

Efficient Search of Informational Cores in Complex Systems

-Application to Brain Networks-

Jun Kitazono, Ryota Kanai, Masafumi Oizumi

February 21, 2020

Abstract

To understand the nature of complex behavior of the brain, it is an important step to identify “cores” in the brain network, where neurons or brain areas strongly interact with each other. Cores can be considered as essential sub-networks for brain functions and cognition. In the last decades, an information-theoretic approach to identifying cores has been developed. In this approach, many-to-many nonlinear interactions between parts are measured by an information loss function, which quantifies how much information would be lost if interactions between parts of a system were removed. Then, a core called a “complex” is defined as a subsystem where the amount of information loss is locally maximal. Although identifying complexes can be a novel and useful approach to reveal essential properties of the brain network, its computation time grows exponentially with the system size, which has hindered practical applications. In this study, we propose a fast and exact algorithm for finding complexes, hierarchical partitioning for complex search (HPC). HPC finds complexes by hierarchically partitioning systems to narrow down candidates for complexes. The computation time of HPC is polynomial, which is dramatically smaller than exponential. We prove that HPC is exact when an information loss function satisfies a mathematical property, monotonicity. We show that mutual information is one such information loss function. We also show that a broad class of submodular functions can be considered as such information loss functions, indicating the expandability of our framework to the class. In simulations, we show that HPC can find complexes in large systems (up to several hundred) in a practical amount of time when mutual information is used as an information loss function. Finally, we demonstrate HPC in electrocorticogram recordings from monkeys. HPC revealed temporally stable and characteristic complexes, which indicates HPC can be reliably utilized for characterizing brain networks.

1 Author Summary

To understand the nature of the brain, it is an important step to identify “cores” in the brain network, which can be considered as essential areas for brain functions and cognition. In the last decades, a novel definition of cores has been developed, which can take account of many-to-many interactions among elements of the network. Although considering many-to-many interactions can be important to understand the complex brain network, identifying the cores has been impossible in large systems because of extremely large computational costs. Here, we proposed a fast and exact algorithm for finding cores. We showed that the proposed algorithm enables us to find cores in large systems consisting of several hundred elements in a practical amount of time. We applied our algorithm to electrocorticogram recordings from a monkey that monitored electrical activity of the brain with electrodes placed directly on the brain surface and demonstrated that there are stable and characteristic core structures in the brain network. This result indicates our algorithm can be reliably applied for uncovering the essential network structures of the brain.

2 Introduction

In the brain, many neurons interact with each other to realize sophisticated function and cognition. To understand the neural mechanisms of brain function and cognition, it is important to uncover network structures in the brain [1, 2, 3]. An effective way to characterize the network structures is to identify “cores” of the network where neurons strongly interact with each other. The cores can be considered as the most important sub-networks for various functions and cognition [2, 4]. In the literature, cores have been defined in many ways using different methods such as modularity [5], hubs [5, 6], and k-core decomposition [5]. Most of those methods are based on graph representations of systems, where the dependence between two nodes is represented as the weight of the edge connecting the two nodes. Although the graph-based analysis is easy to use, there is a limitation. Since a graph is fully described by one-to-one relations between nodes, graph-based analysis inevitably omits the effects of many-to-many interactions, which are not predicted by one-to-one interaction alone, but still can be important for understanding complex interactions in the brain.

To overcome the limitations of the graph-based analysis, we utilize an information-theoretic approach [7, 8] to cores that takes into account many-to-many interactions. In the approach, the degree of the interactions between parts in a system is measured by an information-theoretic measure such as mutual information. More specifically, it is quantified by how much information would be lost if a system was cut into parts and interactions between the parts were removed. Then, roughly speaking, a core is defined as a subsystem where the amount of information loss is larger than all of its supersets. We call such cores defined in this way, “informational cores” or “complexes” [7, 8].

The original idea of the complex was proposed in the Integrated Informa-

tion Theory (IIT) of consciousness [9, 10, 7, 8, 11, 12]. IIT hypothesized that complexes in the brain correspond to the loci of consciousness, i.e., the essential parts supporting consciousness. Nevertheless, we can utilize the complex not only for studying consciousness but for analyzing other systems unrelated to consciousness. This is because the complex is based on the information theory and therefore it can be, in principle, applied to any stochastic systems.

Despite the general applicability of the complex to stochastic systems and the potential superiority to other graph-based approaches, only a few studies utilized complexes to analyze systems, and the sizes of the analyzed systems were small [13, 14, 15]. This is because searching for complexes is extremely difficult in large systems. The computation time for searching for complexes grows exponentially with the number of elements in the system. Even identifying complexes in a system with dozens of elements is virtually impossible.

In this paper, we propose a fast and exact algorithm, which we call “hierarchical partitioning for complex search” (HPC). HPC searches for complexes by hierarchically dividing subsystems. HPC is exact when the measure of the information loss satisfies a certain mathematical property, i.e., monotonicity. Mutual information satisfies this property and therefore we use mutual information in this study. We also show that a class of submodular functions, where “submodular” is a mathematical property of set functions [16], satisfies the monotonicity, indicating the expandability of our framework to the class of functions. HPC can identify complexes in polynomial time. More specifically, HPC enables us to find complexes of a system consisting of several hundreds of elements, which makes the search of complexes possible in, for example, multi-channel EEG or ECoG typically consisting of several hundreds of electrodes, in a practical amount of time.

The rest of the paper is organized as follows. In Section 3, we outline important concepts that are needed to define the complex. In Section 4, we explain information loss, in particular, the mutual information, where the proposed algorithm for searching for complexes exactly works. In Section 5, we explain how the information loss is utilized to quantify the strength of interactions in a system. In Section 6, we show the definition of the complex. Then in Section 7, we introduce our new algorithm HPC. In Section 8, we show a relation between the monotonicity and submodularity. In experiments in Section 9, we first demonstrate how HPC works, by taking a simple model as an example. Second, we evaluate the computation time of HPC. Third, we demonstrate how the algorithm can be applied to real neural data using an open ECoG dataset as an example [17]. Finally, in Section 10, we discuss limitations of HPC and the relation to other definitions of cores and prospects of this study.

The MATLAB codes of HPC is available at **.

3 Outline of important concepts

“Complex” is a rather complicated concept to understand. Before we give a formal mathematical definition of complexes, we first outline two important

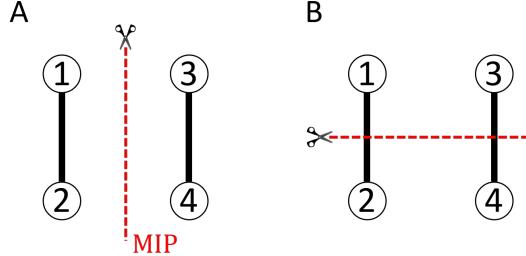


Figure 1: Schematic of Minimum Information Partition (MIP). We consider a subsystem consisting of four elements $\{1, 2, 3, 4\}$. The nodes connected by edges are dependent on each other, while others are not. If the subsystem is vertically cut into the two parts $\{1, 2\}$ and $\{3, 4\}$ as shown in Fig. 1A, no information is lost. In contrast, if the subsystem is partitioned horizontally as shown in Fig. 1B, some amount of information is lost. In this case, the vertical cut in Fig. 1A is the MIP.

concepts, information loss functions and minimum information partitions, which are needed to define complexes, and then outline complexes.

Information loss function Information loss functions measure the strength of dependence between parts from an information-theoretic viewpoint. Information loss functions quantify how much information is lost if a subsystem is “cut” into smaller two parts, where “cut” means removing the interactions between the two parts. If the two parts are independent of each other, no information is lost while if the parts are strongly dependent on each other, much information is lost. As we will explain in detail in the next section, mutual information and integrated information in IIT can be interpreted as information loss functions.

Minimum information partition (MIP) The amount of information loss depends on how a subsystem is partitioned. As an example, suppose a subsystem $\{1, 2, 3, 4\}$ consisting of two independent parts $\{1, 2\}$ and $\{3, 4\}$ (Fig. 1). If the subsystem is vertically cut into the two parts $\{1, 2\}$ and $\{3, 4\}$ as shown in Fig. 1A, no information is lost. In contrast, if the subsystem is partitioned differently as shown in Fig. 1B, some amount of information is lost. The minimum information partition (MIP) of a subsystem is the partition where the information loss is the minimum among all the possible partitions [10, 7, 8]. In this example, the vertical cut in Fig. 1A is the MIP. Thus, the MIP cuts the system with its weakest link.

Complex Complexes are defined by using the concept of MIP: a complex is a subsystem such that the amount of information loss when it is partitioned with its MIP is larger than those of all its supersets [7, 8]. If we add

elements to a complex, the amount of information loss of the extended subsystem measured with its MIP is smaller than that of the complex. Intuitively speaking, a complex is more strongly unified than any of its supersets, and adding elements to a complex inevitably includes a weaker link.

4 Information loss function

In this section, we explain the definition of information loss functions. Information loss functions measure the amount of information loss when interactions between parts are removed. After introducing a general definition of information loss functions, we consider the mutual information as an example of information loss functions. As we describe in Section 7, the mutual information satisfies essential properties for the proposed algorithm for complex search. For this reason, we use the mutual information as an information loss function in this paper.

We consider a probabilistic system consisting of N elements with a distribution $p(\mathbf{x}_V) = p(x_1, \dots, x_N)$. Here, V denotes the set of indices ($V = \{1, \dots, N\}$) and \mathbf{x}_V denotes $(x_1, \dots, x_N) = (x_i)_{i \in V}$. Similarly, for a subset $S \subseteq V$, \mathbf{x}_S denotes $(x_i)_{i \in S}$. For example in the case of a multi-agent system, each variable x_i can be a state of an agent, and N can be the number of agents. In the case of an EEG measurement of brain activity, x_i can be a signal from an electrode and N can be the number of electrodes.

We consider a bi-partition (S_L, S_R) of a subset S that partitions S into two disjoint parts S_L and S_R ($S_R = S \setminus S_L$). Then, we consider a “disconnected” distribution $q(\mathbf{x}_S) = q(\mathbf{x}_{S_L}, \mathbf{x}_{S_R})$, where some form of interactions between S_L and S_R is removed. We define the amount of information loss caused by the partitioning, $f(S_L; S_R)$, as the Kullback-Leibler divergence between the original distribution p and the disconnected distribution q :

$$\begin{aligned} f(S_L; S_R) &= D_{\text{KL}}(p(\mathbf{x}_S) \parallel q(\mathbf{x}_S)), \\ &= \int p(\mathbf{x}_S) \log \frac{p(\mathbf{x}_S)}{q(\mathbf{x}_S)} d\mathbf{x}_S. \end{aligned} \quad (1)$$

The Kullback-Leibler divergence, in general, can be interpreted as information loss when q is used to approximate p [18]. Thus, the information loss function f can be interpreted as information loss caused by removing interactions between parts.

Since S_L is the complement of S_R in S , determining S_L automatically specifies S_R . Therefore, an information loss function $f(S_L; S_R)$ can be considered as a set function of S_L whose domain is the powerset of S . To emphasize this, we introduce the following notation:

$$f_S(S_L) := f(S_L; S \setminus S_L) = f(S_L; S_R). \quad (2)$$

There are various types of information loss functions, depending on what kind of “interactions” are removed [19, 20, 21, 22, 23]. An example is integrated

information, which was originally introduced in IIT [10, 8, 11] and several variants were also proposed by different researchers [24, 25, 20, 19, 21, 22]. Integrated information can be interpreted as information loss when “causal” interactions between subsystems are removed [19]. Integrated information becomes large when two parts strongly affect each other across different time points (see [24, 25, 20, 19, 21] for more details).

Another example of information loss functions is mutual information, which we use in this study. Mutual information quantifies statistical dependence between parts. Let us consider the following disconnected probability distribution q ,

$$q(\mathbf{x}_S) = p(\mathbf{x}_{S_L})p(\mathbf{x}_{S_R}). \quad (3)$$

In this model q , the two parts S_L and S_R are independent, which means any kind of the dependence between S_L and S_R are completely removed. Then, by substituting Eq. (3) into Eq. (1), the mutual information $I(S_L; S_R)$ between the two parts S_L and S_R is given by

$$I(S_L; S_R) = \int p(\mathbf{x}_S) \log \frac{p(\mathbf{x}_S)}{p(\mathbf{x}_{S_L})p(\mathbf{x}_{S_R})} d\mathbf{x}_S. \quad (4)$$

The mutual information measures how strongly two subsystems are statistically dependent on each other. It becomes large when the two parts are strongly dependent on each other, and becomes 0 when two parts are independent, that is, $p(\mathbf{x}_S) = p(\mathbf{x}_{S_L})p(\mathbf{x}_{S_R})$. The mutual information Eq. (4) is also represented as

$$I(S_L; S_R) = H(S_L) + H(S_R) - H(S), \quad (5)$$

where $H(\cdot)$ represents the entropy, e.g., $H(S) = -\int p(\mathbf{x}_S) \log p(\mathbf{x}_S) d\mathbf{x}_S$. Equivalently, we can rewrite the above equation as a set function of S_L :

$$\begin{aligned} I_S(S_L) &= I(S_L; S \setminus S_L) \\ &= H(S_L) + H(S \setminus S_L) - H(S). \end{aligned} \quad (6)$$

We use these Eqs. (5) and (6) in Sections 5 and 8.

The mutual information has two important mathematical properties utilized in this study, as will be shown later. One is symmetric-submodularity, which enables a fast and exact search for minimum information partitions (MIPs) [26, 27] (Section 5). The other is monotonicity, which is a basis of our new algorithm HPC (Section 7).

5 Minimum information partition

In this section, we introduce the Minimum Information Partition (MIP) and the efficient algorithm for searching MIPs.

5.1 Definition of MIPs

The MIP is the bi-partition where the amount of information loss is minimum among all the bi-partitions. Mathematically, the MIP $(S_L^{\text{MIP}}, S_R^{\text{MIP}})$ of a subsystem S is defined as follows:

$$(S_L^{\text{MIP}}, S_R^{\text{MIP}}) = \arg \min_{(S_L, S_R) \in \mathcal{P}_S} f(S_L; S_R), \quad (7)$$

where \mathcal{P}_S denotes the set of all the bi-partitions of the subset S . Since the information loss function $f(S_L; S_R)$ is represented as a set function $f_S(S_L)$, the search of the MIP can be equivalently formulated as a minimization problem of the set function $f_S(S_L)$:

$$S_L^{\text{MIP}} = \arg \min_{S_L \subseteq S, S_L \neq \emptyset} f_S(S_L), \quad S_R^{\text{MIP}} = S \setminus S_L^{\text{MIP}}. \quad (8)$$

We represent the amount of information loss measured with the MIP as

$$f_S^{\text{MIP}} := f(S_L^{\text{MIP}}; S_R^{\text{MIP}}) = f_S(S_L^{\text{MIP}}). \quad (9)$$

Similarly, we represent the mutual information measured with the MIP as I_S^{MIP} . Hereinafter, we refer to a bi-partition as a partition, for simplicity.

By f_S^{MIP} , we can evaluate the “irreducibility” of the subsystem S . When $f_S^{\text{MIP}} = 0$, this means the subsystem S consists of independent parts, i.e., the subsystem S can be reduced to independent parts. In contrast, when f_S^{MIP} is nonzero, the subsystem S cannot be reduced to independent parts. No matter how the subsystem S is cut into parts, at least as much information as f_S^{MIP} is lost.

5.2 Algorithm for searching MIPs

If we search MIPs by exhaustively comparing all the partitions, the computation time grows exponentially with the number of elements in the system. In previous studies, we utilized a mathematical concept, submodularity, to reduce the computation time [28, 26, 27]. In particular, we used a submodular-based algorithm called Queyranne’s algorithm. In this subsection, we first introduce the definition of submodularity and then introduce Queyranne’s algorithm.

5.2.1 Submodularity

The submodularity is a property of set functions that is analogous to the concavity of continuous functions. Specifically, the submodularity is defined as follows.

Definition 1 (Submodularity). *A set function $f : 2^S \rightarrow \mathbb{R}$ is submodular if it satisfies the following inequality for any $A, B \subseteq S$:*

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B).$$

The mutual information $I_S(S_L)$ is submodular as a function of S_L , and the entropy H is also submodular. The submodularity of the mutual information can be easily derived from the submodularity of the entropy by using Eq. (6) [16].

If a submodular function $f : 2^V \rightarrow \mathbb{R}$ satisfies $f(S) = f(V \setminus S)$ for any subset $S \subseteq V$, the function f is called symmetric-submodular function. The mutual information $I_S(S_L)$ is a symmetric-submodular function defined over the powerset of S .

In general, given any submodular function $f : 2^S \rightarrow \mathbb{R}$, a function $g_S : 2^S \rightarrow \mathbb{R}$ defined as

$$g_S(S_L) = f(S_L) + f(S \setminus S_L) - f(S) \quad (S_L \subseteq S) \quad (10)$$

is symmetric-submodular [16]. We call, in this paper, this type of symmetric-submodular functions “symmetrized” submodular functions. The mutual information, which is symmetric-submodular, is especially “symmetrized”-submodular. We use the concept of symmetrized submodular functions in Section 8.

5.2.2 Queyranne’s algorithm

If a set function $f : 2^S \rightarrow \mathbb{R}$ is symmetric-submodular, we can exactly and efficiently find the minimum of the function by Queyranne’s algorithm [29]. Thus, we can use Queyranne’s algorithm to find MIPs, when the mutual information is used as an information loss function [28, 26]. The computation time of the algorithm is $O(|S|^3)$, where $|S|$ indicates the number of elements of S . This is much smaller than the exhaustive search, where the computation time is $O(2^{|S|})$.

6 Complex: informational core of a system

In this section, we introduce the definition of a complex [7, 8]. We also introduce a main complex, which is a stronger definition of a complex [7, 8].

A subsystem is called a complex if the amount of information loss measured with its MIP is nonzero and larger than those of all its supersets. The mathematical definition of a complex is given as follows.

Definition 2 (Complex). *A subset $S \subseteq V$ is called a complex if it satisfies $f_S^{\text{MIP}} > 0$ and $f_S^{\text{MIP}} > f_T^{\text{MIP}}$ for any of its superset T ($T \supset S$ and $T \subseteq V$).*

A schematic explanation of the definition of complexes is shown in Fig. 2. The subsystem $\{3, 4, 5\}$ is a complex, if it has greater f^{MIP} than all its supersets, that is, $f_{\{3, 4, 5\}}^{\text{MIP}}$ is larger than $f_{\{1, 3, 4, 5\}}^{\text{MIP}}$, $f_{\{3, 4, 5, 7\}}^{\text{MIP}}$, $f_{\{3, 4, 5, 6, 7\}}^{\text{MIP}}$, $f_{\{1, 2, 3, 4, 5, 6, 7\}}^{\text{MIP}}$, and so on.

The whole system V is a complex if it satisfies $f_V^{\text{MIP}} > 0$ by definition. We define $f^{\text{MIP}} = 0$ for single elements because we cannot consider partitions of a single element. Therefore, single elements cannot be complexes.

A subsystem is called a main complex if the amount of information loss measured with its MIP is larger than those of all its supersets, and is also larger than or equal to those of its subsets. In other words, a complex is called a main

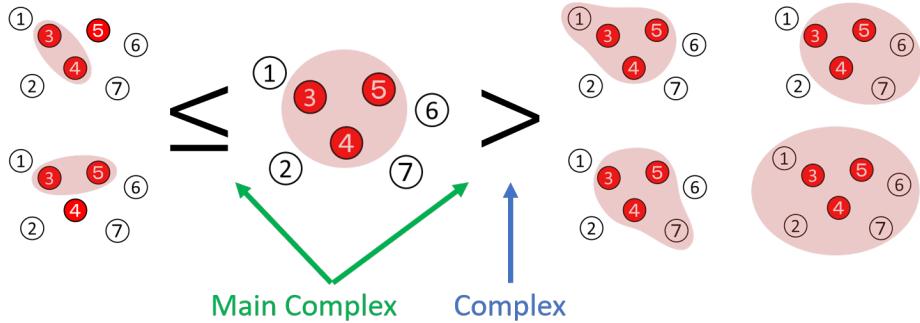


Figure 2: Schematic of the definitions of complex and main complex (Definitions 2 and 3). A subsystem S is a complex, if S has larger f^{MIP} than any supersets of S . In this example, the subsystem $\{3, 4, 5\}$ is a complex if $f_{\{3,4,5\}}^{\text{MIP}}$ is larger than $f_{\{1,3,4,5\}}^{\text{MIP}}$, $f_{\{3,4,5,7\}}^{\text{MIP}}$, $f_{\{3,4,5,6,7\}}^{\text{MIP}}$, $f_{\{1,2,3,4,5,6,7\}}^{\text{MIP}}$, and so on. The subsystem S is a main complex if S is a complex, and S has larger f^{MIP} than any subsets of S . The subsystem $\{3, 4, 5\}$ is a main complex if $\{3, 4, 5\}$ is a complex and $f_{\{3,4,5\}}^{\text{MIP}}$ is larger than $f_{\{3,4\}}^{\text{MIP}}$, $f_{\{3,5\}}^{\text{MIP}}$, and $f_{\{4,5\}}^{\text{MIP}}$.

complex if the amount of information loss with its MIP is larger than or equal to those of all its subsets.

Definition 3 (Main complex). *A complex is called a main complex if it satisfies $f_S^{\text{MIP}} \geq f_R^{\text{MIP}}$ for any of its subset R ($R \subset S$).*

A schematic explanation of the definition of main complexes is shown in Fig. 2. The subsystem $\{3, 4, 5\}$ is a main complex if it has f^{MIP} larger than those of all its supersets, and also, it has f^{MIP} equal to or larger than those of all its subsets, i.e., $\{3, 4\}$, $\{3, 5\}$, and $\{4, 5\}$.

A complex can be regarded as an informational core in the sense that adding elements to a complex always decreases irreducibility of the subsystem, which is quantified by f^{MIP} . A main complex can be also regarded as an informational core but in a stronger sense, because not only adding elements but also removing elements decreases irreducibility of the subsystem. Thus, a main complex can be considered as a locally-most-irreducible subsystem.

We end this section by mentioning a property of main complexes, which will be utilized in HPC. A main complex does not include smaller complexes by definition of main complexes and complexes. Conversely, If a complex S does not include smaller complexes, then S is a main complex. It can be easily shown by contradiction. Thus, a main complex is a complex that does not include smaller complexes. This property of main complexes is stated as a lemma below.

Lemma 4. *A complex is a main complex if and only if the complex does not include other complexes.*

7 Searching complexes

If we search (main) complexes by brute force, we need to find the MIPs of all the $O(2^N)$ subsets, and then compare f^{MIP} of the subsets to check if each subset satisfies the definition of (main) complexes (Defs. 2 and 3). In contrast, by using hierarchical partitioning for complex search (HPC), we need to find the MIPs of only $N - 1$ subsets.

In what follows, we describe the algorithm HPC. We first explain that the mutual information satisfies an important mathematical property, i.e., monotonicity. Then we show that, by taking advantage of the monotonicity of the mutual information, HPC enables us to efficiently find complexes.

7.1 Monotonicity of mutual information

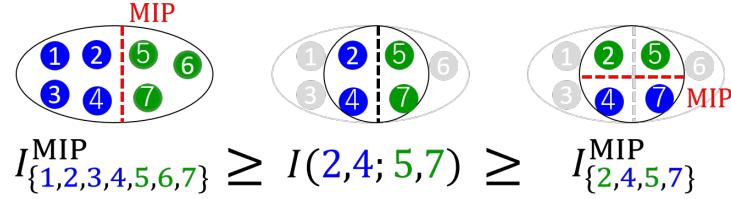


Figure 3: Schematic explanation of the proof of Proposition 5. Here we consider a system $T = \{1, 2, 3, 4, 5, 6, 7\}$. This system T is partitioned by its MIP into two subsystems $T_L = \{1, 2, 3, 4\}$ and $T_R = \{5, 6, 7\}$. The boundary of the MIP of T is indicated by a red-dashed vertical line. Then, we consider a subsystem $S = \{2, 4, 5, 7\} \subset T$. Since S intersects both T_L and T_R ($S \cap T_L = \{2, 4\} \neq \emptyset$ and $S \cap T_R = \{5, 7\} \neq \emptyset$), the first inequality $I_{\{1,2,3,4,5,6,7\}}^{\text{MIP}} = I_T^{\text{MIP}} \geq I(S \cap T_L; S \cap T_R) = I(2, 4; 5, 7)$ holds. The second inequality $I(2, 4; 5, 7) = I(S \cap T_L; S \cap T_R) \geq I_S^{\text{MIP}} = I_{\{2,4,5,7\}}^{\text{MIP}}$ holds by definition of the MIP. The boundary of the MIP of S is indicated by a red-dashed horizontal line.

The mutual information has a well-known mathematical property, i.e., monotonicity, as described below. Let us consider the mutual information between A and B , $I(A; B)$, where A and B are sets of elements. Then, if we add another set of elements C to A , the mutual information never decreases. That is,

$$I(A \cup C; B) \geq I(A; B). \quad (11)$$

Also if we add C to B , $I(A; B \cup C) \geq I(A; B)$. This inequality means that the mutual information monotonically increases as elements are added. We refer this mathematical property to ‘‘monotonicity.’’ The above inequality is equivalent to the non-negativity of the conditional mutual information

$$I(B; C|A) = I(A \cup C; B) - I(A; B) \geq 0. \quad (12)$$

To utilize this concept in HPC, we restate the monotonicity in the context of partitioning a system. Let T be a subset of V and (T_L, T_R) be a partition of T . For any non-empty subset S_L of T_L ($S_L \subset T_L$), the following inequality holds:

$$I(T_L; T_R) \geq I(S_L; T_R). \quad (13)$$

7.2 Auxiliary theorems

In this subsection, we show auxiliary theorems that are the basis of our new algorithm HPC.

By using the monotonicity of the mutual information Eq. (13), we can show an inequality that particularly holds for the mutual information measured with MIPs, which is denoted by I^{MIP} , as follows.

Proposition 5. *Let T be a subset of V , and (T_L, T_R) be the MIP of T . If a subset S of T ($S \subset T$) intersects both T_L and T_R ($S \cap T_L \neq \emptyset$ and $S \cap T_R \neq \emptyset$), then $I_T^{\text{MIP}} \geq I_S^{\text{MIP}}$.*

Proof. We show $I_T^{\text{MIP}} \geq I(S \cap T_L; S \cap T_R) \geq I_S^{\text{MIP}}$. The first inequality immediately follows from the monotonicity of the mutual information Eq. (13):

$$\begin{aligned} I_T^{\text{MIP}} &= I(T_L; T_R) \\ &\geq I(S \cap T_L; T_R) \\ &\geq I(S \cap T_L; S \cap T_R). \end{aligned} \quad (14)$$

Then, the second inequality holds by definition of the MIP: Since $(S \cap T_L, S \cap T_R)$ is a partition of S , that is, $(S \cap T_L) \cup (S \cap T_R) = S$ and $(S \cap T_L) \cap (S \cap T_R) = \emptyset$, the mutual information $I(S \cap T_L; S \cap T_R)$ is always larger than or equal to I_S^{MIP} . \square

A schematic explanation of (the proof of) Proposition 5 is shown in Fig. 3. In this example, a system $T = \{1, 2, 3, 4, 5, 6, 7\}$ is partitioned into $T_L = \{1, 2, 3, 4\}$ and $T_R = \{5, 6, 7\}$ with its MIP. A subsystem $S = \{2, 4, 5, 7\}$ is a subset of T ($S \subset T$). Then, $I_{\{1,2,3,4,5,6,7\}}^{\text{MIP}} = I_T^{\text{MIP}} \geq I_S^{\text{MIP}} = I_{\{2,4,5,7\}}^{\text{MIP}}$, because S intersects both T_L and T_R , that is, $S \cap T_L = \{2, 4, 5, 7\} \cap \{1, 2, 3, 4\} = \{2, 4\} \neq \emptyset$ and $S \cap T_R = \{2, 4, 5, 7\} \cap \{5, 6, 7\} = \{5, 7\} \neq \emptyset$.

The inequality in Proposition 5 means that the subset S has smaller I^{MIP} than one of its superset T , and therefore S does not satisfy the definition of complexes (Def. 2). Thus, the lemma below follows immediately.

Lemma 6. *Let T be a subset of V , and (T_L, T_R) be the MIP of T . If a subset S of T ($S \subset T$) intersects both T_L and T_R ($S \cap T_L \neq \emptyset$ and $S \cap T_R \neq \emptyset$), then S is not a complex.*

Lemma 6 means that a proper subset S of T ($S \subset T$) must be included in T_L or T_R to be a complex. In other words, any complex that is a subset of T is included in T_L or T_R . Therefore, the following proposition holds.

Proposition 7. Let T be a subset of V , and (T_L, T_R) be the MIP of T . Then any complex S that is a subset of T ($S \subseteq T$) satisfies one of the following mutually exclusive conditions (a)–(c).

- (a) $S = T$,
- (b) $S \subseteq T_L$,
- (c) $S \subseteq T_R$.

Proposition 7 is the main basis of our algorithm, hierarchical partitioning for complex search (HPC), as will be shown in Subsection 7.3.

7.3 Hierarchical partitioning for complex search

In this subsection, we explain our algorithm, hierarchical partitioning for complex search (HPC). HPC mainly consists of two steps. The first step is listing candidates of (main) complexes. HPC narrows down candidates for (main) complexes by hierarchically partitioning a system. The second step is screening the candidates to find (main) complexes. We explain the first step in Subsubsection 7.3.1, and then the second step for finding complexes and main complexes in Subsubsections 7.3.2 and 7.3.3, respectively.

7.3.1 Hierarchical partitioning for listing candidates of complexes

Hierarchical partitioning for complex search (HPC) is schematically shown in Fig. 4, and a pseudo-code is given in Algorithm 1. HPC enables finding all the complexes and main complexes by hierarchically partitioning the system with the minimum information partitions.

As is shown in Fig. 4, HPC starts by dividing the whole system with its MIP, and then repeatedly divides the subsystems with their MIPs until the whole system is completely decomposed into single elements. This procedure in HPC is summarized as follows:

1. Find the MIP (V_L, V_R) of the whole system V and divide the whole system V into two subsystems V_L and V_R .
2. Find the MIPs of the subsystems found in the previous step, V_L and V_R , and divide them into (V_{LL}, V_{LR}) and (V_{RL}, V_{RR}) , respectively.
3. Repeat this dividing until the whole system is decomposed into single elements.

After the procedure above, we obtain the set of hierarchically partitioned subsystems, i.e., $V, V_L, V_R, V_{LL}, V_{LR}, V_{RL}, V_{RR}$, and so on. We consider all the set of subsystems $\mathcal{V} = \{V, V_L, V_R, V_{LL}, V_{LR}, V_{RL}, V_{RR}, \dots\}$, excluding single elements. Then, the following theorem holds.

Theorem 8. Any complex $S \subseteq V$ belongs to \mathcal{V} ($S \in \mathcal{V}$).

Proof. By repeatedly applying Proposition 7 to $V, V_L, V_R, V_{LL}, V_{LR}, V_{RL}, V_{RR}$, and so on, we obtain the desired result. \square

Thus from Theorem 8, \mathcal{V} can be seen as the set of candidates of complexes. Also, Theorem 8 means that complexes in a system form a “nested” hierarchy: if there are two complexes, one includes the other, or the two are disjoint. They never partially overlap each other. This nested hierarchy of complexes cannot be derived only from the definition of complexes (Def. 2), because the definition does not specify the relation between complexes. For the nested hierarchy to hold, the inequality of the mutual information quantified at MIPs (Proposition 5) is necessary.

As described above, HPC decomposes the whole system into N single elements, by dividing a subsystem into two subsystems every step. This means that HPC divides the system $N - 1$ times. That is, HPC evaluates MIPs of $N - 1$ subsets. This number is much smaller than the number of subsets evaluated in the exhaustive search, $2^N - N - 1$, which is the number of subsets consisting of more than one element.

Next, we describe the way to select complexes and main complexes from the set of candidates \mathcal{V} .

7.3.2 Selection of complexes from the candidates

After the hierarchical partitioning procedure described above, we need to check whether each candidate of complexes belonging to \mathcal{V} is actually a complex or not in accordance with Def. 2. We can efficiently check this by taking advantage of the hierarchical (tree) structure. Please see Appendix A.1 for details.

7.3.3 Selection of main complexes from the candidates

As is stated in Lemma 4, a main complex is a complex that does not include smaller complexes. Thus, a main complex is a locally farthest complex from the root (the whole system) in the tree. Based on this, we can easily find main complexes. Please see Appendix A.2 for details.

7.4 Time complexity of HPC

In this subsection, we derive a rough estimate of the total time complexity for searching complexes by HPC. Here we only consider the first step of HPC (hierarchical partitioning for listing candidates of complexes, described in Subsubsection 7.3.1) and omit the second step (screening the candidates, described in Subsubsections 7.3.2 and 7.3.3), because the computation time of the second step is negligible.

The total time complexity of searching complexes depends on three factors, i.e., (1) computing the mutual information, (2) searching MIPs, and (3) searching complexes. The total computational cost is roughly bounded above by the products of these three factors (1)–(3). Each factor is given as follows.

- (1) The time complexity of computing the mutual information depends on the type of probability distribution. If the probability distribution is a Gaus-

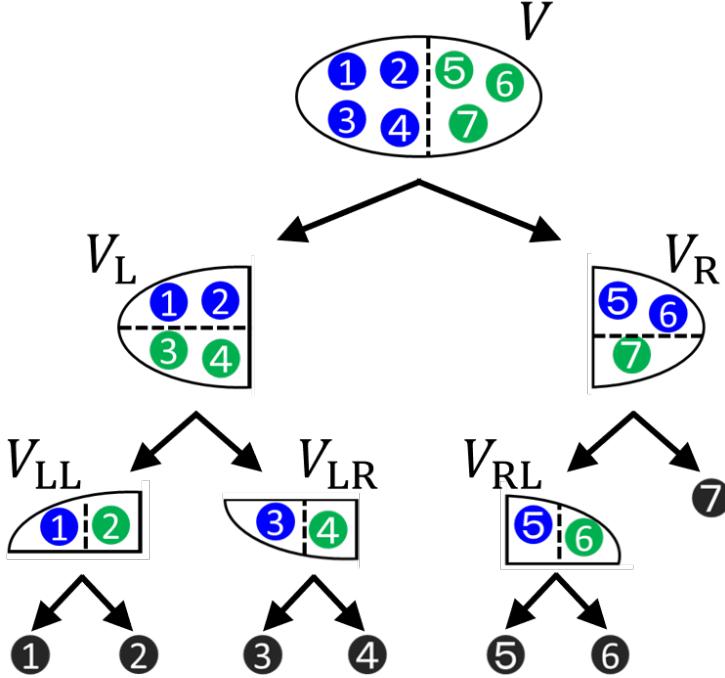


Figure 4: Schematic of hierarchical partitioning for complex search (HPC). In HPC, a system is hierarchically partitioned by MIPs until the system is decomposed into single elements. In this example, the whole system $V = \{1, 2, 3, 4, 5, 6, 7\}$ is divided by its MIP (indicated by a dashed line) into $V_L = \{1, 2, 3, 4\}$ and $V_R = \{5, 6, 7\}$. Then, V_L is divided into V_{LL} and V_{LR} , and V_R into V_{RL} and {7}. Finally, the whole system V is decomposed into seven single elements.

sian, as we assume in Experiments in Section 9, then the time complexity is $O(N^3)$.

- (2) For searching MIPs, we utilize Queyranne's algorithm in this study. The time complexity of the Queyranne's algorithm is known to be $O(N^3)$ [29].
- (3) HPC finds MIPs of $N - 1$ subsets as shown in Section 7. Thus, the time complexity of HPC is $O(N)$.

Therefore, the total time complexity is about $O(N^3) \times O(N^3) \times O(N) = O(N^7)$. We evaluate actual computation time in Subsection 9.2 by a simulation.

Algorithm 1 Hierarchical Partitioning for Complex Search (HPC)

```

procedure HPC( $V$ )
     $\mathcal{V}, \mathcal{I}, parents \leftarrow \text{ListCandidates}(V)$ 
     $cs, mcs \leftarrow \text{ScreenCandidates}(\mathcal{V}, \mathcal{I}, parents)$ 
    return  $cs, mcs$                                  $\triangleright cs$ : complexes,  $mcs$ : main complexes
end procedure

procedure LISTCANDIDATES( $V$ )
    if  $|V| = 1$  then                             $\triangleright V$  is indivisible, i.e., a single element
         $\mathcal{V}, \mathcal{I}, parents \leftarrow \{\}$ 
    else
         $V_L, V_R, I_V^{\text{MIP}} \leftarrow \text{MIP}(V)$            $\triangleright$  Find the MIP of  $V$ 
         $parents(V_L) \leftarrow V$ 
         $parents(V_R) \leftarrow V$ 
         $\mathcal{V}_L, \mathcal{I}_L, parents_L \leftarrow \text{ListCandidates}(V_L)$ 
         $\mathcal{V}_R, \mathcal{I}_R, parents_R \leftarrow \text{ListCandidates}(V_R)$ 
         $\mathcal{V} \leftarrow \{V, \mathcal{V}_L, \mathcal{V}_R\}$ 
         $\mathcal{I} \leftarrow \{I_V^{\text{MIP}}, \mathcal{I}_L, \mathcal{I}_R\}$ 
         $parents \leftarrow \{parents(V_L), parents_L, parents(V_R), parents_R\}$ 
    end if
    return  $\mathcal{V}, \mathcal{I}, parents$                        $\triangleright \mathcal{V}$ : candidates of (main) complexes
end procedure

procedure SCREENCANDIDATES( $\mathcal{V}, \mathcal{I}, parents$ )
     $cs \leftarrow \{V\}, mcs \leftarrow \{V\}, mrac(V) \leftarrow V$        $\triangleright mrac$ : most-recent ancestor
    complex or self
    for all  $T \in \mathcal{V} \setminus \{V\}$  do           $\triangleright$  Tree traversal starting from the root node
        if  $I_T^{\text{MIP}} > \max_{S \in A(parents(T))} I_S^{\text{MIP}}$  then
             $\max_{S \in A(T)} I_S^{\text{MIP}} \leftarrow I_T^{\text{MIP}}$ 
            add  $T$  to  $cs$ 
            remove  $mrac(parents(T))$  from  $mcs$ 
            add  $T$  to  $mcs$ 
             $mrac(T) \leftarrow T$ 
        else
             $\max_{S \in A(T)} I_S^{\text{MIP}} \leftarrow \max_{S \in A(parents(T))} I_S^{\text{MIP}}$ 
             $mrcs(T) \leftarrow mrac(parents(T))$ 
        end if
    end for
    return  $cs, mcs$ 
end procedure

```

8 A relation between the monotonicity and the submodularity

Before moving to the experiments, we show in this section a relation between monotonicity (Eq. (13)) and submodularity (Def. 1). We show that monotonicity is satisfied by a broader class of submodular functions. Based on the relation, we show that we can extend our framework to the class of submodular functions.

As is shown in subsection 7.1, the mutual information has monotonicity, i.e., $I(T_L; T_R) \geq I(S_L; T_R)$ (Eq. (13)). This monotonicity can be derived only from the submodularity of the entropy as follows.

Proof.

$$\begin{aligned}
& I(T_L; T_R) - I(S_L; T_R) \\
&= \{H(T_L) + H(T_R) - H(T)\} - \{H(S_L) + H(T_R) - H(S_L \cup T_R)\} \\
&= H(T_L) + H(S_L \cup T_R) - H(T) - H(S_L) \\
&= H(T_L) + H(S_L \cup T_R) - H(T_L \cup (S_L \cup T_R)) - H(T_L \cap (S_L \cup T_R)) \\
&\geq 0 \quad (\because H \text{ is submodular})
\end{aligned} \tag{15}$$

□

This inequality was shown as the non-negativity of the conditional mutual information in [30]. This proof indicates that the relation between the monotonicity and the submodularity can be stated in a general form [30]:

Proposition 9. *Let $f : 2^V \rightarrow \mathbb{R}$ be a submodular function, and $g : 2^V \times 2^V \rightarrow \mathbb{R}$ be a function defined as*

$$g(A; B) = f(A) + f(B) - f(A \cup B). \tag{16}$$

Let T be a subset of V and (T_L, T_R) be a partition of T . Then, the function g satisfies the following inequality for any non-empty subset S_L of T_L ($S_L \subset T_L$).

$$g(T_L; T_R) \geq g(S_L; T_R). \tag{17}$$

This proposition means that, given any submodular function f , we can define a new function g that satisfies the monotonicity. Consequently, the function g satisfies all the auxiliary theorems shown in Subsection 7.2, as the mutual information does. Therefore, if we regard g as an information loss function (although it does not have to be “informational”), and define complexes using g , we can use HPC to search the complexes. Note that a function with the form Eq. (16) satisfies the monotonicity, but not the other way around, i.e., a function that satisfies the monotonicity is not necessarily represented as the form of Eq. (16).

In addition, the function $g(S_L; S_R) = g_S(S_L)$ is a symmetrized submodular function (Subsection 5.2). This means that we can use Queyranne’s algorithm to search MIPs, when we use g as an information loss function.

Thus, if we regard a symmetrized submodular function as an information loss function, we can utilize Queyranne's algorithm and HPC to search MIPs and complexes, respectively. This is summarized as follows.

Submodular Complex (Complex for symmetrized submodular functions)

Given any submodular function $f : 2^V \rightarrow \mathbb{R}$, we consider a function g defined as

$$g(S_L; S_R) = f(S_L) + f(S_R) - f(S). \quad (18)$$

Then, the function g satisfies the same properties as the mutual information, i.e., symmetric-submodularity and monotonicity. Therefore, if we regard g as an information loss function, we can use Queyranne's algorithm and HPC to search for MIPs and complexes, respectively.

9 Experiments

We first evaluated the performance of the proposed algorithm HPC using simulated data in Subsections 9.1 and 9.2. Throughout the simulations below, we consider a first-order autoregressive (AR) model with Gaussian noise,

$$\mathbf{x}' = A\mathbf{x} + \boldsymbol{\epsilon}, \quad (19)$$

where \mathbf{x} and \mathbf{x}' are the present and the past states of a system consisting of N elements, A is a $N \times N$ matrix called connectivity matrix, and $\boldsymbol{\epsilon}$ is Gaussian noise with mean 0 and covariance $\Sigma(\mathbf{E})$. We consider the stationary distribution of this AR model. The stationary distribution of $p(\mathbf{x})$ is a Gaussian distribution with mean 0 and covariance $\Sigma(\mathbf{X})$. The covariance $\Sigma(\mathbf{X})$ can be computed from the following discrete Lyapunov equation

$$\Sigma(\mathbf{X}) = A\Sigma(\mathbf{X})A^T + \Sigma(\mathbf{E}). \quad (20)$$

By using the covariance matrix $\Sigma(\mathbf{X})$, the mutual information can be analytically calculated (see Appendix B). The details of the parameter settings are described in each subsection.

9.1 A simple example

We demonstrate how the proposed algorithm HPC works by considering a simple exemplary model. We consider an AR model with six elements ($N = 6$). The connectivity matrix A of the model is shown in Fig. 5 as a network. The connectivity matrix A is symmetric and consists of two modules. One module consists of two elements 1 and 2, and the other one consists of four elements 3–6. The intra-connection strength in each group is $0.05/N$, except the connection between the elements 5 and 6. The connection strength between 5 and 6 is $0.1/N$, which is stronger than those between the other pairs. The inter-connections between the groups are $0.01/N$, which are weaker than the intra-connections in each group. The strength of self connections is set to $0.9/N$. The covariance $\Sigma(\mathbf{E})$ of Gaussian noise is set to $0.01I$, where I is an identity matrix.

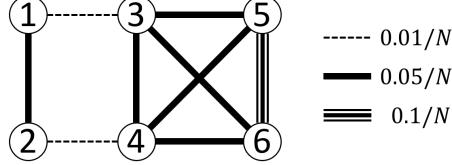


Figure 5: The connectivity matrix A of the AR model used in Subsection 9.1. The matrix A is symmetric, i.e., $a_{ij} = a_{ji}$. The self connections (a_{ii}) are omitted from this figure for simplicity.

Figure 6 shows how HPC hierarchically divides the system with MIPs. First, since the connections between two subsystems $\{1, 2\}$ and $\{3, 4, 5, 6\}$ are weak, the partition that splits the system into $\{1, 2\}$ and $\{3, 4, 5, 6\}$ is the MIP of the entire system. Then, these two subsystems are further divided with their MIPs. The first subsystem $\{1, 2\}$ is divided into single elements $\{1\}$ and $\{2\}$. The second subsystem $\{3, 4, 5, 6\}$ is successively divided into $\{3\}$ and $\{4, 5, 6\}$, $\{4\}$ and $\{5, 6\}$, and $\{5\}$ and $\{6\}$. Among these subsystems, $\{1, 2\}$, $\{3, 4, 5, 6\}$ and $\{5, 6\}$ were identified as complexes (Fig. 7), by comparing the amount of the mutual information of the subsystems measured with their MIPs as described in Section 7. Furthermore, the subsystems $\{1, 2\}$ and $\{5, 6\}$ are identified as main complexes.

Figure 7 visualizes the relation among complexes, main complexes, and subsets that are not complexes. For example, the subsystem $\{4, 5, 6\}$ is not a complex, because it is at a lower position than one of its ancestors $\{3, 4, 5, 6\}$ in the hierarchy. In contrast, the subsystem $\{3, 4, 5, 6\}$ is a complex, because it is at higher position than its only ancestor $\{1, 2, 3, 4, 5, 6\}$. The subsystems $\{1, 2\}$ and $\{5, 6\}$ are main complexes because they are at locally highest positions. HPC evaluates I^{MIP} of only five subsystems, while the exhaustive search evaluates that of 57($= 2^6 - 7$) subsystems. Thus, our algorithm HPC efficiently finds the complexes and the main complexes by hierarchically partitioning the system.

9.2 Computation time of HPC

Next, we empirically measured the computation time of searching complexes by simulation.

We randomly generated the connectivity matrices A in Eq. (19). We determined each element of the connectivity matrix A by sampling from a Gaussian distribution with mean 0 and variance $0.01/N$, where N is the number of elements. The covariance $\Sigma(\mathbf{E})$ of the additive Gaussian noise in AR model was set to $0.01I$. All computation times were measured on a machine with an Intel Xeon Gold 6154 at 3.00GHz. All the calculations were implemented in MATLAB 2019a.

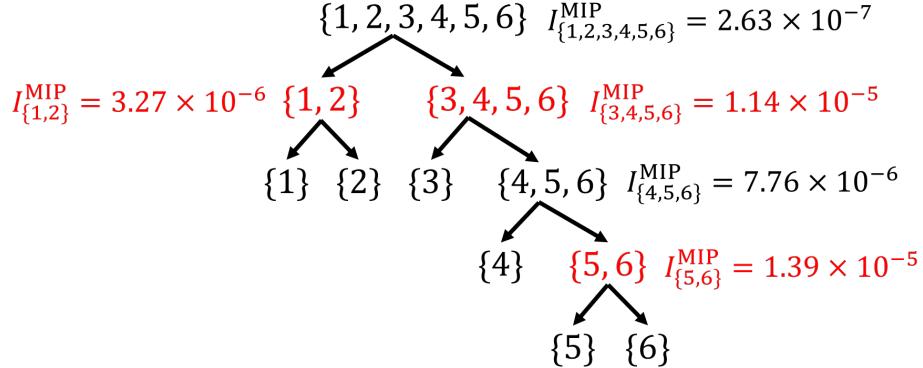


Figure 6: The way how HPC divided the system at MIPs. First, the entire system was divided into two subsystems $\{1, 2\}$ and $\{3, 4, 5, 6\}$. Then, the first subsystem $\{1, 2\}$ was divided into single elements $\{1\}$ and $\{2\}$. The second subsystem $\{3, 4, 5, 6\}$ was successively divided into $\{3\}$ and $\{4, 5, 6\}$, $\{4\}$ and $\{5, 6\}$, and $\{5\}$ and $\{6\}$. Among these subsystems, $\{1, 2\}$, $\{3, 4, 5, 6\}$ and $\{5, 6\}$ are complexes, which are indicated by red color. The subsystems $\{1, 2\}$ and $\{5, 6\}$ are main complexes.

Figure 8 shows the log–log plot of the actual computation time. As a comparison with the HPC algorithm, we also measured the computation time when complexes are exhaustively searched by brute force. The red circles indicate the computation time of the proposed algorithm HPC. To estimate the order of computation time, we fitted a linear function to the red circles by the minimum mean squared error estimation. We discarded the first five circles ($N \leq 50$) from the fitting analysis, because, when N is small, computation time is affected by lower-order terms. We obtained the red solid line, $\log_{10} T = 5.139 \log_{10} N - 7.104$, shown in Fig. 8. As can be seen, the red solid line well approximates the red circles for larger N . This means that the computation time of HPC increases in polynomial order ($T \propto N^{5.139}$). This is reasonably bounded above by the theoretical estimate of the computation time $T \propto N^7$ (Subsection 7.4)¹. In contrast, when we exhaustively searched complexes, the computation time grows exponentially as indicated by the black triangles. We fitted an exponential function to the black triangles by the minimum mean square error estimation. We also discarded the first five points ($N \leq 7$) from the fitting analysis. We obtained the black dashed curve, $T \propto 2.4689^N$ as shown in Fig. 8.

As can be seen from Fig. 8, HPC is much faster than the exhaustive search. For example, when $N = 10^2$, the actual computation time of HPC was about 24

¹A main reason why the empirical computation speed is faster than the theoretical estimate is thanks to the computation of the mutual information by MATLAB. As we mentioned in Subsection 7.4, the time complexity of the computation of the mutual information for Gaussian distributions is $O(N^3)$. However, the actual computation time grows more slowly (from $O(N)$ to $O(N^2)$), when N is up to several hundred.

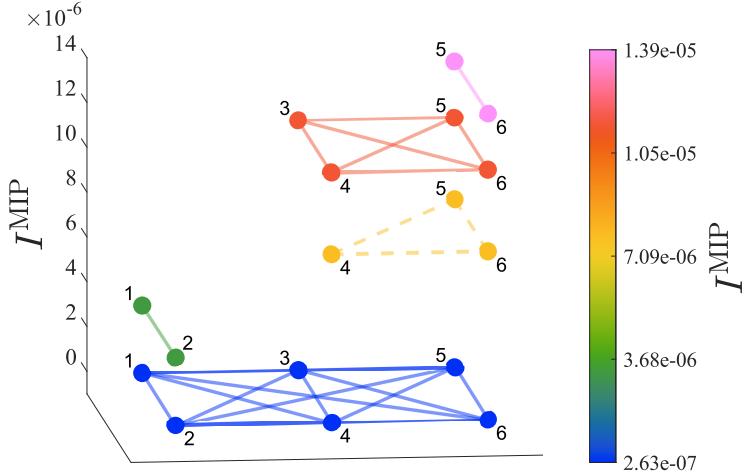


Figure 7: The amount of the mutual information I^{MIP} of subsystems appeared in the dividing process of HPC. The elements in a subsystem are connected by edges in the same color. The color indicates the amount of the mutual information I^{MIP} . The subsystems with solid lines are (main) complexes, and that with dashed lines is a non-complex.

minutes, while that of the exhaustive search would be about 3.36×10^{28} years.

9.3 An application to real neural data

Finally, we applied HPC to a neural dataset to demonstrate how HPC can be used in real settings. We used electrocorticography (ECoG) data recorded from a macaque monkey. The dataset is available at an open database, Neurotycho.org (<http://neurotycho.org/>) [17]. One hundred twenty-eight electrodes were implanted in the left hemisphere. The electrodes were placed at 5 mm intervals, covering the frontal, parietal, temporal, and occipital lobes, and medial frontal and parietal walls. Signals were sampled at a rate of 1 kHz. The monkey was awake with the eyes covered by an eye-mask to restrain visual responses. To remove line noise and artifacts, we performed bipolar re-referencing between adjacent electrode pairs, i.e. subtracting the signal of one electrode from that of the other one. The number of bipolar re-referenced electrodes was 64 in total. Among the 64 channels, two channels were removed from further analysis because of measurement noise.

We extracted 15-minute signals and divided them into 1-minute time windows. Each 1-minute time window consists of $1 \text{ kHz} \times 60 \text{ s} = 60,000$ samples. We searched complexes in each time window. We approximated the probability distribution of the signals with multivariate Gaussian distributions. Under the Gaussian approximation, we can compute the mutual information by using the

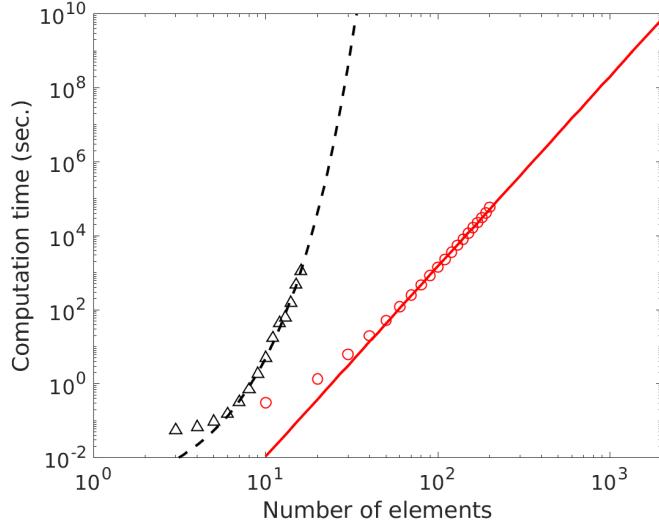


Figure 8: computation time of the proposed algorithm HPC and the exhaustive search. The red circles and the red solid lines indicate the computation time of HPC and the fitted linear function ($\log_{10} T = 5.139 \log_{10} N - 7.104$). The black triangles and black dashed lines indicate the computation time of the exhaustive search and the fitted exponential function ($\log_{10} T = 0.3925N - 3.225$).

equation shown in Appendix B.

Figure 9 shows the main complexes and complexes in the first time window. In each panel, the main complexes and complexes are superimposed in ascending order of the amount of mutual information I^{MIP} . We can see that the main complexes consist of pairs of nearby channels. On the other hand, we can see that the complexes other than the main complexes, by definition, consist of more channels than the main complexes. Especially, the complexes with large I^{MIP} tend to consist of channels clustered in posterior areas: the complex with the smallest I^{MIP} among all the complexes is the whole system (all the 62 channels, shown in blue color). Then, the complexes with moderate I^{MIP} consist of channels clustered in the middle and posterior areas (shown in bluish-green color), and complexes with large I^{MIP} consist of channels clustered in posterior areas (shown in yellowish-green color). Thus, we can see a tendency that the complexes consist of channels in the more posterior area as I^{MIP} increases. Every (main) complex at the first time window is separately shown in Appendix C. We obtained the similar tendency at the different time windows: main complexes consist of pairs or triples of channels, and complexes cover the more posterior area as I^{MIP} increases (Figs. 11 and 12 in Appendix C).

Thus, in our experiment here, since main complexes consist of only pairs or triples of channels, it seems difficult to gain some insights only from the main

complexes. It seems better to analyze not only main complexes but complexes to extract meaningful information.

10 Discussion

In this study, we proposed a fast algorithm for searching informational cores, complexes. We call the proposed algorithm Hierarchical Partitioning for Complex search (HPC) because it narrows down candidates for complexes by hierarchically partitioning a system. We proved that the proposed algorithm HPC can efficiently enumerate all the complexes and the main complexes when the mutual information is used as an information loss function. The number of subsystems whose MIPs are evaluated in HPC is a linear order of system size. This is dramatically smaller than that of an exhaustive search, which is exponential of system size. We also proved that if we regard a specific type of symmetric submodular function, which we call a symmetrized submodular function, as an information loss function, we can apply HPC to search complexes. In the experiments, we numerically evaluated the computation time of the overall complex search process by HPC and showed that the computation time was approximately $O(N^5)$. We applied HPC to a simple model and monkey ECoG data to demonstrate how HPC can be applied.

In monkey ECoG data analyses in Subsection 9.3, the main complexes were small, i.e., pairs or triples of channels. This may be because the channels in a main complex were spatially so close to each other that the correlation among them was strong. Strong correlation results in a large amount of mutual information because the mutual information is a monotonically increasing function of the correlation coefficient. Thus, considering main complexes only may be vulnerable to artifacts, e.g., inhomogeneity in the spatial arrangement of electrodes. In contrast, the complexes were larger than the main complexes, and there was a stable tendency that the complexes covered the posterior area. Therefore, it will be recommended to consider not only main complexes but also complexes to make analyses reliable.

HPC enables us to search (main) complexes in systems consisting of several hundred elements ($N \sim 100\text{--}200$) in a practical amount of time. The number of channels in EEG or ECoG is typically within this range. The (main) complex is a concept that is originally proposed as a locus of consciousness in the integrated information theory of consciousness. However, it can be utilized to analyze brain networks in contexts unrelated consciousness and also other probabilistic systems other than the brain. Therefore, HPC will be beneficial not only for consciousness studies but also for other general research fields.

As shown in Section 7, the inequality in Proposition 5 is a prerequisite for HPC to be exact. While the mutual information satisfies the inequality, other information loss functions, e.g., stochastic interaction [31, 32, 33], integrated information based on mismatched decoding [25], and geometric integrated information [19], do not necessarily satisfy it. Therefore, when those functions are utilized, there is no guarantee that HPC can find all the (main) complexes.

However, HPC might practically work well and might be used as an approximate algorithm. It would be an interesting future work to test to what extent HPC can be applied to those functions.

As shown in Section 8, our framework can be naturally extended by regarding a symmetrized submodular function as an information loss function. An example of such symmetrized submodular functions other than the mutual information is the weight of a graph cut. Since graph representations of systems are useful and important in many scientific fields, we extend our framework to graphs in the next study.

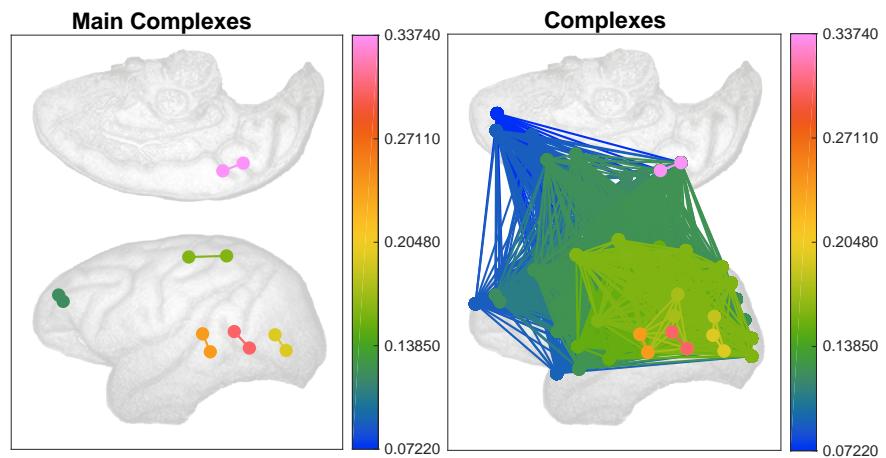


Figure 9: Main complexes and complexes at the fist time window. The channels in each (main) complex are connected by the edges with the same color. The color indicates the amount of the mutual information I^{MIP} of the (main) complex. The main complexes and complexes are superimposed in ascending order of the amount of mutual information I^{MIP} .

A Select complexes and main complexes from the candidate set \mathcal{V}

In this section, we explain how we can select complexes and main complexes from the candidate set \mathcal{V} , which is defined in Subsection 7.3.

A.1 Complex

We need to check whether each candidate of complexes belonging to \mathcal{V} actually satisfies the condition in Def. 2. The condition means that a candidate must have larger I^{MIP} than its supersets to be a complex. To check this, the tree structure obtained by the hierarchical partitioning procedure can be utilized: the candidate needs to be compared only with its ancestor candidates in the tree because its supersets other than the ancestors straddle at least one of the MIP boundaries of the ancestors and therefore have smaller I^{MIP} than the ancestors (Proposition 5). Thus, a candidate is a complex if it has larger I^{MIP} than its ancestor candidates. This comparison can be efficiently done by propagating the maximum value of I^{MIP} among ancestors and self from the root node (the whole system) to the leaf nodes (the single elements) as described below.

Here, let $A(\cdot)$ denote ancestors and self. For example, $A(V_{\text{LLR}})$ indicates $\{V, V_L, V_{\text{LL}}, V_{\text{LLR}}\}$. Starting from the root node V , the amount of the mutual information I_T^{MIP} of each candidate $T \in \mathcal{V}$ is compared with the maximum of the mutual information of T 's ancestors $\max_{S \in A(\text{parents}(T))} I_S^{\text{MIP}}$:

$$\begin{aligned} & \text{if } I_T^{\text{MIP}} > \max_{S \in A(\text{parents}(T))} I_S^{\text{MIP}}, \text{ then } \begin{cases} T \text{ is a complex, and} \\ \max_{S \in A(T)} I_S^{\text{MIP}} = I_T^{\text{MIP}}, \end{cases} \\ & \text{otherwise } \begin{cases} T \text{ is not a complex, and} \\ \max_{S \in A(T)} I_S^{\text{MIP}} = \max_{S \in A(\text{parents}(T))} I_S^{\text{MIP}}. \end{cases} \end{aligned} \quad (21)$$

By repeating this comparison while traversing the entire tree from the root to leaf nodes (i.e., in pre-order), we can list complexes without omission. In total, the comparison is done $|\mathcal{V}|$ times, where $|\mathcal{V}|$ indicates the number of subsystems in \mathcal{V} . We can easily show that $|\mathcal{V}| = N - 1$ as follows. Before the dividing steps, there is only one system (the whole system V) in \mathcal{V} . Then, two subsystems are added to \mathcal{V} every dividing step. Since HPC divides the system $N - 1$ times as described above, there are $1 + 2(N - 1)$ subsystems in \mathcal{V} at the end. Note, however, this number counts N single elements. Therefore, we finally get $|\mathcal{V}| = 1 + 2(N - 1) - N = N - 1$.

A.2 Main complex

As we stated in Subsubsection 7.3.3, a main complex is a locally farthest complex from the root in the tree. Based on this, we can easily find main complexes during the tree traversal as follows. If we find a complex, add the complex to the list of main complexes temporarily. Then, if we find another complex

that is a descendant of a temporal main complex, we remove the temporal main complex from the list and add the new one to the list. After finishing the tree traversal, we obtain the complete list of main complexes.

B Analytical formula of mutual information for Gaussian distribution

We describe the analytical formula of the mutual information when the probability distribution is Gaussian. Let us begin by introducing the notation. We consider a probabilistic system consisting of N elements. We represent the states of the system as $\mathbf{x}_S = (x_1, \dots, x_N)$. We assume that the probability distribution $p(\mathbf{x}_S)$ is Gaussian:

$$\begin{aligned} p(\mathbf{x}_S) &= \frac{1}{\sqrt{|2\pi\Sigma|}} \exp\left(-\frac{1}{2}\mathbf{x}_S^T \Sigma^{-1} \mathbf{x}_S\right), \\ &= \frac{1}{\sqrt{|2\pi\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{x}_{S_L}^T, \mathbf{x}_{S_R}^T) \begin{pmatrix} \Sigma_L & \Sigma_{LR} \\ \Sigma_{LR}^T & \Sigma_R \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{x}_{S_L} \\ \mathbf{x}_{S_R} \end{pmatrix}\right), \end{aligned} \quad (22)$$

where Σ , Σ_L , Σ_R are the covariance matrices of \mathbf{x}_S , \mathbf{x}_{S_L} , and \mathbf{x}_{S_R} , Σ_{LR} is the covariance matrix between \mathbf{x}_{S_L} , and \mathbf{x}_{S_R} , and $|\cdot|$ indicates the determinant. Then, we get the marginalized distributions:

$$\begin{aligned} p(\mathbf{x}_{S_L}) &= \frac{1}{\sqrt{|2\pi\Sigma_L|}} \exp\left(-\frac{1}{2}\mathbf{x}_{S_L}^T \Sigma_L^{-1} \mathbf{x}_{S_L}\right), \\ p(\mathbf{x}_{S_R}) &= \frac{1}{\sqrt{|2\pi\Sigma_R|}} \exp\left(-\frac{1}{2}\mathbf{x}_{S_R}^T \Sigma_R^{-1} \mathbf{x}_{S_R}\right). \end{aligned} \quad (23)$$

Note that we can assume the mean of the Gaussian distribution is zero without loss of generality because the mean value does not affect the values of mutual information.

The entropy of an N -dimensional Gaussian distribution with a covariance matrix Σ is given by $\frac{1}{2} \log |2\pi\Sigma| + \frac{N}{2}$. By substituting this expression to Eq. (5), we get

$$I(S_L; S_R) = \frac{1}{2} \log \frac{|\Sigma_L||\Sigma_R|}{|\Sigma|}. \quad (24)$$

C Results of ECoG data analyses

We show the (main) complexes in the first time window in the experiment in Subsection 9.3. All the (main) complexes in the first time window are separately shown in Figure 10. We can see that the main complexes are small. In contrast, the complexes are larger than the main complexes, and that the channels in the frontal area withdraw from the complexes and the complexes cover more posterior areas with an increasing amount of I^{MIP} .

The main complexes and complexes at all the 15 time windows are shown in Figs 11 and 12. We can see that the main complexes are pairs or triples of channels. The complexes at different time windows are not exactly the same but we can see a common tendency that the complexes cover more posterior areas with an increasing amount of I^{MIP} .

References

- [1] Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186–198, 2009.
- [2] Steven L. Bressler and Vinod Menon. Large-scale brain networks in cognition: emerging methods and principles. *Trends in Cognitive Sciences*, 14(6):277 – 290, 2010.
- [3] Danielle S Bassett and Olaf Sporns. Network neuroscience. *Nature neuroscience*, 20(3):353, 2017.
- [4] M-Marsel Mesulam. Large-scale neurocognitive networks and distributed processing for attention, language, and memory. *Annals of Neurology: Official Journal of the American Neurological Association and the Child Neurology Society*, 28(5):597–613, 1990.
- [5] Patric Hagmann, Leila Cammoun, Xavier Gigandet, Reto Meuli, Christopher J Honey, Van J Wedeen, and Olaf Sporns. Mapping the structural core of human cerebral cortex. *PLOS Biology*, 6(7):1–15, 07 2008.
- [6] Amanda V. Utevsky, David V. Smith, and Scott A. Huettel. Precuneus is a functional core of the default-mode network. *Journal of Neuroscience*, 34(3):932–940, 2014.
- [7] David Balduzzi and Giulio Tononi. Integrated information in discrete dynamical systems: motivation and theoretical framework. *PLoS Comput. Biol.*, 4(6):e1000091, June 2008.
- [8] Giulio Tononi. Consciousness as integrated information: a provisional manifesto. *The Biological Bulletin*, 215(3):216–242, 2008.
- [9] G Tononi, O Sporns, and G M Edelman. A measure for brain complexity: relating functional segregation and integration in the nervous system. *Proc. Natl. Acad. Sci. U. S. A.*, 91(11):5033–7, May 1994.
- [10] Giulio Tononi. An information integration theory of consciousness. *BMC Neurosci.*, 5:42, November 2004.
- [11] Masafumi Oizumi, Larissa Albantakis, and Giulio Tononi. From the phenomenology to the mechanisms of consciousness: integrated information theory 3.0. *PLoS Comput. Biol.*, 10(5):e1003588, May 2014.

- [12] Giulio Tononi, Melanie Boly, Marcello Massimini, and Christof Koch. Integrated information theory: from consciousness to its physical substrate. *Nature Reviews Neuroscience*, 17(7), 2016.
- [13] UnCheol Lee, George A Mashour, Seunghwan Kim, Gyu-Jeong Noh, and Byung-Moon Choi. Propofol induction reduces the capacity for neural information integration: implications for the mechanism of consciousness and general anesthesia. *Consciousness and Cognition*, 18:56–64, 2009.
- [14] Larissa Albantakis, Arend Hintze, Christof Koch, Christoph Adami, and Giulio Tononi. Evolution of integrated causal structures in animats exposed to environments of increasing complexity. *PLoS Computational Biology*, 10(12):e1003966, 2014.
- [15] Larissa Albantakis and Giulio Tononi. The intrinsic cause-effect power of discrete dynamical systemsfrom elementary cellular automata to adapting animats. *Entropy*, 17:54725502, 2015.
- [16] Francis Bach. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends in Machine Learning*, 6(2-3):145–373, 2013.
- [17] Yasuo Nagasaka, Kentaro Shimoda, and Naotaka Fujii. Multidimensional recording (mdr) and data sharing: An ecological open research and educational platform for neuroscience. *PLoS One*, 6(7), 07 2011.
- [18] Kenneth P Burnham and David R Anderson. *Model selection and multi-model inference: a practical information-theoretic approach*. Springer Science & Business Media, 2003.
- [19] Masafumi Oizumi, Naotsugu Tsuchiya, and Shun-ichi Amari. Unified framework for information integration based on information geometry. *Proceedings of the National Academy of Sciences*, 113(51):14817–14822, 2016.
- [20] Max Tegmark. Improved measures of integrated information. *PLoS Computational Biology*, 12(11):e1005123, 2016.
- [21] Shunichi Amari, Naotsugu Tsuchiya, and Masafumi Oizumi. Geometry of information integration. *arXiv preprint arXiv:1709.02050*, 2017.
- [22] Pedro A.M. Mediano, Anil K. Seth, and Adam B. Barrett. Measuring integrated information: Comparison of candidate measures in theory and simulation. *Entropy*, 21(17), 2019.
- [23] Pedro A.M. Mediano, Fernando Rosas, Robin L. Carhart-Harris, Anil K. Seth, and Adam B. Barrett. Beyond integrated information: A taxonomy of information dynamics phenomena. *arXiv*, 2019.
- [24] Adam B Barrett and Anil K Seth. Practical measures of integrated information for time-series data. *PLoS Comput. Biol.*, 7(1):e1001052, January 2011.

- [25] Masafumi Oizumi, Shunichi Amari, Toru Yanagawa, Naotaka Fujii, and Naotsugu Tsuchiya. Measuring integrated information from the decoding perspective. *PLoS Comput. Biol.*, 12(1):e1004654, 2016.
- [26] Shohei Hidaka and Masafumi Oizumi. Fast and exact search for the partition with minimal information loss. *arXiv preprint*, 2017.
- [27] Jun Kitazono, Ryota Kanai, and Masafumi Oizumi. Efficient algorithms for searching the minimum information partition in integrated information theory. *Entropy*, 20(3):173, 2018.
- [28] Mukund Narasimhan, Nebojsa Jojic, and Jeff Bilmes. Q-clustering. In *Advances in Neural Information Processing Systems 18*, pages 979–986. MIT Press, 2006.
- [29] Maurice Queyranne. Minimizing symmetric submodular functions. *Mathematical Programming*, 82(1-2):3–12, 1998.
- [30] Bastian Steudel, Dominik Janzing, and Bernhard Schlkopf. Causal markov condition for submodular information measures. *arXiv preprint arXiv:1002.4020*, 2010.
- [31] Adam B. Barrett, Lionel Barnett, and Anil K. Seth. Multivariate granger causality and generalized variance. *Phys. Rev. E*, 81(4):041907, April 2010.
- [32] Nihat Ay. Information geometry on complexity and stochastic interaction. *MPI MIS PREPRINT 95*, 2001.
- [33] Nihat Ay. Information geometry on complexity and stochastic interaction. *Entropy*, 17(4):2432–2458, April 2015.

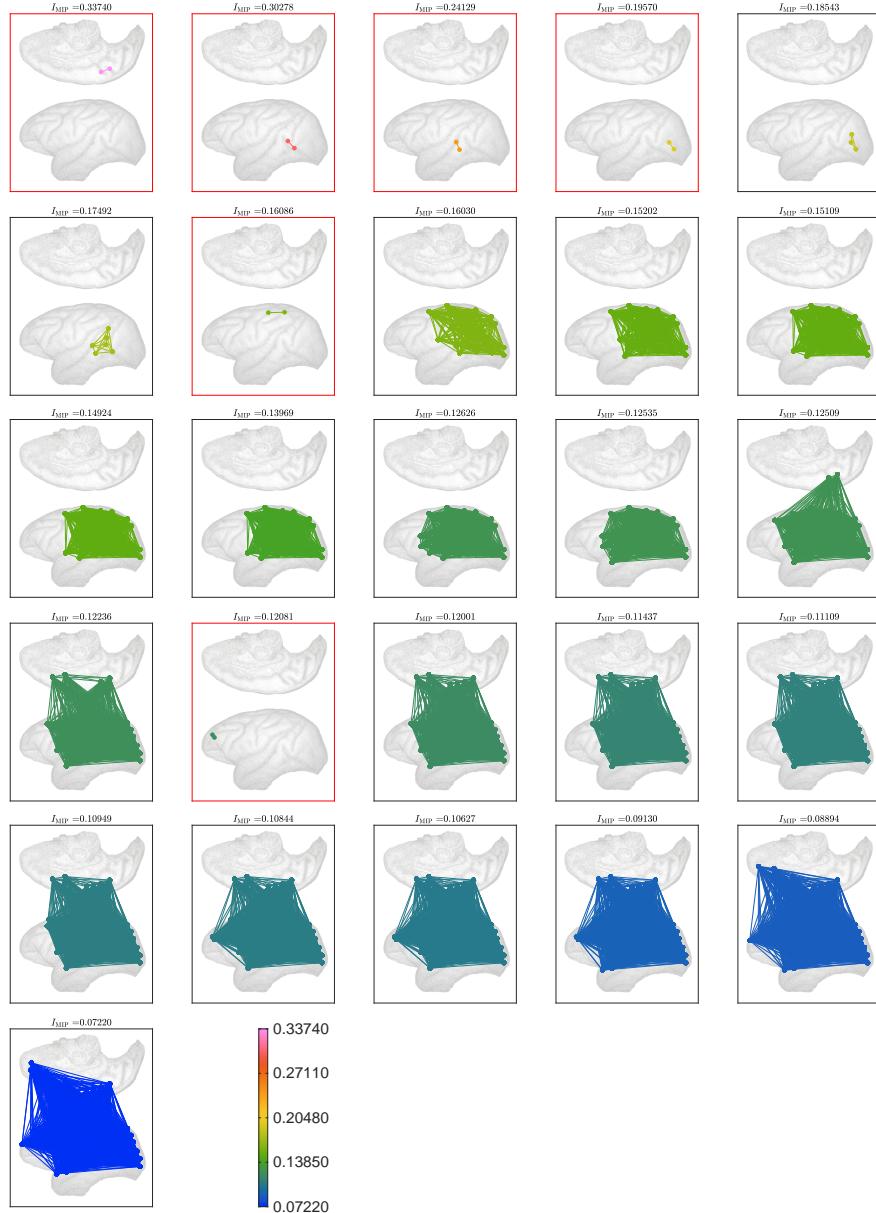


Figure 10: The (main) complexes at the first time window sorted by the amount of mutual information I^{MIP} in descending order. The channels in each complex are connected by edges. The color of the markers and the edges indicates the amount of mutual information I^{MIP} . The panels with red box outline show main complexes and those with black box outline show non-main complexes.

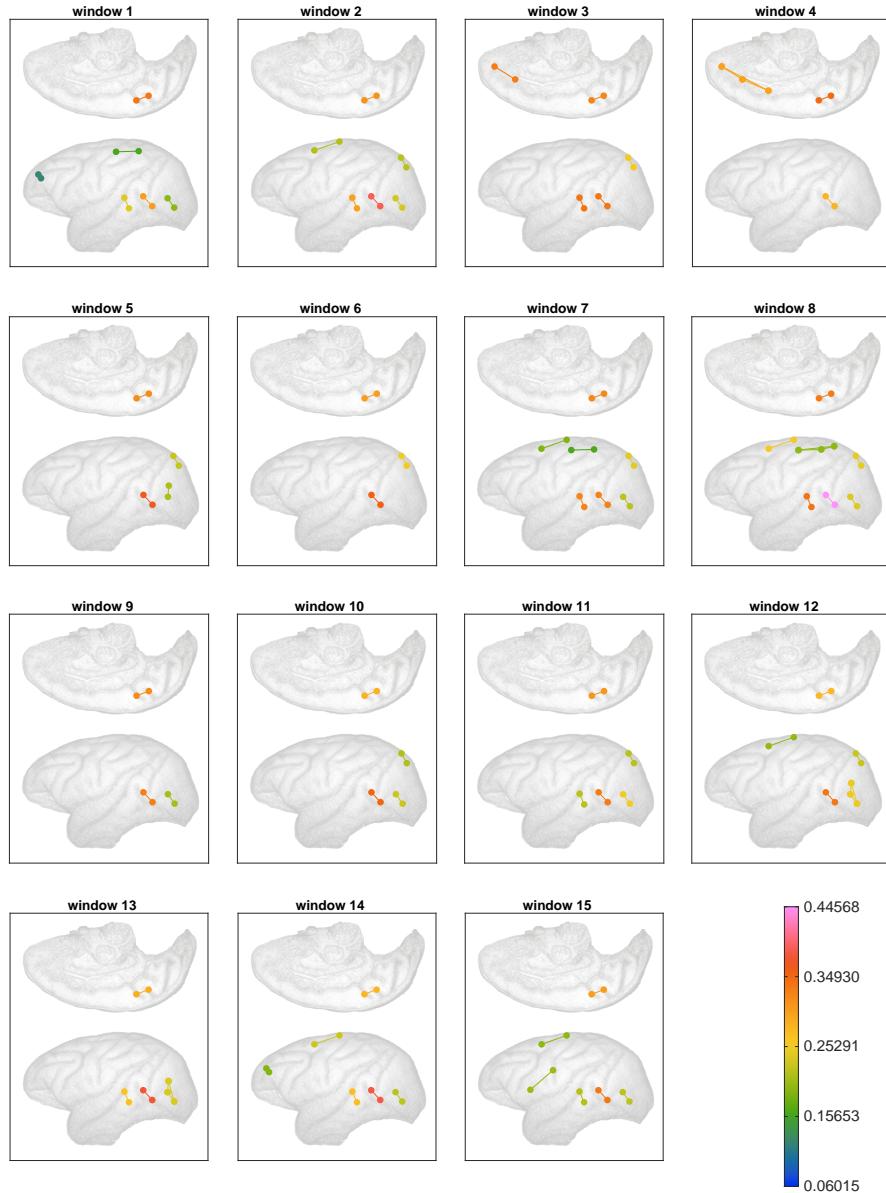


Figure 11: Main complexes at different time windows. The channels in each complex are connected by edges. The color of the markers and the edges indicates the amount of mutual information I^{MIP} .

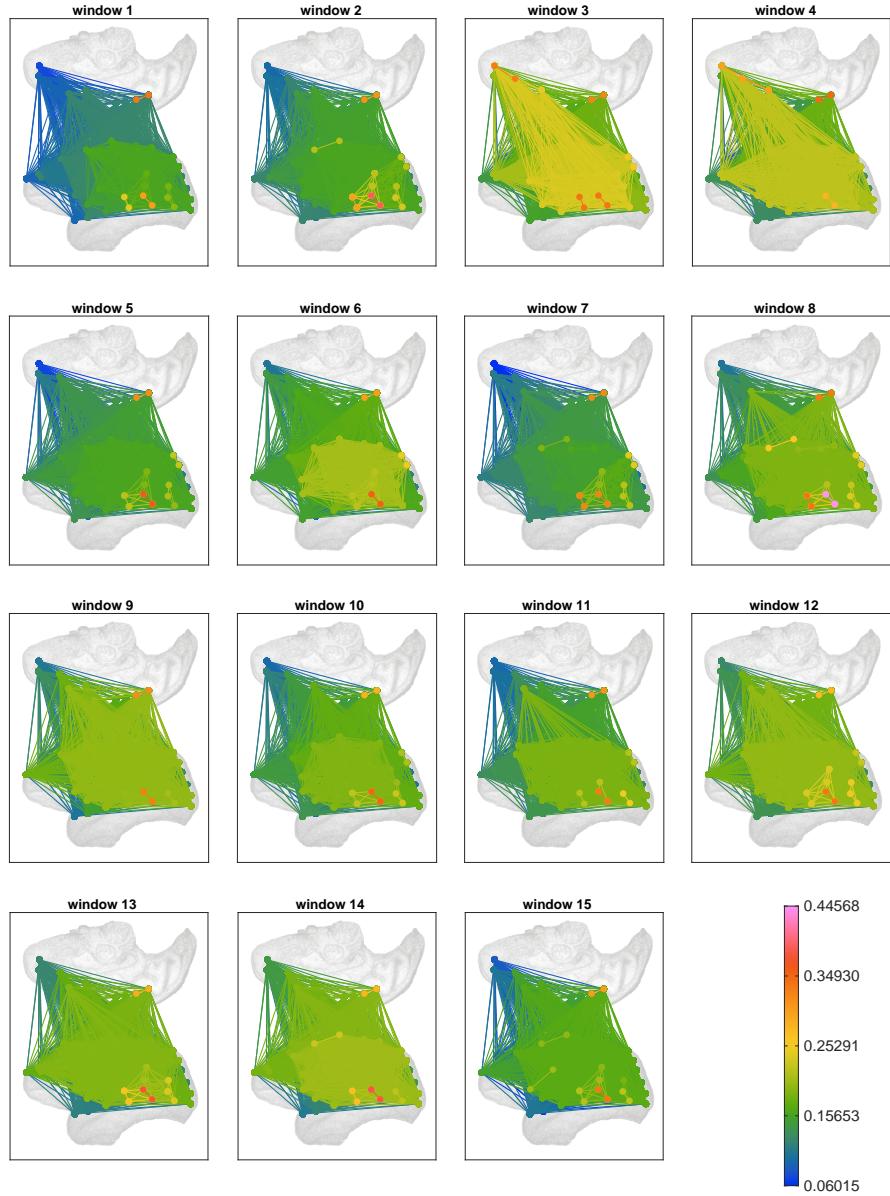


Figure 12: Complexes at all the 15 time windows. The channels in each (main) complex are connected by edges with the same color. The color of the markers and the edges indicates the amount of mutual information I^{MIP} . The main complexes and complexes are superimposed in ascending order of the amount of mutual information I^{MIP} .