination
The Negative Selection Algorithm

Real-Valued Negative Selection in 2D

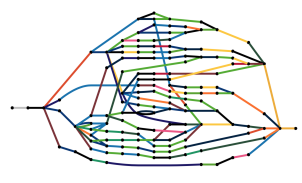


Negative selection

input: self-set $S \subseteq$ SELF
output: detector set D
$D \leftarrow$ random set of circles

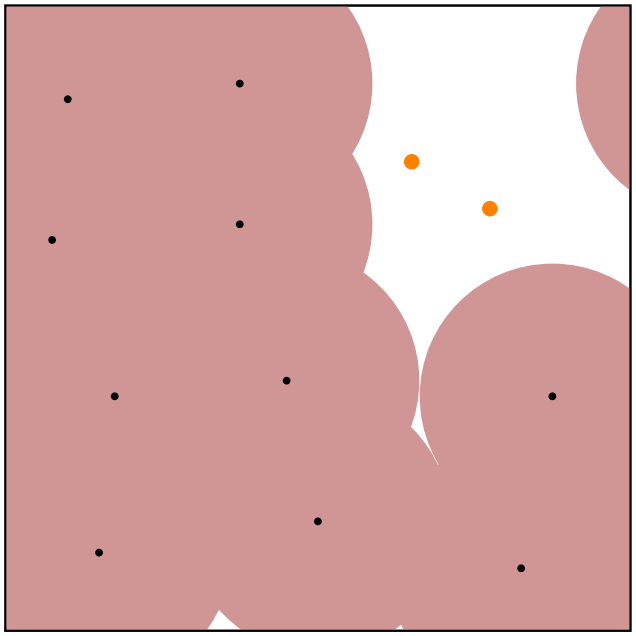
for each $s \in S$
    delete from D all circles
        that contain s
output h

- Learning domain $\mathcal{X}$: $\mathbb{R}^2$
- Concept class $\mathcal{C}$: subregions of $\mathbb{R}^2$
- Hypotheses $h \in \mathcal{H}$: sets of circles with fixed radius r.
- Generalization occurs, and is linked is linked to r.



Self-Nonself Discrimination
The Negative Selection Algorithm

Real-Valued Negative Selection in 2D

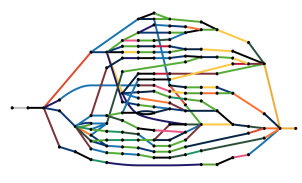


Negative selection

input: self-set $S \subseteq$ SELF
output: detector set D
$D \leftarrow$ random set of circles

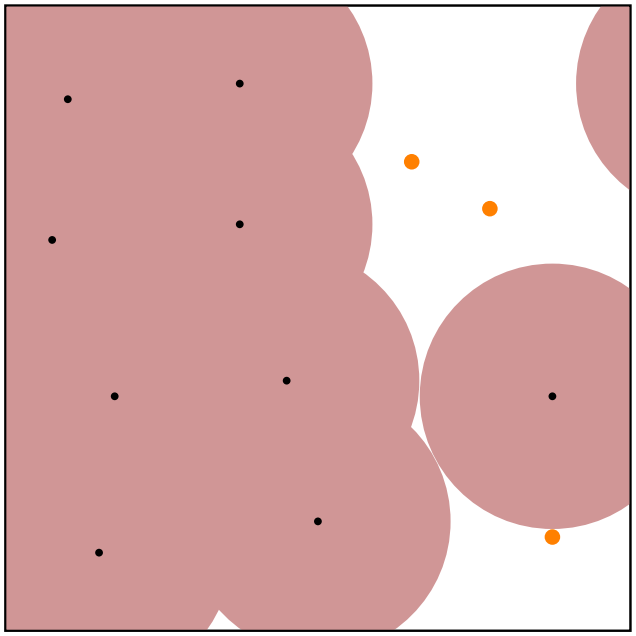
for each $s \in S$
    delete from D all circles
        that contain s
output h

- Learning domain $\mathcal{X}$: $\mathbb{R}^2$
- Concept class $\mathcal{C}$: subregions of $\mathbb{R}^2$
- Hypotheses $h \in \mathcal{H}$: sets of circles with fixed radius r.
- Generalization occurs, and is linked is linked to r.



Self-Nonself Discrimination
The Negative Selection Algorithm

Real-Valued Negative Selection in 2D

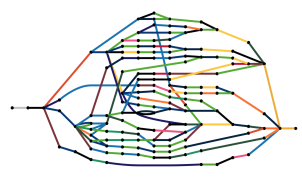


Negative selection

input: self-set $S \subseteq$ SELF
output: detector set D
$D \leftarrow$ random set of circles

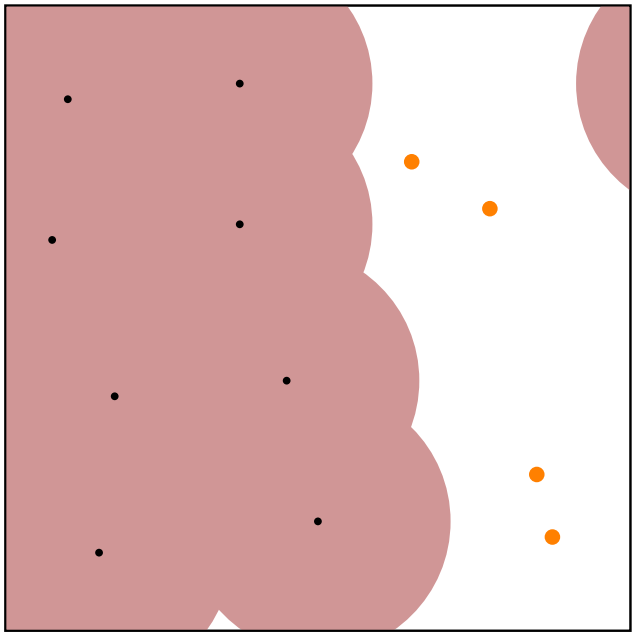
for each $s \in S$
    delete from D all circles
        that contain s
output h

- Learning domain $\mathcal{X}$: $\mathbb{R}^2$
- Concept class $\mathcal{C}$: subregions of $\mathbb{R}^2$
- Hypotheses $h \in \mathcal{H}$: sets of circles with fixed radius r.
- Generalization occurs, and is linked is linked to r.



Self-Nonself Discrimination
The Negative Selection Algorithm

Real-Valued Negative Selection in 2D

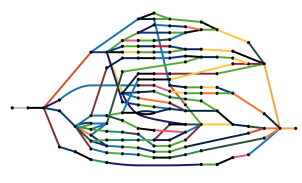


Negative selection

input: self-set $S \subseteq$ SELF
output: detector set D
$D \leftarrow$ random set of circles

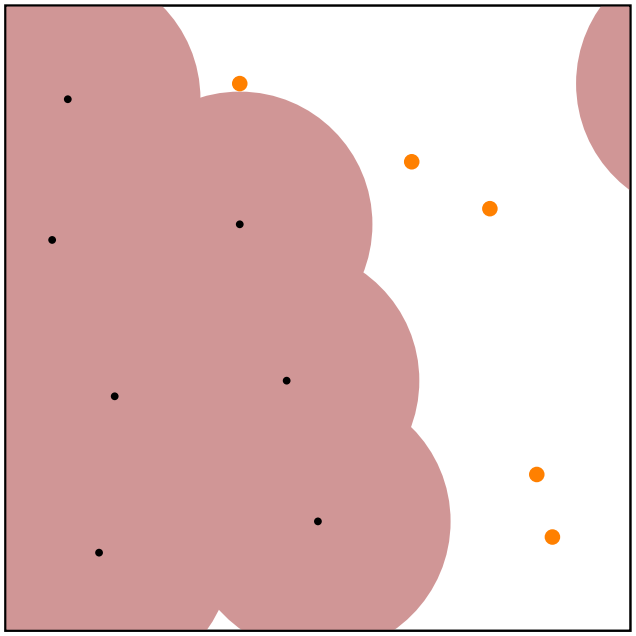
for each $s \in S$
    delete from D all circles
        that contain s
output h

- Learning domain $\mathcal{X}$: $\mathbb{R}^2$
- Concept class $\mathcal{C}$: subregions of $\mathbb{R}^2$
- Hypotheses $h \in \mathcal{H}$: sets of circles with fixed radius r.
- Generalization occurs, and is linked is linked to r.



Self-Nonself Discrimination
The Negative Selection Algorithm

Real-Valued Negative Selection in 2D

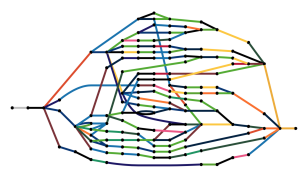


Negative selection

input: self-set $S \subseteq$ SELF
output: detector set D
$D \leftarrow$ random set of circles

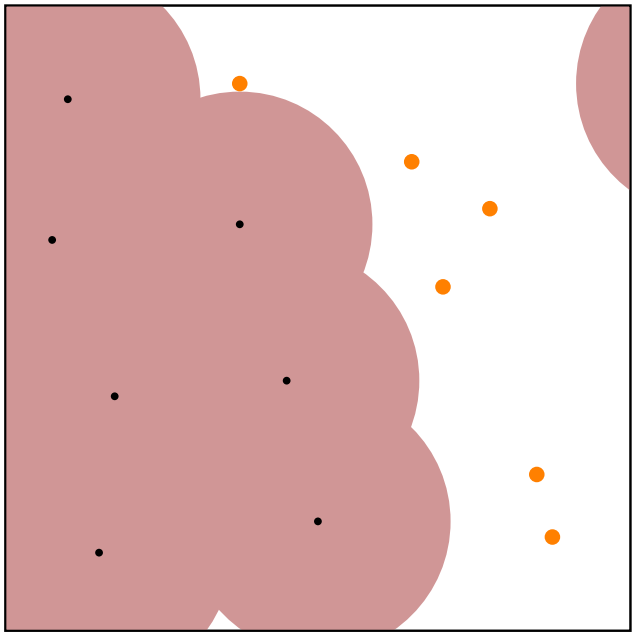
for each $s \in S$
    delete from D all circles
        that contain s
output h

- Learning domain $\mathcal{X}$: $\mathbb{R}^2$
- Concept class $\mathcal{C}$: subregions of $\mathbb{R}^2$
- Hypotheses $h \in \mathcal{H}$: sets of circles with fixed radius r.
- Generalization occurs, and is linked is linked to r.



Self-Nonself Discrimination
The Negative Selection Algorithm

Real-Valued Negative Selection in 2D



Negative selection

input: self-set $S \subseteq$ SELF
output: detector set D
$D \leftarrow$ random set of circles

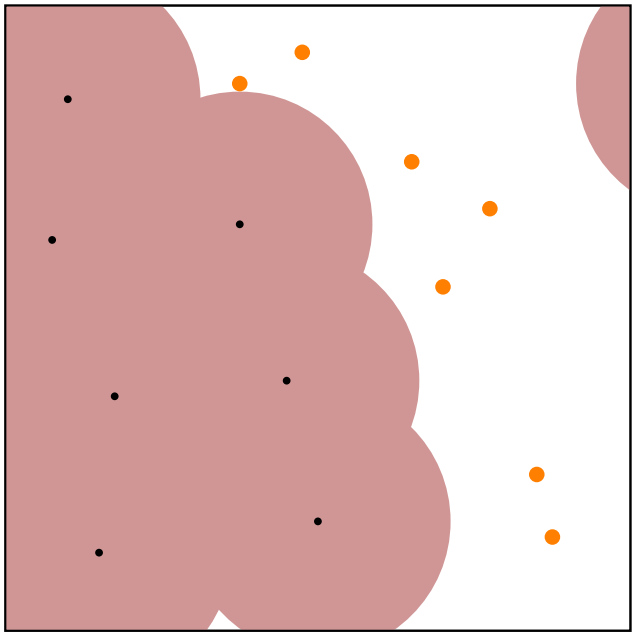
for each $s \in S$
    delete from D all circles
        that contain s
output h

- Learning domain $\mathcal{X}$: $\mathbb{R}^2$
- Concept class $\mathcal{C}$: subregions of $\mathbb{R}^2$
- Hypotheses $h \in \mathcal{H}$: sets of circles with fixed radius r.
- Generalization occurs, and is linked is linked to r.

Self-Nonself Discrimination

The Negative Selection Algorithm

Real-Valued Negative Selection in 2D



Negative selection

input: self-set $S \subseteq$ SELF
output: detector set D
$D \leftarrow$ random set of circles

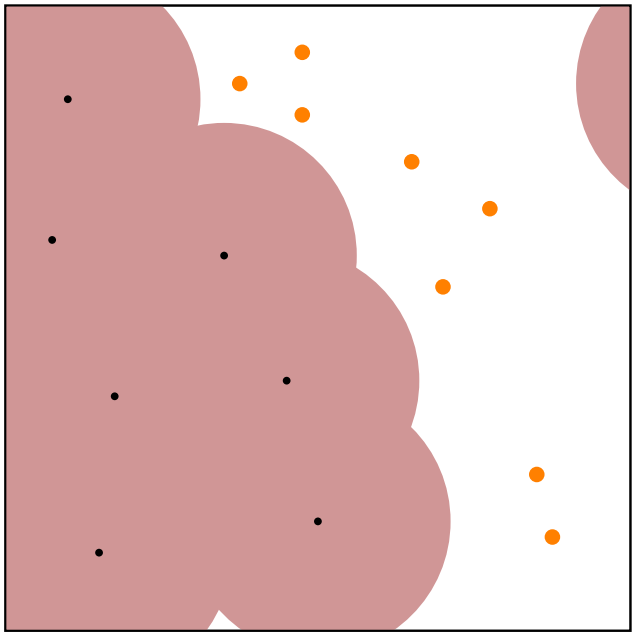
for each $s \in S$
    delete from D all circles
        that contain s
output h

- Learning domain $\mathcal{X}$: $\mathbb{R}^2$
- Concept class $\mathcal{C}$: subregions of $\mathbb{R}^2$
- Hypotheses $h \in \mathcal{H}$: sets of circles with fixed radius r.
- Generalization occurs, and is linked is linked to r.

Self-Nonself Discrimination

The Negative Selection Algorithm

Real-Valued Negative Selection in 2D

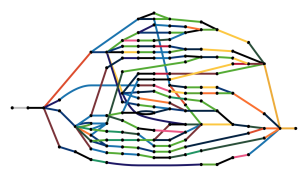


Negative selection

input: self-set $S \subseteq$ SELF
output: detector set D
$D \leftarrow$ random set of circles

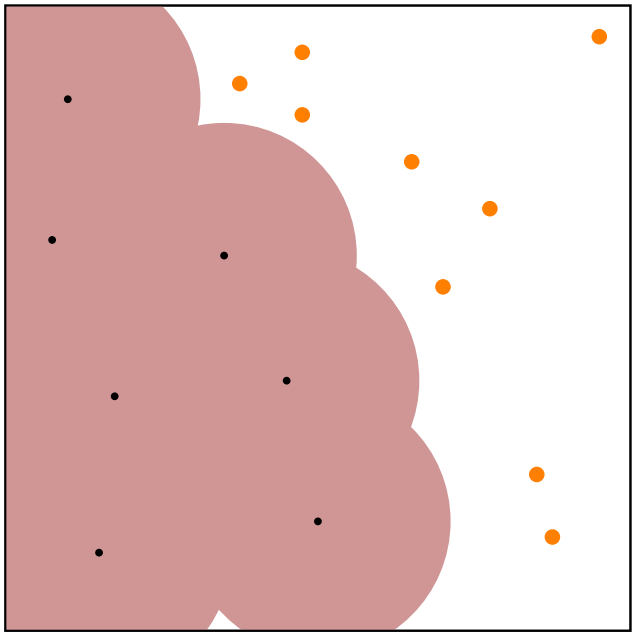
for each $s \in S$
    delete from D all circles
        that contain s
output h

- Learning domain $\mathcal{X}$: $\mathbb{R}^2$
- Concept class $\mathcal{C}$: subregions of $\mathbb{R}^2$
- Hypotheses $h \in \mathcal{H}$: sets of circles with fixed radius r.
- Generalization occurs, and is linked is linked to r.



Self-Nonself Discrimination
The Negative Selection Algorithm

Real-Valued Negative Selection in 2D

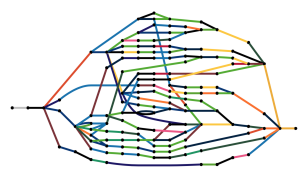


Negative selection

input: self-set $S \subseteq$ SELF
output: detector set D
$D \leftarrow$ random set of circles

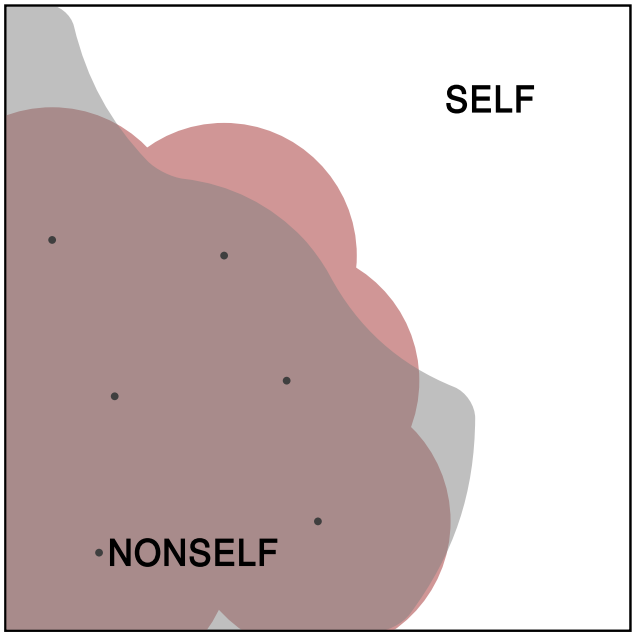
for each $s \in S$
    delete from D all circles
        that contain s
output h

- Learning domain $\mathcal{X}$: $\mathbb{R}^2$
- Concept class $\mathcal{C}$: subregions of $\mathbb{R}^2$
- Hypotheses $h \in \mathcal{H}$: sets of circles with fixed radius r.
- Generalization occurs, and is linked is linked to r.



Self-Nonself Discrimination
The Negative Selection Algorithm

Real-Valued Negative Selection in 2D

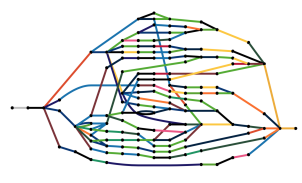


Negative selection

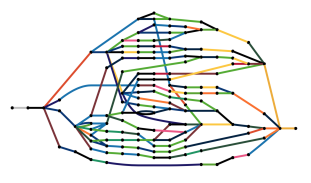
input: self-set $S \subseteq$ SELF
output: detector set D
$D \leftarrow$ random set of circles

for each $s \in S$
    delete from D all circles
        that contain s
output h

- Learning domain $\mathcal{X}$: $\mathbb{R}^2$
- Concept class $\mathcal{C}$: subregions of $\mathbb{R}^2$
- Hypotheses $h \in \mathcal{H}$: sets of circles with fixed radius r.
- Generalization occurs, and is linked is linked to r.



Self-Nonself Discrimination
The Negative Selection Algorithm



5-Minute Exercise

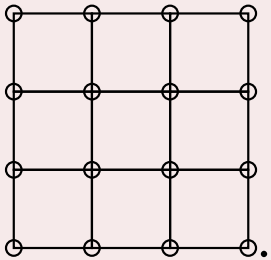
Exercise

Consider the following learning scenario.

- Learning domain $\mathcal{X}$: 2D integer lattice $\{0,\ldots,3\} \times \{0,\ldots,3\}$.
- Concept class: sets of points on $\mathcal{X}$.
- Detectors: unit-area integer squares in $\mathcal{X}$.

Suppose we run negative selection on the input $S = \{(0,0),(1,1),(2,2),(3,3)\}$ and generate an initial detector set D containing all possible squares, i.e.,

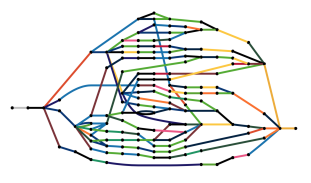
$$S = \begin{array}{cccc} \circ & \circ & \circ & \bullet \\ \circ & \circ & \bullet & \circ \\ \circ & \bullet & \circ & \circ \\ \bullet & \circ & \circ & \circ \end{array} \quad ; \quad D =$$



What is the output detector set? Does generalization occur?

(Formally, each square is a set $\{(x,y),(x+1,y),(x,y+1),(x+1,y+1)\}$ where $x,y \in \{0,1,2\}$, and it matches all points contained in it.)

String-Based Negative Selection



Consider the learning domain of fixed-length binary strings: $\mathcal{X} = \{0, 1\}^\ell$.

Self-set S

01000	11010
01001	11011
01010	
01011	

r-chunk detector

Pair of string s of length r and position i. Matches a string s' of length ℓ, if s occurs in s' at position i.

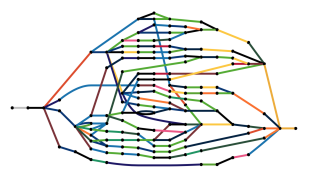
E.g. (010,3); (111,2)

r-contiguous detector

String s of length ℓ. Matches another string s', if s and s' contain the same letters at r subsequent positions.

E.g. 00011; 00100

String-Based Negative Selection



Consider the learning domain of fixed-length binary strings: $\mathcal{X} = \{0, 1\}^\ell$.

Self-set S

01000	11010
01001	11011
01010	
01011	

r-chunk detector

Pair of string s of length r and position i. Matches a string s' of length ℓ, if s occurs in s' at position i.

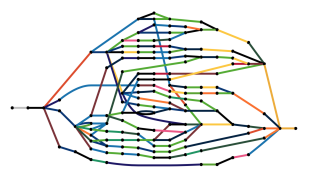
E.g. (010,3); (111,2)

r-contiguous detector

String s of length ℓ. Matches another string s', if s and s' contain the same letters at r subsequent positions.

E.g. 00011; 00100

String-Based Negative Selection



Consider the learning domain of fixed-length binary strings: $\mathcal{X} = \{0, 1\}^\ell$.

Self-set S

01000	11010
01001	11011
01010	
01011	

r-chunk detector

Pair of string s of length r and position i. Matches a string s' of length ℓ, if s occurs in s' at position i.

E.g. (010,3); (111,2)

r-contiguous detector

String s of length ℓ. Matches another string s', if s and s' contain the same letters at r subsequent positions.

E.g. 00011; 00100

More Detectors and Matching Rules

r-chunk

non-jokers in block of length r

*	1	0	1	*	*	pattern, $r = 3$
0	1	0	1	1	0	string

(Balthrop et al., GECCO '02)

r-contiguous

match at $\geq r$ contiguous positions

1	1	0	1	0	0	pattern, $r = 3$
0	1	0	1	1	0	string

(Percus et al., PNAS '93)

monomial

jokers allowed everywhere

0	*	*	1	*	*	pattern
0	1	0	1	1	0	string

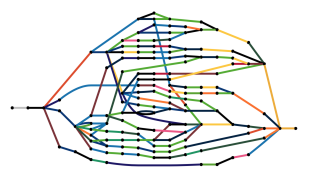
(Liśkiewicz & Textor, GECCO '10)

r-distributed

match at $\geq r$ positions

0	1	0	1	1	0	pattern, $r = 3$
0	1	1	0	0	0	string

(Harmer, IEEE Trans EC '02)



Self-Nonself Discrimination

The Negative Selection Algorithm

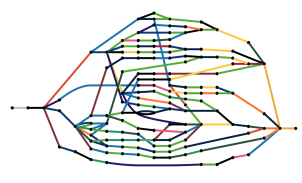
String-Based Negative Selection: Example

Negative selection with $\mathcal{X} = \{0, 1\}^8$, r-contiguous detectors, $r = 5$.

- Generate detector at random.
- Discard if it matches any self string.
- Repeat until enough detectors are generated.

Self-Set S
1 1 0 0 0 1 0 1
0 0 1 1 0 0 1 0
0 0 1 1 1 1 0 0
0 0 1 0 0 1 0 0
1 0 1 1 1 0 1 0

Detector-Set D
1 1 0 1 1 0 1 1
1 1 1 0 1 1 1 1
0 1 0 0 1 0 0 1
0 1 1 1 0 0 0 1



Self-Nonself Discrimination

The Negative Selection Algorithm

Controlling Generalization

The parameter r (matching radius) can be used to control generalization.

Here, we show the inferred self and nonself (gray background) for r-chunk detectors, $\mathcal{X} = \{a, b\}^5$.

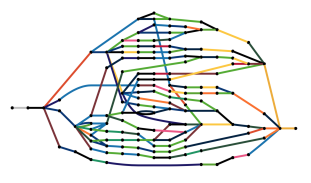
The input S is shown in **bold**. All possible detectors are used as the initial detector set.

$r = 2$

aaaaa	abaaa	**baaaa**	bbaaa
aaaab	abaab	**baaab**	bbaab
aaaba	ababa	**baaba**	bbaba
aaabb	ababb	baabb	bbabb
aabaa	abbaa	babaa	bbbaa
aabab	abbab	babab	bbbab
aabba	abbba	**babba**	bbbba
aabbb	**abbbb**	babbb	**bbbbb**

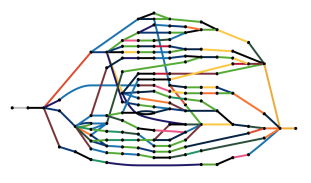
$r = 3$

aaaaa	abaaa	**baaaa**	bbaaa
aaaab	abaab	**baaab**	bbaab
aaaba	ababa	**baaba**	bbaba
aaabb	ababb	baabb	bbabb
aabaa	abbaa	babaa	bbbaa
aabab	abbab	babab	bbbab
aabba	abbba	**babba**	bbbba
aabbb	**abbbb**	babbb	**bbbbb**



Self-Nonself Discrimination

The Negative Selection Algorithm



Anomaly Detection in Unix Processes

Idea: trace unix system calls of a daemon process like `sendmail`, coding each system call as an alphabet symbol. Generate a database of system call sequences in normal circumstances.

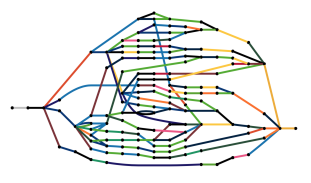
Apply negative selection to these sequences and generate detectors, which may be capable to detect anomalous operations.

Example

`iKEKEEFEEgKEgKEEFEggFKvafvaeggggj`	normal
`6LgggggngngnggLLnnPPggnnggLggiKEKE`	expoit

In your assignment, you will apply negative selection to such data.

S. Forrest, S. A. Hofmeyr, A. Somayaji, T. A. Longstaff:
A sense of self for unix processes.
IEEE Symposium on Security and Privacy, 1996



Summary

- The immune system's T cells are negatively selected to match anything but self.
- A negative selection algorithm is any one-class classifier that is based on this principle.
- Negative selection algorithms are able to learn from examples and to generalize.
- Substantial work is needed to implement a negative selection algorithm efficiently, but that is out of the scope of this lecture.
- Negative selection algorithms can be applied to many data domains, but most "interesting" applications involve strings (see assignment).