On Permutation Masks in Hamming Negative Selection

Thomas Stibor[1] and Jonathan Timmis[2] and Claudia Eckert[1]

[1] Department of Computer Science
Darmstadt University of Technology
{*stibor,eckert*}*@sec.informatik.tu-darmstadt.de*
[2] Departments of Electronics and Computer Science
University of York, Heslington, York
jtimmis@cs.york.ac.uk

Abstract. Permutation masks were proposed for reducing the number of holes in Hamming negative selection when applying the r-contiguous or r-chunk matching rule. Here, we show that (randomly determined) permutation masks re-arrange the semantic representation of the underlying data and therefore shatter self-regions. As a consequence, detectors do not cover areas around self regions, instead they cover randomly distributed elements across the space. In addition, we observe that the resulting holes occur in regions where actually no self regions should occur.

1 Introduction

Applying negative selection for anomaly detection problems has been undertaken extensively [1,2,3,4]. Anomaly detection problems, also termed one-class classification, can be considered as a type of pattern classification problem, where one tries to describe a single class of objects, and distinguish that from all other possible objects. More formally, one-class classification is a problem of generating decision boundaries that can successfully distinguish between the normal and anomalous class. Hamming negative selection is an immune-inspired technique for one-class classification problems. Recent results, however, have revealed several problems concerning algorithm complexity of generating detectors [5,6,7] and determining the proper matching threshold to allow for the generation of correct generalization regions [8]. In this paper we investigate an extended technique for Hamming negative selection: permutation masks. Permutation masks are immunologically motivated by lymphocyte diversity. Lymphocyte diversity is an important property of the immune system, as it enables a lymphocyte to reacting to many substances, i.e. it induces diversity and generalization. This kind of generalization process inspired Hofmeyr [3,9] to propose a similar counterpart for use in Hamming negative selection. Hofmeyr introduced permutation masks in order to reduce the number of undetectable elements. It was argued that permutation masks could be useful for covering the non-self space efficiently when varying the representation by means of permutation masks (see Fig. 1).

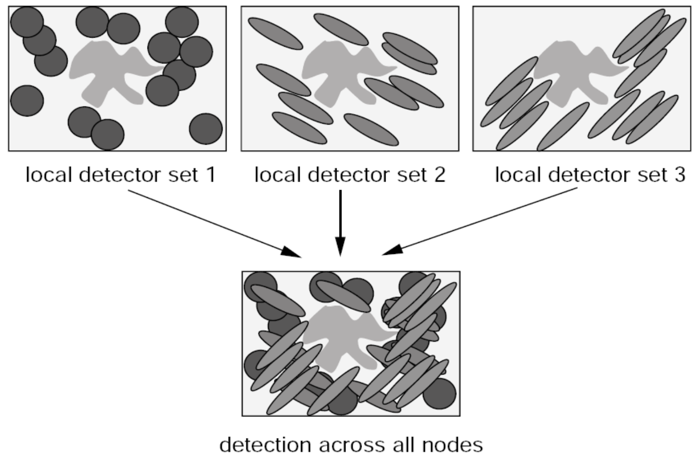


Fig. 1. Visualized concept of varying representations by means of permutation masks to reduce the number of undetectable elements. The light gray shaded area in the middle represents the self regions (normal class in terms of anomaly detection). The dark gray shaded shapes represent areas which are covered by detectors with varying representations. The white area represents the non-self space (anomalous class in terms of anomaly detection). This figure is taken from [9].

In the following two sections we briefly introduce the standard negative selection inspired anomaly detection technique.

2 Artificial Immune System

An artificial immune system (AIS) [10] is a paradigm inspired by the immune system and are used for solving computational and information processing problems. An AIS can be described, and developed, using a framework [10] which contains the following basic elements:

- A representation for the artificial immune elements.
- A set of functions, which quantifies the interactions of the artificial immune elements (affinity).
- A set of algorithms which based on observed immune principles and methods.

This 3-step abstraction (representation, affinity, algorithm) for using the AIS framework is discussed in the following sections.

2.1 Hamming Shape-Space

The notion of *shape-space* was introduced by Perelson and Oster [11] and allows a quantitative affinity description between immune components known as antibodies and antigens. More precisely, a shape-space is a metric space with an associated distance (affinity) function.

The Hamming shape-space U_l^{Σ} is built from all elements of length l over a finite alphabet Σ.

Example 1.

$$
\begin{array}{ccc}
\Sigma = \{0, 1\} & \qquad & \Sigma = \{A, C, G, T\} \\
 & & \\
000\ldots000 & & AAA\ldots AAA \\
000\ldots001 & & AAA\ldots AAC \\
\ldots\ldots\ldots\ldots & & \ldots\ldots\ldots\ldots \\
\ldots\ldots\ldots\ldots & & \ldots\ldots\ldots\ldots \\
\underbrace{111\ldots111}_{l} & & \underbrace{TTT\ldots TTT}_{l}
\end{array}
$$

In example 1 two Hamming shape-spaces for different alphabets and alphabet sizes are presented. On the left, a Hamming shape-space defined over the binary alphabet of length l is shown. On the right, a Hamming shape-space defined over the DNA bases alphabet (Adenine, Cytosine, Guanine, Thymine) is presented.

2.2 R-contiguous and R-chunk Matching

A formal description of antigen-antibody interactions not only requires a representation (encoding), but also appropriate affinity functions. Percus et. al [12] proposed the *r-contiguous* matching rule for abstracting the affinity of an antibody needed to recognize an antigen.

Definition 1. *An element $e \in U_l^{\Sigma}$ with $e = e_1 e_2 \ldots e_l$ and detector $d \in U_l^{\Sigma}$ with $d = d_1 d_2 \ldots d_l$, match with r-contiguous rule, if a position p exists where $e_i = d_i$ for $i = p, \ldots, p + r - 1$, $p \leq l - r + 1$.*

Informally, two elements, with the *same length*, match if at least r contiguous characters are identical.

An additional rule, which subsumes[3] the r-contiguous rule, is the r-chunk matching rule [13].

Definition 2. *An element $e \in U_l^{\Sigma}$ with $e = e_1 e_2 \ldots e_l$ and detector $d \in \mathbb{N} \times D_r^{\Sigma}$ with $d = (p \mid d_1 d_2 \ldots d_r)$, for $r \leq l$, $p \leq l - r + 1$ match with r-chunk rule, if a position p exists where $e_i = d_i$ for $i = p, \ldots, p + r - 1$.*

Informally, element e and detector d match if a position p exists, where all characters of e and d are identical over a sequence of length r.

We use the term *subsume* as any r-contiguous detector can be represented as a set of r-chunk detectors. This implicates that any set of elements from U_l^{Σ} that can be recognized with a set of r-contiguous detectors can also be recognized with some set of r-chunk detectors. The converse statement is surprisingly *not*

[3] include within a larger entity

true, i.e. there exists a set of elements from U_l^{Σ} that can be recognized with a set of r-chunk detectors, but *not* recognized with any set of r-contiguous detectors. We demonstrate this converse statement on an example, a formal approach is provided in [14].

Example 2. Given a Hamming shape-space $U_5^{\{0,1\}}$, a set $S = \{01011, 01100, 01110, 10010, 10100, 11100\}$ of self elements and a detector length $r = 3$.

All possible generable r-contiguous detectors for the complementary space $U_5^{\{0,1\}} \setminus S$ are $D_{r-contiguous} = \{00000, 00001, 00111, 11000, 11001\}$.

All possible generable r-chunk detectors are
$D_{r-chunk} = \{1|000, 1|001, 1|110, 2|000, 2|011, 2|100, 3|000, 3|001, 3|101, 3|111\}$.

The set $D_{r-contiguous}$ recognizes the elements
$\mathcal{P}_1 = U_5^{\{0,1\}} \setminus (S \cup \{01010, 01101, 10011, 10101, 11101, 11110\})$,
whereas the set $D_{r-chunk}$ recognizes the elements
$\mathcal{P}_2 = U_5^{\{0,1\}} \setminus (S \cup \{10011, 01010, 11110\})$. Hence $|\mathcal{P}_1| \leq |\mathcal{P}_2|$.

Example 2 shows, that the set of r-chunk detectors $D_{r-chunk}$ recognizes more elements of $U_5^{\{0,1\}}$ than the set of r-contiguous detectors $D_{r-contiguous}$ and therefore the r-chunk matching rule subsumes the r-contiguous rule.

3 Hamming Negative Selection

Forrest et al. [1] proposed a (generic[4]) negative selection algorithm for detecting changes in data streams. Given a shape-space $U = S_{seen} \cup S_{unseen} \cup N$ which is partitioned into training data S_{seen} and testing data $(S_{seen} \cup S_{unseen} \cup N)$. The basic idea is to generate a number of detectors for the complementary space $U \setminus S_{seen}$ and then to apply these detectors to classify new (unseen) data as self (no data manipulation) or non-self (data manipulation).

Algorithm 1: Generic Negative Selection Algorithm

---

input : S_{seen} = set of self seen elements
output: D = set of generated detectors
begin
  1. Define self as a set S_{seen} of elements in shape-space U
  2. Generate a set D of detectors, such that each fails to match any element in S_{seen}
  3. Monitor (seen and unseen) data $\delta \subseteq U$ by continually matching the detectors in D against δ.
end

The generic negative selection algorithm can be used with arbitrary shape-spaces and affinity functions. In this paper, we focus on Hamming negative selection,

[4] applicable to arbitrary shape-spaces

i.e. the negative selection algorithm which operates on Hamming shape-space and employs the r-chunk matching rule and permutation masks.

3.1 Holes as Generalization Regions

The r-contiguous and r-chunk matching rule induce undetectable elements — termed holes (see Fig. 2). In general, all matching rules which match over a certain element length induce holes. This statement is theoretically investigated in [15,14] and empirically explored[5] in [16]. Holes are some[6] elements from $U \setminus S_{seen}$, i.e. elements not seen during the training phase. For these elements, no detectors can be generated and therefore they cannot be recognized and classified as non-self elements. However, the term holes is not an accurate expression, as holes are *necessary* to generalize beyond the training set. A detector set which generalizes well ensures that seen and unseen self elements are *not* recognized by any detector, whereas all other elements are recognized by detectors and classified as non-self. Hence, holes must represent unseen self elements; or in other words, holes must represent *generalization regions* in the shape-space U_l^{Σ}.

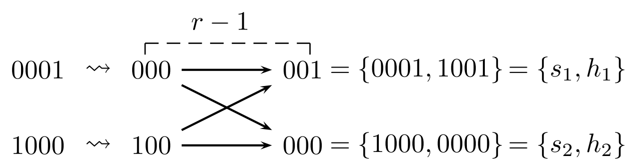


Fig. 2. Self elements $s_1 = 0001$ and $s_2 = 1000$ induce holes h_1, h_2, i.e. elements which are not detectable with r-contiguous and r-chunk matching rules for $r = 3$.

4 Permutation Masks

Permutation masks were proposed by Hofmeyr [3,9] for reducing the number of holes. A permutation mask is a bijective mapping π that specifies a reordering for all elements $a_i \in U_l^{\Sigma}$, i.e. $a_1 \to \pi(a_1), a_2 \to \pi(a_2), \ldots, a_{|\Sigma|^l} \to \pi(a_{|\Sigma|^l})$. More formally, a permutation $\pi \in S_n$, where $n \in \mathbb{N}$, can be written as a $2 \times n$ matrix, where the first row are elements $a_1, a_2, \ldots, a_n$ and the second row the new arrangement $\pi(a_1), \pi(a_2), \ldots, \pi(a_n)$, i.e.

$$\begin{pmatrix} a_1 & a_2 & \ldots & a_n \\ \pi(a_1) & \pi(a_2) & \ldots & \pi(a_n) \end{pmatrix}$$

For the sake of simplicity we will use the equivalent *cycle notation* [17] to specify a permutation. A permutation in cycle notation can be written as $(b_1\, b_2\, \ldots\, b_n)$ and means "b_1 becomes b_2, …, b_{n-1} becomes b_n, b_n becomes b_1. In addition, this notation allows the identity and non-cyclic mappings, for instance $(b_1)\,(b_2\, b_3)\,(b_4)$ means : $b_1 \to b_1$, $b_2 \to b_3$, $b_3 \to b_2$ and $b_4 \to b_4$.

[5] Hamming, r-contiguous, r-chunk and Rogers & Tanimoto matching rule

[6] the number of holes is controlled by the matching threshold r

4.1 Permutation Masks for Inducing other Holes

As explained above, a permutation mask is a bijective mapping and therefore can *increase* or *reduce* the number of holes — there also exists permutation masks which results in self elements which neither increase nor reduce the number of holes. The simplest example is the identity permutation mask.

For reducing the number of holes, π must be chosen at an appropriate value, and a certain number of detectors must be generable.

Reconsider the self elements $s_1 = 0001$, $s_2 = 1000$ in figure (2). One can see that elements $h_1 = 1001$ and $h_2 = 0000$ are not detectable by the r-contiguous and r-chunk matching rule. However, after applying the permutation mask $\pi_0 = (1\,2\,4\,3)$, i.e.

$$\pi_0(s_1) = 0010, \quad \pi_0(s_2) = 0100$$

one can verify (see Fig. 3) that holes h_1, h_2 are eliminated.

$$\pi_0(0001) \rightsquigarrow \overbrace{001 \longrightarrow 010}^{r-1} = \{0010\} = \{\pi_0(s_1)\}$$

$$\pi_0(1000) \rightsquigarrow 010 \longrightarrow 100 = \{0100\} = \{\pi_0(s_2)\}$$

Fig. 3. The permutated self elements $\pi_0(s_1)$ and $\pi_0(s_2)$ induce no holes by r-contiguous and r-chunk matching rule.

However, it is also clear that $(1\,2\,4\,3)\,(2\,4\,3\,1)$, $(4\,3\,1\,2)$ and $(3\,1\,2\,4)$ represent the same permutation, namely the cycle permutation of $\pi_0 = (1\,2\,4\,3)$. Specifically, all cycle permutations of an arbitrary selected π leads, in terms of the r-chunk and r-contiguous matching, to the same holes.

On the other hand, there do exist permutation masks which do not reduce holes, i.e. $\pi(s_i) = s_j$, for $i \neq j$ and self elements $s_1, s_2, \ldots, s_{|S|}$. An example is the permutation $\pi_1 = (14)(2)(3)$, as $\pi_1(s_1) = s_2$ and $\pi_1(s_2) = s_1$.

Furthermore, as mentioned above, a permutation mask can also increase the number of holes. In our subsequent presented experiments this is illustrated for instance in figures[7] 6(c) and 6(d) and more obviously illustrated in figure 4.

[7] with and without permutation mask

$$\begin{tikzcd} \pi_0(s_3) = \pi_0(0100) \rightsquigarrow 000 \arrow[r] \arrow[dr] & 001 = \{0001, 1001\} = \{s_1, h_1\} \\ \pi_0(s_4) = \pi_0(0010) \rightsquigarrow 100 \arrow[r] \arrow[ur] & 000 = \{1000, 0000\} = \{s_2, h_2\} \end{tikzcd}$$

Fig. 4. Let be $s_3 = 0100 = \pi_0(s_1)$ and $s_4 = 0010 = \pi_0(s_2)$. These two self elements induce no holes (see Fig. 3). The permutated self elements $\pi_0(s_3) = 0001 = s_1$ and $\pi_0(s_4) = 1000 = s_2$ induce two holes h_1 and h_2, although s_3 and s_4 induce no holes. This constructed example shows, that a permutation mask can also increase the number of holes.

5 Permutation Masks Experiments in Hamming Negative Selection

In [18,8] results were presented which demonstrated the coherence between the matching threshold r and generalization regions when the r-chunk matching rule in Hamming negative selection is applied. Recall, as holes are not detectable by any detector, holes must represent unseen self elements, or in other words holes must represent generalization regions. In the following experiment we will investigate how randomly determined permutation masks will influence the occurrence of holes (generalization regions). More specifically, we will empirically explore if holes occur in suitable generalization regions when a randomly determined permutation mask is applied. Finally, we explore empirically whether randomly determined permutation masks reduce the number of holes.

Stibor et al. [8] have shown in prior experiments that the matching threshold r is a crucial parameter and is inextricably linked to the input data being analyzed. However, permutation masks were not considered in [8]. In order to study the impact of permutation masks on generalization regions, and to obtain comparable results to previously performed experiments [8], we will utilize the same mapping function and data set. Furthermore, we will explore the impact of permutation masks on an additional data set (see Fig. 5).

5.1 Experiments Settings

The first self data set contains 1000 Gaussian ($\mu = 0.5, \sigma = 0.1$) generated points $p = (x, y) \in [0, 1]^2$. Each point p is mapped to a binary string

$$\underbrace{b_1, b_2, \ldots, b_8}_{b_x}, \underbrace{b_9, b_{10}, \ldots, b_{16}}_{b_y},$$

where the first 8 bits encode the integer x-value $i_x := \lceil 255 \cdot x + 0.5 \rceil$ and the last 8 bits the integer y-value $i_y := \lceil 255 \cdot y + 0.5 \rceil$, i.e.

$$[0, 1]^2 \rightarrow (i_x, i_y) \in [1, \ldots, 256 \times 1, \ldots, 256] \rightarrow (b_x, b_y) \in U_8^{\{0,1\}} \times U_8^{\{0,1\}}$$

This mapping is proposed in [18] and also utilized in [8] — it satisfies a straightforward visualization of real-valued encoded points in Hamming negative selection. The second data set (termed banana data set) is depicted in figure (5) and is a commonly used benchmark for anomaly detection problems [19]. The banana data set is taken from [20] and consists of 5300 points in total. These points are partitioned in two different classes, $\mathcal{C}_+$ which represents points inside the "banana-shape" and class $\mathcal{C}_-$ which contains points outside of the "banana-shape". In this experiment we have taken points from $\mathcal{C}_+$ only for simulating one self-region (similar to figure 1). More specifically, we have normalized with *min-max* method all points from $\mathcal{C}_+$ to the unitary square $[0,1]^2$. We then sampled 1000 random points from $\mathcal{C}_+$ and mapped those sampled points to bit-strings of length 16.

As the r-chunk matching rule subsumes the r-contiguous rule, i.e. recognize at least as many elements as the r-contiguous matching rule (see section 2.2), we have performed all experiments with the r-chunk matching rule. Furthermore, as proposed in [3,9] we have *randomly* determined permutation masks $\pi \in S_{16}$.

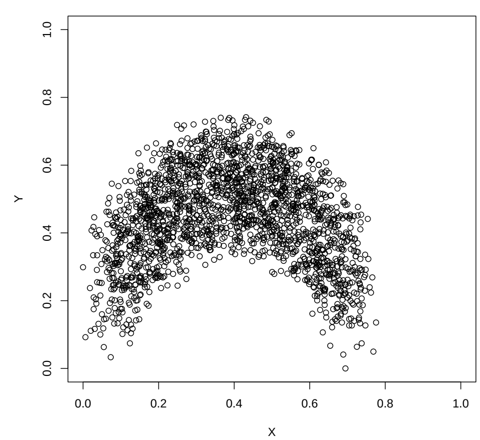


Fig. 5. Banana data set (points from class $\mathcal{C}_+$), *min-max* normalized to $[0,1]^2$. In an perfect case (error-less detection), the r-chunk detectors should cover regions outside the "banana" shape. The region within the "banana" shape is the generalization region and should consists of undetectable elements, i.e. holes and self elements.

5.2 Experimental Results

In figures (6,7,8,9) experimental results are presented. The black points represent the 1000 sampled self elements, the white points are holes, and the grey points represent areas which are covered by r-chunk detectors. It is not surprising that for both data sets, holes occur as they should in generalization regions when

$8 \leq r \leq 10$. This phenomena is discussed and explained in [8]. To summarize results from [8], a detector matching length which is not at least as long as the semantical representation of the underlying data — in this case 8 bits for x and y coordinates — results in incorrect generalization regions.

What is more interesting though, is the observation that a (randomly determined) permutation mask shatters the semantical representation of the underlying data (see Fig. 6-9 (b,d,f,h,j,l,n,p,r,t)) and therefore, holes are randomly distributed across the space instead of being concentrated inside or close to self regions. This observation also means that detectors are not covering areas around the self regions, instead they recognize elements which are also randomly distributed across the space. Furthermore one can see that the number of holes — when applying permutation masks (see Fig. 6-9 (b,d,f,h,j,l,n,p,r,t)) — is in some cases significantly higher than without permutation masks (see Fig. 6-9 (a,c,d,e,g,i,k,m,q,s)). This observation could be explained with the previous observation, that permutation masks distort the underlying data and therefore shatter self regions. As a consequence the underlying data is transformed into a collection of random chunks. For randomly determined self elements, Stibor et al. [6] showed that the number of holes increase exponentially for $r := l \to 0$.

Of course this shattering effect is linked very strongly to the mapping function employed. However it is clear that each permutation mask — except the identity permutation — semantically (more or less) distort the data. Furthermore, we believe that finding a permutation mask which does not significantly distort the semantical representation of the data may be computational intractable[8].

In order to obtain representative results, we performed 50 simulation runs, each with a randomly determined permutation mask for both data sets. Due to the lack of space to present all 50 simulation runs, we have selected two simulation results at random for each data set (see Fig. 6,7,8,9). The remaining simulation results are closely comparable to results in figures (6,7,8,9).

6 Conclusion

Lymphocyte diversity is an important property of the immune system for recognizing a huge amount of diverse substances. This property has been abstracted in terms of permutation masks in the Hamming negative selection detection technique. In this paper we have shown that (randomly determined) permutation masks in Hamming negative selection, distort the semantic meaning of the underlying data — the shape of the distribution — and as a consequence shatter self regions. Furthermore, the distorted data is transformed into a collection of random chunks. Hence, detectors are not covering areas around the self regions, instead they are randomly distributed across the space. Moreover the resulting holes (the generalization) occur in regions where actually no self regions should occur. Additionally we believe that it is computational infeasible to find permutation masks which correctly capture the semantical representation of the data

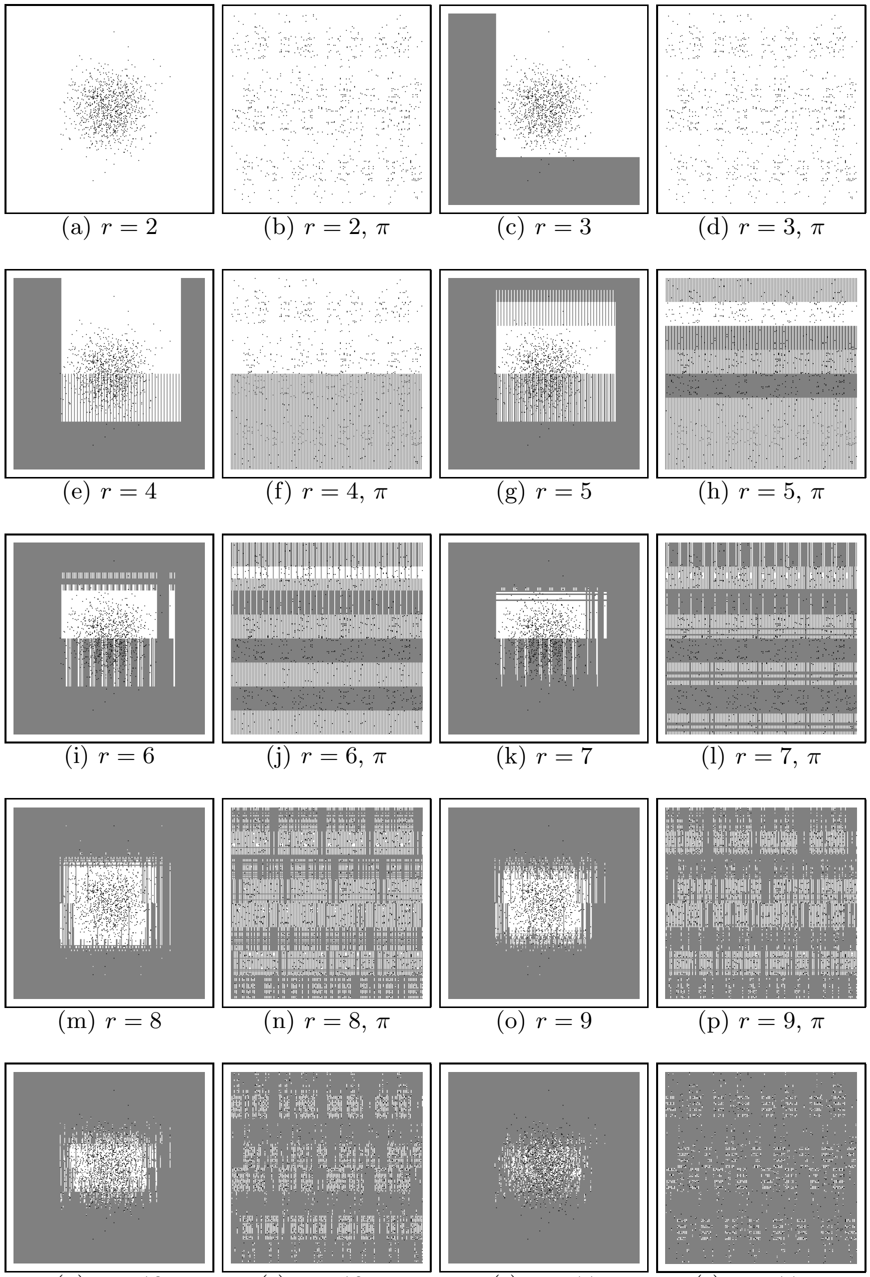
[8] in the worst-case, one have to check all $n!$ permutations of S_n

— if one exists at all. We conclude that the use of permutation masks casts doubt on the appropriateness of abstracting diversity in Hamming negative selection.

References

1. Forrest, S., Perelson, A.S., Allen, L., Cherukuri, R.: Self-nonself discrimination in a computer. In: Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy, IEEE Computer Society Press (1994)
2. Dasgupta, D., Forrest, S.: Novelty detection in time series data using ideas from immunology. In: Proceedings of the 5th International Conference on Intelligent Systems. (1996)
3. Hofmeyr, S.A.: An Immunological Model of Distributed Detection and its Application to Computer Security. PhD thesis, University of New Mexico (1999)
4. Singh, S.: Anomaly detection using negative selection based on the r-contiguous matching rule. In: Proceedings of the 1st International Conference on Artificial Immune Systems (ICARIS), Unversity of Kent at Canterbury Printing Unit (2002) 99–106
5. Kim, J., Bentley, P.J.: An evaluating of negative selection in an artificial immune system for network intrusion detection. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-2001. (2001) 1330–1337
6. Stibor, T., Timmis, J., Eckert, C.: On the appropriateness of negative selection defined over hamming shape-space as a network intrusion detection system. In: Congress On Evolutionary Computation – CEC 2005, IEEE Press (2005) 995–1002
7. Stibor, T., Timmis, J., Eckert, C.: The link between r-contiguous detectors and k-cnf satisfiability. In: Congress On Evolutionary Computation – CEC 2006, IEEE Press (2006 (to appear))
8. Stibor, T., Timmis, J., Eckert, C.: Generalization regions in hamming negative selection. In: Intelligent Information Processing and Web Mining. Advances in Soft Computing, Springer-Verlag (2006) 447–456
9. Hofmeyr, S., Forrest, S.: Architecture for an artificial immune system. Evolutionary Computation **8** (2000) 443–473
10. de Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer Verlag (2002)
11. Perelson, A.S., Oster, G.: Theoretical studies of clonal selection: minimal antibody repertoire size and reliability of self-nonself discrimination. In: J. Theor. Biol. Volume 81. (1979) 645–670
12. Percus, J.K., Percus, O.E., Perelson, A.S.: Predicting the size of the T-cell receptor and antibody combining region from consideration of efficient self-nonself discrimination. Proceedings of National Academy of Sciences USA **90** (1993) 1691–1695
13. Balthrop, J., Esponda, F., Forrest, S., Glickman, M.: Coverage and generalization in an artificial immune system. In: GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference, New York, Morgan Kaufmann Publishers (2002) 3–10
14. Esponda, F., Forrest, S., Helman, P.: A formal framework for positive and negative detection schemes. IEEE Transactions on Systems, Man and Cybernetics Part B: Cybernetics **34** (2004) 357–373
15. D'haeseleer, P., Forrest, S., Helman, P.: An immunological approach to change detection: algorithms, analysis, and implications. In: Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy, IEEE Computer Society, IEEE Computer Society Press (1996) 110–119

16. González, F., Dasgupta, D., Niño, L.F.: A randomized real-valued negative selection algorithm. In: Proceedings of the 2nd International Conference on Artificial Immune Systems (ICARIS). Volume 2787 of Lecture Notes in Computer Science., Edinburgh, UK, Springer-Verlag (2003) 261–272
17. Knuth, D.E.: The Art of Computer Programming. third edn. Volume 1. Addison-Wesley (2002)
18. González, F., Dasgupta, D., Gómez, J.: The effect of binary matching rules in negative selection. In: Genetic and Evolutionary Computation – GECCO-2003. Volume 2723 of Lecture Notes in Computer Science., Chicago, Springer-Verlag (2003) 195–206
19. Tax, D.M.J.: One-class classification. PhD thesis, Technische Universiteit Delft (2001)
20. Rätsch, G.: Benchmark repository (1998) http://ida.first.fraunhofer.de/projects/bench/benchmarks.htm.



(a) $r = 2$ (b) $r = 2$, π (c) $r = 3$ (d) $r = 3$, π

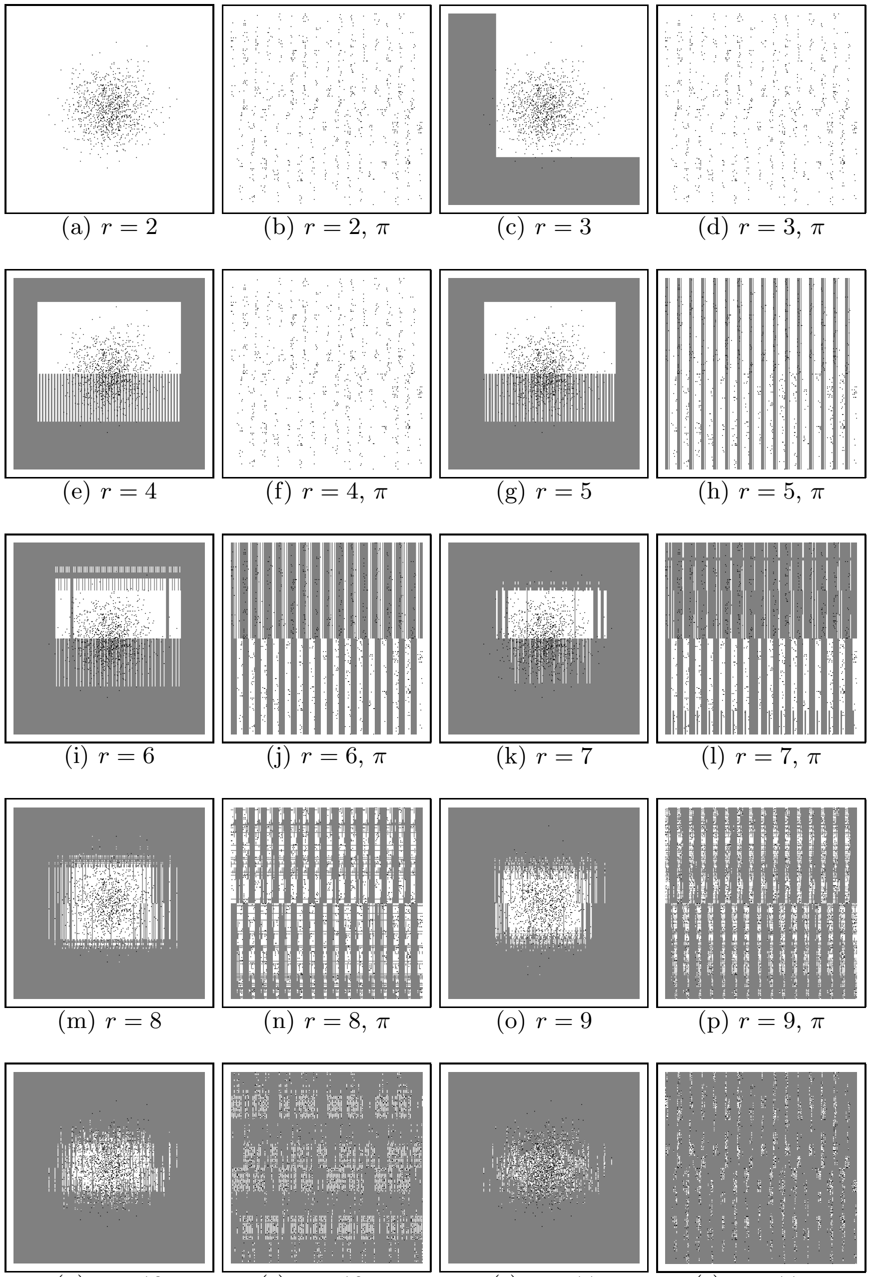
(e) $r = 4$ (f) $r = 4$, π (g) $r = 5$ (h) $r = 5$, π

(i) $r = 6$ (j) $r = 6$, π (k) $r = 7$ (l) $r = 7$, π

(m) $r = 8$ (n) $r = 8$, π (o) $r = 9$ (p) $r = 9$, π

(q) $r = 10$ (r) $r = 10$, π (s) $r = 11$ (t) $r = 11$, π

Fig. 6. A visualized simulation run, with 1000 random (self) points generated by a Gaussian distribution with mean $\mu = 0.5$ and variance $\sigma = 0.1$. The grey shaded area is covered by the generated r-chunk detectors, the white areas are holes. The black points are self elements. The captions which include a "π" are simulations results with the randomly determined permutation mask $\pi \in S_{16}$.



(a) $r = 2$ (b) $r = 2$, π (c) $r = 3$ (d) $r = 3$, π

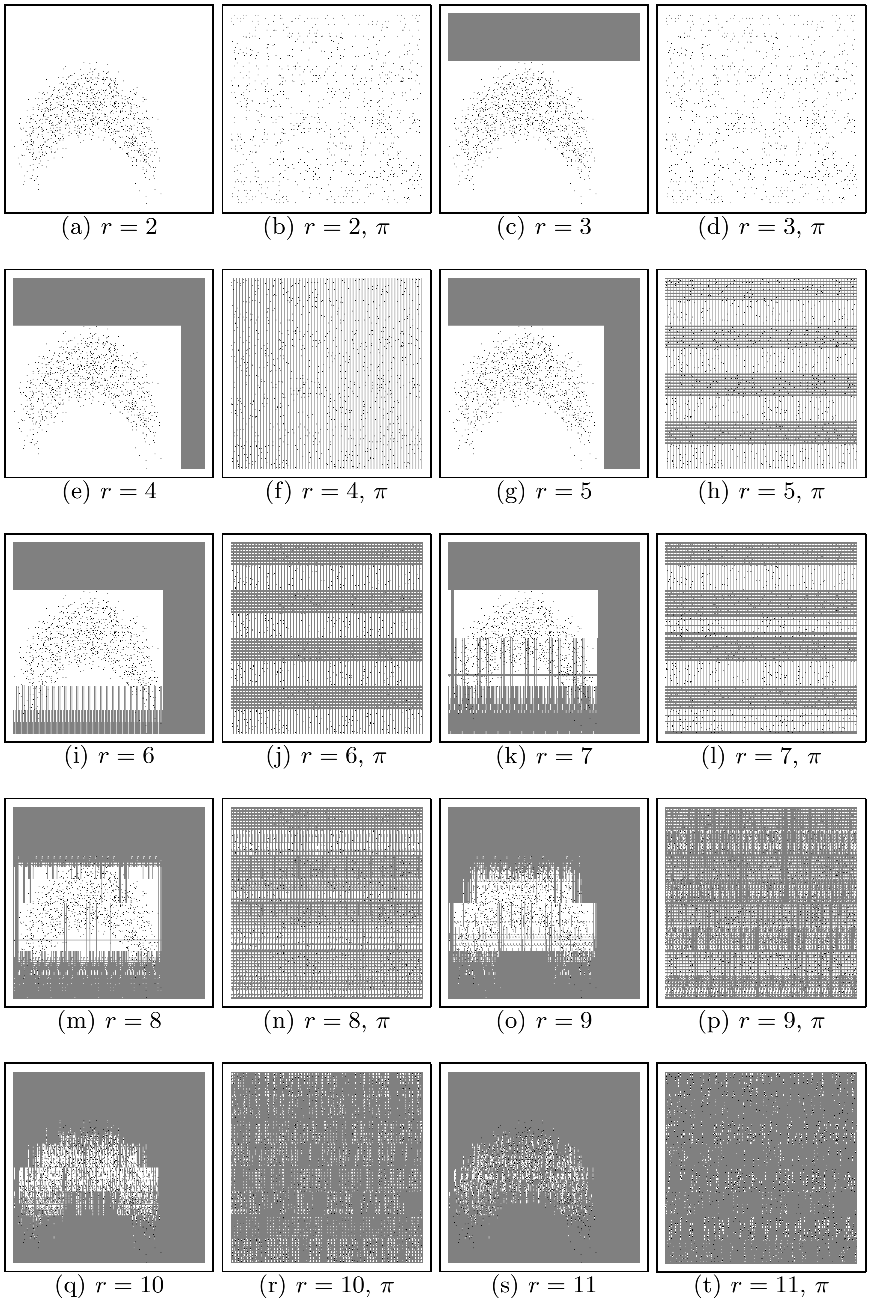
(e) $r = 4$ (f) $r = 4$, π (g) $r = 5$ (h) $r = 5$, π

(i) $r = 6$ (j) $r = 6$, π (k) $r = 7$ (l) $r = 7$, π

(m) $r = 8$ (n) $r = 8$, π (o) $r = 9$ (p) $r = 9$, π

(q) $r = 10$ (r) $r = 10$, π (s) $r = 11$ (t) $r = 11$, π

Fig. 7. An additional visualized simulation run, with 1000 random (self) points generated by a Gaussian distribution with mean $\mu = 0.5$ and variance $\sigma = 0.1$. The grey shaded area is covered by the generated r-chunk detectors, the white areas are holes. The black points are self elements. The captions which include a "π" are simulations results with the randomly determined permutation mask $\pi \in S_{16}$.



(a) $r = 2$ (b) $r = 2$, π (c) $r = 3$ (d) $r = 3$, π

(e) $r = 4$ (f) $r = 4$, π (g) $r = 5$ (h) $r = 5$, π

(i) $r = 6$ (j) $r = 6$, π (k) $r = 7$ (l) $r = 7$, π

(m) $r = 8$ (n) $r = 8$, π (o) $r = 9$ (p) $r = 9$, π

(q) $r = 10$ (r) $r = 10$, π (s) $r = 11$ (t) $r = 11$, π

Fig. 8. A visualized simulation run, 1000 randomly sampled (self) points from banana data set. The grey shaded area is covered by the generated r-chunk detectors, the white areas are holes. The black points are self elements. The captions which include a "π" are simulations results with the randomly determined permutation mask $\pi \in S_{16}$.

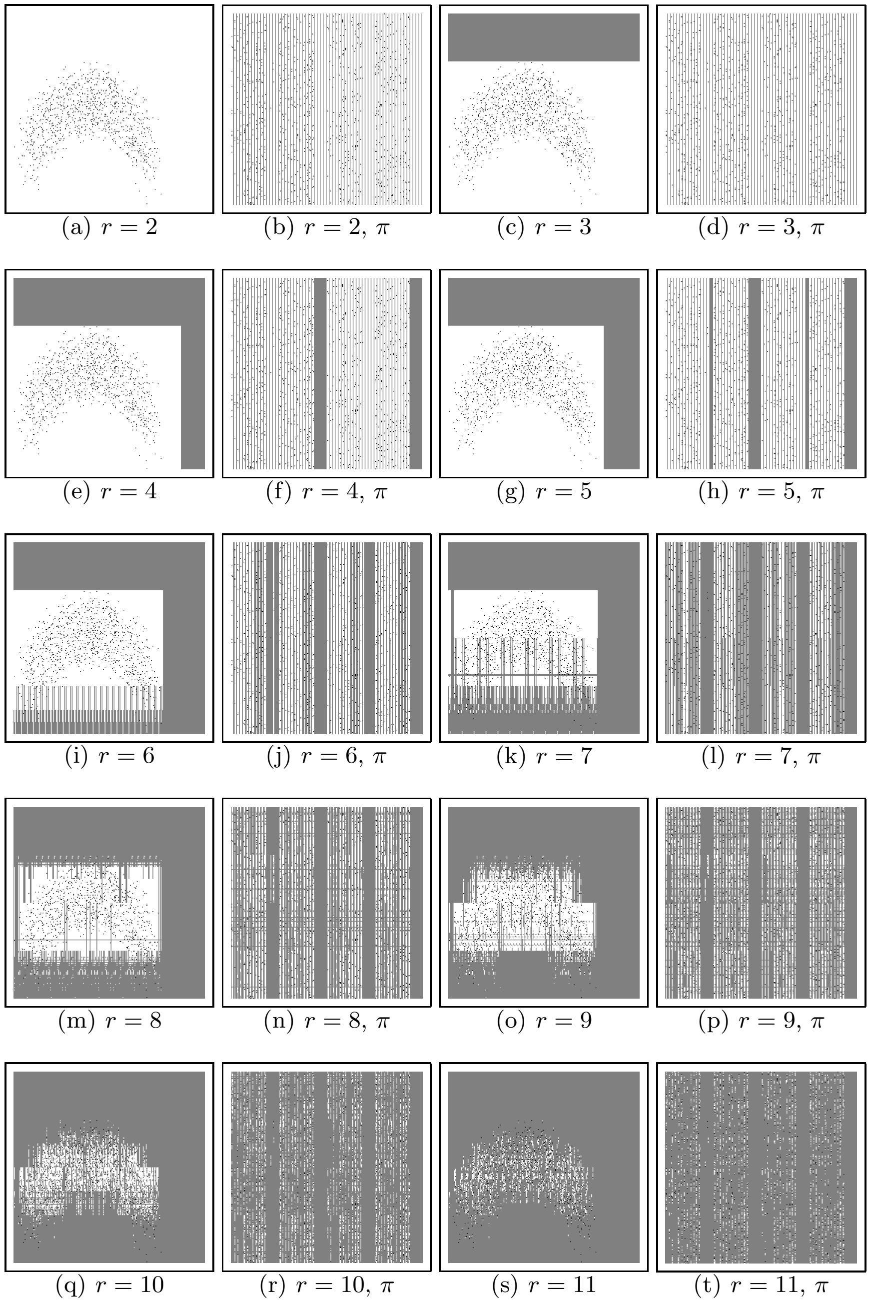


Fig. 9. An additional visualized simulation run, with 1000 randomly sampled (self) points from banana data set. The grey shaded area is covered by the generated r-chunk detectors, the white areas are holes. The black points are self elements. The captions which include a "π" are simulations results with the randomly determined permutation mask $\pi \in S_{16}$.