Society for
Mathematical
Biology

ORIGINAL ARTICLE

The Neural Ring: An Algebraic Tool for Analyzing the Intrinsic Structure of Neural Codes

**Carina Curto · Vladimir Itskov · Alan Veliz-Cuba ·
Nora Youngs**

Abstract Neurons in the brain represent external stimuli via neural codes. These codes often arise from stereotyped stimulus-response maps, associating to each neuron a convex receptive field. An important problem confronted by the brain is to infer properties of a represented stimulus space without knowledge of the receptive fields, using only the intrinsic structure of the neural code. How does the brain do this? To address this question, it is important to determine what stimulus space features can—in principle—be extracted from neural codes. This motivates us to define the *neural ring* and a related *neural ideal*, algebraic objects that encode the full combinatorial data of a neural code. Our main finding is that these objects can be expressed in a "canonical form" that directly translates to a minimal description of the receptive field structure intrinsic to the code. We also find connections to Stanley–Reisner rings, and use ideas similar to those in the theory of monomial ideals to obtain an algorithm for computing the primary decomposition of pseudo-monomial ideals. This allows us to algorithmically extract the canonical form associated to any neural code, providing the groundwork for inferring stimulus space features from neural activity alone.

1 Introduction

Building accurate representations of the world is one of the basic functions of the brain. It is well known that when a stimulus is paired with pleasure or pain, an animal quickly learns the association. Animals also learn, however, the (neutral) relationships between stimuli of the same type. For example, a bar held at a 45-degree angle

C. Curto (✉) · V. Itskov · A. Veliz-Cuba · N. Youngs
Department of Mathematics, University of Nebraska–Lincoln, Lincoln, USA
e-mail: ccurto2@math.unl.edu

appears more similar to one held at 50 degrees than to a perfectly vertical one. Upon hearing a triple of distinct pure tones, one seems to fall "in between" the other two. An explored environment is perceived not as a collection of disjoint physical locations, but as a spatial map. In summary, we do not experience the world as a stream of unrelated stimuli; rather, our brains organize different types of stimuli into highly structured *stimulus spaces*.

The relationship between neural activity and stimulus space structure has, nonetheless, received remarkably little attention. In the field of neural coding, much has been learned about the coding properties of individual neurons by investigating stimulus-response functions, such as place fields (O'Keefe and Dostrovsky 1971; McNaughton et al. 2006), orientation tuning curves (Watkins and Berkley 1974; Ben-Yishai et al. 1995), and other examples of "receptive fields" obtained by measuring neural activity in response to experimentally-controlled stimuli. Moreover, numerous studies have shown that neural activity, together with knowledge of the appropriate stimulus-response functions, can be used to accurately estimate a newly presented stimulus (Brown et al. 1998; Deneve et al. 1999; Ma et al. 2006). This paradigm is being actively extended and revised to include information present in populations of neurons, spurring debates on the role of correlations in neural coding (Nirenberg and Latham 2003; Averbeck et al. 2006; Schneidman et al. 2006a). In each case, however, the underlying structure of the stimulus space is assumed to be known, and is not treated as itself emerging from the activity of neurons. This approach is particularly problematic when one considers that the brain does *not* have access to stimulus-response functions, and must represent the world *without* the aid of dictionaries that lend meaning to neural activity (Curto and Itskov 2008). In coding theory parlance, the brain does not have access to the encoding map, and must therefore represent stimulus spaces via the intrinsic structure of the neural code.

How does the brain do this? In order to eventually answer this question, we must first tackle a simpler one:

Question *What can be inferred about the underlying stimulus space from neural activity alone*? That is, what stimulus space features are encoded in the *intrinsic structure* of the neural code, and can thus be extracted without knowing the individual stimulus-response functions?

Recently, we have shown that, in the case of hippocampal place cell codes, certain topological features of the animal's environment can be inferred from the neural code alone, without knowing the place fields (Curto and Itskov 2008). As will be explained in the next section, this information can be extracted from a simplicial complex associated to the neural code. *What other stimulus space features can be inferred from the neural code*? For this, we turn to algebraic geometry. Algebraic geometry provides a useful framework for inferring geometric and topological characteristics of spaces by associating rings of functions to these spaces. All relevant features of the underlying space are encoded in the intrinsic structure of the ring, where coordinate functions become indeterminates, and the space itself is defined in terms of ideals in the ring. Inferring features of a space from properties of functions—without specified

domains—is similar to the task confronted by the brain, so it is natural to expect that this framework may shed light on our question.

In this article, we introduce the *neural ring*, an algebro-geometric object that can be associated to any combinatorial neural code. Much like the simplicial complex of a code, the neural ring encodes information about the underlying stimulus space in a way that discards specific knowledge of receptive field maps, and thus gets closer to the essence of how the brain might represent stimulus spaces. Unlike the simplicial complex, the neural ring retains the full combinatorial data of a neural code, packaging this data in a more computationally tractable manner. We find that this object, together with a closely related *neural ideal*, can be used to algorithmically extract a compact, minimal description of the *receptive field structure* dictated by the code. This enables us to more directly tie combinatorial properties of neural codes to features of the underlying stimulus space, a critical step toward answering our motivating question.

Although the use of an algebraic construction such as the neural ring is quite novel in the context of neuroscience, the neural code (as we define it) is at its core a combinatorial object, and there is a rich tradition of associating algebraic objects to combinatorial ones (Miller and Sturmfels 2005). The most well-known example is perhaps the Stanley–Reisner ring (Stanley 2004), which turns out to be closely related to the neural ring. Within mathematical biology, associating polynomial ideals to combinatorial data has also been fruitful. Recent examples include inferring wiring diagrams in gene-regulatory networks (Jarrah et al. 2007; Veliz-Cuba 2012) and applications to chemical reaction networks (Shiu and Sturmfels 2010). Our work also has parallels to the study of design ideals in algebraic statistics (Pistone et al. 2001).

The organization of this paper is as follows. In Sect. 2, we introduce *receptive field codes*, and explore how the requirement of convexity enables these codes to constrain the structure of the underlying stimulus space. In Sect. 3, we define the neural ring and the neural ideal, and find explicit relations that enable us to compute these objects for any neural code. Section 4 is the heart of this paper. Here, we present an alternative set of relations for the neural ring, and demonstrate how they enable us to "read off" receptive field structure from the neural ideal. We then introduce pseudo-monomials and pseudo-monomial ideals, by analogy to monomial ideals; this allows us to define a natural "canonical form" for the neural ideal. Using this, we can extract minimal relationships among receptive fields that are dictated by the structure of the neural code. Finally, we present an algorithm for finding the canonical form of a neural ideal, and illustrate how to use our formalism for inferring receptive field structure in a detailed example. Section 5 describes the primary decomposition of the neural ideal and, more generally, of pseudo-monomial ideals. Computing the primary decomposition of the neural ideal is a critical step in our canonical form algorithm, and it also yields a natural decomposition of the neural code in terms of intervals of the Boolean lattice. We end this section with an algorithm for finding the primary decomposition of any pseudo-monomial ideal, using ideas similar to those in the theory of square-free monomial ideals. All longer proofs can be found in Appendix 1. A detailed classification of neural codes on three neurons is given in Appendix 2.

2 Background and Motivation

2.1 Preliminaries

In this section, we introduce the basic objects of study: neural codes, receptive field codes, and convex receptive field codes. We then discuss various ways in which the structure of a convex receptive field code can constrain the underlying stimulus space. These constraints emerge most obviously from the simplicial complex of a neural code, but (as will be made clear) there are also constraints that arise from aspects of a neural code's structure that go well beyond what is captured by the simplicial complex of the code.

Given a set of neurons labeled $\{1, \ldots, n\} \overset{\text{def}}{=} [n]$, we define a *neural code* $\mathcal{C} \subset \{0, 1\}^n$ as a set of binary patterns of neural activity. An element of a neural code is called a *codeword*, $c = (c_1, \ldots, c_n) \in \mathcal{C}$, and corresponds to a subset of neurons

$$\text{supp}(c) \overset{\text{def}}{=} \{i \in [n] \mid c_i = 1\} \subset [n].$$

Similarly, the entire code $\mathcal{C}$ can be identified with a set of subsets of neurons,

$$\text{supp}\,\mathcal{C} \overset{\text{def}}{=} \{\text{supp}(c) \mid c \in \mathcal{C}\} \subset 2^{[n]},$$

where $2^{[n]}$ denotes the set of all subsets of $[n]$. Because we discard the details of the precise timing and/or rate of neural activity, what we mean by neural code is often referred to in the neural coding literature as a *combinatorial code* (Schneidman et al. 2006b; Osborne et al. 2008).

A set of subsets $\Delta \subset 2^{[n]}$ is an (abstract) *simplicial complex* if $\sigma \in \Delta$ and $\tau \subset \sigma$ implies $\tau \in \Delta$. We will say that a neural code $\mathcal{C}$ is a simplicial complex if $\text{supp}\,\mathcal{C}$ is a simplicial complex. In cases where the code is *not* a simplicial complex, we can complete the code to a simplicial complex by simply adding in missing subsets of codewords. This allows us to define the *simplicial complex of the code* as

$$\Delta(\mathcal{C}) \overset{\text{def}}{=} \{\sigma \subset [n] \mid \sigma \subseteq \text{supp}(c) \text{ for some } c \in \mathcal{C}\}.$$

Clearly, $\Delta(\mathcal{C})$ is the smallest simplicial complex that contains $\text{supp}\,\mathcal{C}$.

2.2 Receptive Field Codes (RF Codes)

Neurons in many brain areas have activity patterns that can be characterized by receptive fields.[1] Abstractly, a *receptive field* is a map $f_i : X \to \mathbb{R}_{\geq 0}$ from a space of stimuli, X, to the average firing rate of a single neuron, i, in response to each stimulus. Receptive fields are computed by correlating neural responses to independently measured external stimuli. We follow a common abuse of language, where both the map and its support (i.e., the subset $U_i \subset X$ where f_i takes

---

[1]In the vision literature, the term "receptive field" is reserved for subsets of the visual field; we use the term in a more general sense, applicable to any modality.
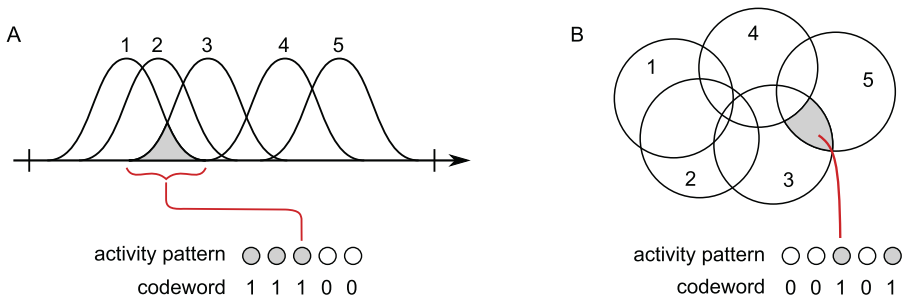
Fig. 1 Receptive field overlaps determine codewords in 1D and 2D RF codes. (**A**) Neurons in a 1D RF code have receptive fields that overlap on a *line segment* (or *circle*, in the case of orientation-tuning). Each stimulus on the *line* corresponds to a binary codeword. Gaussians depict graded firing rates for neural responses; this additional information is discarded by the RF code. (**B**) Neurons in a 2D RF code, such as a place field code, have receptive fields that partition a two-dimensional stimulus space into nonoverlapping regions, as illustrated by the shaded area. All stimuli within one of these regions will activate the same set of neurons, and hence have the same corresponding codeword

on positive values) are referred to as "receptive fields." *Convex* receptive fields are convex[2] subsets of the stimulus space, for $X \subset \mathbb{R}^d$. The paradigmatic examples are orientation-selective neurons in visual cortex (Watkins and Berkley 1974; Ben-Yishai et al. 1995) and hippocampal place cells (O'Keefe and Dostrovsky 1971; McNaughton et al. 2006). Orientation-selective neurons have *tuning curves* that reflect a neuron's preference for a particular angle (see Fig. 1A). Place cells are neurons that have *place fields*; i.e., each neuron has a preferred (convex) region of the animal's physical environment where it has a high firing rate (see Fig. 1B). Both tuning curves and place fields are examples of receptive fields.

A receptive field code (RF code) is a neural code that corresponds to the brain's representation of the stimulus space covered by the receptive fields. When a stimulus lies in the intersection of several receptive fields, the corresponding neurons may cofire while the rest remain silent. The active subset σ of neurons can be identified with a binary codeword $c \in \{0, 1\}^n$ via $\sigma = \mathrm{supp}(c)$. Unless otherwise noted, a stimulus space X need only be a topological space. However, we usually have in mind $X \subset \mathbb{R}^d$, and this becomes important when we consider *convex* RF codes.

Definition Let X be a stimulus space (e.g., $X \subset \mathbb{R}^d$), and let $\mathcal{U} = \{U_1, \ldots, U_n\}$ be a collection of open sets, with each $U_i \subset X$ the receptive field of the ith neuron in a population of n neurons. The *receptive field code (RF code)* $\mathcal{C}(\mathcal{U}) \subset \{0, 1\}^n$ is the set of all binary codewords corresponding to stimuli in X:

$$
\mathcal{C}(\mathcal{U}) \overset{\text{def}}{=} \left\{ c \in \{0, 1\}^n \;\middle|\; \left(\bigcap_{i \in \mathrm{supp}(c)} U_i \right) \setminus \left(\bigcup_{j \notin \mathrm{supp}(c)} U_j \right) \neq \emptyset \right\}.
$$

[2]A subset $B \subset \mathbb{R}^n$ is *convex* if, given any pair of points $x, y \in B$, the point $z = tx + (1-t)y$ is contained in B for any $t \in [0, 1]$.

If $X \subset \mathbb{R}^d$ and each of the U_is is also a *convex* subset of X, then we say that $\mathcal{C}(\mathcal{U})$ is a *convex* RF code.

Our convention is that the empty intersection is $\bigcap_{i \in \emptyset} U_i = X$, and the empty union is $\bigcup_{i \in \emptyset} U_i = \emptyset$. This means that if $\bigcup_{i=1}^{n} U_i \subsetneq X$, then $\mathcal{C}(\mathcal{U})$ includes the all-zeros codeword corresponding to an "outside" point not covered by the receptive fields; on the other hand, if $\bigcap_{i=1}^{n} U_i \neq \emptyset$, then $\mathcal{C}(\mathcal{U})$ includes the all-ones codeword. Figure 1 shows examples of convex receptive fields covering one- and two-dimensional stimulus spaces, and examples of codewords corresponding to regions defined by the receptive fields.

Returning to our discussion in the Introduction, we have the following question: If we can assume $\mathcal{C} = \mathcal{C}(\mathcal{U})$ is a RF code, then *what can be learned about the underlying stimulus space X from knowledge only of $\mathcal{C}$, and not of $\mathcal{U}$?* The answer to this question will depend critically on whether or not we can assume that the RF code is convex. In particular, if we do not assume convexity of the receptive fields, then any code can be realized as a RF code in any dimension.

Lemma 2.1 *Let $\mathcal{C} \subset \{0, 1\}^n$ be a neural code. Then, for any $d \geq 1$, there exists a stimulus space $X \subset \mathbb{R}^d$ and a collection of open sets $\mathcal{U} = \{U_1, \ldots, U_n\}$ (not necessarily convex), with $U_i \subset X$ for each $i \in [n]$, such that $\mathcal{C} = \mathcal{C}(\mathcal{U})$.*

Proof Let $\mathcal{C} \subset \{0, 1\}^n$ be any neural code, and order the elements of $\mathcal{C}$ as $\{c_1, \ldots, c_m\}$, where $m = |\mathcal{C}|$. For each $c \in \mathcal{C}$, choose a distinct point $x_c \in \mathbb{R}^d$ and an open neighborhood N_c of x_c such that no two neighborhoods intersect. Define $U_j \stackrel{\text{def}}{=} \bigcup_{j \in \text{supp}(c_k)} N_{c_k}$, let $\mathcal{U} = \{U_1, \ldots, U_n\}$, and $X = \bigcup_{i=1}^{m} N_{c_i}$. Observe that if the all-zeros codeword is in $\mathcal{C}$, then $N_0 = X \setminus \bigcup_{i=1}^{n} U_i$ corresponds to the "outside point" not covered by any of the U_is. By construction, $\mathcal{C} = \mathcal{C}(\mathcal{U})$. $\qquad \square$

Although any neural code $\mathcal{C} \subseteq \{0, 1\}^n$ can be realized as a RF code, it is *not* true that any code can be realized as a *convex* RF code. Counterexamples can be found in codes having as few as three neurons.

Lemma 2.2 *The neural code $\mathcal{C} = \{0, 1\}^3 \setminus \{111, 001\}$ on three neurons cannot be realized as a convex RF code.*

Proof Assume the converse, and let $\mathcal{U} = \{U_1, U_2, U_3\}$ be a set of convex open sets in $\mathbb{R}^d$ such that $\mathcal{C} = \mathcal{C}(\mathcal{U})$. The code necessitates that $U_1 \cap U_2 \neq \emptyset$ (since $110 \in \mathcal{C}$), $(U_1 \cap U_3) \setminus U_2 \neq \emptyset$ (since $101 \in \mathcal{C}$), and $(U_2 \cap U_3) \setminus U_1 \neq \emptyset$ (since $011 \in \mathcal{C}$). Let $p_1 \in (U_1 \cap U_3) \setminus U_2$ and $p_2 \in (U_2 \cap U_3) \setminus U_1$. Since $p_1, p_2 \in U_3$ and U_3 is convex, the line segment $\ell = (1 - t)p_1 + tp_2$ for $t \in [0, 1]$ must also be contained in U_3. There are just two possibilities. Case 1: ℓ passes through $U_1 \cap U_2$ (see Fig. 2, left). This implies $U_1 \cap U_2 \cap U_3 \neq \emptyset$, and hence $111 \in \mathcal{C}$, a contradiction. Case 2: ℓ does not intersect $U_1 \cap U_2$. Since U_1, U_2 are open sets, this implies ℓ passes outside of $U_1 \cup U_2$ (see Fig. 2, right), and hence $001 \in \mathcal{C}$, a contradiction. $\qquad \square$

Fig. 2 Two cases in the proof of Lemma 2.2
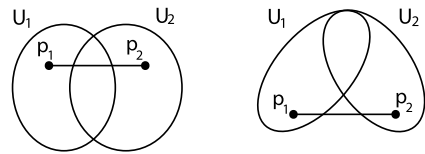
2.3 Stimulus Space Constraints Arising from Convex RF Codes

It is clear from Lemma 2.1 that there is essentially no constraint on the stimulus space for realizing a code as a RF code. However, if we demand that $\mathcal{C}$ is a *convex RF code*, then the overlap structure of the U_is sharply constrains the geometric and topological properties of the underlying stimulus space X. To see how this works, we first consider the simplicial complex of a neural code, $\Delta(\mathcal{C})$. Classical results in convex geometry and topology provide constraints on the underlying stimulus space X for convex RF codes, based on the structure of $\Delta(\mathcal{C})$. We will discuss these next. We then turn to the question of constraints that arise from combinatorial properties of a neural code $\mathcal{C}$ that are *not* captured by $\Delta(\mathcal{C})$.

2.3.1 Helly's Theorem and the Nerve Theorem

Here, we briefly review two classical and well-known theorems in convex geometry and topology, Helly's theorem and the Nerve theorem, as they apply to convex RF codes. Both theorems can be used to relate the structure of the simplicial complex of a code, $\Delta(\mathcal{C})$, to topological features of the underlying stimulus space X.

Suppose $\mathcal{U} = \{U_1, \ldots, U_n\}$ is a finite collection of convex open subsets of $\mathbb{R}^d$, with dimension $d < n$. We can associate to $\mathcal{U}$ a simplicial complex $N(\mathcal{U})$ called the *nerve* of $\mathcal{U}$. A subset $\{i_1, \ldots, i_k\} \subset [n]$ belongs to $N(\mathcal{U})$ if and only if the appropriate intersection $\bigcap_{\ell=1}^{k} U_{i_\ell}$ is nonempty. If we think of the U_is as receptive fields, then $N(\mathcal{U}) = \Delta(\mathcal{C}(\mathcal{U}))$. In other words, the nerve of the cover corresponds to the simplicial complex of the associated (convex) RF code.

Helly's Theorem *Consider k convex subsets, $U_1, \ldots, U_k \subset \mathbb{R}^d$, for $d < k$. If the intersection of every $d+1$ of these sets is nonempty, then the full intersection $\bigcap_{i=1}^{k} U_i$ is also nonempty.*

A nice exposition of this theorem and its consequences can be found in Danzer et al. (1963). One straightforward consequence is that the nerve $N(\mathcal{U})$ is completely determined by its d-skeleton, and corresponds to the largest simplicial complex with that d-skeleton. For example, if $d = 1$, then $N(\mathcal{U})$ is a clique complex (fully determined by its underlying graph). Since $N(\mathcal{U}) = \Delta(\mathcal{C}(\mathcal{U}))$, Helly's theorem imposes constraints on the minimal dimension of the stimulus space X when $\mathcal{C} = \mathcal{C}(\mathcal{U})$ is assumed to be a convex RF code.

Nerve Theorem *The homotopy type of $X(\mathcal{U}) \stackrel{\text{def}}{=} \bigcup_{i=1}^{n} U_i$ is equal to the homotopy type of the nerve of the cover, $N(\mathcal{U})$. In particular, $X(\mathcal{U})$ and $N(\mathcal{U})$ have exactly the same homology groups.*

Fig. 3 Four arrangements of
three convex receptive fields,
$\mathcal{U} = \{U_1, U_2, U_3\}$, each having
$\Delta(\mathcal{C}(\mathcal{U})) = 2^{[3]}$. *Square boxes*
denote the stimulus space X in
cases where $U_1 \cup U_2 \cup U_3 \subsetneq X$.
(**A**) $\mathcal{C}(\mathcal{U}) = 2^{[3]}$, including the
all-zeros codeword 000.
(**B**) $\mathcal{C}(\mathcal{U}) = \{111, 101, 011,$
$001\}$, with $X = U_3$. (**C**) $\mathcal{C}(\mathcal{U}) =$
$\{111, 011, 001, 000\}$.
(**D**) $\mathcal{C}(\mathcal{U}) = \{111, 101, 011, 110,$
$100, 010\}$, and $X = U_1 \cup U_2$.
The minimal embedding
dimension for the codes in
panels (**A**) and (**D**) is $d = 2$,
while for panels (**B**) and (**C**) it is
$d = 1$
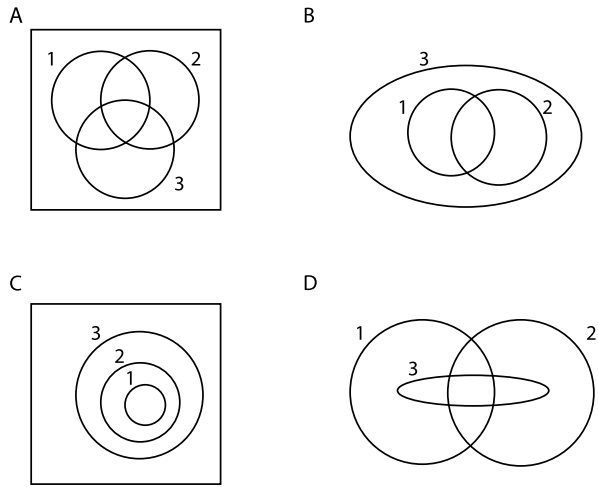
The Nerve theorem is an easy consequence of Hatcher (2002, Corollary 4G.3). This is a powerful theorem relating the simplicial complex of a RF code, $\Delta(\mathcal{C}(\mathcal{U})) = N(\mathcal{U})$, to topological features of the underlying space, such as homology groups and other homotopy invariants. Note, however, that the similarities between $X(\mathcal{U})$ and $N(\mathcal{U})$ only go so far. In particular, $X(\mathcal{U})$ and $N(\mathcal{U})$ typically have very different dimension. It is also important to keep in mind that the Nerve theorem concerns the topology of $X(\mathcal{U}) = \bigcup_{i=1}^n U_i$. In our setup, if the stimulus space X is larger, so that $\bigcup_{i=1}^n U_i \subsetneq X$, then the Nerve theorem tells us only about the homotopy type of $X(\mathcal{U})$, not of X. Since the U_i are open sets, however, conclusions about the dimension of X can still be inferred.

In addition to Helly's theorem and the Nerve theorem, there is a great deal known about $\Delta(\mathcal{C}(\mathcal{U})) = N(\mathcal{U})$ for collections of convex sets in $\mathbb{R}^d$. In particular, the f-vectors of such simplicial complexes have been completely characterized by Kalai (1984, 1986).

2.3.2 Beyond the Simplicial Complex of the Neural Code

We have just seen how the simplicial complex of a neural code, $\Delta(\mathcal{C})$, yields constraints on the stimulus space X if we assume $\mathcal{C}$ can be realized as a convex RF code. The example described in Lemma 2.2, however, implies that other kinds of constraints on X may emerge from the combinatorial structure of a neural code, even if there is no obstruction stemming from $\Delta(\mathcal{C})$.

In Fig. 3, we show four possible arrangements of three convex receptive fields in the plane. Each convex RF code has the same corresponding simplicial complex $\Delta(\mathcal{C}) = 2^{[3]}$, since $111 \in \mathcal{C}$ for each code. Nevertheless, the arrangements clearly have different combinatorial properties. In Fig. 3C, for instance, we have $U_1 \subset U_2 \subset U_3$, while Fig. 3A has no special containment relationships among the receptive fields. This "receptive field structure" (RF structure) of the code has implications for the underlying stimulus space.

Let d be the minimal integer for which the code can be realized as a convex RF code in $\mathbb{R}^d$; we will refer to this as the *minimal embedding dimension* of $\mathcal{C}$. Note that the codes in Fig. 3A, D have $d = 2$, whereas the codes in Fig. 3B, C have $d = 1$. The simplicial complex, $\Delta(\mathcal{C})$, is thus not sufficient to determine the minimal embedding dimension of a convex RF code, but this information *is* somehow present in the RF structure of the code. Similarly, in Lemma 2.2 we saw that $\Delta(\mathcal{C})$ does not provide sufficient information to determine whether or not $\mathcal{C}$ can be realized as a convex RF code; after working out the RF structure, however, it was easy to see that the given code was not realizable.

2.3.3 The Receptive Field Structure (RF Structure) of a Neural Code

As we have just seen, the intrinsic structure of a neural code contains information about the underlying stimulus space that cannot be inferred from the simplicial complex of the code alone. This information is, however, present in what we have loosely referred to as the "RF structure" of the code. We now explain more carefully what we mean by this term.

Given a set of receptive fields $\mathcal{U} = \{U_1, \ldots, U_n\}$ in a stimulus space X, there are certain containment relations between intersections and unions of the U_is that are "obvious," and carry no information about the particular arrangement in question. For example, $U_1 \cap U_2 \subseteq U_2 \cup U_3 \cup U_4$ is always guaranteed to be true, because it follows from $U_2 \subseteq U_2$. On the other hand, a relationship such as $U_3 \subseteq U_1 \cup U_2$ (as in Fig. 3D) is *not* always present, and thus reflects something about the structure of a particular receptive field arrangement.

Let $\mathcal{C} \subset \{0, 1\}^n$ be a neural code, and let $\mathcal{U} = \{U_1, \ldots, U_n\}$ be any arrangement of receptive fields in a stimulus space X such that $\mathcal{C} = \mathcal{C}(\mathcal{U})$ (this is guaranteed to exist by Lemma 2.1). The *RF structure* of $\mathcal{C}$ refers to the set of relations among the U_is that are not "obvious," and have the form:

$$\bigcap_{i \in \sigma} U_i \subseteq \bigcup_{j \in \tau} U_j, \quad \text{for } \sigma \cap \tau = \emptyset.$$

In particular, this includes any empty intersections $\bigcap_{i \in \sigma} U_i = \emptyset$ (here $\tau = \emptyset$). In the Fig. 3 examples, the panel A code has no RF structure relations; while panel B has $U_1 \subset U_3$ and $U_2 \subset U_3$; panel C has $U_1 \subset U_2 \subset U_3$; and panel D has $U_3 \subset U_1 \cup U_2$.

The central goal of this paper is to develop a method to algorithmically extract a minimal description of the RF structure directly from a neural code $\mathcal{C}$, without first realizing it as $\mathcal{C}(\mathcal{U})$ for some arrangement of receptive fields. We view this as a first step toward inferring stimulus space features that cannot be obtained from the simplicial complex $\Delta(\mathcal{C})$. To do this we turn to an algebro-geometric framework, that of neural rings and ideals. These objects are defined in Sect. 3 so as to capture the full combinatorial data of a neural code, but in a way that allows us to naturally and algorithmically infer a compact description of the desired RF structure, as shown in Sect. 4.

3 Neural Rings and Ideals

In this section, we define the neural ring R_C and a closely-related neural ideal, J_C. First, we briefly review some basic algebraic geometry background needed throughout this paper.

3.1 Basic Algebraic Geometry Background

The following definitions are standard (see, for example, Cox et al. 1997).

Rings and Ideals   Let R be a commutative ring. A subset $I \subseteq R$ is an *ideal* of R if it has the following properties:

(i) I is a subgroup of R under addition.
(ii) If $a \in I$, then $ra \in I$ for all $r \in R$.

An ideal I is said to be *generated by* a set A, and we write $I = \langle A \rangle$, if

$$I = \{r_1 a_1 + \cdots + r_n a_n \mid a_i \in A, r_i \in R, \text{ and } n \in \mathbb{N}\}.$$

In other words, I is the set of all finite combinations of elements of A with coefficients in R.

An ideal $I \subset R$ is *proper* if $I \subsetneq R$. An ideal $I \subset R$ is *prime* if it is proper and satisfies: if $rs \in I$ for some $r, s \in R$, then $r \in I$ or $s \in I$. An ideal $m \subset R$ is *maximal* if it is proper and for any ideal I such that $m \subseteq I \subseteq R$, either $I = m$ or $I = R$. An ideal $I \subset R$ is *radical* if $r^n \in I$ implies $r \in I$, for any $r \in R$ and $n \in \mathbb{N}$. An ideal $I \subset R$ is *primary* if $rs \in I$ implies $r \in I$ or $s^n \in I$ for some $n \in \mathbb{N}$. A *primary decomposition* of an ideal I expresses I as an intersection of finitely many primary ideals.

Ideals and Varieties   Let k be a field, n the number of neurons, and $k[x_1, \ldots, x_n]$ a polynomial ring with one indeterminate x_i for each neuron. We will consider k^n to be the neural activity space, where each point $v = (v_1, \ldots, v_n) \in k^n$ is a vector tracking the state v_i of each neuron. Note that any polynomial $f \in k[x_1, \ldots, x_n]$ can be evaluated at a point $v \in k^n$ by setting $x_i = v_i$ each time x_i appears in f. We will denote this value $f(v)$.

Let $J \subset k[x_1, \ldots, x_n]$ be an ideal, and define the variety

$$V(J) \stackrel{\text{def}}{=} \{v \in k^n \mid f(v) = 0 \text{ for all } f \in J\}.$$

Similarly, given a subset $S \subset k^n$, we can define the ideal of functions that vanish on this subset as

$$I(S) \stackrel{\text{def}}{=} \{f \in k[x_1, \ldots, x_n] \mid f(v) = 0 \text{ for all } v \in S\}.$$

The ideal-variety correspondence (Cox et al. 1997) gives us the usual order-reversing relationships: $I \subseteq J \Rightarrow V(J) \subseteq V(I)$, and $S \subseteq T \Rightarrow I(T) \subseteq I(S)$. Furthermore, $V(I(V)) = V$ for any variety V, but it is not always true that $I(V(J)) = J$ for an ideal J (see Sect. 6.1). We will regard neurons as having only two states, "on" or "off," and thus choose $k = \mathbb{F}_2 = \{0, 1\}$.

3.2 Definition of the Neural Ring

Let $\mathcal{C} \subset \{0, 1\}^n = \mathbb{F}_2^n$ be a neural code, and define the ideal $I_{\mathcal{C}}$ of $\mathbb{F}_2[x_1, \ldots, x_n]$ corresponding to the set of polynomials that vanish on all codewords in $\mathcal{C}$:

$$I_{\mathcal{C}} \overset{\text{def}}{=} I(\mathcal{C}) = \big\{ f \in \mathbb{F}_2[x_1, \ldots, x_n] \,\big|\, f(c) = 0 \text{ for all } c \in \mathcal{C} \big\}.$$

By design, $V(I_{\mathcal{C}}) = \mathcal{C}$ and hence $I(V(I_{\mathcal{C}})) = I_{\mathcal{C}}$. Note that the ideal generated by the *Boolean relations*,

$$\mathcal{B} \overset{\text{def}}{=} \big\langle x_1^2 - x_1, \ldots, x_n^2 - x_n \big\rangle,$$

is automatically contained in $I_{\mathcal{C}}$, irrespective of $\mathcal{C}$.

The *neural ring* $R_{\mathcal{C}}$ corresponding to the code $\mathcal{C}$ is the quotient ring

$$R_{\mathcal{C}} \overset{\text{def}}{=} \mathbb{F}_2[x_1, \ldots, x_n]/I_{\mathcal{C}},$$

together with the set of indeterminates $x_1, \ldots, x_n$. We say that two neural rings are *equivalent* if there is a bijection between the sets of indeterminates that yields a ring homomorphism.

Remark Due to the Boolean relations, any element $y \in R_{\mathcal{C}}$ satisfies $y^2 = y$ (cross-terms vanish because $2 = 0$ in $\mathbb{F}_2$), so the neural ring is a *Boolean ring* isomorphic to $\mathbb{F}_2^{|\mathcal{C}|}$. It is important to keep in mind, however, that $R_{\mathcal{C}}$ comes equipped with a privileged set of functions, $x_1, \ldots, x_n$; this allows the ring to keep track of considerably more structure than just the size of the neural code.

3.3 The Spectrum of the Neural Ring

We can think of $R_{\mathcal{C}}$ as the ring of functions of the form $f : \mathcal{C} \to \mathbb{F}_2$ on the neural code, where each function assigns a 0 or 1 to each codeword $c \in \mathcal{C}$ by evaluating $f \in \mathbb{F}_2[x_1, \ldots, x_n]/I_{\mathcal{C}}$ through the substitutions $x_i = c_i$ for $i = 1, \ldots, n$. Quotienting the original polynomial ring by $I_{\mathcal{C}}$ ensures that there is only one zero function in $R_{\mathcal{C}}$. The spectrum of the neural ring, $\text{Spec}(R_{\mathcal{C}})$, consists of all prime ideals in $R_{\mathcal{C}}$. We will see shortly that the elements of $\text{Spec}(R_{\mathcal{C}})$ are in one-to-one correspondence with the elements of the neural code $\mathcal{C}$. Indeed, our definition of $R_{\mathcal{C}}$ was designed for this to be true.

For any point $v \in \{0, 1\}^n$ of the neural activity space, let

$$m_v \overset{\text{def}}{=} I(v) = \big\{ f \in \mathbb{F}_2[x_1, \ldots, x_n] \,\big|\, f(v) = 0 \big\}$$

be the maximal ideal of $\mathbb{F}_2[x_1, \ldots, x_n]$ consisting of all functions that vanish on v. We can also write $m_v = \langle x_1 - v_1, \ldots, x_n - v_n \rangle$ (see Lemma 6.3 in Sect. 6.1). Using this, we can characterize the spectrum of the neural ring.

Lemma 3.1 $\text{Spec}(R_{\mathcal{C}}) = \{\bar{m}_v \mid v \in \mathcal{C}\}$, *where* $\bar{m}_v$ *is the quotient of* m_v *in* $R_{\mathcal{C}}$.

The proof is given in Sect. 6.1. Note that because $R_{\mathcal{C}}$ is a Boolean ring, the maximal ideal spectrum and the prime ideal spectrum coincide.

3.4 The Neural Ideal and an Explicit Set of Relations for the Neural Ring

The definition of the neural ring is rather impractical, as it does not give us explicit relations for generating I_C and R_C. Here, we define another ideal, J_C, via an explicit set of generating relations. Although J_C is closely related to I_C, it turns out that J_C is a more convenient object to study, which is why we will use the term *neural ideal* to refer to J_C rather than I_C.

For any $v \in \{0, 1\}^n$, consider the function $\rho_v \in \mathbb{F}_2[x_1, \ldots, x_n]$ defined as

$$\rho_v \overset{\text{def}}{=} \prod_{i=1}^n (1 - v_i - x_i) = \prod_{\{i \,|\, v_i = 1\}} x_i \prod_{\{j \,|\, v_j = 0\}} (1 - x_j) = \prod_{i \in \text{supp}(v)} x_i \prod_{j \notin \text{supp}(v)} (1 - x_j).$$

Note that $\rho_v(x)$ can be thought of as a characteristic function for v, since it satisfies $\rho_v(v) = 1$ and $\rho_v(x) = 0$ for any other $x \in \mathbb{F}_2^n$. Now consider the ideal $J_C \subseteq \mathbb{F}_2[x_1, \ldots, x_n]$ generated by all functions ρ_v, for $v \notin C$:

$$J_C \overset{\text{def}}{=} \langle \{\rho_v \mid v \notin C\} \rangle.$$

We call J_C the *neural ideal* corresponding to the neural code C. If $C = 2^{[n]}$ is the complete code, we simply set $J_C = 0$, the zero ideal. J_C is related to I_C as follows, giving us explicit relations for the neural ring.

Lemma 3.2 *Let $C \subset \{0, 1\}^n$ be a neural code. Then*

$$I_C = J_C + \mathcal{B} = \langle \{\rho_v \mid v \notin C\}, \{x_i(1 - x_i) \mid i \in [n]\} \rangle,$$

where $\mathcal{B} = \langle \{x_i(1 - x_i) \mid i \in [n]\} \rangle$ is the ideal generated by the Boolean relations, and J_C is the neural ideal.

The proof is given in Sect. 6.1.

4 How to Infer RF Structure Using the Neural Ideal

This section is the heart of the paper. We begin by presenting an alternative set of relations that can be used to define the neural ring. These relations enable us to easily interpret elements of I_C as receptive field relationships, clarifying the connection between the neural ring and ideal and the RF structure of the code. We next introduce pseudo-monomials and pseudo-monomial ideals, and use these notions to obtain a minimal description of the neural ideal, which we call the "canonical form." Theorem 4.3 enables us to use the canonical form of J_C in order to "read off" a minimal description of the RF structure of the code. Finally, we present an algorithm that inputs a neural code C and outputs the canonical form $\text{CF}(J_C)$, and illustrate its use in a detailed example.

4.1 An Alternative Set of Relations for the Neural Ring

Let $\mathcal{C} \subset \{0, 1\}^n$ be a neural code, and recall by Lemma 2.1 that $\mathcal{C}$ can always be realized as a RF code $\mathcal{C} = \mathcal{C}(\mathcal{U})$, provided we don't require the U_is to be convex. Let X be a stimulus space and $\mathcal{U} = \{U_i\}_{i=1}^n$ a collection of open sets in X, and consider the RF code $\mathcal{C}(\mathcal{U})$. The neural ring corresponding to this code is $R_{\mathcal{C}(\mathcal{U})}$.

Observe that the functions $f \in R_{\mathcal{C}(\mathcal{U})}$ can be evaluated at any point $p \in X$ by assigning

$$x_i(p) = \begin{cases} 1 & \text{if } p \in U_i, \\ 0 & \text{if } p \notin U_i, \end{cases}$$

each time x_i appears in the polynomial f. The vector $(x_1(p), \ldots, x_n(p)) \in \{0, 1\}^n$ represents the neural response to the stimulus p. Note that if $p \notin \bigcup_{i=1}^n U_i$, then $(x_1(p), \ldots, x_n(p)) = (0, \ldots, 0)$ is the all-zeros codeword. For any $\sigma \subset [n]$, define

$$U_\sigma \overset{\text{def}}{=} \bigcap_{i \in \sigma} U_i \quad \text{and} \quad x_\sigma \overset{\text{def}}{=} \prod_{i \in \sigma} x_i.$$

Our convention is that $x_\emptyset = 1$ and $U_\emptyset = X$, even in cases where $X \supsetneq \bigcup_{i=1}^n U_i$. Note that for any $p \in X$,

$$x_\sigma(p) = \begin{cases} 1 & \text{if } p \in U_\sigma, \\ 0 & \text{if } p \notin U_\sigma. \end{cases}$$

The relations in $I_{\mathcal{C}(\mathcal{U})}$ encode the combinatorial data of $\mathcal{U}$. For example, if $U_\sigma = \emptyset$ then we cannot have $x_\sigma = 1$ at any point of the stimulus space X, and must therefore impose the relation x_σ to "knock off" those points. On the other hand, if $U_\sigma \subset U_i \cup U_j$, then $x_\sigma = 1$ implies either $x_i = 1$ or $x_j = 1$, something that is guaranteed by imposing the relation $x_\sigma(1 - x_i)(1 - x_j)$. These observations lead us to an alternative ideal, $I_\mathcal{U} \subset \mathbb{F}_2[x_1, \ldots, x_n]$, defined directly from the arrangement of receptive fields $\mathcal{U} = \{U_1, \ldots, U_n\}$:

$$I_\mathcal{U} \overset{\text{def}}{=} \left\langle \left\{ x_\sigma \prod_{i \in \tau}(1 - x_i) \mid U_\sigma \subseteq \bigcup_{i \in \tau} U_i \right\} \right\rangle.$$

Note that if $\tau = \emptyset$, we only get a relation for $U_\sigma = \emptyset$, and this is x_σ. If $\sigma = \emptyset$, then $U_\sigma = X$, and we only get relations of this type if X is contained in the union of the U_is. This is equivalent to the requirement that there is no "outside point" corresponding to the all-zeros codeword.

Perhaps unsurprisingly, it turns out that $I_\mathcal{U}$ and $I_{\mathcal{C}(\mathcal{U})}$ exactly coincide, so $I_\mathcal{U}$ provides an alternative set of relations that can be used to define $R_{\mathcal{C}(\mathcal{U})}$.

Theorem 4.1 $I_\mathcal{U} = I_{\mathcal{C}(\mathcal{U})}$.

The proof is given in Sect. 6.2.

4.2 Interpreting Neural Ring Relations as Receptive Field Relationships

Theorem 4.1 suggests that we can interpret elements of $I_{\mathcal{C}}$ in terms of relationships between receptive fields.

Lemma 4.2 *Let $\mathcal{C} \subset \{0,1\}^n$ be a neural code, and let $\mathcal{U} = \{U_1, \ldots, U_n\}$ be any collection of open sets (not necessarily convex) in a stimulus space X such that $\mathcal{C} = \mathcal{C}(\mathcal{U})$. Then, for any pair of subsets $\sigma, \tau \subset [n]$,*

$$x_\sigma \prod_{i \in \tau}(1 - x_i) \in I_{\mathcal{C}} \quad \Leftrightarrow \quad U_\sigma \subseteq \bigcup_{i \in \tau} U_i.$$

Proof ($\Leftarrow$) This is a direct consequence of Theorem 4.1. ($\Rightarrow$) We distinguish two cases, based on whether or not σ and τ intersect. If $x_\sigma \prod_{i \in \tau}(1 - x_i) \in I_{\mathcal{C}}$ and $\sigma \cap \tau \neq \emptyset$, then $x_\sigma \prod_{i \in \tau}(1 - x_i) \in \mathcal{B}$, where $\mathcal{B} = \langle \{x_i(1 - x_i) \mid i \in [n]\} \rangle$ is the ideal generated by the Boolean relations. Consequently, the relation does not give us any information about the code, and $U_\sigma \subseteq \bigcup_{i \in \tau} U_i$ follows trivially from the observation that $U_i \subseteq U_i$ for any $i \in \sigma \cap \tau$. If, on the other hand, $x_\sigma \prod_{i \in \tau}(1 - x_i) \in I_{\mathcal{C}}$ and $\sigma \cap \tau = \emptyset$, then $\rho_v \in I_{\mathcal{C}}$ for each $v \in \{0,1\}^n$ such that $\mathrm{supp}(v) \supseteq \sigma$ and $\mathrm{supp}(v) \cap \tau = \emptyset$. Since $\rho_v(v) = 1$, it follows that $v \notin \mathcal{C}$ for any v with $\mathrm{supp}(v) \supseteq \sigma$ and $\mathrm{supp}(v) \cap \tau = \emptyset$. To see this, recall from the original definition of $I_{\mathcal{C}}$ that for all $c \in \mathcal{C}$, $f(c) = 0$ for any $f \in I_{\mathcal{C}}$; it follows that $\rho_v(c) = 0$ for all $c \in \mathcal{C}$. Because $\mathcal{C} = \mathcal{C}(\mathcal{U})$, the fact that $v \notin \mathcal{C}$ for any v such that $\mathrm{supp}(v) \supseteq \sigma$ and $\mathrm{supp}(v) \cap \tau = \emptyset$ implies $\bigcap_{i \in \sigma} U_i \setminus \bigcup_{j \in \tau} U_j = \emptyset$. We can thus conclude that $U_\sigma \subseteq \bigcup_{j \in \tau} U_j$. $\square$

Lemma 4.2 allows us to extract RF structure from the different types of relations that appear in $I_{\mathcal{C}}$:

- Boolean relations: $\{x_i(1 - x_i)\}$. The relation $x_i(1 - x_i)$ corresponds to $U_i \subseteq U_i$, which does not contain any information about the code $\mathcal{C}$.
- Type 1 relations: $\{x_\sigma\}$. The relation x_σ corresponds to $U_\sigma = \emptyset$.
- Type 2 relations: $\{x_\sigma \prod_{i \in \tau}(1 - x_i) \mid \sigma, \tau \neq \emptyset, \sigma \cap \tau = \emptyset, U_\sigma \neq \emptyset$ and $\bigcup_{i \in \tau} U_i \neq X\}$. The relation $x_\sigma \prod_{i \in \tau}(1 - x_i)$ corresponds to $U_\sigma \subseteq \bigcup_{i \in \tau} U_i$.
- Type 3 relations: $\{\prod_{i \in \tau}(1 - x_i)\}$. The relation $\prod_{i \in \tau}(1 - x_i)$ corresponds to $X \subseteq \bigcup_{i \in \tau} U_i$.

The somewhat complicated requirements on the Type 2 relations ensure that they do not include polynomials that are multiples of Type 1, Type 3, or Boolean relations. Note that the constant polynomial 1 may appear as both a Type 1 and a Type 3 relation, but only if $X = \emptyset$. The four types of relations listed above are otherwise disjoint. Type 3 relations only appear if X is fully covered by the receptive fields, and there is thus no all-zeros codeword corresponding to an "outside" point.

Not all elements of $I_{\mathcal{C}}$ are one of the above types, of course, but we will see that these are sufficient to generate $I_{\mathcal{C}}$. This follows from the observation (see Lemma 6.6) that the neural ideal $J_{\mathcal{C}}$ is generated by the Type 1, Type 2, and Type 3 relations, and recalling that $I_{\mathcal{C}}$ is obtained from $J_{\mathcal{C}}$ be adding in the Boolean relations (Lemma 3.2). At the same time, not all of these relations are necessary to generate the neural ideal. Can we eliminate redundant relations to come up with a "minimal" list of generators

for $J_\mathcal{C}$, and hence $I_\mathcal{C}$, that captures the essential RF structure of the code? This is the goal of the next section.

4.3 Pseudo-monomials and a Canonical Form for the Neural Ideal

The Type 1, Type 2, and Type 3 relations are all products of linear terms of the form x_i and $1 - x_i$, and are thus very similar to monomials. By analogy with square-free monomials and square-free monomial ideals (Miller and Sturmfels 2005), we define the notions of pseudo-monomials and pseudo-monomial ideals. Note that we do not allow repeated indices in our definition of pseudo-monomial, so the Boolean relations are explicitly excluded.

Definition If $f \in \mathbb{F}_2[x_1, \ldots, x_n]$ has the form $f = \prod_{i \in \sigma} x_i \prod_{j \in \tau} (1 - x_j)$ for some $\sigma, \tau \subset [n]$ with $\sigma \cap \tau = \emptyset$, then we say that f is a *pseudo-monomial*.

Definition An ideal $J \subset \mathbb{F}_2[x_1, \ldots, x_n]$ is a *pseudo-monomial ideal* if J can be generated by a finite set of pseudo-monomials.

Definition Let $J \subset \mathbb{F}_2[x_1, \ldots, x_n]$ be an ideal, and $f \in J$ a pseudo-monomial. We say that f is a *minimal* pseudo-monomial of J if there does not exist another pseudo-monomial $g \in J$ with $\deg(g) < \deg(f)$ such that $f = hg$ for some $h \in \mathbb{F}_2[x_1, \ldots, x_n]$.

By considering the set of *all* minimal pseudo-monomials in a pseudo-monomial ideal J, we obtain a unique and compact description of J, which we call the "canonical form" of J.

Definition We say that a pseudo-monomial ideal J is in *canonical form* if we present it as $J = \langle f_1, \ldots, f_l \rangle$, where the set $\mathrm{CF}(J) \overset{\mathrm{def}}{=} \{f_1, \ldots, f_l\}$ is the set of *all* minimal pseudo-monomials of J. Equivalently, we refer to $\mathrm{CF}(J)$ as the *canonical form* of J.

Clearly, for any pseudo-monomial ideal $J \subset \mathbb{F}_2[x_1, \ldots, x_n]$, $\mathrm{CF}(J)$ is unique and $J = \langle \mathrm{CF}(J) \rangle$. On the other hand, it is important to keep in mind that although $\mathrm{CF}(J)$ consists of minimal pseudo-monomials, it is not necessarily a minimal set of generators for J. To see why, consider the pseudo-monomial ideal $J = \langle x_1(1 - x_2), x_2(1 - x_3) \rangle$. This ideal in fact contains a third minimal pseudo-monomial: $x_1(1 - x_3) = (1 - x_3) \cdot [x_1(1 - x_2)] + x_1 \cdot [x_2(1 - x_3)]$. It follows that $\mathrm{CF}(J) = \{x_1(1 - x_2), x_2(1 - x_3), x_1(1 - x_3)\}$, but clearly we can remove $x_1(1 - x_3)$ from this set and still generate J.

For any code $\mathcal{C}$, the neural ideal $J_\mathcal{C}$ is a pseudo-monomial ideal because $J_\mathcal{C} = \langle \{\rho_v \mid v \notin \mathcal{C}\} \rangle$, and each of the ρ_vs is a pseudo-monomial. (In contrast, $I_\mathcal{C}$ is rarely a pseudo-monomial ideal, because it is typically necessary to include the Boolean relations as generators.) Theorem 4.3 describes the canonical form of $J_\mathcal{C}$. In what follows, we say that $\sigma \subseteq [n]$ is *minimal with respect to* property P if σ satisfies P, but P is not satisfied for any $\tau \subsetneq \sigma$. For example, if $U_\sigma = \emptyset$ and for all $\tau \subsetneq \sigma$ we have $U_\tau \neq \emptyset$, then we say that "σ is minimal w.r.t. $U_\sigma = \emptyset$."

Theorem 4.3 *Let $\mathcal{C} \subset \{0, 1\}^n$ be a neural code, and let $\mathcal{U} = \{U_1, \ldots, U_n\}$ be any collection of open sets (not necessarily convex) in a nonempty stimulus space X such that $\mathcal{C} = \mathcal{C}(\mathcal{U})$. The canonical form of $J_{\mathcal{C}}$ is:*

$$J_{\mathcal{C}} = \Big\langle \{x_\sigma \mid \sigma \text{ is minimal w.r.t. } U_\sigma = \emptyset\},$$

$$\Big\{ x_\sigma \prod_{i \in \tau} (1 - x_i) \,\Big|\, \sigma, \tau \neq \emptyset,\ \sigma \cap \tau = \emptyset,\ U_\sigma \neq \emptyset,\ \bigcup_{i \in \tau} U_i \neq X,$$

$$\text{and } \sigma, \tau \text{ are each minimal w.r.t. } U_\sigma \subseteq \bigcup_{i \in \tau} U_i \Big\},$$

$$\Big\{ \prod_{i \in \tau} (1 - x_i) \,\Big|\, \tau \text{ is minimal w.r.t. } X \subseteq \bigcup_{i \in \tau} U_i \Big\} \Big\rangle.$$

We call the above three (disjoint) sets of relations comprising $\mathrm{CF}(J_{\mathcal{C}})$ *the minimal Type 1 relations, the minimal Type 2 relations, and the minimal Type 3 relations, respectively.*

The proof is given in Sect. 6.3. Note that, because of the uniqueness of the canonical form, if we are given $\mathrm{CF}(J_{\mathcal{C}})$ then Theorem 4.3 allows us to read off the corresponding (minimal) relationships that must be satisfied by any receptive field representation of the code as $\mathcal{C} = \mathcal{C}(\mathcal{U})$:

- Type 1: $x_\sigma \in \mathrm{CF}(J_{\mathcal{C}})$ implies that $U_\sigma = \emptyset$, but all lower-order intersections U_γ with $\gamma \subsetneq \sigma$ are non-empty.
- Type 2: $x_\sigma \prod_{i \in \tau} (1 - x_i) \in \mathrm{CF}(J_{\mathcal{C}})$ implies that $U_\sigma \subseteq \bigcup_{i \in \tau} U_i$, but no lower-order intersection is contained in $\bigcup_{i \in \tau} U_i$, and all the U_is are necessary for $U_\sigma \subseteq \bigcup_{i \in \tau} U_i$.
- Type 3: $\prod_{i \in \tau} (1 - x_i) \in \mathrm{CF}(J_{\mathcal{C}})$ implies that $X \subseteq \bigcup_{i \in \tau} U_i$, but X is not contained in any lower-order union $\bigcup_{i \in \gamma} U_i$ for $\gamma \subsetneq \tau$.

The canonical form $\mathrm{CF}(J_{\mathcal{C}})$ thus provides a minimal description of the RF structure dictated by the code $\mathcal{C}$.

The Type 1 relations in $\mathrm{CF}(J_{\mathcal{C}})$ can be used to obtain a (crude) lower bound on the minimal embedding dimension of the neural code, as defined in Sect. 2.3.2. Recall Helly's theorem (Sect. 2.3.1), and observe that if $x_\sigma \in \mathrm{CF}(J_{\mathcal{C}})$ then σ is minimal with respect to $U_\sigma = \emptyset$; this in turn implies that $|\sigma| \leq d + 1$. (If $|\sigma| > d + 1$, by minimality all $d + 1$ subsets intersect and by Helly's theorem we must have $U_\sigma \neq \emptyset$.) We can thus obtain a lower bound on the minimal embedding dimension d as

$$d \geq \max_{\{\sigma \mid x_\sigma \in \mathrm{CF}(J_{\mathcal{C}})\}} |\sigma| - 1,$$

where the maximum is taken over all σ such that x_σ is a Type 1 relation in $\mathrm{CF}(J_{\mathcal{C}})$. This bound only depends on $\Delta(\mathcal{C})$, however, and does not provide any insight regarding the different minimal embedding dimensions observed in the examples of Fig. 3. These codes have no Type 1 relations in their canonical forms, but they are nicely

differentiated by their minimal Type 2 and Type 3 relations. From the receptive field arrangements depicted in Fig. 3, we can easily write down $\mathrm{CF}(J_{\mathcal{C}})$ for each of these codes.

A. $\mathrm{CF}(J_{\mathcal{C}}) = \{0\}$. There are no relations here because $\mathcal{C} = 2^{[3]}$.
B. $\mathrm{CF}(J_{\mathcal{C}}) = \{1 - x_3\}$. This Type 3 relation reflects the fact that $X = U_3$.
C. $\mathrm{CF}(J_{\mathcal{C}}) = \{x_1(1 - x_2), x_2(1 - x_3), x_1(1 - x_3)\}$. These Type 2 relations correspond to $U_1 \subset U_2$, $U_2 \subset U_3$, and $U_1 \subset U_3$. Note that the first two of these receptive field relationships imply the third; correspondingly, the third canonical form relation satisfies: $x_1(1 - x_3) = (1 - x_3) \cdot [x_1(1 - x_2)] + x_1 \cdot [x_2(1 - x_3)]$.
D. $\mathrm{CF}(J_{\mathcal{C}}) = \{(1 - x_1)(1 - x_2)\}$. This Type 3 relation reflects $X = U_1 \cup U_2$, and implies $U_3 \subset U_1 \cup U_2$.

Nevertheless, we do not yet know how to infer the minimal embedding dimension from $\mathrm{CF}(J_{\mathcal{C}})$. In Appendix 2, we provide a complete list of neural codes on three neurons, up to permutation, and their respective canonical forms.

4.4 Comparison to the Stanley–Reisner Ideal

Readers familiar with the Stanley–Reisner ideal (Miller and Sturmfels 2005; Stanley 2004) will recognize that this kind of ideal is generated by the Type 1 relations of a neural code $\mathcal{C}$. The corresponding simplicial complex is $\Delta(\mathcal{C})$, the smallest simplicial complex that contains the code.

Lemma 4.4 *Let $\mathcal{C} = \mathcal{C}(\mathcal{U})$. The ideal generated by the Type 1 relations, $\langle x_\sigma \mid U_\sigma = \emptyset \rangle$, is the Stanley–Reisner ideal of $\Delta(\mathcal{C})$. Moreover, if $\mathrm{supp}\,\mathcal{C}$ is a simplicial complex, then $\mathrm{CF}(J_{\mathcal{C}})$ contains no Type 2 or Type 3 relations, and $J_{\mathcal{C}}$ is thus the Stanley–Reisner ideal for $\mathrm{supp}\,\mathcal{C}$.*

Proof To see the first statement, observe that the *Stanley–Reisner ideal* of a simplicial complex Δ is the ideal

$$I_\Delta \overset{\mathrm{def}}{=} \langle x_\sigma \mid \sigma \notin \Delta \rangle,$$

and recall that $\Delta(\mathcal{C}) = \{\sigma \subseteq [n] \mid \sigma \subseteq \mathrm{supp}(c) \text{ for some } c \in \mathcal{C}\}$. As $\mathcal{C} = \mathcal{C}(\mathcal{U})$, an equivalent characterization is $\Delta(\mathcal{C}) = \{\sigma \subseteq [n] \mid U_\sigma \neq \emptyset\}$. Since these sets are equal, so are their complements in $2^{[n]}$:

$$\big\{\sigma \subseteq [n] \,\big|\, \sigma \notin \Delta(\mathcal{C})\big\} = \big\{\sigma \subseteq [n] \,\big|\, U_\sigma = \emptyset\big\}.$$

Thus, $\langle x_\sigma \mid U_\sigma = \emptyset \rangle = \langle x_\sigma \mid \sigma \notin \Delta(\mathcal{C}) \rangle$, which is the Stanley–Reisner ideal for $\Delta(\mathcal{C})$.

To prove the second statement, suppose that $\mathrm{supp}\,\mathcal{C}$ is a simplicial complex. Note that $\mathcal{C}$ must contain the all-zeros codeword, so $X \supsetneq \bigcup_{i=1}^{n} U_i$ and there can be no Type 3 relations. Suppose the canonical form of $J_{\mathcal{C}}$ contains a Type 2 relation $x_\sigma \prod_{i \in \tau}(1 - x_i)$, for some $\sigma, \tau \subset [n]$ satisfying $\sigma, \tau \neq \emptyset$, $\sigma \cap \tau = \emptyset$ and $U_\sigma \neq \emptyset$. The existence of this relation indicates that $\sigma \notin \mathrm{supp}\,\mathcal{C}$, while there does exist an $\omega \in \mathcal{C}$ such that $\sigma \subset \omega$. This contradicts the assumption that $\mathrm{supp}\,\mathcal{C}$ is a simplicial complex. We conclude that $J_{\mathcal{C}}$ has no Type 2 relations. $\quad\square$
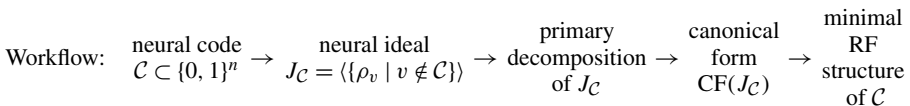
The canonical form of $J_\mathcal{C}$ thus enables us to immediately read off, via the Type 1 relations, the minimal forbidden faces of the simplicial complex $\Delta(\mathcal{C})$ associated to the code, and also the minimal deviations of $\mathcal{C}$ from being a simplicial complex, which are captured by the Type 2 and Type 3 relations.

4.5 An Algorithm for Obtaining the Canonical Form

Now that we have established that a minimal description of the RF structure can be extracted from the canonical form of the neural ideal, the most pressing question is the following:

Question How do we find the canonical form $\mathrm{CF}(J_\mathcal{C})$ if all we know is the code $\mathcal{C}$, and we are *not* given a representation of the code as $\mathcal{C} = \mathcal{C}(\mathcal{U})$?

In this section, we describe an algorithmic method for finding $\mathrm{CF}(J_\mathcal{C})$ from knowledge only of $\mathcal{C}$. It turns out that computing the primary decomposition of $J_\mathcal{C}$ is a key step toward finding the minimal pseudo-monomials. This parallels the situation for monomial ideals, although there are some additional subtleties in the case of pseudo-monomial ideals. As previously discussed, from the canonical form we can read off the RF structure of the code, so the overall workflow is as follows:

$$\text{Workflow:} \quad \begin{array}{c} \text{neural code} \\ \mathcal{C} \subset \{0, 1\}^n \end{array} \to \begin{array}{c} \text{neural ideal} \\ J_\mathcal{C} = \langle\{\rho_v \mid v \notin \mathcal{C}\}\rangle \end{array} \to \begin{array}{c} \text{primary} \\ \text{decomposition} \\ \text{of } J_\mathcal{C} \end{array} \to \begin{array}{c} \text{canonical} \\ \text{form} \\ \mathrm{CF}(J_\mathcal{C}) \end{array} \to \begin{array}{c} \text{minimal} \\ \text{RF} \\ \text{structure} \\ \text{of } \mathcal{C} \end{array}$$

Canonical Form Algorithm

Input: A neural code $\mathcal{C} \subset \{0, 1\}^n$.

Output: The canonical form of the neural ideal, $\mathrm{CF}(J_\mathcal{C})$.

Step 1: From $\mathcal{C} \subset \{0, 1\}^n$, compute $J_\mathcal{C} = \langle\{\rho_v \mid v \notin \mathcal{C}\}\rangle$.

Step 2: Compute the primary decomposition of $J_\mathcal{C}$. It turns out (see Theorem 5.4 in the next section) that this decomposition yields a unique representation of the ideal as

$$J_\mathcal{C} = \bigcap_{a \in \mathcal{A}} \mathbf{p}_a,$$

where each $a \in \mathcal{A}$ is an element of $\{0, 1, *\}^n$, and $\mathbf{p}_a$ is defined as

$$\mathbf{p}_a \overset{\mathrm{def}}{=} \langle\{x_i - a_i \mid a_i \neq *\}\rangle = \langle\{x_i \mid a_i = 0\}, \{1 - x_j \mid a_j = 1\}\rangle.$$

Note that the $\mathbf{p}_a$s are all prime ideals. We will see later how to compute this primary decomposition algorithmically, in Sect. 5.3.

Step 3: Observe that any pseudo-monomial $f \in J_\mathcal{C}$ must satisfy $f \in \mathbf{p}_a$ for each $a \in \mathcal{A}$. It follows that f is a multiple of one of the linear generators of $\mathbf{p}_a$ for each $a \in \mathcal{A}$. Compute the following set of elements of $J_\mathcal{C}$:

$$\mathcal{M}(J_\mathcal{C}) = \left\{ \prod_{a \in \mathcal{A}} g_a \;\middle|\; g_a = x_i - a_i \text{ for some } a_i \neq * \right\}.$$

$\mathcal{M}(J_\mathcal{C})$ consists of all polynomials obtained as a product of linear generators g_a, one for each prime ideal $\mathbf{p}_a$ of the primary decomposition of $J_\mathcal{C}$.

Step 4: Reduce the elements of $\mathcal{M}(J_\mathcal{C})$ by imposing $x_i(1 - x_i) = 0$. This eliminates elements that are not pseudo-monomials. It also reduces the degrees of some of the remaining elements, as it implies $x_i^2 = x_i$ and $(1 - x_i)^2 = (1 - x_i)$. We are left with a set of pseudo-monomials of the form $f = \prod_{i \in \sigma} x_i \prod_{j \in \tau} (1 - x_j)$ for $\tau \cap \sigma = \emptyset$. Call this new reduced set $\widetilde{\mathcal{M}}(J_\mathcal{C})$.

Step 5: Finally, remove all elements of $\widetilde{\mathcal{M}}(J_\mathcal{C})$ that are multiples of lower-degree elements in $\widetilde{\mathcal{M}}(J_\mathcal{C})$.

Proposition 4.5 *The resulting set is the canonical form* $\mathrm{CF}(J_\mathcal{C})$.

The proof is given in Sect. 6.4.

4.6 An Example

Now we are ready to use the canonical form algorithm in an example, illustrating how to obtain a possible arrangement of convex receptive fields from a neural code.

Suppose a neural code $\mathcal{C}$ has the following 13 codewords, and 19 missing words:

$$\mathcal{C} = \{00000, \ 10000, \ 01000, \ 00100, \ 00001, \ 11000, \ 10001,$$
$$01100, \ 00110, \ 00101, 00011, \ 11100, \ 00111\},$$
$$\{0, 1\}^5 \backslash \mathcal{C} = \{00010, \ 10100, \ 10010, \ 01010, \ 01001, \ 11010, \ 11001,$$
$$10110, \ 10101, \ 10011, \ 01110, \ 01101, \ 01011, \ 11110,$$
$$11101, \ 11011, \ 10111, \ 01111, \ 11111\}.$$

Thus, the neural ideal $J_\mathcal{C}$ has 19 generators, using the original definition $J_\mathcal{C} = \langle \{\rho_v \mid v \notin \mathcal{C}\} \rangle$:

$$J_\mathcal{C} = \big\langle x_4(1 - x_1)(1 - x_2)(1 - x_3)(1 - x_5), x_1x_3(1 - x_2)(1 - x_4)(1 - x_5),$$
$$x_1x_4(1 - x_2)(1 - x_3)(1 - x_5), x_2x_4(1 - x_1)(1 - x_3)(1 - x_5),$$
$$x_2x_5(1 - x_1)(1 - x_3)(1 - x_4), x_1x_2x_4(1 - x_3)(1 - x_5),$$
$$x_1x_2x_5(1 - x_3)(1 - x_4), x_1x_3x_4(1 - x_2)(1 - x_5), x_1x_3x_5(1 - x_2)(1 - x_4),$$
$$x_1x_4x_5(1 - x_2)(1 - x_3), x_2x_3x_4(1 - x_1)(1 - x_5), x_2x_3x_5(1 - x_1)(1 - x_4),$$
$$x_2x_4x_5(1 - x_1)(1 - x_3), x_1x_2x_3x_4(1 - x_5), x_1x_2x_3x_5(1 - x_4),$$
$$x_1x_2x_4x_5(1 - x_3), x_1x_3x_4x_5(1 - x_2), x_2x_3x_4x_5(1 - x_1), x_1x_2x_3x_4x_5 \big\rangle.$$

Despite the fact that we are considering only five neurons, this looks like a complicated ideal. Considering the canonical form of $J_\mathcal{C}$ will help us to extract the relevant combinatorial information and allow us to create a possible arrangement of receptive

fields $\mathcal{U}$ that realizes this code as $\mathcal{C} = \mathcal{C}(\mathcal{U})$. Following step 2 of our canonical form algorithm, we take the primary decomposition of $J_{\mathcal{C}}$:

$$J_{\mathcal{C}} = \langle x_1, x_2, x_4 \rangle \cap \langle x_1, x_2, 1 - x_3 \rangle \cap \langle x_1, x_2, 1 - x_5 \rangle \cap \langle x_2, x_3, x_4 \rangle$$

$$\cap \langle x_3, x_4, x_5 \rangle \cap \langle x_1, x_4, x_5 \rangle \cap \langle 1 - x_2, x_4, x_5 \rangle.$$

Then, as described in steps 3–5 of the algorithm, we take all possible products amongst these seven larger ideals, reducing by the relation $x_i(1 - x_i) = 0$ (note that this gives us $x_i = x_i^2$, and hence we can say $x_i^k = x_i$ for any $k > 1$). We also remove any polynomials that are multiples of smaller-degree pseudo-monomials in our list. This process leaves us with six minimal pseudo-monomials, yielding the canonical form:

$$J_{\mathcal{C}} = \langle \mathrm{CF}(J_{\mathcal{C}}) \rangle = \langle x_1 x_3 x_5, \, x_2 x_5, \, x_1 x_4, \, x_2 x_4, \, x_1 x_3(1 - x_2), \, x_4(1 - x_3)(1 - x_5) \rangle.$$

Note in particular that every generator we originally put in $J_{\mathcal{C}}$ is a multiple of one of the six relations in $\mathrm{CF}(J_{\mathcal{C}})$. Next, we consider what the relations in $\mathrm{CF}(J_{\mathcal{C}})$ tell us about the arrangement of receptive fields that would be needed to realize the code as $\mathcal{C} = \mathcal{C}(\mathcal{U})$.

1. $x_1 x_3 x_5 \in \mathrm{CF}(J_{\mathcal{C}}) \Rightarrow U_1 \cap U_3 \cap U_5 = \emptyset$, while $U_1 \cap U_3$, $U_3 \cap U_5$ and $U_1 \cap U_5$ are all nonempty.
2. $x_2 x_5 \in \mathrm{CF}(J_{\mathcal{C}}) \Rightarrow U_2 \cap U_5 = \emptyset$, while U_2, U_5 are both nonempty.
3. $x_1 x_4 \in \mathrm{CF}(J_{\mathcal{C}}) \Rightarrow U_1 \cap U_4 = \emptyset$, while U_1, U_4 are both nonempty.
4. $x_2 x_4 \in \mathrm{CF}(J_{\mathcal{C}}) \Rightarrow U_2 \cap U_4 = \emptyset$, while U_2, U_4 are both nonempty.
5. $x_1 x_3(1 - x_2) \in \mathrm{CF}(J_{\mathcal{C}}) \Rightarrow U_1 \cap U_3 \subseteq U_2$, while $U_1 \not\subseteq U_2, U_3 \not\subseteq U_2$, and $U_1 \cap U_3 \neq \emptyset$.
6. $x_4(1 - x_3)(1 - x_5) \in \mathrm{CF}(J_{\mathcal{C}}) \Rightarrow U_4 \subseteq U_3 \cup U_5$, while $U_4 \neq \emptyset$, and that $U_4 \not\subseteq U_3, U_4 \not\subseteq U_5$.

The minimal Type 1 relations (1–4) tell us that we should draw U_1, U_3, and U_5 with all pairwise intersections, but leaving a "hole" in the middle since the triple intersection is empty. Then U_2 should be drawn to intersect U_1 and U_3, but not U_5. Similarly, U_4 should intersect U_3 and U_5, but not U_1 or U_2. The minimal Type 2 relations (5–6) tell us that U_2 should be drawn to contain the intersection $U_1 \cap U_3$, while U_4 lies in the union $U_3 \cup U_5$, but is not contained in U_3 or U_5 alone. There are no minimal Type 3 relations, as expected for a code that includes the all-zeros codeword.

Putting all this together, and assuming convex receptive fields, we can completely infer the receptive field structure, and draw the corresponding picture (see Fig. 4). It is easy to verify that the code $\mathcal{C}(\mathcal{U})$ of the pictured arrangement indeed coincides with $\mathcal{C}$.

5 Primary Decomposition

Let $\mathcal{C} \subset \{0, 1\}^n$ be a neural code. The primary decomposition of $I_{\mathcal{C}}$ is boring:

$$I_{\mathcal{C}} = \bigcap_{c \in \mathcal{C}} m_c,$$

Fig. 4 An arrangement of five sets that realizes $\mathcal{C}$ as $\mathcal{C}(\mathcal{U})$
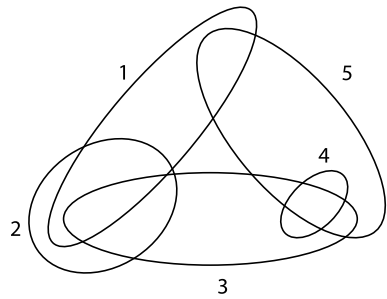
where m_v for any $v \in \{0, 1\}^n$ is the maximal ideal $I(v)$ defined in Sect. 3.3. This simply expresses $I_{\mathcal{C}}$ as the intersection of all maximal ideals m_c for $c \in \mathcal{C}$, because the variety $\mathcal{C} = V(I_{\mathcal{C}})$ is just a finite set of points and the primary decomposition reflects no additional structure of the code.

On the other hand, the primary decomposition of the neural ideal $J_{\mathcal{C}}$ retains the full combinatorial structure of $\mathcal{C}$. Indeed, we have seen that computing this decomposition is a critical step toward obtaining $\mathrm{CF}(J_{\mathcal{C}})$, which captures the receptive field structure of the neural code. In this section, we describe the primary decomposition of $J_{\mathcal{C}}$ and discuss its relationship to some natural decompositions of the neural code. We end with an algorithm for obtaining primary decomposition of any pseudo-monomial ideal.

5.1 Primary Decomposition of the Neural Ideal

We begin by defining some objects related to $\mathbb{F}_2[x_1, \ldots, x_n]$ and $\{0, 1\}^n$, without reference to any particular neural code. For any $a \in \{0, 1, *\}^n$, we define the variety

$$V_a \stackrel{\text{def}}{=} \{v \in \{0, 1\}^n \mid v_i = a_i \text{ for all } i \text{ s.t. } a_i \neq *\} \subseteq \{0, 1\}^n.$$

This is simply the subset of points compatible with the word "a", where $*$ is viewed as a "wild card" symbol. Note that $V_v = \{v\}$ for any $v \in \{0, 1\}^n$. We can also associate a prime ideal to a,

$$\mathbf{p}_a \stackrel{\text{def}}{=} \langle \{x_i - a_i \mid a_i \neq *\} \rangle \subseteq \mathbb{F}_2[x_1, \ldots, x_n],$$

consisting of polynomials in $\mathbb{F}_2[x_1, \ldots, x_n]$ that vanish on all points compatible with a. To obtain all such polynomials, we must add in the Boolean relations (see Sect. 6.1):

$$\mathbf{q}_a \stackrel{\text{def}}{=} I(V_a) = \mathbf{p}_a + \langle x_1^2 - x_1, \ldots, x_n^2 - x_n \rangle.$$

Note that $V_a = V(\mathbf{p}_a) = V(\mathbf{q}_a)$.

Next, let us relate this all to a code $\mathcal{C} \subset \{0, 1\}^n$. Recall the definition of the neural ideal,

$$J_{\mathcal{C}} \stackrel{\text{def}}{=} \langle \{\rho_v \mid v \notin \mathcal{C}\} \rangle = \left\langle \left\{ \prod_{i=1}^n ((x_i - v_i) - 1) \,\middle|\, v \notin \mathcal{C} \right\} \right\rangle.$$

We have the following correspondences.

Lemma 5.1 $J_{\mathcal{C}} \subseteq \mathbf{p}_a \Leftrightarrow V_a \subseteq \mathcal{C}$.

Proof ($\Rightarrow$) $J_{\mathcal{C}} \subseteq \mathbf{p}_a \Rightarrow V(\mathbf{p}_a) \subseteq V(J_{\mathcal{C}})$. Recalling that $V(\mathbf{p}_a) = V_a$ and $V(J_{\mathcal{C}}) = \mathcal{C}$, this gives $V_a \subseteq \mathcal{C}$. ($\Leftarrow$) $V_a \subseteq \mathcal{C} \Rightarrow I(\mathcal{C}) \subseteq I(V_a) \Rightarrow I_{\mathcal{C}} \subseteq \mathbf{q}_a$. Recalling that both $I_{\mathcal{C}}$ and $\mathbf{q}_a$ differ from $J_{\mathcal{C}}$ and $\mathbf{p}_a$, respectively, by the addition of the Boolean relations, we obtain $J_{\mathcal{C}} \subseteq \mathbf{p}_a$. $\qquad\square$

Lemma 5.2 *For any* $a, b \in \{0, 1, *\}^n$, $V_a \subseteq V_b \Leftrightarrow \mathbf{p}_b \subseteq \mathbf{p}_a$.

Proof ($\Rightarrow$) Suppose $V_a \subseteq V_b$. Then, for any i such that $b_i \neq *$ we have $a_i = b_i$. It follows that each generator of $\mathbf{p}_b$ is also in $\mathbf{p}_a$, so $\mathbf{p}_b \subseteq \mathbf{p}_a$. ($\Leftarrow$) Suppose $\mathbf{p}_b \subseteq \mathbf{p}_a$. Then, $V_a = V(\mathbf{p}_a) \subseteq V(\mathbf{p}_b) = V_b$. $\qquad\square$

Recall that a an ideal $\mathbf{p}$ is said to be a *minimal prime* over J if $\mathbf{p}$ is a prime ideal that contains J, and there is no other prime ideal $\mathbf{p}'$ such that $\mathbf{p} \supsetneq \mathbf{p}' \supseteq J$. Minimal primes $\mathbf{p}_a \supseteq J_{\mathcal{C}}$ correspond to maximal varieties V_a such that $V_a \subseteq \mathcal{C}$. Consider the set

$$\mathcal{A}_{\mathcal{C}} \stackrel{\text{def}}{=} \{a \in \{0, 1, *\}^n \mid V_a \subseteq \mathcal{C}\}.$$

We say that $a \in \mathcal{A}_{\mathcal{C}}$ is *maximal* if there does not exist another element $b \in \mathcal{A}_{\mathcal{C}}$ such that $V_a \subsetneq V_b$ (i.e., $a \in \mathcal{A}_{\mathcal{C}}$ is maximal if V_a is maximal such that $V_a \subseteq \mathcal{C}$).

Lemma 5.3 *The element* $a \in \mathcal{A}_{\mathcal{C}}$ *is maximal if and only if* $\mathbf{p}_a$ *is a minimal prime over* $J_{\mathcal{C}}$.

Proof Recall that $a \in \mathcal{A}_{\mathcal{C}} \Rightarrow V_a \subseteq \mathcal{C}$, and hence $J_{\mathcal{C}} \subseteq \mathbf{p}_a$ (by Lemma 5.1). ($\Rightarrow$) Let $a \in \mathcal{A}_{\mathcal{C}}$ be maximal, and choose $b \in \{0, 1, *\}$ such that $J_{\mathcal{C}} \subseteq \mathbf{p}_b \subseteq \mathbf{p}_a$. By Lemmas 5.1 and 5.2, $V_a \subseteq V_b \subseteq \mathcal{C}$. Since a is maximal, we conclude that $b = a$, and hence $\mathbf{p}_b = \mathbf{p}_a$. It follows that $\mathbf{p}_a$ is a minimal prime over $J_{\mathcal{C}}$. ($\Leftarrow$) Suppose $\mathbf{p}_a$ is a minimal prime over $J_{\mathcal{C}}$. Then by Lemma 5.1, $a \in \mathcal{A}_{\mathcal{C}}$. Let b be a maximal element of $\mathcal{A}_{\mathcal{C}}$ such that $V_a \subseteq V_b \subseteq \mathcal{C}$. Then $J_{\mathcal{C}} \subseteq \mathbf{p}_b \subseteq \mathbf{p}_a$. Since $\mathbf{p}_a$ is a minimal prime over $J_{\mathcal{C}}$, $\mathbf{p}_b = \mathbf{p}_a$, and hence $b = a$. Thus, a is maximal in $\mathcal{A}_{\mathcal{C}}$. $\qquad\square$

We can now describe the primary decomposition of $J_{\mathcal{C}}$. Here we assume the neural code $\mathcal{C} \subseteq \{0, 1\}^n$ is nonempty, so that $J_{\mathcal{C}}$ is a proper pseudo-monomial ideal.

Theorem 5.4 $J_{\mathcal{C}} = \bigcap_{i=1}^{\ell} \mathbf{p}_{a_i}$ *is the unique irredundant primary decomposition of* $J_{\mathcal{C}}$, *where* $\mathbf{p}_{a_1}, \ldots, \mathbf{p}_{a_\ell}$ *are the minimal primes over* $J_{\mathcal{C}}$.

The proof is given in Sect. 6.6. Combining this theorem with Lemma 5.3, we have the following corollary.

Corollary 5.5 $J_{\mathcal{C}} = \bigcap_{i=1}^{\ell} \mathbf{p}_{a_i}$ *is the unique irredundant primary decomposition of* $J_{\mathcal{C}}$, *where* $a_1, \ldots, a_\ell$ *are the maximal elements of* $\mathcal{A}_{\mathcal{C}}$.

5.2 Decomposing the Neural Code via Intervals of the Boolean Lattice

From the definition of $\mathcal{A}_\mathcal{C}$, it is easy to see that the maximal elements yield a kind of "primary" decomposition of the neural code $\mathcal{C}$ as a union of maximal V_as.

Lemma 5.6 $\mathcal{C} = \bigcup_{i=1}^{\ell} V_{a_i}$, where $a_1, \ldots, a_\ell$ are the maximal elements of $\mathcal{A}_\mathcal{C}$. (I.e., $\mathbf{p}_{a_1}, \ldots, \mathbf{p}_{a_\ell}$ are the minimal primes in the primary decomposition of $J_\mathcal{C}$.)

Proof Since $V_a \subseteq \mathcal{C}$ for any $a \in \mathcal{A}_\mathcal{C}$, clearly $\bigcup_{i=1}^{\ell} V_{a_i} \subseteq \mathcal{C}$. To see the reverse inclusion, note that for any $c \in \mathcal{C}$, $c \in V_c \subseteq V_a$ for some maximal $a \in \mathcal{A}_\mathcal{C}$. Hence, $\mathcal{C} \subseteq \bigcup_{i=1}^{\ell} V_{a_i}$. $\qquad\square$

Note that Lemma 5.6 could also be regarded as a corollary of Theorem 5.4, since $\mathcal{C} = V(J_\mathcal{C}) = V(\bigcap_{i=1}^{\ell} \mathbf{p}_{a_i}) = \bigcup_{i=1}^{\ell} V(\mathbf{p}_{a_i}) = \bigcup_{i=1}^{\ell} V_{a_i}$, and the maximal $a \in \mathcal{A}_\mathcal{C}$ correspond to minimal primes $\mathbf{p}_a \supseteq J_\mathcal{C}$. Although we were able to prove Lemma 5.6 directly, in practice we use the primary decomposition in order to find (algorithmically) the maximal elements $a_1, \ldots, a_\ell \in \mathcal{A}_\mathcal{C}$, and thus determine the V_as for the above decomposition of the code.

It is worth noting here that the decomposition of $\mathcal{C}$ in Lemma 5.6 is not necessarily *minimal*. This is because one can have fewer $\mathbf{q}_a$s such that

$$\bigcap_{i \in \sigma \subsetneq [\ell]} \mathbf{q}_{a_i} = \bigcap_{i \in [\ell]} \mathbf{p}_{a_i}.$$

Since $V(\mathbf{q}_{a_i}) = V(\mathbf{p}_{a_i}) = V_{a_i}$, this would lead to a decomposition of $\mathcal{C}$ as a union of fewer V_{a_i}s. In contrast, the primary decomposition of $J_\mathcal{C}$ in Theorem 5.4 is irredundant, and hence none of the minimal primes can be dropped from the intersection.

5.2.1 Neural Activity "Motifs" and Intervals of the Boolean Lattice

We can think of an element $a \in \{0, 1, *\}^n$ as a neural activity "motif". That is, a is a pattern of activity and silence for a subset of the neurons, while V_a consists of all activity patterns on the full population of neurons that are consistent with this motif (irrespective of what the code is). For a given neural code $\mathcal{C}$, the set of maximal $a_1, \ldots, a_l \in \mathcal{A}_\mathcal{C}$ corresponds to a set of minimal motifs that define the code (here "minimal" is used in the sense of having the fewest number of neurons that are constrained to be "on" or "off" because $a_i \neq *$). If $a \in \{0, *\}^n$, we refer to a as a neural *silence* motif, since it corresponds to a pattern of silence. In particular, silence motifs correspond to simplices in $\text{supp}\,\mathcal{C}$, since $\text{supp}\, V_a$ is a simplex in this case. If $\text{supp}\,\mathcal{C}$ is a simplicial complex, then Lemma 5.6 gives the decomposition of $\mathcal{C}$ as a union of minimal silence motifs (corresponding to *facets*, or maximal simplices, of $\text{supp}\,\mathcal{C}$).

More generally, V_a corresponds to an *interval* of the Boolean lattice $\{0, 1\}^n$. Recall the poset structure of the Boolean lattice: for any pair of elements $v_1, v_2 \in \{0, 1\}^n$, we have $v_1 \leq v_2$ if and only if $\text{supp}(v_1) \subseteq \text{supp}(v_2)$. An *interval* of the Boolean lattice is thus a subset of the form:

$$[u_1, u_2] \overset{\text{def}}{=} \{v \in \{0, 1\}^n \mid u_1 \leq v \leq u_2\}.$$

Given an element $a \in \{0, 1, *\}^n$, we have a natural interval consisting of all Boolean lattice elements "compatible" with a. Letting $a^0 \in \{0, 1\}^n$ be the element obtained from a by setting all $*$s to 0, and $a^1 \in \{0, 1\}^n$ the element obtained by setting all $*$s to 1, we find that

$$V_a = [a^0, a^1] = \{v \in \{0, 1\}^n \mid a^0 \leq v \leq a^1\}.$$

Simplices correspond to intervals of the form $[0, a^1]$, where 0 is the bottom "all-zeros" element in the Boolean lattice.

While the primary decomposition of $J_{\mathcal{C}}$ allows a neural code $\mathcal{C} \subseteq \{0, 1\}^n$ to be decomposed as a union of intervals of the Boolean lattice, as indicated by Lemma 5.6, the canonical form $\mathrm{CF}(J_{\mathcal{C}})$ provides a decomposition of the *complement* of $\mathcal{C}$ as a union of intervals. First, notice that to any pseudo-monomial $f \in \mathrm{CF}(J_{\mathcal{C}})$ we can associate an element $b \in \{0, 1, *\}$ as follows: $b_i = 1$ if $x_i | f$, $b_i = 0$ if $(1 - x_i) | f$, and $b_i = *$ otherwise. In other words,

$$f = f_b \stackrel{\mathrm{def}}{=} \prod_{\{i \mid b_i = 1\}} x_i \prod_{\{j \mid b_j = 0\}} (1 - x_j).$$

As before, b corresponds to an interval $V_b = [b^0, b^1] \subset \{0, 1\}^n$. Recalling the $J_{\mathcal{C}}$ is generated by pseudo-monomials corresponding to non-codewords, it is now easy to see that the complement of $\mathcal{C}$ in $\{0, 1\}^n$ can be expressed as the union of V_bs, where each b corresponds to a pseudo-monomial in the canonical form. The canonical form thus provides an alternative description of the code, nicely complementing Lemma 5.6.

Lemma 5.7 $\mathcal{C} = \{0, 1\}^n \setminus \bigcup_{i=1}^{k} V_{b_i}$, *where* $\mathrm{CF}(J_{\mathcal{C}}) = \{f_{b_1}, \ldots, f_{b_k}\}$.

We now illustrate both decompositions of the neural code with an example.

Example 1 Consider the neural code $\mathcal{C} = \{000, 001, 011, 111\} \subset \{0, 1\}^3$ corresponding to a set of receptive fields satisfying $U_1 \subsetneq U_2 \subsetneq U_3 \subsetneq X$. The primary decomposition of $J_{\mathcal{C}} \subset \mathbb{F}_2[x_1, x_2, x_3]$ is given by

$$\langle x_1, x_2 \rangle \cap \langle x_1, 1 - x_3 \rangle \cap \langle 1 - x_2, 1 - x_3 \rangle,$$

while the canonical form is

$$\mathrm{CF}(J_{\mathcal{C}}) = \langle x_1(1 - x_2), x_2(1 - x_3), x_1(1 - x_3) \rangle.$$

From the primary decomposition, we can write $\mathcal{C} = V_{a_1} \cup V_{a_2} \cup V_{a_3}$ for $a_1 = 00*$, $a_2 = 0*1$, and $a_3 = *11$. The corresponding Boolean lattice intervals are $[000, 001]$, $[001, 011]$, and $[011, 111]$, respectively, and are depicted in black in Fig. 5. As noted before, this decomposition of the neural code need not be minimal; indeed, we could also write $\mathcal{C} = V_{a_1} \cup V_{a_3}$, as the middle interval is not necessary to cover all codewords in $\mathcal{C}$.

From the canonical form, we obtain $\mathcal{C} = \{0, 1\}^3 \setminus (V_{b_1} \cup V_{b_2} \cup V_{b_3})$, where $b_1 = 10*$, $b_2 = *10$, and $b_3 = 1*0$. The corresponding Boolean lattice intervals spanning

Fig. 5 Boolean interval
decompositions of the code
$\mathcal{C} = \{000, 001, 011, 111\}$ (in
black) and of its complement (in
gray), arising from the primary
decomposition and canonical
form of $J_{\mathcal{C}}$, respectively
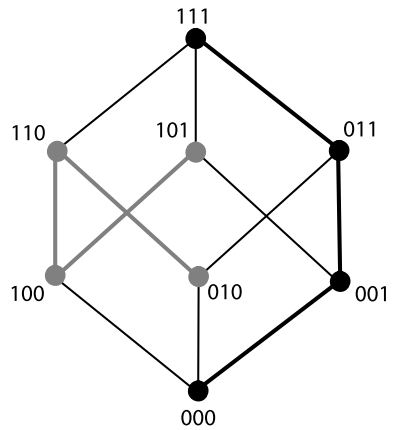
the complement of $\mathcal{C}$ are [100, 101], [010, 110], and [100, 110], respectively; these
are depicted in gray in Fig. 5. Again, notice that this decomposition is not minimal—
namely, $V_{b_3} = [100, 110]$ could be dropped.

5.3 An Algorithm for Primary Decomposition of Pseudo-monomial Ideals

We have already seen that computing the primary decomposition of the neural ideal
$J_{\mathcal{C}}$ is a critical step toward extracting the canonical form CF($J_{\mathcal{C}}$), and that it also
yields a meaningful decomposition of $\mathcal{C}$ in terms of neural activity motifs. Recall
from Sect. 4.3 that $J_{\mathcal{C}}$ is always a *pseudo-monomial ideal*—i.e., $J_{\mathcal{C}}$ is generated by
pseudo-monomials, which are polynomials $f \in \mathbb{F}_2[x_1, \ldots, x_n]$ of the form

$$f = \prod_{i \in \sigma} z_i, \quad \text{where } z_i \in \{x_i, 1 - x_i\} \text{ for any } i \in [n].$$

In this section, we provide an explicit algorithm for finding the primary decomposi-
tion of such ideals.

In the case of *monomial* ideals, there are many algorithms for obtaining the pri-
mary decomposition, and there are already fast implementations of such algorithms
in algebraic geometry software packages such as Singular and Macaulay 2 (Eisenbud
et al. 2002). Pseudo-monomial ideals are closely related to square-free monomial
ideals, but there are some differences which require a bit of care. In particular, if
$J \subseteq F_2[x_1, \ldots, x_n]$ is a pseudo-monomial ideal and $z \in \{x_i, 1 - x_i\}$ for some $i \in [n]$,
then for f a pseudo-monomial:

$$f \in \langle J, z \rangle \quad \not\Rightarrow \quad f \in J \quad \text{or} \quad f \in \langle z \rangle.$$

To see why, observe that $x_1 \in \langle x_1(1 - x_2), x_2 \rangle$, because $x_1 = 1 \cdot x_1(1 - x_2) + x_1 \cdot x_2$,
but x_1 is not a multiple of either $x_1(1 - x_2)$ or x_2. We can nevertheless adapt ideas
from (square-free) monomial ideals to obtain an algorithm for the primary decom-
position of pseudo-monomial ideals. The following lemma allows us to handle the
above complication.

Lemma 5.8 *Let $J \subset \mathbb{F}_2[x_1, \ldots, x_n]$ be a pseudo-monomial ideal, and let $z \in \{x_i, 1 - x_i\}$ for some $i \in [n]$. For any pseudo-monomial f,*

$$f \in \langle J, z \rangle \quad \Rightarrow \quad f \in J \quad or \quad f \in \langle z \rangle \quad or \quad (1 - z)f \in J.$$

The proof is given in Sect. 6.5. Using Lemma 5.8, we can prove the following key lemma for our algorithm, which mimics the case of square-free monomial ideals.

Lemma 5.9 *Let $J \subset \mathbb{F}_2[x_1, \ldots, x_n]$ be a pseudo-monomial ideal, and let $\prod_{i \in \sigma} z_i$ be a pseudo-monomial, with $z_i \in \{x_i, 1 - x_i\}$ for each i. Then*

$$\left\langle J, \prod_{i \in \sigma} z_i \right\rangle = \bigcap_{i \in \sigma} \langle J, z_i \rangle.$$

The proof is given in Sect. 6.5. Note that if $\prod_{i \in \sigma} z_i \in J$, then this lemma implies $J = \bigcap_{i \in \sigma} \langle J, z_i \rangle$, which is the key fact we will use in our algorithm. This is similar to Lemma 2.1 in (Eisenbud et al. 2002, Monomial Ideals Chapter), and suggests a recursive algorithm along similar lines to those that exist for monomial ideals.

The following observation will add considerable efficiency to our algorithm for pseudo-monomial ideals.

Lemma 5.10 *Let $J \subset \mathbb{F}_2[x_1, \ldots, x_n]$ be a pseudo-monomial ideal. For any $z_i \in \{x_i, 1 - x_i\}$ we can write*

$$J = \langle z_i g_1, \ldots, z_i g_k, (1 - z_i) f_1, \ldots, (1 - z_i) f_\ell, h_1, \ldots, h_m \rangle,$$

where the g_j, f_j and h_j are pseudo-monomials that contain no z_i or $1 - z_i$ term. (Note that k, ℓ or m may be zero if there are no generators of the corresponding type.) Then

$$\langle J, z_i \rangle = \langle J|_{z_i = 0}, z_i \rangle = \langle z_i, f_1, \ldots, f_\ell, h_1, \ldots, h_m \rangle.$$

Proof Clearly, the addition of z_i in $\langle J, z_i \rangle$ renders the $z_i g_j$ generators unnecessary. The $(1 - z_i) f_j$ generators can be reduced to just f_j because $f_j = 1 \cdot (1 - z_i) f_j + f_j \cdot z_i$. □

We can now state our algorithm. Recall that an ideal $I \subseteq R$ is *proper* if $I \neq R$.

Algorithm for Primary Decomposition of Pseudo-monomial Ideals

Input: A proper pseudo-monomial ideal $J \subset \mathbb{F}_2[x_1, \ldots, x_n]$. This is presented as $J = \langle g_1, \ldots, g_r \rangle$ with each generator g_i a pseudo-monomial.

Output: Primary decomposition of J. This is returned as a set $\mathcal{P}$ of prime ideals, with $J = \bigcap_{I \in \mathcal{P}} I$.

Step 1 (Initializion Step): Set $\mathcal{P} = \emptyset$ and $D = \{J\}$. Eliminate from the list of generators of J those that are multiples of other generators.

Step 2 (Splitting Step): For each ideal $I \in D$ compute D_I as follows.

Step 2.1: Choose a nonlinear generator $z_{i_1} \cdots z_{i_m} \in I$, where each $z_i \in \{x_i, 1 - x_i\}$, and $m \geq 2$. (Note: the generators of I should always be pseudo-monomials.)

Step 2.2: Set $D_I = \{\langle I, z_{i_1} \rangle, \ldots, \langle I, z_{i_m} \rangle\}$. By Lemma 5.9, we know that

$$I = \bigcap_{k=1}^{m} \langle I, z_{i_k} \rangle = \bigcap_{K \in D_I} K.$$

Step 3 (Reduction Step): For each D_I and each ideal $\langle I, z_i \rangle \in D_I$, reduce the set of generators as follows.

Step 3.1: Set $z_i = 0$ in each generator of I. This yields a "0" for each multiple of z_i, and removes $1 - z_i$ factors in each of the remaining generators. By Lemma 5.10, $\langle I, z_i \rangle = \langle I|_{z_i=0}, z_i \rangle$.

Step 3.2: Eliminate 0s and generators that are multiples of other generators.

Step 3.3: If there is a "1" as a generator, eliminate $\langle I, z_i \rangle$ from D_I as it is not a proper ideal.

Step 4 (Update Step): Update D and $\mathcal{P}$, as follows.

Step 4.1: Set $D = \bigcup D_I$, and remove redundant ideals in D. That is, remove an ideal if it has the same set of generators as another ideal in D.

Step 4.2: For each ideal $I \in D$, if I has only linear generators (and is thus prime), move I to $\mathcal{P}$ by setting $\mathcal{P} = \mathcal{P} \cup I$ and $D = D \setminus I$.

Step 5 (Recursion Step): Repeat steps 2–4 until $D = \emptyset$.

Step 6 (Final Step): Remove redundant ideals of $\mathcal{P}$. That is, remove ideals that are not necessary to preserve the equality $J = \bigcap_{I \in \mathcal{P}} I$.

Proposition 5.11 *This algorithm is guaranteed to terminate, and the final $\mathcal{P}$ is a set of irredundant prime ideals such that $J = \bigcap_{I \in \mathcal{P}} I$.*

Proof For any pseudo-monomial ideal $I \in D$, let $\deg(I)$ be the sum of the degrees of all generating monomials of I. To see that the algorithm terminates, observe that for each ideal $\langle I, z_i \rangle \in D_I$, $\deg(\langle I, z_i \rangle) < \deg(I)$ (this follows from Lemma 5.10). The degrees of elements in D thus steadily decrease with each recursive iteration, until they are removed as prime ideals that are appended to $\mathcal{P}$. At the same time, the size of D is strictly bounded at $|D| \leq 2^{\binom{n}{3}}$, since there are only $\binom{n}{3}$ pseudo-monomials in $\mathbb{F}_2[x_1, \ldots, x_n]$, and thus at most $2^{\binom{n}{3}}$ distinct pseudo-monomial ideals.

By construction, the final $\mathcal{P}$ is an irredundant set of prime ideals. Throughout the algorithm, however, it is always true that $J = (\bigcap_{I \in D} I) \cap (\bigcap_{I \in \mathcal{P}} I)$. Since the final $D = \emptyset$, the final $\mathcal{P}$ satisfies $J = \bigcap_{I \in \mathcal{P}} I$. □

Appendix 1: Proofs

6.1 Proof of Lemmas 3.1 and 3.2

To prove Lemmas 3.1 and 3.2, we need a version of the Nullstellensatz for finite fields. The original "Hilbert's Nullstellensatz" applies when k is an algebraically closed field. It states that if $f \in k[x_1, \ldots, x_n]$ vanishes on $V(J)$, then $f \in \sqrt{J}$. In other words,

$$I(V(J)) = \sqrt{J}.$$

Because we have chosen $k = \mathbb{F}_2 = \{0, 1\}$, we have to be a little careful about the usual ideal-variety correspondence, as there are some subtleties introduced in the case of finite fields. In particular, $J = \sqrt{J}$ in $\mathbb{F}_2[x_1, \ldots, x_n]$ does not imply $I(V(J)) = J$.

The following lemma and theorem are well known. Let $\mathbb{F}_q$ be a finite field of size q, and $\mathbb{F}_q[x_1, \ldots, x_n]$ the n-variate polynomial ring over $\mathbb{F}_q$.

Lemma 6.1 *For any ideal $J \subseteq \mathbb{F}_q[x_1, \ldots, x_n]$, the ideal $J + \langle x_1^q - x_1, \ldots, x_n^q - x_n \rangle$ is a radical ideal.*

Theorem 6.2 (Strong Nullstellensatz in Finite Fields) *For an arbitrary finite field $\mathbb{F}_q$, let $J \subseteq \mathbb{F}_q[x_1, \ldots, x_n]$ be an ideal. Then*

$$I(V(J)) = J + \langle x_1^q - x_1, \ldots, x_n^q - x_n \rangle.$$

6.1.1 Proof of Lemma 3.1

We begin by describing the maximal ideals of $\mathbb{F}_2[x_1, \ldots, x_n]$. Recall that

$$m_v \overset{\text{def}}{=} I(v) = \{ f \in \mathbb{F}_2[x_1, \ldots, x_n] \mid f(v) = 0 \}$$

is the maximal ideal of $\mathbb{F}_2[x_1, \ldots, x_n]$ consisting of all functions that vanish on $v \in \mathbb{F}_2^n$. We will use the notation $\bar{m}_v$ to denote the quotient of m_v in $R_\mathcal{C}$, in cases where $m_v \supset I_\mathcal{C}$.

Lemma 6.3 $m_v = \langle x_1 - v_1, \ldots, x_n - v_n \rangle \subset \mathbb{F}_2[x_1, \ldots, x_n]$, *and is a radical ideal.*

Proof Denote $A_v = \langle x_1 - v_1, \ldots, x_n - v_n \rangle$, and observe that $V(A_v) = \{v\}$. It follows that $I(V(A_v)) = I(v) = m_v$. On the other hand, using the Strong Nullstellensatz in Finite Fields we have

$$I(V(A_v)) = A_v + \langle x_1^2 - x_1, \ldots, x_n^2 - x_n \rangle = A_v,$$

where the last equality is obtained by observing that, since $v_i \in \{0, 1\}$ and $x_i^2 - x_i = x_i(1 - x_i)$, each generator of $\langle x_1^2 - x_1, \ldots, x_n^2 - x_n \rangle$ is already contained in A_v. We conclude that $A_v = m_v$, and the ideal is radical by Lemma 6.1. $\square$

In the proof of Lemma 3.1, we make use of the following correspondence: for any quotient ring R/I, the maximal ideals of R/I are exactly the quotients $\bar{m} = m/I$, where m is a maximal ideal of R that contains I (Atiyah and Macdonald 1969).

Proof of Lemma 3.1 First, recall that because $R_{\mathcal{C}}$ is a Boolean ring, $\text{Spec}(R_{\mathcal{C}}) = \text{maxSpec}(R_{\mathcal{C}})$, the set of all maximal ideals of $R_{\mathcal{C}}$. We also know that the maximal ideals of $\mathbb{F}_2[x_1, \ldots, x_n]$ are exactly those of the form m_v for $v \in \mathbb{F}_2^n$. By the correspondence stated above, to show that $\text{maxSpec}(R_{\mathcal{C}}) = \{\bar{m}_v \mid v \in \mathcal{C}\}$ it suffices to show $m_v \supset I_{\mathcal{C}}$ if and only if $v \in \mathcal{C}$. To see this, note that for each $v \in \mathcal{C}$, $I_{\mathcal{C}} \subseteq m_v$ because, by definition, all elements of $I_{\mathcal{C}}$ are functions that vanish on each $v \in \mathcal{C}$. On the other hand, if $v \notin \mathcal{C}$ then $m_v \not\supseteq I_{\mathcal{C}}$; in particular, the characteristic function $\rho_v \in I_{\mathcal{C}}$ for $v \notin \mathcal{C}$, but $\rho_v \notin m_v$ because $\rho_v(v) = 1$. Hence, the maximal ideals of $R_{\mathcal{C}}$ are exactly those of the form $\bar{m}_v$ for $v \in \mathcal{C}$. $\qquad\square$

We have thus verified that the points in $\text{Spec}(R_{\mathcal{C}})$ correspond to codewords in $\mathcal{C}$. This was expected given our original definition of the neural ring, and suggests that the relations on $\mathbb{F}_2[x_1, \ldots, x_n]$ imposed by $I_{\mathcal{C}}$ are simply relations ensuring that $V(\bar{m}_v) = \emptyset$ for all $v \notin \mathcal{C}$.

6.1.2 Proof of Lemma 3.2

Here we find explicit relations for $I_{\mathcal{C}}$ in the case of an arbitrary neural code. Recall that

$$\rho_v = \prod_{i=1}^{n} ((x_i - v_i) - 1) = \prod_{\{i \mid v_i = 1\}} x_i \prod_{\{j \mid v_j = 0\}} (1 - x_j),$$

and that $\rho_v(x)$ can be thought of as a characteristic function for v, since it satisfies $\rho_v(v) = 1$ and $\rho_v(x) = 0$ for any other $x \in \mathbb{F}_2^n$. This immediately implies that

$$V(J_{\mathcal{C}}) = V\big(\langle \{\rho_v \mid v \notin \mathcal{C}\} \rangle\big) = \mathcal{C}.$$

We can now prove Lemma 3.2.

Proof of Lemma 3.2 Observe that $I_{\mathcal{C}} = I(\mathcal{C}) = I(V(J_{\mathcal{C}}))$, since $V(J_{\mathcal{C}}) = \mathcal{C}$. On the other hand, the Strong Nullstellensatz in Finite Fields implies $I(V(J_{\mathcal{C}})) = J_{\mathcal{C}} + \langle x_1^2 - x_1, \ldots, x_n^2 - x_n \rangle = J_{\mathcal{C}} + \mathcal{B}$. $\qquad\square$

6.2 Proof of Theorem 4.1

Recall that for a given set of receptive fields $\mathcal{U} = \{U_1, \ldots, U_n\}$ in some stimulus space X, the ideal $I_{\mathcal{U}} \subset \mathbb{F}_2[x_1, \ldots, x_n]$ was defined as

$$I_{\mathcal{U}} \stackrel{\text{def}}{=} \left\langle \left\{ x_\sigma \prod_{i \in \tau} (1 - x_i) \;\middle|\; U_\sigma \subseteq \bigcup_{i \in \tau} U_i \right\} \right\rangle.$$

The Boolean relations are present in $I_{\mathcal{U}}$ irrespective of $\mathcal{U}$, as it is always true that $U_i \subseteq U_i$ and this yields the relation $x_i(1 - x_i)$ for each i. By analogy with our definition

of $J_\mathcal{C}$, it makes sense to define an ideal $J_\mathcal{U}$ which is obtained by stripping away the Boolean relations. This will then be used in the proof of Theorem 4.1.

Note that if $\sigma \cap \tau \neq \emptyset$, then for any $i \in \sigma \cap \tau$ we have $U_\sigma \subseteq U_i \subseteq \bigcup_{j \in \tau} U_i$, and the corresponding relation is a multiple of the Boolean relation $x_i(1 - x_i)$. We can thus restrict attention to relations in $I_\mathcal{U}$ that have $\sigma \cap \tau = \emptyset$, so long as we include separately the Boolean relations. These observations are summarized by the following lemma.

Lemma 6.4 $I_\mathcal{U} = J_\mathcal{U} + \langle x_1^2 - x_1, \ldots, x_n^2 - x_n \rangle$, where

$$J_\mathcal{U} \stackrel{\text{def}}{=} \left\langle \left\{ x_\sigma \prod_{i \in \tau} (1 - x_i) \,\middle|\, \sigma \cap \tau = \emptyset \text{ and } U_\sigma \subseteq \bigcup_{i \in \tau} U_i \right\} \right\rangle.$$

Proof of Theorem 4.1 We will show that $J_\mathcal{U} = J_{\mathcal{C}(\mathcal{U})}$ (and thus that $I_\mathcal{U} = I_{\mathcal{C}(\mathcal{U})}$) by showing that each ideal contains the generators of the other.

First, we show that all generating relations of $J_{\mathcal{C}(\mathcal{U})}$ are contained in $J_\mathcal{U}$. Recall that the generators of $J_{\mathcal{C}(\mathcal{U})}$ are of the form

$$\rho_v = \prod_{i \in \text{supp}(v)} x_i \prod_{j \notin \text{supp}(v)} (1 - x_j) \quad \text{for } v \notin \mathcal{C}(\mathcal{U}).$$

If ρ_v is a generator of $J_{\mathcal{C}(\mathcal{U})}$, then $v \notin \mathcal{C}(\mathcal{U})$ and this implies (by the definition of $\mathcal{C}(\mathcal{U})$) that $U_{\text{supp}(v)} \subseteq \bigcup_{j \notin \text{supp}(v)} U_j$. Taking $\sigma = \text{supp}(v)$ and $\tau = [n] \setminus \text{supp}(v)$, we have $U_\sigma \subseteq \bigcup_{j \in \tau} U_j$ with $\sigma \cap \tau = \emptyset$. This in turn tells us (by the definition of $J_\mathcal{U}$) that $x_\sigma \prod_{j \in \tau} (1 - x_j)$ is a generator of $J_\mathcal{U}$. Since $\rho_v = x_\sigma \prod_{j \in \tau} (1 - x_j)$ for our choice of σ and τ, we conclude that $\rho_v \in J_\mathcal{U}$. Hence, $J_{\mathcal{C}(\mathcal{U})} \subseteq J_\mathcal{U}$.

Next, we show that all generating relations of $J_\mathcal{U}$ are contained in $J_{\mathcal{C}(\mathcal{U})}$. If $J_\mathcal{U}$ has generator $x_\sigma \prod_{i \in \tau} (1 - x_i)$, then $U_\sigma \subseteq \bigcup_{i \in \tau} U_i$ and $\sigma \cap \tau = \emptyset$. This in turn implies that $\bigcap_{i \in \sigma} U_i \setminus \bigcup_{j \in \tau} U_j = \emptyset$, and thus (by the definition of $\mathcal{C}(\mathcal{U})$) we have $v \notin \mathcal{C}(\mathcal{U})$ for any v such that $\text{supp}(v) \supseteq \sigma$ and $\text{supp}(v) \cap \tau = \emptyset$. It follows that $J_{\mathcal{C}(\mathcal{U})}$ contains the relation $x_{\text{supp}(v)} \prod_{j \notin \text{supp}(v)} (1 - x_j)$ for any such v. This includes all relations of the form $x_\sigma \prod_{j \in \tau} (1 - x_j) \prod_{k \notin \sigma \cup \tau} P_k$, where $P_k \in \{x_k, 1 - x_k\}$. Taking $f = x_\sigma \prod_{j \in \tau} (1 - x_j)$ in Lemma 6.5 (below), we can conclude that $J_{\mathcal{C}(\mathcal{U})}$ contains $x_\sigma \prod_{j \in \tau} (1 - x_j)$. Hence, $J_\mathcal{U} \subseteq J_{\mathcal{C}(\mathcal{U})}$. $\qquad \square$

Lemma 6.5 *For any $f \in k[x_1, \ldots, x_n]$ and $\tau \subseteq [n]$, the ideal $\langle \{ f \prod_{i \in \tau} P_i \mid P_i \in \{x_i, 1 - x_i\} \} \rangle = \langle f \rangle$.*

Proof First, denote $I_f(\tau) \stackrel{\text{def}}{=} \langle \{ f \prod_{i \in \tau} P_i \mid P_i \in \{x_i, 1 - x_i\} \} \rangle$. We wish to prove that $I_f(\tau) = \langle f \rangle$, for any $\tau \subseteq [n]$. Clearly, $I_f(\tau) \subseteq \langle f \rangle$, since every generator of $I_f(\tau)$ is a multiple of f. We will prove $I_f(\tau) \supseteq \langle f \rangle$ by induction on $|\tau|$.

If $|\tau| = 0$, then $\tau = \emptyset$ and $I_f(\tau) = \langle f \rangle$. If $|\tau| = 1$, so that $\tau = \{i\}$ for some $i \in [n]$, then $I_f(\tau) = \langle f(1 - x_i), f x_i \rangle$. Note that $f(1 - x_i) + f x_i = f$, so $f \in I_f(\tau)$, and thus $I_f(\tau) \supseteq \langle f \rangle$.

Now, assume that for some $\ell \geq 1$ we have $I_f(\sigma) \supseteq \langle f \rangle$ for any $\sigma \subseteq [n]$ with $|\sigma| \leq \ell$. If $\ell \geq n$, we are done, so we need only show that if $\ell < n$, then $I_f(\tau) \supseteq \langle f \rangle$

for any τ of size $\ell + 1$. Consider $\tau \subseteq [n]$ with $|\tau| = \ell + 1$, and let $j \in \tau$ be any element. Define $\tau' = \tau \backslash \{j\}$, and note that $|\tau'| = \ell$. By our inductive assumption, $I_f(\tau') \supseteq \langle f \rangle$. We will show that $I_f(\tau) \supseteq I_f(\tau')$, and hence $I_f(\tau) \supseteq \langle f \rangle$.

Let $g = f \prod_{i \in \tau'} P_i$ be any generator of $I_f(\tau')$ and observe that both $f(1 - x_j) \prod_{i \in \tau'} P_i$ and $f x_j \prod_{i \in \tau'} P_i$ are both generators of $I_f(\tau)$. It follows that their sum, g, is also in $I_f(\tau)$, and hence $g \in I_f(\tau)$ for any generator g of $I_f(\tau')$. We conclude that $I_f(\tau) \supseteq I_f(\tau')$, as desired. $\qquad \square$

6.3 Proof of Theorem 4.3

We begin by showing that $J_{\mathcal{U}}$, first defined in Lemma 6.4, can be generated using the Type 1, Type 2, and Type 3 relations introduced in Sect. 4.2. From the proof of Theorem 4.1, we know that $J_{\mathcal{U}} = J_{\mathcal{C}(\mathcal{U})}$, so the following lemma in fact shows that $J_{\mathcal{C}(\mathcal{U})}$ is generated by the Type 1, 2, and 3 relations as well.

Lemma 6.6 *For $\mathcal{U} = \{U_1, \ldots, U_n\}$ a collection of sets in a stimulus space X,*

$$J_{\mathcal{U}} = \left\langle \{x_\sigma \mid U_\sigma = \emptyset\}, \left\{ \prod_{i \in \tau} (1 - x_i) \;\middle|\; X \subseteq \bigcup_{i \in \tau} U_i \right\}, \right.$$

$$\left\{ x_\sigma \prod_{i \in \tau} (1 - x_i) \;\middle|\; \sigma, \tau \neq \emptyset, \; \sigma \cap \tau = \emptyset, \; U_\sigma \neq \emptyset, \right.$$

$$\left. \left. \bigcup_{i \in \tau} U_i \neq X, \; and \; U_\sigma \subseteq \bigcup_{i \in \tau} U_i \right\} \right\rangle.$$

$J_{\mathcal{U}}$ (equivalently, $J_{\mathcal{C}(\mathcal{U})}$) is thus generated by the Type 1, Type 3 and Type 2 relations, respectively.

Proof Recall that in Lemma 6.4 we defined $J_{\mathcal{U}}$ as

$$J_{\mathcal{U}} \overset{\text{def}}{=} \left\langle \left\{ x_\sigma \prod_{i \in \tau} (1 - x_i) \;\middle|\; \sigma \cap \tau = \emptyset \; and \; U_\sigma \subseteq \bigcup_{i \in \tau} U_i \right\} \right\rangle.$$

Observe that if $U_\sigma = \emptyset$, then we can take $\tau = \emptyset$ to obtain the Type 1 relation x_σ, where we have used the fact that $\prod_{i \in \emptyset}(1 - x_i) = 1$. Any other relation with $U_\sigma = \emptyset$ and $\tau \neq \emptyset$ would be a multiple of x_σ. We can thus write:

$$J_{\mathcal{U}} = \left\langle \{x_\sigma \mid U_\sigma = \emptyset\}, \right.$$

$$\left. \left\{ x_\sigma \prod_{i \in \tau} (1 - x_i) \;\middle|\; \tau \neq \emptyset, \; \sigma \cap \tau = \emptyset, \; U_\sigma \neq \emptyset, \; and \; U_\sigma \subseteq \bigcup_{i \in \tau} U_i \right\} \right\rangle.$$

Next, if $\sigma = \emptyset$ in the second set of relations above, then we have the relation $\prod_{i \in \tau}(1 - x_i)$ with $U_\emptyset = X \subseteq \bigcup_{i \in \tau} U_i$. Splitting off these Type 3 relations, and removing multiples of them that occur if $\bigcup_{i \in \tau} U_i = X$, we obtain the desired result. $\quad \square$

Next, we show that $J_{\mathcal{U}}$ can be generated by reduced sets of the Type 1, Type 2, and Type 3 relations given above. First, consider the Type 1 relations in Lemma 6.6, and observe that if $\tau \subseteq \sigma$, then x_σ is a multiple of x_τ. We can thus reduce the set of Type 1 generators needed by taking only those corresponding to minimal σ with $U_\sigma = \emptyset$:

$$\langle \{x_\sigma \mid U_\sigma = \emptyset\}\rangle = \langle \{x_\sigma \mid \sigma \text{ is minimal w.r.t. } U_\sigma = \emptyset\}\rangle.$$

Similarly, we find for the Type 3 relations:

$$\left\langle \left\{ \prod_{i\in\tau}(1-x_i) \;\middle|\; X \subseteq \bigcup_{i\in\tau}U_i \right\} \right\rangle = \left\langle \left\{ \prod_{i\in\tau}(1-x_i) \;\middle|\; \tau \text{ is minimal w.r.t. } X \subseteq \bigcup_{i\in\tau}U_i \right\} \right\rangle.$$

Finally, we reduce the Type 2 generators. If $\rho \subseteq \sigma$ and $x_\rho \prod_{i\in\tau}(1-x_i) \in J_{\mathcal{U}}$, then we also have $x_\sigma \prod_{i\in\tau}(1-x_i) \in J_{\mathcal{U}}$. So we can restrict ourselves to only those generators for which σ is minimal with respect to $U_\sigma \subseteq \bigcup_{i\in\tau}U_i$. Similarly, we can reduce to minimal τ such that $U_\sigma \subseteq \bigcup_{i\in\tau}U_i$. In summary:

$$\left\langle \left\{ x_\sigma \prod_{i\in\tau}(1-x_i) \;\middle|\; \sigma, \tau \neq \emptyset,\ \sigma \cap \tau = \emptyset,\ U_\sigma \neq \emptyset,\ \bigcup_{i\in\tau}U_i \neq X, \text{ and } U_\sigma \subseteq \bigcup_{i\in\tau}U_i \right\} \right\rangle$$

$$= \left\langle \left\{ x_\sigma \prod_{i\in\tau}(1-x_i) \;\middle|\; \sigma, \tau \neq \emptyset,\ \sigma \cap \tau = \emptyset,\ U_\sigma \neq \emptyset, \right.\right.$$

$$\left.\left. \bigcup_{i\in\tau}U_i \neq X, \text{ and } \sigma, \tau \text{ are each minimal w.r.t. } U_\sigma \subseteq \bigcup_{i\in\tau}U_i \right\} \right\rangle.$$

We can now prove Theorem 4.3.

Proof of Theorem 4.3 Recall that $\mathcal{C} = \mathcal{C}(\mathcal{U})$, and that by the proof of Theorem 4.1 we have $J_{\mathcal{C}(\mathcal{U})} = J_{\mathcal{U}}$. By the reductions given above for the Type 1, 2, and 3 generators, we also know that $J_{\mathcal{U}}$ can be reduced to the form given in the statement of Theorem 4.3. We conclude that $J_{\mathcal{C}}$ can be expressed in the desired form.

To see that $J_{\mathcal{C}}$, as given in the statement of Theorem 4.3, is in canonical form, we must show that the given set of generators is exactly the complete set of minimal pseudo-monomials for $J_{\mathcal{C}}$. First, observe that the generators are all pseudo-monomials. If x_σ is one of the Type 1 relations, and $x_\sigma \in \langle g \rangle$ with $\langle x_\sigma \rangle \neq \langle g \rangle$, then $g = \prod_{i\in\tau}x_i$ for some $\tau \subsetneq \sigma$. Since $U_\tau \neq \emptyset$, however, it follows that $g \notin J_{\mathcal{C}}$ and hence x_σ is a minimal pseudo-monomial of $J_{\mathcal{C}}$. By a similar argument, the Type 2 and Type 3 relations above are also minimal pseudo-monomials in $J_{\mathcal{C}}$.

It remains only to show that there are no additional minimal pseudo-monomials in $J_{\mathcal{C}}$. Suppose $f = x_\sigma \prod_{i\in\tau}(1-x_i)$ is a minimal pseudo-monomial in $J_{\mathcal{C}}$. By Lemma 4.2, $U_\sigma \subseteq \bigcup_{i\in\tau}U_i$ and $\sigma \cap \tau = \emptyset$, so f is a generator in the original definition of $J_{\mathcal{U}}$ (Lemma 6.4). Since f is a minimal pseudo-monomial of $J_{\mathcal{C}}$, there does not exist a $g \in J_{\mathcal{C}}$ such that $g = x_{\sigma'}\prod_{i\in\tau'}(1-x_i)$ with either $\sigma' \subsetneq \sigma$ or $\tau' \subsetneq \tau$. Therefore, σ and τ are each minimal with respect to $U_\sigma \subseteq \bigcup_{i\in\tau}U_i$. We conclude that f is one of the generators for $J_{\mathcal{C}}$ given in the statement of Theorem 4.3. It is a minimal

Type 1 generator if $\tau = \emptyset$, a minimal Type 3 generator if $\sigma = \emptyset$, and is otherwise a minimal Type 2 generator. The three sets of minimal generators are disjoint because the Type 1, Type 2, and Type 3 relations are disjoint, provided $X \neq \emptyset$. $\qquad\square$

6.4 Proof of Proposition 4.5

Note that every polynomial obtained by the canonical form algorithm is a pseudo-monomial of $J_{\mathcal{C}}$. This is because the algorithm constructs products of factors of the form x_i or $1 - x_i$, and then reduces them in such a way that no index is repeated in the final product, and there are no powers of any x_i or $1 - x_i$ factor; we are thus guaranteed to end up with pseudo-monomials. Moreover, since the products each have at least one factor in each prime ideal of the primary decomposition of $J_{\mathcal{C}}$, the pseudo-monomials are all in $J_{\mathcal{C}}$. Proposition 4.5 states that this set of pseudo-monomials is precisely the canonical form $\mathrm{CF}(J_{\mathcal{C}})$.

To prove Proposition 4.5, we will make use of the following technical lemma. Here, z_i, $y_i \in \{x_i, 1 - x_i\}$, and thus any pseudo-monomial in $\mathbb{F}_2[x_1, \ldots, x_n]$ is of the form $\prod_{j \in \sigma} z_j$ for some index set $\sigma \subseteq [n]$.

Lemma 6.7 *If $y_{i_1} \cdots y_{i_m} \in \langle z_{j_1}, \ldots, z_{j_\ell} \rangle$ where $\{i_k\}$ and $\{j_r\}$ are each distinct sets of indices, then $y_{i_k} = z_{j_r}$ for some $k \in [m]$ and $r \in [\ell]$.*

Proof Let $f = y_{i_1} \cdots y_{i_m}$ and $P = \{z_{j_1}, \ldots, z_{j_\ell}\}$. Since $f \in \langle P \rangle$, then $\langle P \rangle = \langle P, f \rangle$, and so $V(\langle P \rangle) = V(\langle P, f \rangle)$. We need to show that $y_{i_k} = z_{j_r}$ for some pair of indices i_k, j_r. Suppose by way of contradiction that there is no i_k, j_r such that $y_{i_k} = z_{j_r}$.

Select $a \in \{0, 1\}^n$ as follows: for each $j_r \in \{j_1, \ldots, j_\ell\}$, let $a_{j_r} = 0$ if $z_{j_r} = x_{j_r}$, and let $a_{j_r} = 1$ if $z_{j_r} = 1 - x_{j_r}$; when evaluating at a, we thus have $z_{j_r}(a) = 0$ for all $r \in [\ell]$. Next, for each $i_k \in \omega \stackrel{\mathrm{def}}{=} \{i_1, \ldots, i_m\} \backslash \{j_1, \ldots, j_\ell\}$, let $a_{i_k} = 1$ if $y_{i_k} = x_{i_k}$, and let $a_{i_k} = 0$ if $y_{i_k} = 1 - x_{i_k}$, so that $y_{i_k}(a) = 1$ for all $i_k \in \omega$. For any remaining indices t, let $a_t = 1$. Because we have assumed that $y_{i_k} \neq z_{j_r}$ for any i_k, j_r pair, we have for any $i \in \{i_1, \ldots, i_m\} \cap \{j_1, \ldots, j_\ell\}$ that $y_i(a) = 1 - z_i(a) = 1$. It follows that $f(a) = 1$.

Now, note that $a \in V(\langle P \rangle)$ by construction. We must therefore have $a \in V(\langle P, f \rangle)$, and hence $f(a) = 0$, a contradiction. We conclude that there must be some i_k, j_r with $y_{i_k} = z_{j_r}$, as desired. $\qquad\square$

We can now prove the proposition.

Proof of Proposition 4.5 It suffices to show that after step 4 of the algorithm, the reduced set $\widetilde{\mathcal{M}}(J_{\mathcal{C}})$ consists entirely of pseudo-monomials of $J_{\mathcal{C}}$, and includes all *minimal* pseudo-monomials of $J_{\mathcal{C}}$. If this is true, then after removing multiples of lower-degree elements in step 5 we are guaranteed to obtain the set of minimal pseudo-monomials, $\mathrm{CF}(J_{\mathcal{C}})$, since it is precisely the nonminimal pseudo-monomials that will be removed in the final step of the algorithm.

Let $J_{\mathcal{C}} = \bigcap_{i=1}^s P_i$ be the primary decomposition of $J_{\mathcal{C}}$, with each P_i a prime ideal of the form $P_i = \langle z_{j_1}, \ldots, z_{j_\ell} \rangle$. Recall that $\mathcal{M}(J_{\mathcal{C}})$, as defined in step 3 of the

algorithm, is precisely the set of all polynomials g that are obtained by choosing one linear factor from the generating set of each P_i:

$$\mathcal{M}(J_\mathcal{C}) = \{g = z_{p_1} \cdots z_{p_s} \mid z_{p_i} \text{ is a linear generator of } P_i\}.$$

Furthermore, recall that $\widetilde{\mathcal{M}}(J_\mathcal{C})$ is obtained from $\mathcal{M}(J_\mathcal{C})$ by the reductions in step 4 of the algorithm. Clearly, all elements of $\widetilde{\mathcal{M}}(J_\mathcal{C})$ are pseudo-monomials that are contained in $J_\mathcal{C}$.

To show that $\widetilde{\mathcal{M}}(J_\mathcal{C})$ contains all *minimal* pseudo-monomials of $J_\mathcal{C}$, we will show that if $f \in J_\mathcal{C}$ is a pseudo-monomial, then there exists another pseudo-monomial $h \in \widetilde{\mathcal{M}}(J_\mathcal{C})$ (possibly the same as f) such that $h|f$. To see this, let $f = y_{i_1} \cdots y_{i_m}$ be a pseudo-monomial of $J_\mathcal{C}$. Then, $f \in P_i$ for each $i \in [s]$. For a given $P_i = \langle z_{j_1}, \ldots, z_{j_\ell} \rangle$, by Lemma 6.7 we have $y_{i_k} = z_{j_r}$ for some $k \in [m]$ and $r \in [\ell]$. In other words, each prime ideal P_i has a generating term, call it z_{p_i}, that appears as one of the linear factors of f. Setting $g = z_{p_1} \cdots z_{p_s}$, it is clear that $g \in \mathcal{M}(J_\mathcal{C})$ and that either $g|f$, or $z_{p_i} = z_{p_j}$ for some distinct pair i, j. By removing repeated factors in g one obtains a pseudo-monomial $h \in \widetilde{\mathcal{M}}(J_\mathcal{C})$ such that $h|g$ and $h|f$. If we take f to be a minimal pseudo-monomial, we find $f = h \in \widetilde{\mathcal{M}}(J_\mathcal{C})$. □

6.5 Proof of Lemmas 5.8 and 5.9

Here, we prove Lemmas 5.8 and 5.9, which underlie the primary decomposition algorithm.

Proof of Lemma 5.8 Assume $f \in \langle J, z \rangle$ is a pseudo-monomial. Then $f = z_{i_1} z_{i_2} \cdots z_{i_r}$, where $z_i \in \{x_i, 1 - x_i\}$ for each i, and the i_k are distinct. Suppose $f \notin \langle z \rangle$. This implies $z_{i_k} \neq z$ for all factors appearing in f. We will show that either $f \in J$ or $(1 - z)f \in J$.

Since J is a pseudo-monomial ideal, we can write

$$J = \langle zg_1, \ldots, zg_k, (1-z)f_1, \ldots, (1-z)f_l, h_1, \ldots, h_m \rangle,$$

where the g_j, f_j, and h_j are pseudo-monomials that contain no z or $1 - z$ term. This means

$$f = z_{i_1} z_{i_2} \cdots z_{i_r} = z \sum_{j=1}^{k} u_j g_j + (1-z) \sum_{j=1}^{l} v_j f_j + \sum_{j=1}^{m} w_j h_j + yz,$$

for polynomials u_j, v_j, w_j, and $y \in \mathbb{F}_2[x_1, \ldots, x_n]$. Now consider what happens if we set $z = 0$ in f:

$$f|_{z=0} = z_{i_1} z_{i_2} \cdots z_{i_r}|_{z=0} = \sum_{j=1}^{l} v_j \Big|_{z=0} f_j + \sum_{j=1}^{m} w_j \Big|_{z=0} h_j.$$

Next, observe that after multiplying the above by $(1 - z)$ we obtain an element of J:

$$(1-z)f|_{z=0} = (1-z) \sum_{j=1}^{l} v_j \Big|_{z=0} f_j + (1-z) \sum_{j=1}^{m} w_j \Big|_{z=0} h_j \in J,$$

since $(1 - z)f_j \in J$ for $j = 1, \ldots, l$ and $h_j \in J$ for $j = 1, \ldots, m$. There are two cases:

Case 1: If $1 - z$ is a factor of f, say $z_{i_1} = 1 - z$, then $f|_{z=0} = z_{i_2} \cdots z_{i_r}$ and thus $f = (1 - z)f|_{z=0} \in J$.

Case 2: If $1 - z$ is *not* a factor of f, then $f = f|_{z=0}$. Multiplying by $1 - z$ we obtain $(1 - z)f \in J$.

We thus conclude that $f \notin \langle z \rangle$ implies $f \in J$ or $(1 - z)f \in J$. $\qquad\square$

Proof of Lemma 5.9 Clearly, $\langle J, z_\sigma \rangle \subseteq \bigcap_{i \in \sigma} \langle J, z_i \rangle$. To see the reverse inclusion, consider $f \in \bigcap_{i \in \sigma} \langle J, z_i \rangle$. We have three cases.

Case 1: $f \in J$. Then $f \in \langle J, z_\sigma \rangle$.

Case 2: $f \notin J$, but $f \in \langle z_i \rangle$ for all $i \in \sigma$. Then $f \in \langle z_\sigma \rangle$, and hence $f \in \langle J, z_\sigma \rangle$.

Case 3: $f \notin J$ and $f \notin \langle z_i \rangle$ for all $i \in \tau \subset \sigma$, but $f \in \langle z_j \rangle$ for all $j \in \sigma \setminus \tau$. Without loss of generality, we can rearrange indices so that $\tau = \{1, \ldots, m\}$ for $m \geq 1$. By Lemma 5.8, we have $(1 - z_i)f \in J$ for all $i \in \tau$. We can thus write

$$f = (1 - z_1)f + z_1(1 - z_2)f + \cdots + z_1 \cdots z_{m-1}(1 - z_m)f + z_1 \cdots z_m f.$$

Observe that the first m terms are each in J. On the other hand, $f \in \langle z_j \rangle$ for each $j \in \sigma \setminus \tau$ implies that the last term is in $\langle z_\tau \rangle \cap \langle z_{\sigma \setminus \tau} \rangle = \langle z_\sigma \rangle$. Hence, $f \in \langle J, z_\sigma \rangle$.

We may thus conclude that $\bigcap_{i \in \sigma} \langle J, z_i \rangle \subseteq \langle J, z_\sigma \rangle$, as desired. $\qquad\square$

6.6 Proof of Theorem 5.4

Recall that $J_\mathcal{C}$ is always a proper pseudo-monomial ideal for any nonempty neural code $\mathcal{C} \subseteq \{0, 1\}^n$. Theorem 5.4 is thus a direct consequence of the following proposition.

Proposition 6.8 *Suppose $J \subset \mathbb{F}_2[x_1, \ldots, x_n]$ is a proper pseudo-monomial ideal. Then, J has a unique irredundant primary decomposition of the form $J = \bigcap_{a \in \mathcal{A}} \mathbf{p}_a$, where $\{\mathbf{p}_a\}_{a \in \mathcal{A}}$ are the minimal primes over J.*

Proof By Proposition 5.11, we can always (algorithmically) obtain an irredundant set $\mathcal{P}$ of prime ideals such that $J = \bigcap_{I \in \mathcal{P}} I$. Furthermore, each $I \in \mathcal{P}$ has the form $I = \langle z_{i_1}, \ldots, z_{i_k} \rangle$, where $z_i \in \{x_i, 1 - x_i\}$ for each i. Clearly, these ideals are all prime ideals of the form $\mathbf{p}_a$ for $a \in \{0, 1, *\}$. It remains only to show that this primary decomposition is unique, and that the ideals $\{\mathbf{p}_a\}_{a \in \mathcal{A}}$ are the minimal primes over J. This is a consequence of some well-known facts summarized in Lemmas 6.9 and 6.10, below. First, observe by Lemma 6.9 that J is a radical ideal. Lemma 6.10 then tells us that the decomposition in terms of minimal primes is the unique irredundant primary decomposition for J. $\qquad\square$

Lemma 6.9 *If J is the intersection of prime ideals, $J = \bigcap_{i=1}^{\ell} \mathbf{p}_i$, then J is a radical ideal.*

Proof Suppose $p^n \in J$. Then $p^n \in \mathbf{p}_i$ for all $i \in [\ell]$, and hence $p \in \mathbf{p}_i$ for all $i \in [\ell]$. Therefore, $p \in J$. $\qquad\square$

The following fact about the primary decomposition of radical ideals is true over any field, as a consequence of the Lasker–Noether theorems (Cox et al. 1997, pp. 204–209).

Lemma 6.10 *If J is a proper radical ideal, then it has a unique irredundant primary decomposition consisting of the minimal prime ideals over J.*

Appendix 2: Neural Codes on Three Neurons

See Table 1 and Figs. 6 and 7.

Table 1 Forty permutation-inequivalent codes, each containing 000, on three neurons

Label	Code $\mathcal{C}$	Canonical Form CF($J_{\mathcal{C}}$)
A1	000, 100, 010, 001, 110, 101, 011, 111	$\emptyset$
A2	000, 100, 010, 110, 101, 111	$x_3(1-x_1)$
A3	000, 100, 010, 001, 110, 101, 111	$x_2 x_3(1-x_1)$
A4	000, 100, 010, 110, 101, 011, 111	$x_3(1-x_1)(1-x_2)$
A5	000, 100, 010, 110, 111	$x_3(1-x_1),\, x_3(1-x_2)$
A6	000, 100, 110, 101, 111	$x_2(1-x_1),\, x_3(1-x_1)$
A7	000, 100, 010, 101, 111	$x_3(1-x_1),\, x_1 x_2(1-x_3)$
A8	000, 100, 010, 001, 110, 111	$x_1 x_3(1-x_2),\, x_2 x_3(1-x_1)$
A9	000, 100, 001, 110, 011, 111	$x_3(1-x_2),\, x_2(1-x_1)(1-x_3)$
A10	000, 100, 010, 101, 011, 111	$x_3(1-x_1)(1-x_2),\, x_1 x_2(1-x_3)$
A11	000, 100, 110, 101, 011, 111	$x_2(1-x_1)(1-x_3),\, x_3(1-x_1)(1-x_2)$
A12	000, 100, 110, 111	$x_3(1-x_1),\, x_3(1-x_2),\, x_2(1-x_1)$
A13	000, 100, 010, 111	$x_3(1-x_1),\, x_3(1-x_2),\, x_1 x_2(1-x_3)$
A14	000, 100, 010, 001, 111	$x_1 x_2(1-x_3),\, x_2 x_3(1-x_1),\, x_1 x_3(1-x_2)$
A15	000, 110, 101, 011, 111	$x_1(1-x_2)(1-x_3),\, x_2(1-x_1)(1-x_3),\, x_3(1-x_1)(1-x_2)$
A16*	000, 100, 011, 111	$x_2(1-x_3),\, x_3(1-x_2)$
A17*	000, 110, 101, 111	$x_2(1-x_1),\, x_3(1-x_1),\, x_1(1-x_2)(1-x_3)$
A18*	000, 100, 111	$x_2(1-x_1),\, x_2(1-x_3),\, x_3(1-x_1),\, x_3(1-x_2)$
A19*	000, 110, 111	$x_3(1-x_1),\, x_3(1-x_2),\, x_1(1-x_2),\, x_2(1-x_1)$
A20*	000, 111	$x_1(1-x_2),\, x_2(1-x_3),\, x_3(1-x_1),\, x_1(1-x_3),\, x_2(1-x_1),\, x_3(1-x_2)$
B1	000, 100, 010, 001, 110, 101	$x_2 x_3$
B2	000, 100, 010, 110, 101	$x_2 x_3,\, x_3(1-x_1)$
B3	000, 100, 010, 101, 011	$x_1 x_2,\, x_3(1-x_1)(1-x_2)$

Table 1 (*Continued*)

Label	Code $\mathcal{C}$	Canonical Form $\mathrm{CF}(J_\mathcal{C})$
B4	000, 100, 110, 101	$x_2x_3, x_2(1-x_1), x_3(1-x_1)$
B5	000, 100, 110, 011	$x_1x_3, x_3(1-x_2), x_2(1-x_1)(1-x_3)$
B6*	000, 110, 101	$x_2x_3, x_2(1-x_1), x_3(1-x_1), x_1(1-x_2)(1-x_3)$
C1	000, 100, 010, 001, 110	x_1x_3, x_2x_3
C2	000, 100, 010, 101	$x_1x_2, x_2x_3x_3, x_3(1-x_1)$
C3*	000, 100, 011	$x_1x_2, x_1x_3, x_2(1-x_3), x_3(1-x_2)$
D1	000, 100, 010, 001	x_1x_2, x_2x_3, x_1x_3
E1	000, 100, 010, 001, 110, 101, 011	$x_1x_2x_3$
E2	000, 100, 010, 110, 101, 011	$x_1x_2x_3, x_3(1-x_1)(1-x_2)$
E3	000, 100, 110, 101, 011	$x_1x_2x_3, x_2(1-x_1)(1-x_2), x_3(1-x_1)(1-x_2)$
E4	000, 110, 011, 101	$x_1x_2x_3, x_1(1-x_2)(1-x_3), x_2(1-x_1)(1-x_3), x_3(1-x_1)(1-x_2)$
F1*	000, 100, 010, 110	x_3
F2*	000, 100, 110	$x_3, x_2(1-x_1)$
F3*	000, 110	$x_3, x_1(1-x_2), x_2(1-x_1)$
G1*	000, 100	x_2, x_3
H1*	000	x_1, x_2, x_3
I1*	000, 100, 010	x_3, x_1x_2

Note: Labels A–I indicate the various families of Type 1 relations present in $\mathrm{CF}(J_\mathcal{C})$, organized as follows (up to permutation of indices): (A) None, (B) $\{x_1x_2\}$, (C) $\{x_1x_2, x_2x_3\}$, (D) $\{x_1x_2, x_2x_3, x_1x_3\}$, (E) $\{x_1x_2x_3\}$, (F) $\{x_1\}$, (G) $\{x_1, x_2\}$, (H) $\{x_1, x_2, x_3\}$, (I) $\{x_1, x_2, x_3\}$. All codes within the same A–I series share the same simplicial complex, $\Delta(\mathcal{C})$. The *s denote codes that have $U_i = \emptyset$ for at least one receptive field (as in the F, G, H, and I series) as well as codes that require $U_1 = U_2$ or $U_1 = U_2 \cup U_3$ (up to permutation of indices); these are considered to be highly degenerate. The remaining 27 codes are depicted with receptive field diagrams (Fig. 6) and Boolean lattice diagrams (Fig. 7)
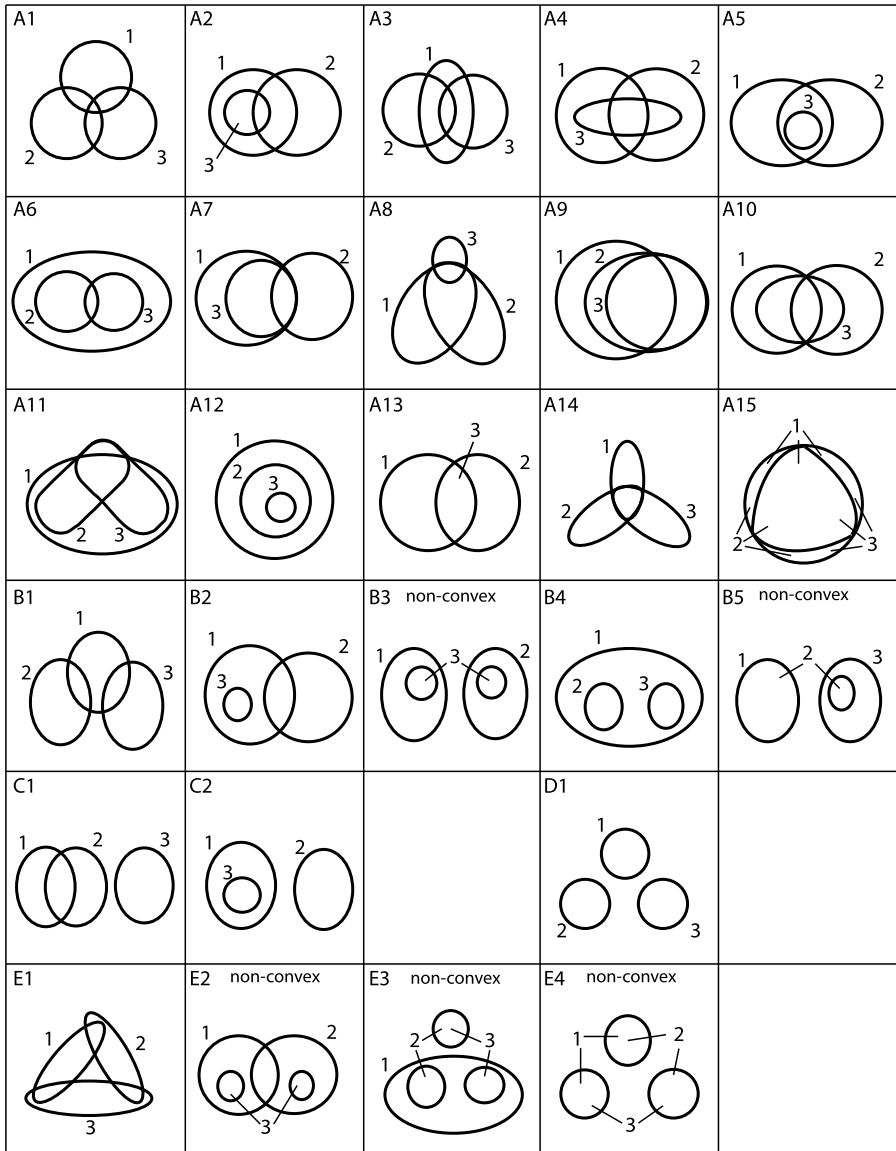
Fig. 6 Receptive field diagrams for the 27 non-∗ codes on three neurons listed in Table 1. Codes that admit no realization as a convex RF code are labeled "non-convex." The code E2 is the one from Lemma 2.2, while A1 and A12 are permutation-equivalent to the codes in Fig. 3A and C, respectively. Deleting the all-zeros codeword from A6 and A4 yields codes permutation-equivalent to those in Fig. 3B and D, respectively
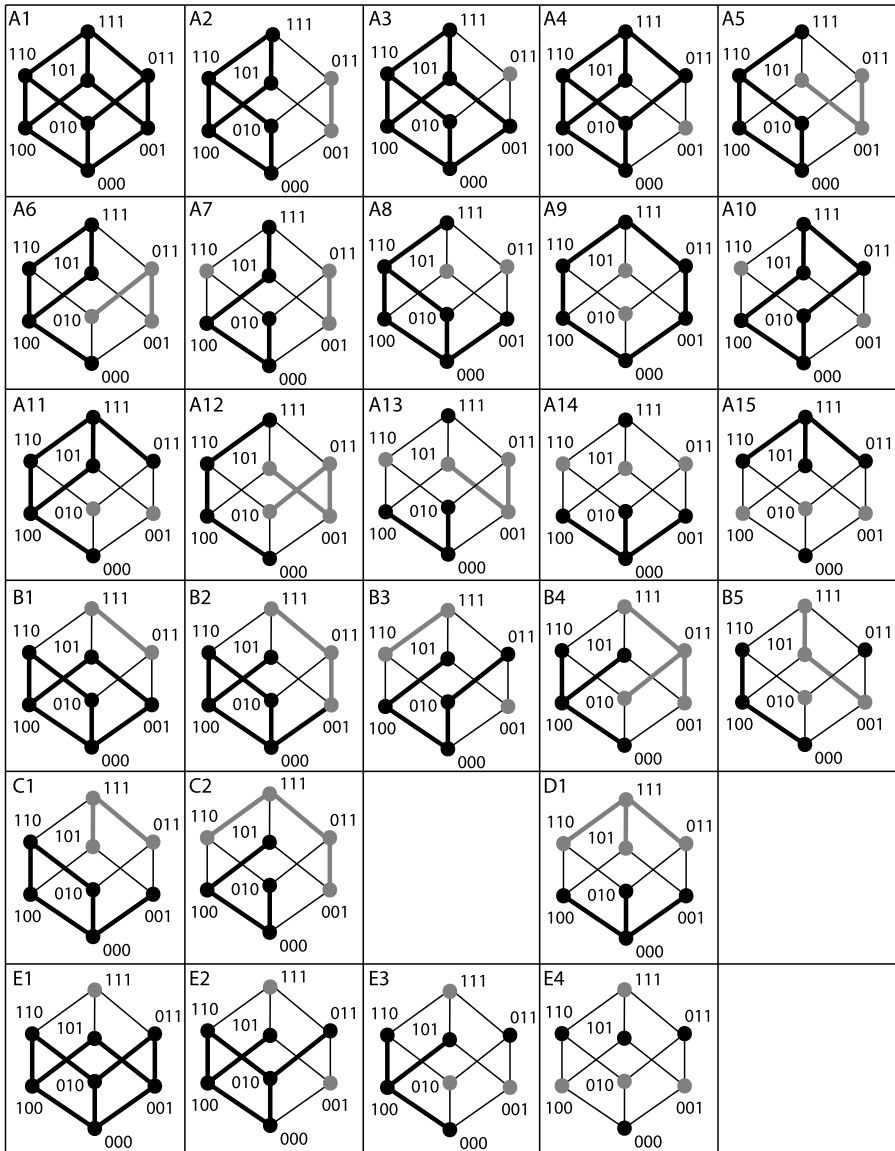
Fig. 7 Boolean lattice diagrams for the 27 non-∗ codes on three neurons listed in Table 1. Interval decompositions (see Sect. 5.2) for each code are depicted in *black*, while decompositions of code complements, arising from CF($J_{\mathcal{C}}$), are shown in *gray*. Thin *black lines* connect elements of the Boolean lattice that are Hamming distance 1 apart. Note that the lattice in A12 is permutation-equivalent to the one depicted in Fig. 5

References

Atiyah, M. F., & Macdonald, I. G. (1969). *Introduction to commutative algebra*. Reading: Addison–Wesley

Averbeck, B. B., Latham, P. E., & Pouget, A. (2006). Neural correlations, population coding and computation. *Nat. Rev. Neurosci.*, *7*(5), 358–366.

Ben-Yishai, R., Bar-Or, R. L., & Sompolinsky, H. (1995). Theory of orientation tuning in visual cortex. *Proc. Natl. Acad. Sci. USA*, *92*(9), 3844–3848.

Brown, E. N., Frank, L. M., Tang, D., Quirk, M. C., & Wilson, M. A. (1998). A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells. *J. Neurosci.*, *18*(18), 7411–7425.

Cox, D., Little, J., & O'Shea, D. (1997). An introduction to computational algebraic geometry and commutative algebra. In *Undergraduate texts in mathematics: Ideals, varieties, and algorithms* (2nd ed.). New York: Springer.

Curto, C., & Itskov, V. (2008). Cell groups reveal structure of stimulus space. *PLoS Comput. Biol.*, *4*(10).

Danzer, L., Grünbaum, B., & Klee, V. (1963). Helly's theorem and its relatives. In *Proc. sympos. pure math.* (Vol. VII, pp. 101–180). Providence: Am. Math. Soc.

Deneve, S., Latham, P. E., & Pouget, A. (1999). Reading population codes: a neural implementation of ideal observers. *Nat. Neurosci.*, *2*(8), 740–745.

Eisenbud, D., Grayson, D. R., Stillman, M., & Sturmfels, B. (Eds.) (2002). *Algorithms and computation in mathematics*: *Vol. 8. Computations in algebraic geometry with Macaulay 2*. Berlin: Springer.

Hatcher, A. (2002). *Algebraic topology*. Cambridge: Cambridge University Press.

Jarrah, A., Laubenbacher, R., Stigler, B., & Stillman, M. (2007). Reverse-engineering of polynomial dynamical systems. *Adv. Appl. Math.*, *39*, 477–489.

Kalai, G. (1984). Characterization of f-vectors of families of convex sets in $\mathbf{R}^d$. I. Necessity of Eckhoff's conditions. *Isr. J. Math.*, *48*(2–3), 175–195.

Kalai, G. (1986). Characterization of f-vectors of families of convex sets in $\mathbf{R}^d$. II. Sufficiency of Eckhoff's conditions. *J. Comb. Theory, Ser. A*, *41*(2), 167–188.

Ma, W. J., Beck, J. M., Latham, P. E., & Pouget, A. (2006). Bayesian inference with probabilistic population codes. *Nat. Neurosci.*, *9*(11), 1432–1438.

McNaughton, B. L., Battaglia, F. P., Jensen, O., Moser, E. I., & Moser, M. B. (2006). Path integration and the neural basis of the 'cognitive map'. *Nat. Rev. Neurosci.*, *7*(8), 663–678.

Miller, E., & Sturmfels, B. (2005). *Graduate texts in mathematics: Combinatorial commutative algebra*. Berlin: Springer.

Nirenberg, S., & Latham, P. E. (2003). Decoding neuronal spike trains: how important are correlations? *Proc. Natl. Acad. Sci. USA*, *100*(12), 7348–7353.

O'Keefe, J., & Dostrovsky, J. (1971). The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Res.*, *34*(1), 171–175.

Osborne, L., Palmer, S., Lisberger, S., & Bialek, W. (2008). The neural basis for combinatorial coding in a cortical population response. *J. Neurosci.*, *28*(50), 13522–13531.

Pistone, G., Riccomagno, E., & Wynn, H. P. (2001). Computational commutative algebra in statistics. In *Monographs on statistics and applied probability.: Vol. 89. Algebraic statistics*, Boca Raton: Chapman & Hall/CRC Press.

Schneidman, E., Berry, M. II., Segev, R., & Bialek, W. (2006a). Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, *440*(20), 1007–1012.

Schneidman, E., Puchalla, J., Segev, R., Harris, R., Bialek, W., & Berry II, M. (2006b). Synergy from silence in a combinatorial neural code. arXiv:q-bio.NC/0607017.

Shiu, A., & Sturmfels, B. (2010). Siphons in chemical reaction networks. *Bull. Math. Biol.*, *72*(6), 1448–1463.

Stanley, R. (2004). *Progress in mathematics: Combinatorics and commutative algebra*. Boston: Birkhäuser.

Veliz-Cuba, A. (2012). An algebraic approach to reverse engineering finite dynamical systems arising from biology. *SIAM J. Appl. Dyn. Syst.*, *11*(1), 31–48.

Watkins, D. W., & Berkley, M. A. (1974). The orientation selectivity of single neurons in cat striate cortex. *Exp. Brain Res.*, *19*, 433–446.