University College Cork

MS4090 - Final Year Project

Neural Codes and Neural Ideals

Author:
Matthew Kiely

Supervisor:
Dr. Anca Mustata

December 20, 2023

Abstract

Neuroscience is a subject which, in recent years, has been undergoing a period of experimental expansion by utilising methods in algebraic geometry and topology. The neural ideal is one tool in algebraic geometry that has been applied to representations of neural activity known as neural codes in an effort to characterise two conceptual objects - receptive fields $\{U_1, ...U_n\}$ and the stimulus space $X \in \mathbb{R}^d$ with $U_i \in X$. Experimentally and mathematically, it has been found that convex receptive fields are of particular significance. This final year project introduces these objects, as well as the mathematical framework for the neural ideal and its ability to derive convex receptive fields relations. The canonical form of the neural ideal is developed, which outlines a minimal set of convex receptive field relationships. Methods in algebraic geometry and algebraic topology are presented which determine the convexity or non-convexity of neural code subtypes. Finally, some recent developments in this subfield of mathematical neuroscience are addressed.

Contents

Acknowledgements

I would like to express my deepest gratitude to my supervisor Anca Mustata. Her patience, time and support throughout this project has proven extremely invaluable.

I would also like to thank the UCC School of Mathematical Sciences department, without whom this experience would not have been possible.

1 Introduction and Motivation

A central axiom of neuroscience is that the brain forms perceptions of the world by responding to external stimuli. Stimuli come in a huge variety of forms (light, sound etc.), while the response always takes the form of electrical neural activity. Neuroscientists have found that there exist certain families of neuronal cells that respond strongly to specific types of stimuli. For instance, place cells are a type of neuron studied in mice that respond to position within a fixed environment [1]. In 2014, the Nobel Prize in Physiology and Medicine was awarded to John O'Keefe (who received an honourary degree from UCC, also in 2014), May-Britt Moser and Edvard I. Moser "for their discoveries of (such) cells that constitute a positioning system in the brain" [2].

In some cases, it is very easy to conceptualise a set of stimuli as a space. Indeed, in the example above, we can regard the *stimulus space* as the animal's fixed environment - a subset of $\mathbb{R}^2$ or $\mathbb{R}^3$. As another example, researchers have discovered neurons that respond to pure frequency tones [3] - the stimulus space in this case would resemble a frequency range - a subset of $\mathbb{R}$. However it is not always so simple to describe the stimulus space for a given stimulus. Take for instance the olfaction system, the stimulus space of which is likely to be highly-dimensional and is currently poorly understood.

Another object of interest to neuroscientists are receptive fields, regions of the stimulus space that correspond to the activation of individual neurons. Receptive fields are also measurable by monitoring the activity of single neurons for different stimuli (points in the stimulus space). In many studied cases, including the case of place cells, the receptive fields are well approximated by convex sets. Hence, stimulus spaces containing convex receptive fields are of particular interest in neuroscience.

The central question of this report is as follows - given a dataset of neural activity, what can we conclude about the underlying stimulus space? And how do our conclusions change if we assume the associated receptive fields to be convex? In 2013 Curto et al. presented algebraic tools with the aim of characterising the stimulus space in these cases [4]. Besides this paper, within mathematical biology, associating algebraic objects to combinatorial ones has yielded significant results [5].

This report presents the algebraic tools introduced by Curto et.al. which are being used to understand stimulus space structure and receptive field convexity. We begin by presenting a representation of neural activity known as the *neural code* and move on to *receptive fields*. Tools in algebraic geometry such as the *neural ideal* are then introduced to establish the relationship between neural activity and receptive field structure. We describe methods to characterise neural codes with convex receptive field representations, including a method sourced from algebraic topology. Finally, we briefly outline some advances made in this field since the work of Curto et al.

2 The Neural Code

Given a set of neurons labelled $\{1, 2, ..., n\} = [n]$, the associated neural code consists of binary codewords describing their neural activity.

Definition 2.1. A *neural codeword* or *codeword* is a vector in $\{0,1\}^n$:

$$c = (c_1, c_2, ..., c_n) \tag{2.1}$$

where $c_i = 0$ or $c_i = 1$. We henceforth write c in the form $c = c_1c_2...c_n$ for convenience. E.g. $c = 101$. For a neuron i, the component c_i describes its activity[1].

Definition 2.2. A collection of codewords C is called a *neural code* or *code*, and is a subset of $\{0,1\}^n$.

Definition 2.3. We also define the *support* of codewords and codes:

$$\text{supp}(c) = \{i \in [n] \,|\, c_i = 1\} \subseteq [n] \tag{2.2}$$

$$\text{supp}(C) = \{\text{supp}(c) \,|\, c \in C\} \subseteq 2^{[n]} \tag{2.3}$$

The support gives us flexibility to view neural activity in terms of binary digits, or in terms of active neurons.

2.1 Receptive Fields

For each neuron, we can assign it a region in the stimulus space in which it is active. We call such a region a *receptive field*. Receptive fields are defined to be open [2].

We can also define the *receptive field code* or *RF code*:

Definition 2.4. Let X be a stimulus space and let $\mathcal{U} = \{U_1, U_2, ..., U_n\}$ be a collection of open sets with each $U_i \subseteq X$ the receptive field of neuron i. The receptive field code $\mathcal{C}(\mathcal{U})$ is the set of all codewords corresponding to stimuli in X:

$$\mathcal{C}(\mathcal{U}) = \{c \in \{0,1\}^n \,|\, (\bigcap_{i \in \text{supp}(c)} U_i) \setminus (\bigcup_{j \notin \text{supp}(c)} U_j) \neq \emptyset\} \tag{2.4}$$

This construction begs the question as to whether it is even possible to represent a code C as an RF code $C(\mathcal{U})$ i.e. if $C = C(\mathcal{U})$. Intuitively, we should expect this to be dependent on the properties of the receptive fields. In the case where we simply demand the receptive fields to be open, the question becomes trivial:

Theorem 2.5. *For a neural code C, there exists a stimulus space $X \subseteq \mathbb{R}^d$ and a collection of open sets (receptive fields) $\mathcal{U} = \{U_i\}_{i=1}^n$ in X such that $C = C(\mathcal{U})$.*

Proof. Let $C = \{c_1, ..., c_m\}$. For each $c_i \in C$ choose a distinct point $x_{c_i} \in \mathbb{R}$ with open neighbourhood N_{c_i} such that no two neighbourhoods intersect.

Construct the receptive field $U_j = \bigcup_{j \in \text{supp}(c_k)} N_{c_k}$.
U_j is the union of all open neighbourhoods whose associated codewords have their j-th neuron activated. Then $\mathcal{U} = \{U_i\}_{i=1}^n$ and $X = \bigcup_{i=1}^m N_{c_i}$ By construction, $C = C(\mathcal{U})$.

□

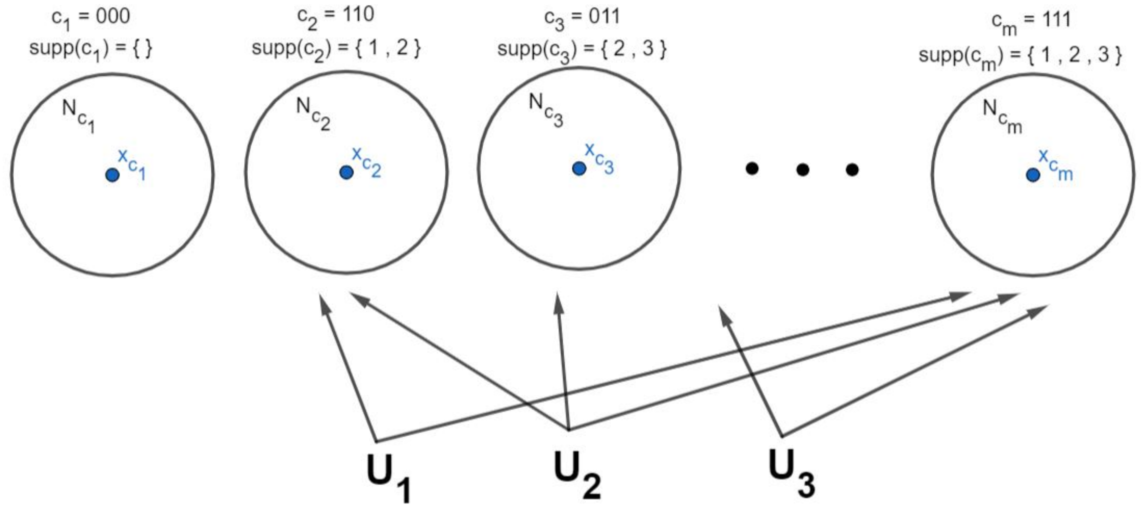
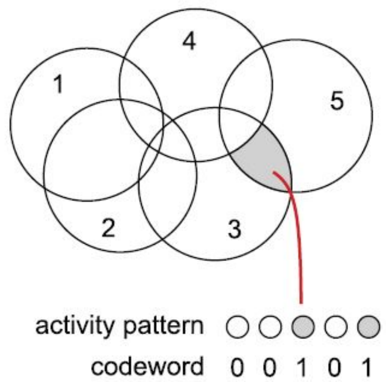


Figure 1: Construction from Theorem 2.5 for $n = 3$ and a code C such that $\{000, 110, 011, 111\} \subset C$.

By requiring the receptive fields to satisfy openness only, we don't learn anything new about the stimulus space for a code C. The dimension d is arbitrary for any code, and no topological information about X is gained.

Thus, if we wish to characterise X, it makes sense to require an additional property of the receptive fields. The property we will choose is *convexity*. Henceforth, for a collection of convex receptive fields $\mathcal{U}$, if $C = C(\mathcal{U})$ then we say C is *convex*. As mentioned previously, the choice of convexity as a new property has grounding in neuroscientific research [1, 3].

Figure 2: Example of convex receptive fields $\mathcal{U}$ for $n = 5$. The shaded region can be written:
$(\bigcap_{i\in\{3,5\}} U_i)\backslash(\bigcup_{j\in\{1,2,4\}} U_j)$
And hence represents the codeword 00101 $\in C(\mathcal{U})$.



2.2 Convexity and Stimulus Space Structure

Adding convexity as a property allows us to extract information about the underlying stimulus space, including dimensional bounds and topological features. We do this by

[1]'Neural activity' is generally encoded as follows - monitor the neural activity of interest over a period of time, then split the time period into intervals. If neuron i is 'silent' in a given interval, assign its codeword component $c_i = 0$, and if it 'fires' at least once, assign $c_i = 1$. Hence, a complete codeword can be viewed as the activity of all neurons for a chosen time interval, and a code can be viewed as samplings of neural activity over the total time period.

[2]Work has also been done on closed receptive fields. See 5

Helly's Theorem and the Nerve Theorem.

Helly's Theorem

Theorem 2.6. ***(Helly's Theorem)*** *For convex sets $U_1, ..., U_k \in \mathbb{R}^d$ with $d < k$, if the intersection of every $d+1$ of these sets is non-empty, then the full intersection $\bigcap_{i=1}^{k} U_i$ is also non-empty.*

Definition 2.7. For a code C, we say the *minimal embedding dimension* of C is the minimal positive integer d for which the code has a convex realisation in $X \subseteq \mathbb{R}^d$.

Using Helly's theorem, we can identify a lower bound for the minimal embedding dimension of X:

Lemma 2.8. *The code $C = \{000, 110, 101, 011\}$ does not have a convex realisation in a stimulus space $X \subseteq \mathbb{R}$.*

Proof. Assume $C = C(\mathcal{U})$ convex for $X \subseteq \mathbb{R}$ (d = 1).
There exist 3 neurons for the code C and hence 3 receptive fields U_1, U_2, U_3. Notice that the codeword corresponding to the intersection of any $d+1 = 2$ of these receptive fields is in C, but critically, the codeword $\{111\}$ corresponding to the full intersection $\bigcap_{i=1}^{3} U_i$ is not contained in C, so the full intersection must be empty.
But by Helly's theorem, the full intersection must be non-empty. By contradiction, our initial assumption is false, and the minimal embedding dimension of C is at least 2. □

By contrast, the Nerve theorem is concerned with topological features of the stimulus space.

The Nerve Theorem

Definition 2.9. A set $\Delta \subseteq 2^{[n]}$ is an *abstract simplicial complex*[3] if $\sigma \in \Delta$ and $\tau \subset \sigma$ implies $\tau \in \Delta$. In other words, a simplicial complex contains the subsets of its elements. We can also define the *simplicial complex of a code* as the smallest simplicial complex that contains supp(C):

$$\Delta(C) = \{\sigma \subset [n] \,|\, \sigma \subseteq \text{supp}(c) \qquad \text{for some } \ c \in C\} \tag{2.5}$$

Theorem 2.10. ***(Nerve Theorem)*** *The homotopy type of the union of convex receptive fields $\bigcup_{i=1}^{n} U_i$ is equal to the homotopy type of $\Delta(C(\mathcal{U}))$.*

Where the homotopy type of a set X is the set of all homotopy groups $\Pi_k(X)$, where $\Pi_k(X)$ is the set of all pointed maps $f : S^k \to X$ (where S^k is a k-dimensional sphere) up to homotopy.

Note that if $X = \bigcup_{i=1}^{n} U_i$ (which is the case if the all-zeroes codeword is not in C), then we get direct information about the topology of X. Later in the report, we develop our discussion on how homotopy groups relate to neural codes, and we introduce a method for computing homotopy types of simplicial complexes.

These theorems only go so far in characterising the stimulus space. For instance, codes with the same simplicial complex may have different minimal embedding dimensions, and may not have a convex realisation at all. Therefore there must be additional information inherent in the combinatorial structure of the neural code, or equivalently in the arrangement of its receptive fields.

[3]referred to hereon as a 'simplicial complex'

A central goal of this report is to introduce algebraic tools to generate a minimal description of the receptive field structure, which can be defined as non-trivial inclusion relations of the form:

$$\bigcap_{i\in\sigma} U_i \subseteq \bigcup_{j\in\tau} U_j \quad \text{for } \sigma \cap \tau = \emptyset \tag{2.6}$$

Obtaining these relations is a starting point for extracting stimulus space features that Helly's theorem and the Nerve theorem cannot capture.

3 The Neural Ideal and RF Structure

3.1 Neural Ideals

One of the central claims of this report is the usefulness of the *neural ideal* and the *neural ring* in determining stimulus space structure. Indeed, we will devote most of this chapter to developing a proof that a direct relation between these algebraic objects and the receptive fields exists.

First some background in algebraic geometry that will allow us to motivate the neural ideal and neural ring:

Definition 3.1. For some field k and ideal $J \subset k[x_1, ..., x_n]$, we can define the *variety* of J:

$$V(J) = \{v \in k^n \mid f(v) = 0 \text{ for all } f \in J\} \tag{3.1}$$

We also introduce the ideal of functions that vanish on some subset $S \subset k^n$:

$$I(S) = \{f \in k[x_1, ..., x_n] \mid f(v) = 0 \text{ for all } v \in S\} \tag{3.2}$$

If we let $k = \mathbb{F}_2$, and n be the number of neurons, we may conclude that the neural code C is a variety in $\mathbb{F}_q^n$ for some suitable ideal. This provides us with a natural definition for the neural ideal:

Definition 3.2. The neural ideal of a code C is the set of polynomials that vanish on all codewords in C:

$$I_C = \{f \in \mathbb{F}_2[x_1, ..., x_n] \,|\, f(c) = 0, \ \forall\, c \in C\} \tag{3.3}$$

Note that $V(I_C) = C$, and also $I(C) = I_C$.

One of the main tenets of algebraic geometry is that the geometric properties of a space should be retrieved from the algebraic properties of its ideal. Hence, we expect the combinatorial properties of C to be reflected in its ideal I_C.

A word about the polynomial ring $\mathbb{F}_2[x_1, ..., x_n]$. For every $f \in \mathbb{F}_2[x_1, ..., x_n]$ we have the Boolean relation $f^2(c) = f(c)$ for all $c \in \mathbb{F}_2^n$, by Fermat's Little Theorem. Hence, we have a whole family of functions which are identically zero on the whole of $\mathbb{F}_2^n$:

$$\mathcal{B} = \langle \{x_i^2 - x_i\}_{i \in [n]} \rangle \tag{3.4}$$

By definition, these functions return zero for any codeword, and hence are included in I_C for any code we choose. This means that we should expect to extract no information about the code (or the stimulus space) from these polynomials in particular. It makes sense then to make a distinction between the parts of the neural ideal that are 'trivial', and the parts that may end up being useful.

Before we construct this distinction, we introduce some convenient notation. We define the following polynomials in $\mathbb{F}_2[x_1, ..., x_n]$:

$$x_\sigma = \prod_{i \in \sigma} x_i \text{ and } \bar{x}_\tau = \prod_{i \in \tau} (1 - x_i). \tag{3.5}$$

We extend this notation analogously to receptive fields:

$$U_\sigma = \bigcap_{i \in \sigma} U_i \tag{3.6}$$

Now we introduce the 'useful' part of the neural ideal:

Definition 3.3. For a neural ideal I_C we can define:

$$J_C = \langle \{\rho_v \mid v \notin C\} \rangle \tag{3.7}$$

where:

$$\rho_v = x_{\text{supp}(v)} \bar{x}_{\text{supp}(v)^c} \tag{3.8}$$

and where $\text{supp}(v)^c$ denotes the complement of $\text{supp}(v)$.
Note that ρ_v can be considered a characteristic function on v, so $\rho_v(c) = 1$ iff $c = v$.

Lemma 3.4.

$$I_C = J_C + \mathcal{B} = \langle \{\rho_v \mid v \notin C\}, \{x_i^2 - x_i\}_{i \in [n]} \rangle \tag{3.9}$$

This lemma is a consequence of Hilbert's Nullstellensatz for Finite Fields:

Theorem 3.5. *Hilbert's Nullstellensatz for Finite Fields*
For any arbitrary finite field $\mathbb{F}_q$, for an ideal $J \subseteq \mathbb{F}_q[x_1, ..., x_n]$:

$$I(V(J)) = J + \langle \{x_i^2 - x_i\}_{i \in [n]} \rangle \tag{3.10}$$

Proof.
Claim:
$J + \mathcal{B} \subseteq I(V(J))$

Clearly true since for all $c \in C$, we have $\rho_v(c) = 0$ and $f(c) = 0$ for $\rho_v \in J_C$ and $f \in \mathcal{B}$.

Claim:
$I(V(J)) \subseteq J + \mathcal{B}$
Let $J = \langle f_1, ..., f_k \rangle$ for some basis of J.
Construct $h = \prod_{i=1}^{k}(1 - f_i^{q-1})$. Note that for some $v \in V(J)$:

$$h(v) = \prod_{i=1}^{k}(1 - f_i^{q-1}(v)) = \prod_{i=1}^{k}(1 - 0) = 1.$$

For some $u \in V(J)^C$ we must have that $f_j(u) \neq 0$ for some f_j, so in $\mathbb{F}_2$, $f_j(u) = 1$. Thus

$$h(u) = \prod_{i=1}^{k}(1 - f_i^{q-1}(u)) = 0.$$

$$\text{In summary: } \begin{cases} h(v) = 1 & v \in V(J) \\ h(u) = 0 & u \in V(J)^c \end{cases}$$

For all $f \in I(V(J))$ have $f = fh + f(1 - h)$.
Moreover, we know that $fh \in \mathcal{B}$ since by definition, $f = 0$ on $V(J)$ and $h = 0$ on $V(J)^c$.
Similarly, $f(1 - h) \in J$ since $1 - h = 0$ on $V(J)$.
Therefore $f \in J + \mathcal{B}$.

□

Since $I_C = I(C) = I(V(I_C)) = I(V(J_C))$ lemma 3.4 holds.

J_C will contain all non-trivial information about our code. An additional advantage of using J_C, is that while it is not immediately clear how to construct I_C from a code, J_C is easily constructible, since it consists of the functions ρ_v, which are derived directly from the codewords.

Definition 3.6. The neural ring of a code C is the quotient ring:

$$R_C = \mathbb{F}_2[x_1, ..., x_n]/I_C \tag{3.11}$$

Loosely speaking, each element of the coded neural ring is a function that has a unique set of outputs on the set of codewords in C. It is an object worth noting, though the neural ideal has been found to be more useful in the context of our aims.

Neural ideals and neural rings consist of functions of the form:

$$f : \mathbb{F}_2^n \to \mathbb{F}_2 \tag{3.12}$$

If we have an RF code $C = C(\mathcal{U})$ then we can introduce an alternative form of these objects that reference the stimuli rather than codes.

3.2 Relation to Receptive Fields

Definition 3.7. For a set of receptive fields $\mathcal{U} = \{U_1, U_2, ..., U_n\}$ contained in a stimulus space X and an RF code $C = C(\mathcal{U})$, the neural ideal can be defined as:

$$I_{C(\mathcal{U})} = \{f \in \mathbb{F}_2[x_1, ..., x_n] \mid f(p) = 0, \ \forall p \in X\} \tag{3.13}$$

Where we evaluate $f(p)$ by assigning:

$$x_i(p) = \begin{cases} 1 & \text{if}\, p \in U_i \\ 0 & \text{if}\, p \notin U_i \end{cases} \tag{3.14}$$

And $R_{C(\mathcal{U})}$ is defined analogously to Definition 3.2. The vector $(x_1(p), x_2(p), ..., x_n(p)) \in \{0, 1\}^n$ represents the neural response to a stimulus p. These neural ideals and rings consist of functions of the form:

$$f : X \to \mathbb{F}_2 \tag{3.15}$$

It is easy to see that $x_\sigma = 1$ if $p \in U_\sigma$, and $x_\sigma = 0$ if $p \notin U_\sigma$.

We introduce a third and final conceptualisation of the neural ideal which brings us closer to our central proof:

Definition 3.8. We define the ideal of a set of receptive fields $\mathcal{U} = \{U_1, U_2, ..., U_n\}$ by

$$I_\mathcal{U} = \langle \{x_\sigma \bar{x}_\tau \mid U_\sigma \subseteq \bigcup_{i \in \tau} U_i\} \rangle.$$

We expect that the geometric properties of the stimulus space X with receptive fields $\mathcal{U}$ should be encoded in this ideal. If we prove that $I_\mathcal{U} = I_{C(\mathcal{U})}$, we can establish a direct correspondence between the geometric properties of $\mathcal{U}$ and its neural code $C(\mathcal{U})$.

Theorem 3.9. *The following equality of ideals holds:*

$$I_{C(\mathcal{U})} = I_\mathcal{U} \tag{3.16}$$

Proof. We begin by defining an ideal $J_\mathcal{U}$ that allows us to separate the Boolean relations from $I_\mathcal{U}$. We then prove that $J_{C(\mathcal{U})} = J_\mathcal{U}$.
Note that for a function $f = x_\sigma \bar{x}_\tau$, if $\sigma \cap \tau \neq \emptyset$, then there exists k such that $x_k \bar{x}_k | f$, and

$f \in \mathcal{B}$. The corresponding receptive field relation is trivial, of the form $U_\sigma \subseteq U_k \subseteq \bigcup_{i\in\tau} U_i$. This motivates our definition of $J_\mathcal{U}$:

$$J_\mathcal{U} = \langle\{x_\sigma \bar{x}_\tau \mid \sigma \cap \tau = \emptyset \text{ and } U_\sigma \subseteq \bigcup_{i\in\tau} U_i\}\rangle; \tag{3.17}$$

thus $I_\mathcal{U} = J_\mathcal{U} + \mathcal{B}$ where $\mathcal{B}$ contains the Boolean relations.

We also utilise the following lemma:

Lemma 3.10. *For any $f \in k[x_1, ..., x_n]$ and $\tau \subseteq [n]$, the following equality of ideals holds:*

$$\langle\{f \prod_{i\in\tau} P_i \mid P_i \in \{x_i, 1 - x_i\}\}\rangle = \langle f\rangle.$$

<u>Claim:</u>
The generators of $J_{C(\mathcal{U})}$ are in $J_\mathcal{U}$.

We have that ρ_v for $v \notin C(\mathcal{U})$ a generator of $J_{C(\mathcal{U})}$. By the definition of $C(\mathcal{U})$, if we define $\sigma = \text{supp}(v)$ and $\tau = \text{supp}(v)^c$ then we get $U_\sigma \subseteq \bigcup_{i\in\tau} U_i$ with $\sigma \cap \tau = \emptyset$.

Hence, ρ_v is a generator for $J_\mathcal{U}$, and $J_{C(\mathcal{U})} \subseteq J_\mathcal{U}$.

<u>Claim:</u>
The generators of $J_\mathcal{U}$ are in $J_{C(\mathcal{U})}$.

Pick a generator $x_\sigma \bar{x}_\tau \in J_\mathcal{U}$. Since U_σ is contained by $\bigcup_{i\in\tau} U_i$, we have

$$U_\sigma \backslash \bigcup_{i\in\tau} U_i = \emptyset \Leftrightarrow \bigcap_{i\in\sigma} U_i \backslash \bigcup_{j\in\tau} U_j = \emptyset.$$

Then $v \notin C(\mathcal{U})$ for any v with $\sigma \subseteq \text{supp}(v)$ and $\tau \cap \text{supp}(v) = \emptyset$. Hence,

$$x_{\text{supp(v)}} \bar{x}_{\text{supp}(v)^c} \in J_{C(\mathcal{U})}.$$

We can decompose this into $x_\sigma \bar{x}_\tau \prod_{k\notin\sigma\cup\tau} P_k$ with $P_k \in \{x_k, 1 - x_k\}$. By Lemma 3.10, $x_\sigma \bar{x}_\tau \in J_{C(\mathcal{U})}$, thus $J_\mathcal{U} \subseteq J_{C(\mathcal{U})}$.

□

Proof. (Lemma 3.10)
Let $I(\tau) = \langle\{f \prod_{i\in\tau} P_i \mid P_i \in \{x_i, 1 - x_i\}\}\rangle$. Want to show that $I(\tau) = \langle f\rangle$ for all $\tau \subseteq [n]$.

Since every generator of $I(\tau)$ is a multiple of f, $I(\tau) \subseteq \langle f\rangle$. We will prove $\langle f\rangle \subseteq I(\tau)$ by induction on $|\tau|$.

If $|\tau| = 0$, then $I(\tau) = \langle f\rangle$. If $|\tau| = 1$, then $I(\tau) = \langle f(1 - x_i), fx_i\rangle$, hence $f \in I(\tau)$ and $\langle f\rangle \subseteq I(\tau)$.

Assume $\langle f\rangle \subseteq I(\tau)$ for $|\tau| = k$. Then for $|\tau| = k + 1$, pick an element $j \in \tau$, and define $\tau' = \tau\backslash\{j\}$.

Let $g = f \prod_{i\in\tau'} P_i$ be any generator of $I(\tau')$ and notice that gx_j and $g(1 - x_j)$ are both generators of $I(\tau)$. Hence, $gx_j + g(1 - x_j) = g \in I(\tau)$. Hence, any generator $g \in I(\tau')$ is in $I(\tau)$.
Therefore $\langle f\rangle \subseteq I(\tau') \subseteq I(\tau)$. □

Now we introduce our main theorem:

Theorem 3.11. *For a neural code $C \subseteq \{0,1\}^n$ and $\mathcal{U} = \{U_1, ..., U_n\}$ in X such that $C = C(\mathcal{U})$, then for any pair of subsets $\sigma, \tau \subset [n]$:*

$$x_\sigma \bar{x}_\tau \in I_C \Leftrightarrow U_\sigma \subseteq \bigcup_{i \in \tau} U_i \tag{3.18}$$

Proof. ($\Leftarrow$) This is a direct consequence of Theorem 3.9
($\Rightarrow$) Case 1: $\sigma \cap \tau \neq \emptyset$.
If $x_\sigma \bar{x}_\tau \in I_C$ then $x_\sigma \bar{x}_\tau = x_k(1 - x_k) x_{\sigma \backslash k} \bar{x}_{\tau \backslash k} \in \mathcal{B}$ for some $k \in \sigma \cap \tau$.
The corresponding receptive field relation would be trivially true.

Case 2: $\sigma \cap \tau = \emptyset$.
If $x_\sigma \bar{x}_\tau \in I_C$, then $\rho_v \in I_C$ for each $v \in \{0,1\}^n$ such that $\sigma \subseteq \operatorname{supp}(v)$ and $\operatorname{supp}(v) \cap \tau = \emptyset$.
Since $\rho_v(v) = 1$, it follows that $v \notin C$ for any such v.
Then, since $C = C(\mathcal{U})$ and $v \notin C$, we have $\bigcap_{i \in \sigma} U_i \backslash \bigcup_{j \in \tau} U_j = \emptyset$.
(Indeed, if $p \in \bigcap_{i \in \sigma} U_i \backslash \bigcup_{j \in \tau} U_j = \emptyset$, then construct v such that $i \in \operatorname{supp}(v)$ iff $p \in U_i$ so that $p \in \bigcap_{i \in \operatorname{supp}(v)} U_i \backslash \bigcup_{j \in \operatorname{supp}(v)^c} U_j = \emptyset$, which would contradict $v \notin C$.)
This is equivalent to the receptive field relation:

$$U_\sigma \subseteq \bigcup_{j \in \tau} U_j \tag{3.19}$$

Hence $x_\sigma \bar{x}_\tau \in I_{\mathcal{U}}$. □

We now have a means of computing receptive field relationships starting only with data of neural activity. We can classify the important relationships into 4 distinct types (Take $\sigma, \tau \neq \emptyset$):

- Boolean relations: $\{x_i(1 - x_i)\}$
  Corresponds to $U_i \subseteq U_i$ which is trivial.
- Type 1 relations: $\{x_\sigma\}$
  Corresponds to $U_\sigma = \emptyset$.
- Type 2 relations; $\{x_\sigma \bar{x}_\tau \mid \sigma \cap \tau = \emptyset\}$
  Corresponds to $U_\sigma \subseteq \bigcup_{i \in \tau} U_i$.
- Type 3 relations: $\{\bar{x}_\tau\}$
  Corresponds to $X \subseteq \bigcup_{i \in \tau} U_i$.

There exist elements of I_C that do not belong to these types, but these types are sufficient to generate I_C. This means that there are 'redundant' elements of I_C that don't provide new information that the generators already provide. If we can find a minimal set of generators for J_C (and hence $I_C = J_C + \mathcal{B}$), we get an object that contains only the essential RF structure of a code. We call this the canonical form of the neural ideal.

3.3 The Canonical Form of the Neural Ideal

We group the polynomials in J_C into a single type called *pseudo-monomials*:

Definition 3.12. A function $f \in \mathbb{F}_2^n[x_1, ..., x_n]$ is a pseudo-monomial if $f = x_\sigma \bar{x}_\tau$ for some $\sigma, \tau \in [n]$ and $\sigma \cap \tau = \emptyset$.

Definition 3.13. For an ideal J, we say a pseudo-monomial $f \in J$ is minimal if there exists no $g \in J$ such that $\deg(g) < \deg(f)$ and $f = hg$ for some $h \in \mathbb{F}_2^n[x_1, ..., x_n]$.

We also say that if J can be generated by a finite set of pseudo-monomials, it is a pseudo-monomial ideal. By generating J from minimal pseudo-monomials, we obtain a description that is unique and compact, providing a concise layout of the receptive field relationships.

The canonical form of the neural ideal appears as follows ($\sigma, \tau \neq \emptyset$):

$$\begin{aligned} J_C = &\langle \{x_\sigma \mid \sigma \text{ minimal for } U_\sigma = \emptyset\}, \\ &\{x_\sigma \bar{x}_\tau \mid \sigma \cap \tau = \emptyset;\ \sigma, \tau \text{ minimal for } U_\sigma \subseteq \bigcup_{i \in \tau} U_i\}, \\ &\{\bar{x}_\tau \mid \tau \text{ minimal for } X \subseteq \bigcup_{i \in \tau} U_i\}\rangle \end{aligned} \tag{3.20}$$

Letting the three distinct sets in J_C correspond to relation types, we have:

- Type 1: $U_\sigma = \emptyset$, but $U_\gamma \neq \emptyset$ for all $\gamma \subset \sigma$.
- Type 2: $U_\sigma \subseteq \bigcup_{i \in \tau} U_i$, but $U_\gamma \not\subseteq \bigcup_{i \in \delta} U_i$ for all $\gamma \subset \sigma$ or $\delta \subset \tau$.
- Type 3: $X \subseteq \bigcup_{i \in \tau} U_i$, but $X \not\subseteq \bigcup_{i \in \delta} U_i$ for all $\delta \subset \tau$.

An extra use of the canonical form is aiding the construction of receptive field diagrams.We present an example of this, starting from a neural code and ending in a receptive field diagram.

Take the neural code $C = \{$ 00000, 10000, 01000, 00100, 00001, 11000, 10001, 01100, 00110, 00101, 00011, 11100, 00111 $\}$.

Then $C^c = \{$00010, 10100, 10010, 01010, 01001, 11010, 11001, 10110, 10101, 10011, 01110, 01101, 01011, 11110, 11101, 11011, 10111, 01111, 11111$\}$.

We can compute J_C from the definition $\langle \{\rho_v \mid v \notin C\}\rangle$:

$$\begin{aligned} J_C = \langle &x_4(1-x_1)(1-x_2)(1-x_3)(1-x_5), x_1x_3(1-x_2)(1-x_4)(1-x_5), \\ &x_1x_4(1-x_2)(1-x_3)(1-x_5), x_2x_4(1-x_1)(1-x_3)(1-x_5), \\ &x_2x_5(1-x_1)(1-x_3)(1-x_4), x_1x_2x_4(1-x_3)(1-x_5), \\ &x_1x_2x_5(1-x_3)(1-x_4), x_1x_3x_4(1-x_2)(1-x_5), x_1x_3x_5(1-x_2)(1-x_4), \\ &x_1x_4x_5(1-x_2)(1-x_3), x_2x_3x_4(1-x_1)(1-x_5), x_2x_3x_5(1-x_1)(1-x_4), \\ &x_2x_4x_5(1-x_1)(1-x_3), x_1x_2x_3x_4(1-x_5), x_1x_2x_3x_5(1-x_4), \\ &x_1x_2x_4x_5(1-x_3), x_1x_3x_4x_5(1-x_2), x_2x_3x_4x_5(1-x_1), \\ &x_1x_2x_3x_4x_5\rangle \end{aligned} \tag{3.21}$$

Presenting J_C in canonical form we get:

$$J_C = \langle x_1x_3x_5, x_2x_5, x_1x_4, x_2x_4, x_1x_3(1-x_2), x_4(1-x_3)(1-x_5)\rangle \tag{3.22}$$

Now we can interpret these minimal pseudo-monomial generators as receptive field relationships:

- $x_1x_3x_5$:
  $U_1 \cap U_3 \cap U_5 = \emptyset$, but any pairwise intersection is non-empty.
- x_2x_5:
  $U_2 \cap U_5 = \emptyset$, while any individual RF is non-empty.
- x_1x_4:
  $U_1 \cap U_4 = \emptyset$, while any individual RF is non-empty.
- x_2x_4:
  $U_2 \cap U_4 = \emptyset$, while any individual RF is non-empty.
- $x_1x_3(1-x_2)$:
  $U_1 \cap U_3 \subseteq U_2$, but $U_1, U_3 \not\subseteq U_2$ and $U_1 \cap U_3$ non-empty.
- $x_4(1-x_3)(1-x_5)$:
  $U_4 \subseteq U_3 \cap U_5$, but $U_3 \not\subseteq U_3, U_5$ and U_4 non-empty.

We can interpret this information and draw the according receptive field diagram:

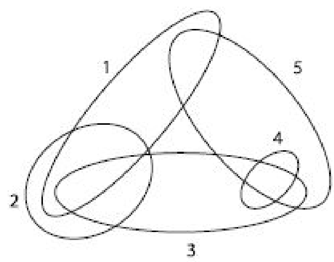


Figure 3: A realisation of the code $C = C(\mathcal{U})$

4 Determining Convexity

Determining the convexity of any neural code is an unsolved problem in mathematical neuroscience. However, there are some properties a code may possess that infer its convexity. Likewise, there is at least one condition discovered that can infer the non-convexity of a code. We present these properties in this chapter, and provide a framework for computing whether a code possesses such properties. This section is inspired by further work by Curto et al. [6].

4.1 Conditions for Convexity

Theorem 4.1. *A code is convex if its support is a simplicial complex.*

There also exist two larger known classes of convex codes:

Definition 4.2. A code C is *intersection-complete* if for any $c, d \in C$, there is a codeword $v \in C$ such that $\text{supp}(v) = \text{supp}(c) \cap \text{supp}(d)$.

Definition 4.3. An element of a simplicial complex $\rho \in \Delta$ is maximal if $\rho \cup \sigma \notin \Delta$ for any $\sigma \nsubseteq \rho$, $\sigma \in \Delta$.

We then say a code C is *max-intersection-complete* if for any collection of maximal sets $\{\rho_i\}_{i=1}^{k} \subseteq \Delta(C)$, we have $\bigcap_{i=1}^{k} \rho_i \in C$.

Theorem 4.4. *A code is convex if it is intersection-complete.*

Theorem 4.5. *A code is convex if it is max-intersection-complete.*

These properties are identifiable by working with the support of the code. However if a code C possesses one of these properties, we should expect the canonical form of its corresponding neural ideal J_C to possess a signature that infers these properties, as it contains all the combinatorial information of C. Indeed, the signatures for the above properties have been discovered, and the simplicial complexes and intersection-complete codes are as follows (the canonical form denoted by CF):

- A code C is a simplicial complex iff $CF(J_C)$ consists only of monomials.
- A code C containing the all-zeros codeword is intersection-complete iff $CF(J_C)$ consists only of monomials and mixed monomials of the form:

$$\prod_{i \in \sigma} x_i(1 - x_j) \tag{4.1}$$

The signature for max-intersection complete codes was discovered most recently and requires some additional exposition. See section 5.

4.2 Links, Minimal Codes, and Contractibility

The non-convexity of a neural code can also be determined upon fulfillment of a specific property. Unlike the properties that imply convexity however, this property is not immediately determinable from looking at the support of the code or the canonical form of its neural ideal. First a definition - given the simplicial complex of a code, we compute its *links*:

$$Lk_\sigma(\Delta) = \{\omega \in \Delta \,|\, \sigma \cap \omega = \emptyset \text{ and } \sigma \cup \omega \in \Delta\} \tag{4.2}$$

Each link is computed from an element $\sigma \in \Delta$. We then introduce a topological property called *contractibility*:

Definition 4.6. A space is *contractible* if it is homotopy-equivalent to a point i.e. it is connected and has no holes.

This property is important as it allows us to define a new code that gets to the heart of determining non-convexity:

Theorem 4.7. *For a code C and its simplicial complex $\Delta(C) = \Delta$, we associate to C a unique minimal code:*

$$C_{min}(\Delta) = \{\sigma \in \Delta \,|\, Lk_\sigma(\Delta) \textit{ is not contractible}\} \cup \{\emptyset\} \tag{4.3}$$

If $C_{min} \not\subseteq C$ then C is not convex. [4].

By computing $C_{\min}$, we have a (sometimes inconclusive) test for the non-convexity of C. In implementing this method, it is easy to compute simplicial complexes and links directly from their definitions. However, the contractibililty of a link is a consequence of the homology groups of that link. Hence if we want to compute $C_{min}(\Delta)$, we need to motivate an appropriate definition for homology groups. The following section addresses this, following work in [7].

Important to note is that since $C_{min}(\Delta)$ is defined by the simplicial complex of a code, we can classify codes into groupings based on their simplicial complexes. It is sufficient to identify the maximal elements of the code, since any two codes with the same maximal elements will have the same simplicial complex.

One caveat is that in general, determining whether a simplicial complex is contractible is an undecidable problem [8]. However, we can still at least compute a subset of C_{min} by considering its elements that are detectable by homological methods.

4.3 Homology Groups

To begin discussing homology groups, we need to introduce some basic spaces and functions.

Definition 4.8. A k-simplex is the collection of points:

$$\{x \in \mathbb{R}^{k+1} \,|\, x_0 + ... + x_k = 1, \text{ and } x_i \geq 0\} \tag{4.4}$$

k-simplexes are generalisations of a triangle to any dimension. Plugging in values of $k =$ 0,1,2,3... we can see the corresponding k-simplexes must be a single point, a line segment, a triangle, and a tetrahedron respectively.

Important to note is that a k-simplex can have an *orientation*, defined by how its 0-simplexes are ordered. We say that two orientations are the same if they differ by an even permutation.

Example 4.9. *We say (v_0, v_1, v_2) and (v_0, v_2, v_1) have different orientations as they differ by an odd permutation, while (v_0, v_1, v_2) and (v_2, v_0, v_1) have the same orientation as they differ by an even permutation.*

We say that an oriented simplex is equal to the negative of the same simplex with opposite orientation.

[4]Note the abuse in notation: $C_{\min}$ is said to contain sets of neurons in its definition rather than codewords - this is done to avoid writing $\Delta(C_{\min}(\Delta))$

We can construct a *geometric simplicial complex*[5] Δ as sets of k-simplexes according to the following rules:

- Any face of a simplex in Δ is also in Δ.
- The intersection of any two simplexes $\sigma_1, \sigma_2 \in \Delta$ is a face of both σ_1 and σ_2.

where the faces of a k-simplex are the lower dimensional simplexes that make up its boundary.

Definition 4.10. For a simplicial complex Δ, and its oriented k-simplexes S_k, the sets of k-chains on Δ are denoted by C_k and are defined:

$$C_k = \{\sum_{i=1}^{N} c_i\sigma_i \,|\, c_i \in \mathbb{Z} \,\text{and}\, \sigma_i \in S_k\} \tag{4.5}$$

with $N = |S_k|$.
The k-chains on a simplicial complex consist of all linear combinations of its k-simplexes.

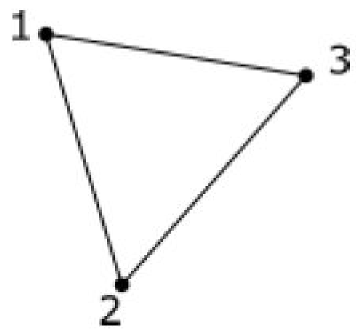


Figure 4: A simplicial complex

For example, the simplicial complex above $(\{\{1\},\{2\},\{3\},\{1,2\},\{1,3\},\{2,3\}\})$ has:

$$3 * [1] + 5 * [2] - 2 * [3] \in C_0 \tag{4.6}$$

and:

$$11 * [1,2] + 1 * [1,3] + 3 * [3,2] = 11 * [1,2] + 1 * [1,3] - 3 * [2,3] \in C_1 \tag{4.7}$$

Definition 4.11. A boundary map is a linear map $\partial_k : C_k \to C_{k-1}$ where for a k-simplex $\sigma = [v_0, ..., v_k]$:

$$\partial_k(\sigma) = \sum_{i=0}^{k}(-1)^i[v_0, ..., \widehat{v_i}, ..., v_k] \tag{4.8}$$

where $\widehat{v_i}$ means that element is 'removed', hence, the output of $\partial_k(\sigma)$ is indeed a $(k-1)$-chain.

∂ is a linear map, so for k-chains:

$$\partial_k(\sum_{i=1}^{N} c_i\sigma_i) = \sum_{i=1}^{N} c_i\partial_k(\sigma_i) \tag{4.9}$$

[5]referred to in this chapter as 'simplicial complexes'

We can also express the boundary maps as matrices, where an entry $(\partial_k)_{ij}$ is the scalar coefficient of some $(k-1)$-chain corresponding to the output of ∂_k on some k-chain. These entries must be in the set $\{-1, 0, 1\}$. E.g.:

$$\partial_1 = \begin{array}{c} \\ [1] \\ [2] \\ [3] \end{array} \begin{array}{c} \begin{array}{ccc} [1,2] & [1,3] & [2,3] \end{array} \\ \begin{pmatrix} -1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \end{pmatrix} \end{array}$$

An important feature of the boundary map is that $\partial_{k+1}\partial_k = 0$.

Proof. We have: $\partial_{k+1}(\sigma) = \sum_{j=0}^{k+1}(-1)^j[v_0, ..., \widehat{v_j}, ..., v_{k+1}]$ With $i < j$, select two terms from the previous sum, and act with ∂_k on them:

$$\begin{aligned} & \partial_k((-1)^j[v_0, ..., \widehat{v_j}, ..., v_{k+1}] + (-1)^i[v_0, ..., \widehat{v_i}, ..., v_{k+1}]) \\ = & \sum_{l_1=0}^{k}(-1)^{l_1}(-1)^j[v_0, ..., \widehat{v_{l_1}}, ..., \widehat{v_j}, ..., v_{k+1}] + \sum_{l_2=0}^{k}(-1)^{l_2-1}(-1)^i[v_0, ..., \widehat{v_i}, ..., \widehat{v_{l_2}}, ..., v_{k+1}]. \end{aligned}$$

Selecting the terms $l_1 = i$ and $l_2 = j$ we get:

$$(-1)^i(-1)^j[v_0, ..., \widehat{v_i}, ..., \widehat{v_j}, ..., v_{k+1}] + (-1)^{j-1}(-1)^i[v_0, ..., \widehat{v_i}, ..., \widehat{v_j}, ..., v_{k+1}] = 0.$$

Since $k(k+1)$ is even, and there are $k(k+1)$ total terms in $\partial_k\partial_{k+1}\sigma$, we can split the terms into pairs which each sum to zero, hence $\partial_k\partial_{k+1}\sigma = 0$.

$\square$

We will be interested in computing the ranks of these boundary maps. There are many ways to do this. Here we will choose to represent the boundary map as a matrix in its Smith normal form. For any $n \times m$ matrix A, there exists an invertible $n \times n$ matrix U and an invertible $m \times m$ matrix V such that $UAV = D$ where:

$$D = \begin{pmatrix} \alpha_1 & 0 & 0 & \cdots & 0 \\ 0 & \alpha_2 & 0 & & 0 \\ 0 & 0 & \ddots & & 0 \\ \vdots & & & \alpha_r & \\ 0 & & & & 0 \end{pmatrix} \tag{4.10}$$

Clearly $\text{rank}(A) = r$. Additionally, if A is an integer matrix, then U, V, D are also integer matrices.

With this toolset prepared, we can begin to discuss homology groups.

Definition 4.12. For an oriented simplicial complex Δ, we define the k-th homology group:

$$H_k(\Delta) = \frac{\text{Ker}(\partial_k)}{\text{Im}(\partial_{k+1})} \tag{4.11}$$

Theorem 4.13. *Let A be an $m \times n$ integer matrix and B be an $l \times m$ integer matrix such that $BA = 0$. Then:*

$$\frac{Ker(B)}{Im(A)} = \bigoplus_{i=1}^{r} \frac{\mathbb{Z}}{\alpha_i} \oplus \mathbb{Z}^{m-r-s} \tag{4.12}$$

Where $r = rank(A)$, $s = rank(B)$ and $\{\alpha_i\}_{i=1}^{r}$ are the non-zero elements on the diagonal of the Smith normal form of A.

Hence:

$$H_k(\Delta) = \frac{\mathrm{Ker}(\partial_k)}{\mathrm{Im}(\partial_{k+1})} = \bigoplus_{i=1}^{r} \frac{\mathbb{Z}}{\alpha_{i,k}} \oplus \mathbb{Z}^{\beta_k} \tag{4.13}$$

with ∂_{k+1} a $m \times n$ matrix, $r = \mathrm{rank}(\partial_{k+1})$, $s = \mathrm{rank}(\partial_k)$ and $\beta_k = m - r - s$. The rank of $H_k(\Delta)$ is defined to be β_k, where $\{\beta_i\}_{i\in\mathbb{N}}$ are called the Betti numbers of the simplicial complex.

Let us now return to the condition from Theorem 4.7. To confirm that our link is homotopic to a point, we require two things:

- $\alpha_{i,k} = \pm 1$ for all i and k.
- $\beta_0 = 1$ and $\beta_k = 0$ for all $k > 1$.

These quantities are directly computable from the matrix representations of the boundary maps, which are themselves computable. Hence, we have successfully produced a means of constructing $C_{\min}$, and thus a way of inferring the non-convexity of the associated codes of $C_{\min}$.

4.4 Program to Check Convexity

The main aims of the program were twofold. Firstly, we wanted to implement the methods described in section 4 for determining the convexity of a code C:

- C is a simplicial complex $\Rightarrow C$ convex.
- C is intersection complete $\Rightarrow C$ convex.
- C is max-intersection complete $\Rightarrow C$ convex.
- $C_{min} \not\subseteq C \Rightarrow C$ not convex.

Secondly, following this implementation, it was desired to find the convexity of all codes of low dimension (n=2,3).

The methods described above were sufficient to categorise all codes of dimension n = 2,3. We present results of the program below in figure 5, namely an assortment of codes represented by graphs, sorted by number of neurons n and maximal codewords $\{\rho\}$ (with equivalence up to permutation of neurons):

Please find the full program under the Ancillary Files section of the submission.

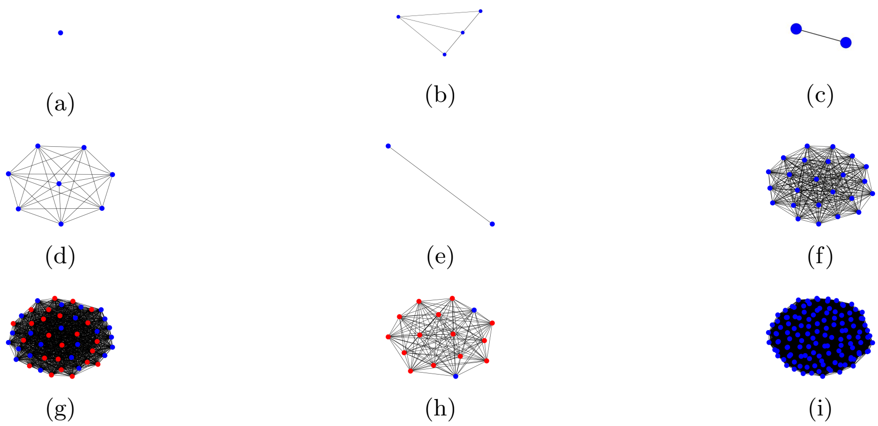


Figure 5: Graphs of codes, where the nodes correspond to codes, the edges link codes of like maximal codewords, and the node colours of blue and red correspond to convex and non-convex codes respectively. The graphs are sorted by dimension and maximal codewords. Neurons: (a)-(d) 2, (e)-(i) 3. Maximal codewords: (a)$\{00\}$, (b)$\{10\}$, (c)$\{10, 01\}$, (d)$\{11\}$, (e)$\{100, 010, 001\}$, (f)$\{110, 001\}$, (g)$\{110, 101\}$, (h)$\{110, 101, 011\}$, (i)$\{111\}$

5 Further Advances

Presented in this section is a brief introduction on some recent advances made that expand on the work produced by Curto et al. [4].

5.1 Algebraic Signature for Max-Intersection Complete Codes

An algebraic signature has been found for max-intersection complete codes [9]. First some definitions:

Definition 5.1. The Stanley-Reisner ideal of a simplicial complex Δ is:

$$I(\Delta) = \langle \prod_{i \in \sigma} x_i \mid \sigma \notin \Delta \rangle = \bigcap_{\sigma \in \mathrm{Facets}(\Delta)} \langle x_i \mid i \notin \sigma \rangle \tag{5.1}$$

It is known that $I(\Delta)$ is a radical ideal and its prime decomposition is the right-hand side of equation 5.1 [10].

Theorem 5.2. *A code C on n neurons is max-intersection complete if and only if for every non-monomial ϕ in the canonical form of the neural ideal of C, there exists $i \in [n]$ such that:*

(i) every associated prime of the Stanley-Reisner ideal of $\Delta(C)$ that contains x_i also contains ϕ, and

(ii) $(1 - x_i) \mid \phi$

This is a useful extension of the ability of the neural ideal to capture qualities of its neural code, especially since this defintion of max-intersection completeness can be checked on a code in a sub-exponential runtime algorithm.

5.2 Gröbner Bases of the Neural Ideal

The canonical form of the neural ideal is one of many possible generating sets of the neural ideal. One alternative generating set is its *Gröbner basis.* Two primary motivating factors for formulating the neural ideal in terms of different bases are as follows:

Dimension	4	5	6	7	8
Canonical form	0.0016	0.0076	0.108	0.621	1.964
Gröbner basis	0.00147	0.00202	0.00496	0.01604	0.16638

Figure 6: Runtime of Canonical form and Gröbner basis computations.

- Generating relationships between these bases and the canonical form, not only for theoretical concerns, but also:
- Once such relationships are established, these alternative bases could yield more efficient algorithms for computing the canonical form.

Indeed, according to Garcia et al. [11], the best algorithms at the time yielded '[efficient computation]' of canonical forms for codes up to 10 dimensions.

For codes of larger dimension, it was found that while Gröbner bases are generally more efficient to compute, the standard deviation of the computation time is much larger.

References

[1] Dostrovsky J O'Keefe J. *The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat.* PMID: 5124915, 1971.

[2] *https://www.nobelprize.org/prizes/medicine/2014/summary/.*

[3] Nevers R. Tank D. Aronov D. *Mapping of a non-spatial dimension by the hippocampal–entorhinal circuit.* Nature 543, 719–722, 2017.

[4] Carina Curto et al. *The neural ring: an algebraic tool for analyzing the intrinsic structure of neural codes.* Bulletin of Mathematical Biology et al., 2013.

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

[6] N. Youngs. C. Curto A. Veliz-Cuba. *Analysis of combinatorial neural codes: an algebraic approach. Book chapter in Algebraic and Combinatorial Computational Biology.* R. Robeva, M. Macaulay (Eds), 2018.

[7] Eric Bunch. *Calculating Homology of a Simplicial Complex Using Smith Normal Form.* Github, 2018.

[8] M. Tancer. *Recognition of collapsible complexes is NP-complete.* Discrete Comput. Geom., 55(1):21–38, 2016.

[9] Alexander Ruys de Perez, Laura Felicia Matusevich, and Anne Shiu. *Neural Codes and the Factor Complex.* 2019. arXiv: `1904.03235 [math.CO]`.

[10] R. Stanley. *Progress in mathematics: Combinatorics and commutative algebra.* Boston: Birkhäuser, 2004.

[11] Rebecca Garcia et al. *Gröbner Bases of Neural Ideals.* 2018. arXiv: `1612.05660 [q-bio.NC]`.