

# Chapter 6

## Sequential Synchronization Analysis

Toshio Murase, Marshall Scott Poole, Raquel Asencio, and Joseph McDonald

### 6.1 Introduction

Sequences have long been a central interest in group research.<sup>1</sup> Sequences capture how group processes unfold over time, and characterization of sequences as a whole and their properties offers valuable insights into group decision-making, conflict management, group cohesion, teamwork, and many other group phenomena.

Sequences have been studied on a variety of levels in group research. Some of the best known sequences are the stages of the group life cycle. While Tuckman's (1965) iconic "Forming, Norming, Storming, and Performing" stage sequence is the best known of these, several dozen models of the group life course have been described (Hare, 1976, 2010; LaCoursiere, 1980). Sequential models of specific group activities such as problem solving (Bales & Strodtbeck, 1951), decision making (Fisher, 1970; Poole & Roth, 1989), conflict (Pondy, 1967), and teamwork

---

<sup>1</sup>Preparation of this chapter was supported by National Science Foundation grant #BCS 0941268 and Army Research Institute grant W5J9CQ-12-C-0017. The contents of this chapter represent the opinions of the authors and not of these organizations.

T. Murase (✉)

Department of Psychology, Roosevelt University, Chicago, IL, USA

e-mail: [toshio.murase@gmail.com](mailto:toshio.murase@gmail.com)

M.S. Poole

University of Illinois, Urbana, IL, USA

e-mail: [mspoole@illinois.edu](mailto:mspoole@illinois.edu)

R. Asencio

Krannert School of Management, Purdue University, Lafayette, IN, USA

e-mail: [rasencio@purdue.edu](mailto:rasencio@purdue.edu)

J. McDonald

Human Interfaces, Inc., Austin, TX, USA

e-mail: [joe@humaninterfaces.net](mailto:joe@humaninterfaces.net)

(Ishak & Ballard, 2012; Marks, Mathieu, & Zaccaro, 2001) have also been advanced. Conceptually, these activity sequences can be thought of as embedded within longer group life cycles. Still other scholars have focused on short cycles of group activity that might be repeated multiple times within episodes of group work, such as Tschan's (1995) orientation-action-evaluation cycles, which are posited to be tied to quality of group work.

In studying sequences, researchers can focus on the entire sequence, as did Tuckman (1965), Bales and Strodtbeck (1951), and Poole and Roth (1989). Relevant research questions include: Do all groups follow the proposed sequence?; What factors determine whether a given sequence occurs?; Is following the sequence related to outcomes such as effectiveness and group cohesion? A second option is to focus on subsequences that make up the entire sequence, as Tschan (1995) and Murase et al. (2015) did. In this case relevant questions include: What types of subsequences occur and what is their frequency?; How do they chain together to generate longer sequences and what types of longer sequences occur?; How are they related to outcomes such as group effectiveness or group cohesion? Finally, researchers may identify characteristics of sequences or subsequences, such their frequency, complexity (Poole & Roth, 1989), or conformity to an ideal sequence (Poole & Roth, 1989) or subsequence (Tschan, 1995). Relevant research questions are: How do various sequences compare in terms of the properties?; What factors govern variability in the characteristics?; How do the characteristics relate to outcomes such as group effectiveness or group cohesiveness?

The approaches described in the previous paragraph focus on the sequence as a property of the group as a whole. Another approach is to decompose the sequential data from the group level to the individual level. In this case the sequence of behaviors of each member is analyzed. Just as with group level sequences, individual sequences can be characterized in terms of their overall structure, subsequences, and characteristics, and the same questions posed for the group as a whole can be posed for the sequences of individual members. But decomposition also enables researchers to explore the processes that lead to the emergence of a group or its properties from the interactions among members.

One of the oldest questions in group research is "What makes a group more than just a collection of individuals?" There has been a long debate over whether a group has an entity beyond the behaviors of its individual members (Davis, 1969; Hewes, 1996; Kozlowski, Chao, Grand, Braun, & Kuljanin, 2013; Kozlowski & Klein, 2000). Kozlowski and Klein (2000) argue that higher level group properties emerge through two processes, *composition* of individual attitudes, knowledge, and/or behaviors into aggregates and *compilation*, which depends on nonlinear combination of individual attitudes, knowledge and/or behaviors. McGrath and Kelly (1986) and Ancona and Chong (1996) consider temporal elements of coordination among individual member sequences. *Entrainment* is defined as cases in which the pace, rhythm, and cycles of individual behaviors come into alignment with one another. In this case, the group's activity takes on a character of a holistic unit greater than the individual members. McGrath and Kelly argued that entrainment depends on an external factor such as the group's task or a leader or events in the environment that the group must respond to. However, it also seems possible that

entrainment might also be driven by members' desire to coordinate and engage one another in internal group interaction. The study of synchronization and entrainment of member behavior enables us to investigate the degree to which the group transcends individual member activities.

This chapter will provide an overview of several methods for sequence analysis that address these questions, including whole sequence methods, short cycle methods, and sequential synchronization analysis. Methods for whole sequence and short cycle analysis have been discussed at length elsewhere, so they will be described in general terms; sequential synchronization analysis has not been previously introduced, so the remainder of the chapter will be devoted to an explanation of how it works and can be conducted.

## 6.2 Sequence Analysis

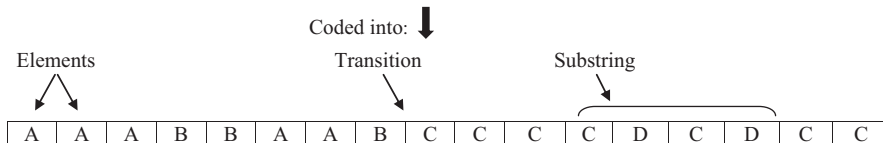
### 6.2.1 *Sequence Data*

Group sequence data can come from a number of sources. It can be directly recorded by observers (e.g., Bales, 1950), or it can be coded from audio or video recordings (e.g., Fisher, 1970; Poole, 1981). Researchers like Axelrod (1976, 2015) used archives of diplomatic notes and negotiations to reconstruct sequences of argument. Data can also be gathered using computerized group or team simulations of, for example, military tasks, emergency patients, or negotiations (e.g., Schiflett, Elliott, Salas, & Coovert, 2004), which capture automatically the choices and actions of each member down to hundredth of a second units. Another data resource for group research is data captured from the internet (e.g., email, social media, text messages) and mobile devices (e.g., geolocation, sociometric badges).

Figure 6.1 presents a general illustration of the type of sequence data that results from the operations described in the previous two paragraphs. The top row shows the basic data units. These units are then coded into meaningful categories (in this case A, B, C, and D), which are the elements of the sequence. As the previous discussion shows, in some cases the coding system defines the units as part of the coding process (e.g., Interaction Process Analysis), while in other cases (e.g., a military simulation) the units are “hard-coded” into the data recording apparatus, while in still others (e.g., server data from a massive multiplayer online game) the units must be retrieved from a more complex data store. Each unit may also be associated with a timestamp, shown in the bottom row of the figure; this timestamp orders the elements and may also be used to determine durations. The timestamp in this figure is based on a “Newtonian” conception of time, in which time can be divided into equal units and proceeds linearly into the future. The top row of the figure portrays a different conception of time, “event time,” in which the occurrence of events marks the units, regardless of how long they were or the intervals between them. In addition to time stamps, this data also indicates the source of or major actor in each unit. Note that a member may engage in several consecutive acts.

Units (U\*)/Members (M\*):

U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	U12	U13	U14	U15	U16	U17
M1	M1	M1	M2	M1	M3	M1	M3	M2	M2	M2	M1	M3	M3	M1	M2	M3



Timestamp:

01	05	11	13	16	22	31	45	51	62	67	73	79	83	88	93	98
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

**Fig. 6.1** Sequences and sequence data

Some properties of sequence data are shown in the second row of Fig. 6.1, transitions from one element to the next. Substrings (or subsequences) are meaningful short-term patterns of acts; they may be defined structurally by repeated sequences of elements or theoretically by specification of meaningful sequences of elements (e.g., plan-act-evaluate). Identification of meaningful units or subsequences sometimes proceeds through a series of hierarchical steps. As the third row of the figure indicates, each series of similar units can be re-coded into a single occurrence or phase of this unit. A phase is a coherent period of group activity of the same type. In this case, the phasic sequence is ABABCDCDC. This can be reduced to a still higher-order pattern, as shown in row four, in which repeating AB substrings are reinterpreted as E phases and CD substrings as F phases. Poole and Roth (1989) used this approach to simplify phase sequences in group decision-making using a procedure formally described in Holmes and Poole (1991).

## 6.2.2 Analyzing Sequences

Many group process studies analyze sequences by “collapsing” them into profiles of the total number of each type of act in the sequence. These profiles are useful because they show general differences between sequences. A sequence with a lot of conflict events is clearly different from one with very few.

Information is lost, however, by synoptic measures of processes such as profiles. Where in a decision process a conflict occurs tells us a lot about the process. A conflict early on may serve to raise issues for the group to discuss and resolve; a conflict at the end may create an impasse that stymies the group. Considering the sequence of activities tells us the “story” of the group process in a way that simple totals cannot.

Rudimentary sequence analysis has often been applied to coded data in the social sciences. Human pattern recognition is powerful and adaptive, making it possible to extract rich information about human interaction and behavior from video sessions (e.g., DeChurch & Marks, 2006; Kozlowski, Chao, Chang, & Fernandez, 2015; Stachowski, Kaplan, & Waller, 2009). Bales and Strodtbeck (1951), for instance, divided their discussions into thirds and constructed graphs of amounts of orientation, evaluation, and control behavior over time to compare sequences of group problem solving sessions. However, if researchers have sequences made up of many units or a large set of sequences, manually identifying critical patterns is a difficult and daunting task. Methods developed in the biological sciences to identify DNA sequences from millions and billions of data points (Koonin & Galperin, 2003) and in computer science, where strings of thousands of digits or lines of code must be compared (Sankoff & Kruskal, 1983) can be brought to bear in this case. To overcome this challenge, these disciplines developed approaches to data mining and large scale analytics designed to find unique patterns of information and to evaluate similarities in structure and function between sequences (Needleman & Wunsch, 1970).

Sequence analysis is particularly aligned with process models that posit that groups develop through a series of distinct stages (Tuckman, 1965) and engage in patterns of phases to make decisions and accomplish their tasks (Bales & Strodtbeck, 1951; Gersick, 1988; Poole & Holmes, 1995; Poole & Roth, 1989; Sambamurthy & Poole, 1992). For example, Marks et al. (2001, see also Ishak & Ballard, 2012) proposed temporally-based team process in which team members engage in two types of phases alternatively to achieve objectives: transition phases—where members engage in planning and strategizing—and action phases—where they engage in activities directly contributing to team performance.

Sequence analysis is also appropriate for models of act-to-act sequences. The assumption commonly shared among these models is that events and behaviors trigger each other to create unique contexts in which one leads to another, which then facilitates the occurrence of more events and behaviors later on (Lehmann-Willenbrock, Meyers, Kauffeld, Neining, & Henschel, 2011). Tschan's (1995) plan-act-evaluate behavioral cycle model of effective team activity is a good example of this approach.

### 6.2.2.1 Whole Sequence Analysis

Poole and his colleagues investigated the phasic sequences groups followed to make decisions (Poole & Holmes, 1995; Poole & Roth, 1989; Sambamurthy & Poole, 1992). Instead of measuring members' perceptions of their decision-making process, Poole and Roth (1989) content-coded 47 decision processes by taking the following steps: (a) identifying major activity (e.g., problem-focused, execution-focused, and solution-focused activities) within each 30-second time segment of a process to create a sequence of the activities; (b) grouping into phases the activities of the same category if they occurred consecutively and also grouping into phases

activities from the different categories if they happened in a row. They used the technique of flexible phase mapping (Holmes & Poole, 1991) to identify various sequences and methods including optimal matching to compare and classify sequences into types. The sequences of activity phases produced by this method provided the fine-detailed picture of when the specific activity phases occurred and in what order. For example, some groups always went through a fixed process of different phases while others moved through different stages and cycled back to the previous stages. The richness of the sequence data helped Poole and Roth uncover that groups did not follow unitary group process but that their processes were much more complex and diverse.

One useful technique in whole sequence analysis is optimal matching (OM), which is designed to compare similarities of pairs of sequences (Abbott & Tsay, 2000; Aisenbrey & Fasang, 2010; Hollister, 2009; Wu, 2000). OM evaluates how similar pairs of sequences are. It assesses the degree of difference (distance) between pairs of sequences using substitution-insertion-deletion transformation operations (INDEL). Suppose one wants to compare two sequences: ABC and ADE. OM calculates the distance between them by using the INDEL operations. First, OM replaces B at the second position of ABC with D; inserts E between D and C of sequence ADC, which turns the sequence into ADEC; and then deletes C at the last position of the sequence. The number of the operations required to convert the first sequence into the second one is 3, which is the distance score between these two sequences. Weights are generally attached to various INDELS based on similarity of elements. For example of A and B both pertain to problem statements and to a solution statement, substituting A for B would make less difference than substituting C for A. So the B-A substitution would be given lower weight (cost) than the C-A substitution. Based on this logic optimal matching algorithms assign weighted differences to each pair of sequences in a set. The number of ways to calculate distance scores between a pair of long sequences increases drastically. Therefore, OM seeks the most optimal ways to calculate distance scores among sequences (Abbott & Tsay, 2000).

The resulting set of distance scores can then be analyzed using multidimensional scaling or clustering techniques to derive sets of sequences with similar structures. For example, Sambamurthy and Poole (1992) derived three different sets of sequences from a sample of 45 conflict management discussions: one in which conflict was suppressed, one in which there were open disagreements that were not resolved, and a third in which there was open discussion and cooperative management of the conflict. The third set had more positive relationships to outcomes than the other two. It is also possible to take a reference sequence—for example, an ideal type sequence—and use optimal matching to determine how similar one or more sequences are to the reference sequence.

There has been much debate over the proper use and benefits and costs of using OM. Readers can refer to Aisenbrey and Fasang (2010) and Herndon and Lewis (2015) for further discussion of these issues.

### 6.2.2.2 Subsequence Analysis

While Poole and colleagues studied entire sequences, Lehmann-Willenbrock et al. (2011) examined whether mood emerges through short-cycles of behavioral patterns in which complaining behavior leads to supporting behavior which leads to complaining behavior. They coded discussions in which 57 company teams discussed solutions to problems in their work activities. Each statement provided by an employee in the conversation was assigned to one of 44 behavioral categories, resulting in a sequence of behaviors for the team. Lehmann-Willenbrock et al. examined how often one behavioral type was followed by another by calculating probability ratings among all possible pairs of behaviors in the  $44 \times 44$  table. Using the probability ratings, they found that team members often engaged in specific cycles of complaining behaviors (e.g., complaining, complaining, and complaining; complaining, supporting, and complaining), and that the cycles of complaining behaviors resulted in unaroused and unpleasant group mood while the cycles of positive behaviors produced pleasant group mood. Methods such as relational event modeling can be used to test hypotheses about short cycle sequences as well (see Chap. 4, this volume).

Murase et al. (2015) took a different approach to obtain sequences of actions from six-person teams participating in a military simulation game. The server recorded in milliseconds various acts which team members performed, producing sequence data consisting of thousands of thousands of acts over time. Murase et al. developed 37 behavioral categories important for the game, each of which contained short sequences of acts that occurred in specific orders. They then wrote scripts to count the number of times subsequences of acts in the log that matched any of the 37 behavioral categories occurred (they employed 30 s windows for sampling purposes). Their sequence data showed which member in the team engaged in what type of behavior in which time segment. This data was subsequently used in an analysis of social entrainment among team members that will be described in the next section.

Poole, Lambert, Murase, Asencio, and McDonald (2017) and Cornwell (2015) summarize these and other sequence analysis techniques, along with theoretical and data related issues. The bibliographies of these two works list a number of references to more detailed descriptions of specific sequence methods. The remainder of this chapter focuses on the method of sequential synchronization analysis, which facilitates identification of emergent processes such as teamwork through the coordination of the behavioral streams of individual members.

## 6.3 Sequential Synchronization Analysis

### 6.3.1 *Individual Sequences into Group Processes*

To conduct sequential synchronization analysis the researcher first decomposes the group sequence into a sequence for each member and then analyzes relationships among individual data sequences to determine team level dynamics.

Two theoretical forms have been advanced to explain how group dynamics emerge at the team level: compositional and compilational models (Chan, 1998; Kozlowski & Klein, 2000; Roberts, Hulin, & Rousseau, 1978). Compositional models argue that a phenomenon at the individual level resembles the same form of the phenomenon at the team level while compilational models argue that the forms of a phenomenon at the individual and team level are different.

Compositional models are based on the logic that each member's behavior can serve as an estimate of the group or team's behavior, because the phenomenon of interest manifests in the same way at the individual and group levels. Averaging the individual estimates thus yields a more reliable measure of the group or team's behavior. For example, in the case of group decision-making, information sharing is such that any information given by a single member can be used by the entire group. So it makes sense to take each members' information sharing (or, in the case of self-report measures, perceptions of group information sharing level) and combine or average them to get an overall measure for the group.

In contrast, compilational models operate under a logic of individual variability that assumes that it is the pattern or variation among members that gives the group process its character (Murase, Doty, Wax, DeChurch, & Contractor, 2012). So, if one member of a team is quarrelsome and difficult, this can disrupt the team's activity no matter what other members do. Or members may specialize, as in a transactive memory system, where one member specializes in remembering past mistakes and serves as devil's advocate, while another specializes in coming up with novel ideas to address the problems raised by the first. Only if the group has individual members who enact these and other key roles, will it make an effective decision. So it is the pattern of members rather than any sort of sum total that characterizes the emergent group, and to capture this emergence, the various types of patterns or at least variance among members must be characterized. Measures for compilation include the standard deviation, minimum and maximum score of the team members, or gini coefficients on various measures such as personality traits, self-efficacy, or member roles (Barrick, Stewart, Neubert, & Mount, 1998; Campion, Medsker, & Higgs, 1993; Stewart, Fulmer, & Barrick, 2005). All of these measures are based on individual characteristics of members or synoptic, summary measures of group interaction, rather than the group process itself. One influential theory that offers a process-oriented, nonsynoptic account of group emergence from individual activities is the theory of social entrainment (Ancona & Chong, 1992; McGrath & Kelly, 1986).

### 6.3.2 *Entrainment*

A great deal of evidence suggests that human behavior—including group and team behavior—is patterned by rhythms and temporal cycles. McGrath and Kelly (1986) summarize evidence that human interaction is characterized by “complex temporal



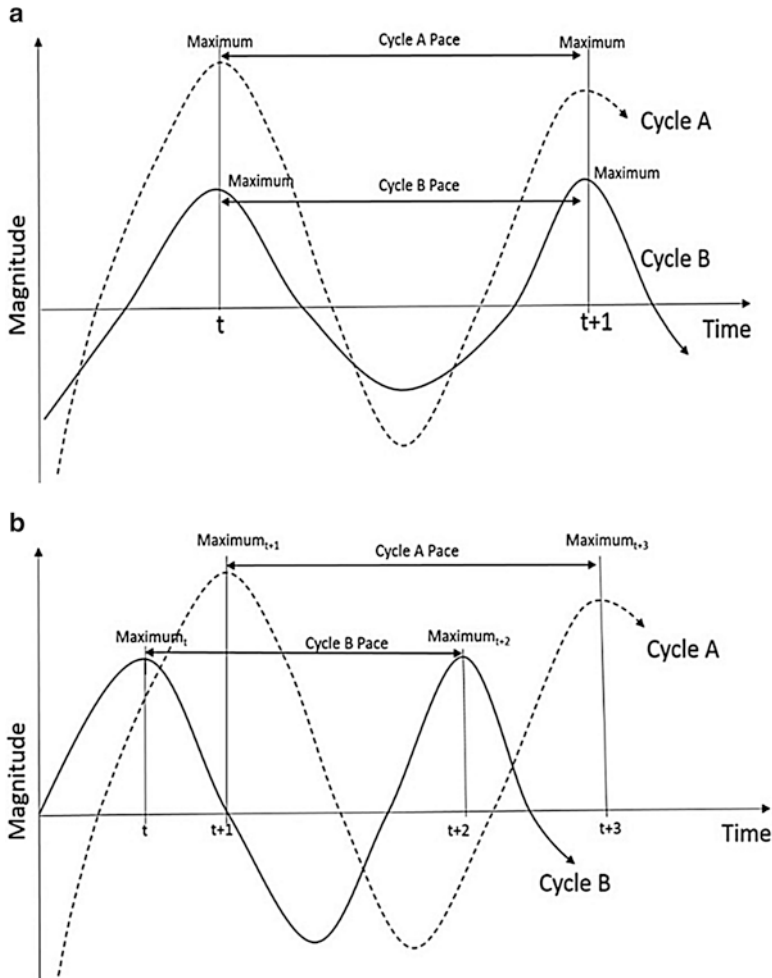
patterning of multiple sets of responses by multiple social actors. These patterns have been expressed by such terms as ‘mutuality,’ ‘reciprocity,’ ‘complementarity,’ ‘dominance,’ ‘similarity,’ ‘simultaneity,’ and ‘alternation’” (p. 7). Cappella (1991) makes a case that at the dyadic level these rhythms and patterns in interaction are biologically determined. Poole and Roth (1989) noted that about 40 % of decision-making groups engaged in repetitive cycles of problem-solution interaction. Tschan (1995) showed that short repetitive cycles of problem-solving were characteristic of effective teams.

McGrath (1990) argues that activities in social systems operate in rhythmic and cyclic forms. Multiple activities, initially operating in different rhythms, eventually get locked into the same rhythmic pattern by influencing one another’s pace or adjusting their activity rhythms to the rhythms of dominant members or external events. For example, project deadlines, unexpected requests from a client, and a competing company’s market entry function as dominant rhythms to which members on teams must adjust their work paces (Ancona & Chong, 1992). Once the activities have settled into a fixed rhythmic pattern, it becomes persistent even when the dominant activity ceases, unless another disrupting event or new dominance pacer emerges to which the activities must start entraining (Harrison, Mohammed, McGrath, Florey, & Vanderstoep, 2003). These studies have demonstrated that synchronization of activities among members is a mechanism underlying the emergence of group-level phenomena.

Most previous research has relied on experimental manipulations and/or measurement of members’ perceptions to capture synchronization. However, it is also possible to identify synchronization from behavioral sequences.

For example, to accomplish a specific objective in a military team exercise, members may increase the level of a relevant behavior (e.g., attacking an enemy unit). Once the objective has been accomplished, the level of the behavior begins to decrease and then eventually cease for a while. This cycle repeats as triggering events (new enemy combatants) occur. In this case, members engage in oscillating activity patterns with one cycle representing a basic behavioral unit, defined as a peak-to-peak period (Cazelles & Stone, 2003). The overlap degree of peak-to-peak periods between pairs of activity cycles essentially determines synchronization degree and type.

If the peaks of multiple members’ oscillating patterns occur at the same time points, or the pace in which the peaks occur is the same (regardless of whether or not the peaks occur at the same time points), those members are said to be entrained to one another (Ancona & Chong, 1992). Ancona and Chong define the former as *synchronic entrainment* and the latter as *tempo entrainment*. If the peaks of pairs or sets of the oscillating patterns at completely at the alternating points, they are defined as *harmonic entrainment*. Figure 6.2a, b demonstrate two types of entrainment where pace is defined as a period from one peak (maximum) at  $t$  time to another peak at  $t+1$  time of a cycle (Cazelles & Stone, 2003). Various statistical measures of the properties of pairs or sets of patterns—discussed below—can be used to determine whether various types of entrainments hold in a group.



**Fig. 6.2** Types of entrainment: (a) synchronic entrainment, (b) temporal entrainment

## 6.4 A Step-by-Step Guide to Sequential Synchronization Analysis

This section is organized to provide step-by-step directions for identifying sequences and then calculating phase-lock scores from a hypothetical time-series data, which are used to capture the degree of synchronization of team behavior. The approach to identifying sequences was used in Murase et al.'s study (2015) which counted frequencies of sequences using the R package TraMineR (Ritschard, Bürgin, & Studer, 2013) and calculated phase-lock scores using the R package synchrony (Gouhier & Guichard, 2014).

We provide a hypothetical study in which four members participate in a military simulation game in which two four-member teams must navigate a course through enemy positions. In order to do perform effectively, their units collect and exchange information important to their mission and also coordinate attacks on enemy units. There are eight events in this scenario: (A) collecting information, (B) member's unit health decrease, (C) attack, (D) enemy health decrease, (E) communication, (F) enemy death, (G) exchanging information, (H) moving with other member, (I) moving alone, and (J) moving close to the enemy. These elements are documented at one-second intervals in the order in which they occurred during the hypothetical mission. The data set is available for download for those who are interested in analyzing it at <http://hdl.handle.net/2142/91573>. The R code for conducting the analysis is referenced below in the example.

The dataset is made up of two teams of four members each. Each row represents a series of events performed by a single member. In this data, the events B, D, and F1 appear across all the members when any of these events occurs to at least one member because they are events that happen to or have impacts on all members of both teams. For example, Member 1 on Team 1 starts engaging the enemy at the 19th position, and the enemy's health decreases at the 21st position. Although this event belongs to Member 1, it is documented across all the members, because the enemy's health decrement is beneficial for any member who encounters this enemy. This irregular, "messy" data structure is typical of sequence data sets, particularly those derived from digital traces. This underscores the value of attending to temporal patterns in data rather than individual acts: focusing on event D alone for Members 2, 3, and 4 might lead us to conclude incorrectly that these members engaged the enemy; but focusing on the sequence CD (attack → enemy health decrease) for Member 1 uncovers the meaning of the event, showing that the result for all was a product of the Member 1's action.

The methods discussed in this section can be applied to simple units like those just defined or to more complex units such as subsequences. In our discussion we will use subsequences as our basic unit of analysis, on the premise discussed in the previous paragraph, that using subsequences or cycles as basic units gives us a more nuanced and accurate description of member behavior.

### ***6.4.1 Step 1: Theoretically Define the Units of Interest***

The first and most important step is to develop a set of theoretically sound units of analysis. When using single acts, the coding system often specifies them. In the case of subsequences construct definition occurs through considering meaningful combinations of acts. Not all subsequences are necessarily meaningful and even when all are, only a few might be of interest given the theory being tested. These serve as basic units of analysis. One challenge lies in the process of putting events in specific orders to create sequence bases because theories in social sciences typically do not specify sets of events and in what exact order those events should unfold. It is the

researcher's responsibility to carefully evaluate what events and behaviors need to be included and in what order they should be placed so that the short sequences can capture the concepts of interest.

For example, in team research, explicit and implicit coordination have been found to influence team performance (Rico, Sánchez-Manzanares, Gil, & Gibson, 2008). Explicit coordination is defined as the process in which members communicate to define responsibilities, make plans and deadlines, and exchange information in order to orchestrate their efforts and activities to achieve common objectives. On the other hand, implicit coordination emphasizes members' ability to predict each other's activities in the process of orchestrating their efforts (Rico et al., 2008). As can be seen, these definitions do not precisely specify what exact behavioral events should be included and in what order. The researcher must choose the behaviors that fit these definitions.

The subsequences of implicit and explicit coordination can include combinations of several different types of behavioral events. For example, using the categories defined above, one subsequence for explicit coordination starts from communication with member A to moving with member A to being close to enemy. On the other hand, a subsequence of implicit coordination starts from moving along to moving with member A to being close to enemy because the definition of implicit coordination emphasizes one's ability to predict other members' behavior (Rico et al., 2008). This definition suggests that communication should not be the essential part of short sequences which capture implicit coordination.

Additionally, the researcher must determine how long the subsequences should be. An appropriate length should be long enough so that below that length a sequence of events should not be complete, but above it a sequence can be broken down into smaller subsequences. For example, it is difficult to determine what type of construct can be captured by a subsequence of two behaviors which starts from moving alone to communication, because depending on what behavioral events come before or after this sequence, the meaning of the sequence changes. If the events, moving with other team and being close to enemy, come after this sequence, the new subsequence with the four events could mean explicit coordination. One member tells another member nearby that he is moving toward the enemy unit, and asks the member to come to his location. Then these two meet and move together toward the enemy unit. If these two behaviors do not come after the original sequence, it can be too short to determine whether it captures explicit coordination or something else.

On the other hand, if a subsequence is too long, it could consist of two or more subsequences, each of which alone could provide sufficient information to capture a theoretical construct. For example, if a sequence is assumed to consist of six actions of moving alone, communicating, moving with another member, being close to enemy, attacking, and enemy health decrease, this sequence can be broken into the first subsequence of four behavioral elements—moving alone, communicating, moving another member, and being close to enemy—and a second subsequence of attacking and enemy health decrease. The first subsequence is explicit coordination, and the second subsequence defines a new construct: engaging enemy. Therefore, the researcher must consider not only the “*what events*” question (what

events need to be included) but also the “*how many events*” question (how many events are necessary to make one complete sequence).

Furthermore, the researcher can create multiple subsequences all of which can belong to the same construct. There is no reason to expect that there should be only one subsequence per construct. For example, psychological scales are comprised of multiple items because having multiple questions is considered necessary to capture different aspects of the same construct (Nunnally & Bernstein, 1994). This perspective can be applied to the sequence-based method. If one subsequence may not be enough to capture the entire construct space, multiple subsequences are necessary to obtain adequate coverage of the construct.

This first step is essential for ensuring legitimacy for this type of method. It is common in computer science to simply mine sequences and use the obtained set. However, if we want to relate our sequence analysis to theory, this “dustbowl empiricist” approach would not be sufficient. For the eight act categories we had above, there would be 56 possible pairs for each individual team member and many more if we consider three and four act sequences. This is simply too many to sort through. Generating the subsequences of interest based upon both theory and empirical findings from the literature provides a solid framework through which the researcher can appropriately interpret the meanings of subsequences uncovered by data mining. Without theoretical guidance, the researcher will be easily overwhelmed by the enormous number of short sequences identified through data mining alone.

Out of hundreds of possible sequences, Murase et al. (2015) defined seven different subsequence types comprised of 37 actual subsequences to represent four key teamwork constructs: implicit coordination, explicit coordination, taskwork, and information gathering. Two subsequence types indicated implicit coordination, two explicit coordination, two taskwork, and one indicated information gathering. In this case they used teamwork theory to guide a multilevel classification scheme that started with 37 meaningful sequences, which were then grouped into seven basic types, which were then mapped onto the four key teamwork constructs.

### ***6.4.2 Step 2: Extract Subsequences from Data***

The next step is to extract subsequences of events from the longer sequence of each participant. The R package TraMineR (Gabadinho, Ritschard, Mueller, & Studer, 2011) can be used to conduct a number of different types of sequence analyses. TraMineR contains numerous R functions with which researchers can create and manipulate data for sequence analysis, mine data to find unique sequences, and visualize results. Researchers who are more familiar with Stata can conduct similar types of sequence analysis using Stata packages such as SAID (Halpin, 2014) and others (e.g., Brzinsky-Fay, Kohler, & Luniak, 2006). The rest of the analytical demonstration will be conducted using TraMineR.

In this case we want to extract subsequences from the data. While we know theoretically which subsequences we are looking for, it is useful to mine the full set of subsequences for additional information. In some cases, additional unanticipated subsequences that correspond to our theoretical constructs may be identified. In other cases one or more subsequences might suggest additional constructs compatible with our theoretical orientation.

To extract subsequences, we use the subsequence function (which is called `seqfsub` in the *Synchrony* package) to mine event sequences in the form of shifts from one type of behavior to another type. One consideration is subsequence length. The length of a sequence could be anywhere from 2 units (i.e.,  $A \rightarrow B$ ) to the entire length of data collected in one's study. A second consideration is how to deal with repeats of the same unit multiple times in a row. When data are documented in every second as they are in a game, the same event can be recorded for a member many times in row; for instance, if the player is moving continuously, then movement will be recorded each second so long as the continuous movement occurs. As a result, the data can contain a long string of the same events with a different element at the end (i.e., `AAAAAAB`), and the repeats are an artifact of the recording. The subsequence function identifies no shift ( $A$ ) and one shift from  $A$  to  $B$  ( $A \rightarrow B$ ) at the end, and ignores the intervening multiple occurrences of the element.

When we employ a subsequence identification technique like `seqfsub` that only identifies shifts from one type of act to another (and ignores successive repeats of the same unit), we recommend that the researcher consider whether to break data into multiple shorter segments to limit the time period over which subsequences can extend. If the original sequence runs over hours, months, or days, techniques like `seqfsub` might identify subsequences which extend over longer stretches of time than humans can realistically act over or attend to. If one's sequence data spans 60 min, for example, mining the entire sequence makes no sense because the subsequence function will pull out many sequences which are not meaningful. For example, the function could identify a shift between two behaviors—communication with member  $A$  during the first 30 s of the session and moving with member  $A$  25 m into the game. Such a shift does not make sense given the nature of teamwork interaction patterns, in which members typically respond relatively immediately to one another. To avoid this issue, the researcher should consider breaking the time into multiple time segments within which shifts between units are considered meaningful. The appropriate length of time segments will vary according to the phenomenon. A reasonable latency period for teamwork is relatively short, while in the case of organizational innovation adoption sequences could extend over days, weeks, or months and still be meaningful.

The second decision point is to determine how many shifts are allowed to be part of short sequences. The subsequence function could completely exhaust the entire list of short sequences, and it could take significant computing resources to complete the identification process if the empirical sequence is very long. For more efficient subsequence identification, the researcher should determine the appropriate number of shifts which are maximally allowed in short sequences. If too many shifts are allowed, they would not be interpretable or can be broken down into

shorter sequences. In our case, we limit the length of sequences to be no more than 3 shifts (i.e.,  $A \rightarrow B$ ,  $B \rightarrow C$ ,  $C \rightarrow D$ ), which is in line with the decisions on this matter made by other researchers (Lehmann-Willenbrock et al., 2011; Murase et al., 2015; Poole & Roth, 1989).

The last decision point is to consider how far apart the behaviors within the same shift or the shifts within the same sequence are allowed to be. Suppose that there is a sequence of As and Bs at 10 positions (AAABAABBAA) and that the researcher is interested in identifying the short sequence  $(A \rightarrow B) - (B \rightarrow A)$ . First, the researcher considers whether the events of the same shift should occur at the positions right next to each other or at the positions somewhat apart from each other. For example, it is important to consider whether  $A_1$  and  $B_4$  (the subscripts indicate the event positions in the sequence) are allowed to define a shift or whether only adjacent acts like  $A_3$  and  $B_4$ , and  $A_6$  and  $B_7$  should be identified as shifts. The same concern must be exercised when the researcher considers which shifts should be included in the same subsequence. Depending on how far apart the behaviors within the same shift and shifts within the same short sequence are allowed to be located, the subsequence function produces different frequencies even for the same short sequence.

To operationalize various choices related to relationships among units in subsequences, there are several different counting operations one can use: one occurrence per object (COBJ), one occurrence per span-window (CWIN), distinct occurrences with possibility of event-timestamp overlap (CDIST\_O), and distinct occurrences with no event-timestamp overlap allowed (CDIST) (Joshi, Karypis, & Kumar, 1999).

COBJ counts a specified sequence only once throughout the entire data even if the sequence appears more than once. This is an appropriate rule to use when once a subsequence occurs its full effect is felt. CWIN uses a moving window within which it evaluates the occurrence of the short sequence. First, the researcher must determine how many units a moving window covers every time it moves. For example, if the moving window is set to cover three units, every time it moves, it assesses whether the sequence occurs in those three units. After the moving window goes through the entire data set, the CWIN function provides the total number of occurrences of the short sequence. This rule is appropriate if every occurrence of the subsequence counts. Finally, CDIST\_O identifies all possible short sequences within the window whose length is specified by the researcher. The CDIST\_O function differs from CDIST in that CDIST counts only one occurrence of the short sequence in a window, whereas CDIST\_O counts all occurrences within the window, even those that overlap. More detailed descriptions and comparisons of the counting operations can be found in Joshi et al. (1999).

### ***6.4.3 Step 3: Revisit Theoretically Defined Subsequences in Light of Sequence Mining Results***

The subsequence functions CWIN and CDIST\_O will identify all possible combinations of subsequences and count their frequencies. In step 1 the researcher makes the decisions that define the types of subsequences that will be identified.

**Table 6.1** Counts of subsequences

Set of events		Base sequence	Short sequence 1	Short sequence 2
A, B	Sequence	$(A \rightarrow B)$	$(A \rightarrow B) - (B \rightarrow A)$	$(A \rightarrow B) - (A)$
	Frequency	7	6	7

Note: The arrow sign indicates a shift from one behavior to another; a hyphen connects two shifts.

No theory allows the researcher to make perfect determinations about all meaningful subsequences that indicate theoretical constructs. Additional promising subsequences may have been identified in the sequence mining process. The next task, then, is to use these results to refine the subsequence indicators that are supposed to capture the target constructs. Only those subsequences which indicate the target constructs or suggest new constructs that fit within the theoretical framework should be retained and all the rest should be discarded. Although this process seems straightforward, it is not.

Table 6.1 presents a scenario with the set of events which any short sequences identified must contain. For example, two other short sequences contain the set of AB events and provide their frequency information. Note that two letters connected by the arrow consist of a shift while the hyphens connect two shifts to create a longer chain. Suppose you have identified A and B as critical events, and the subsequence function has identified two other subsequences  $(A \rightarrow B) - (B \rightarrow A)$  and  $(A \rightarrow B) - (A \rightarrow B)$ . The issue faced in this scenario is that the two latter chains contain the  $A \rightarrow B$  shift as part of their sequence so you wonder how this information can be combined. Because of the same  $A \rightarrow B$  shift in the both short sequences, their frequency counts are not independent of each other but are redundant. As you can see, the base sequence  $(A \rightarrow B)$  occurred seven times. This means that any short sequences containing the base sequence can occur more than seven times. Thus, unless, the specific short sequence  $(A \rightarrow B) - (B \rightarrow A)$  is the target short sequence whose occurrence is 6, the researcher should record 7 for this scenario while discarding the other frequency numbers.

As the length of the original sequence data increases, the number of subsequences one can make exponentially increases and becomes impossible to count manually. Utilizing the data mining approach provides the researcher with the new ability to capture information that the researcher cannot think of without the data mining technique.

#### ***6.4.4 Step 4: Aggregate Frequency Counts of Subsequences for Data Segments***

In step 2 we argued that any long sequence could be broken into shorter segments that reflect realistic latencies in thought and action and also ease computational demands. Once an appropriate set of subsequence indicators have been identified, the next step is to count them in each segment to yield a sequence of counts for each



individual member. Carrying through our example of the categories discussed in Step 1 this would yield values of the number of subsequences devoted to explicit coordination, implicit coordination, taskwork, and information gathering for each segment. The result is four time series, one for each activity, for each member.

### 6.4.5 Step 5: Compute Synchronization Scores

Entrainment can be assessed by calculating the degree and type of synchronization across the individual member time series. The output of values of the algorithm provides a means for calculating the degree to which members remain phase-locked or socially entrained throughout the game. Suppose two members have coordination cycles with the same pace. If they coordinate with each other at the same time throughout the game, the cycle value differences are zero. However, even if their paces are the same, members can engage in coordination at different time points. For example, one member coordinates in every 5 min at the 5th, 10th, and 15th minute, but the other member engages in coordination at the 3rd, 8th, and 13th minute. In this case, the cycle value differences yield a series of non-zero constants. Finally, if members engage in coordination at random time points and change the pace of these cycles, the cycle differences yield a series of random numbers. It is important to note that this third scenario represents members who are not entrained to one another.

Because the phase-lock algorithm produces random numbers for non-entrained members, the phase-lock calculation can determine the degree to which members are entrained by the distribution of the previously calculated cycle differences, with uniformly-distributed values representing low phase-lock (i.e., low entrainment) (Cazelles & Stone, 2003). For every pair of members, cycle difference scores for every time point are calculated to create a distribution. If two members' coordination cycles are in perfect sync, the cycle difference scores are zeros while two members that constantly and randomly change their pace would create a uniform distribution of the difference scores. Therefore, if the distribution of cycle differences has a clear peak, two members are said to be "phase-locked", and if the distribution spreads out and approaches uniformity, phase-lock decreases. We use kurtosis values to represent the degree of "peakedness" of cycle-difference distributions.

Besides the phase-lock technique which is the main synchrony analysis in this chapter, other synchrony analysis techniques which are also available in the synchrony package deserve attention. Community-wide synchrony (Loreau & de Mazancourt, 2008) evaluates the degree to which members' time-series data fluctuate in unison. Kendall's coefficient concordance is a non-parametric statistic which evaluates agreement among members' time-series data (Gouhier, Guichard, & Gonzalez, 2010). Although these statistics can be used to evaluate entrainment, the phase-lock technique is the most appropriate because it captures similarity between peak-to-peak paces of multiple cycles, which we used to define entrainment. When using other techniques, we recommend that researchers carefully consider the definition of entrainment and then select the most appropriate technique.

## 6.5 Example

In this section we analyze the sample dataset mentioned earlier. Table 6.2 summarizes ten basic activity elements team members engaged in the game and describes of which of three coordination sequences the actions should be part. These three sequences are also specified in the Searchcode file at <http://hdl.handle.net/2142/91572>. This file currently allows readers to specify up to six elements that sequences should and should not contain. Elements that the sequences must contain need to be specified in the “action” columns, and TRUES must be specified in the “yesno” column. If there are some elements that should not be part of sequences, they must be specified in the action columns, and FALSEs must be specified at the appropriate positions in the yesno columns. For example, the first row in the Searchcode file contains A and G and two TRUES, meaning that mined sequences must contain A and G. If sequences should not contain, for example, G, the TRUE at the second position should be changed to FALSE. If A should not be contained, the first TRUE should be changed to FALSE.

Two R code scripts for sequence and synchrony analysis are available for download at <http://hdl.handle.net/2142/91573>. The scripts help readers understand how we prepared data for sequential synchrony analysis and conducted the analyses. It is difficult to provide the full description in this chapter for what we did line-by-line given limited space, but we attempt to highlight the main lines important for the analysis and provide explanations. The further explanations for all the script lines are provided directly in the scripts.

We broke the data into 10 20-second time segments as we recommend in the Step 2 section, and identified all sequences within each segment to create time-series data per member. The code to create the time segments is shown in Table 6.3.

**Table 6.2** Coding categories used in the example

Element	Action	Coordination sequence: engaging enemy	Coordination sequence: exchanging information	Coordination sequence: planning
A	Pick up information			✓
B	Health decrease			
C	Attack	✓		
D	Enemy health decrease			
E	Communication		✓	
F1	Enemy death		✓	
G	Exchange information			✓
H	Moving with other team	✓		
I	Moving alone			
J	Move close to enemy	✓		

Note: Check mark indicates of which sequence the element/action is part

**Table 6.3** Dividing data into segments

2	<code>seg&lt;-20</code>
12	<code>for(time_i in c(1:(ncol(teamdat)-3)/seg)){</code>
13	<code>print(paste("MTS",mts_i," Mem",member_i," Seg",time_i,sep=""))</code>
14	<code>lst&lt;-time_i*seg</code>
15	<code>fst&lt;-lst+1-seg</code>
16	<code>subdat&lt;-teammemdat[1,fst:lst]</code>
17	<code>variability&lt;-length(unique(apply(subdat,2,as.character)))</code>

Line 2: Object `seg` indicates the 20-second time window used to divide the data into ten different time segments

Line 12: The `for-in` function specified the number of time the lines that follow should repeat

Lines 14–15: The beginning and ending of each time segment are calculated

Line 16: Object `subdat` is the data segment extracted from a member's entire data set (`teammemdat`)

Next we identified sequences within each segment. First, the CDIST counting operation was used to identify sequences that contained up to three shifts. Once identified, sequences were evaluated for whether they captured team coordination, and their frequency counts were documented if they contained one of the sets of behaviors in the following order: H, J, and C; F1 and E; A and G. These three sets of behaviors indicate different ways in which members engage in team coordination. Sequences containing H, J, and C indicate that members move together to engage enemy. Sequences containing F and E indicate that members plan for the next move after they complete a task (which is removing the enemy threat). Finally, sequences containing A and G indicate that members exchange information as they locate it. Although we could generate more combinations of behaviors, we use only these three sequences in this demonstration. If sequences contained any other behaviors which were not specified in this section, their frequency counts were not documented. Table 6.4 shows the commands given to TraMineR for this operation.

The next step was to examine whether sequences members engaged in within the same time segments were considered as redundant or unique. For example, Member 2 on MTS 1 engaged in three sequences containing H, J, and C in the seventh time segment: (H) – (H→J) – (J→C); (H) – (J→C); and (H→J) – (J→C). If the frequencies of all the three sequences were included, the total count for this segment would be 3. However, if the chain of actions in this segment is evaluated, it is obvious that these three are actually duplicates. The chain is HHHHHHHHJCCCBDCBDC. This member engaged in this type of coordination activity only once in this time segment as indicated in that the member engaged in one series of move activities and one series of attack activities. Therefore, we took only one sequence out of these three and documented its frequency count. Furthermore, we took this approach through the entire data. This is a complex operation that is explained in the code available for download.

Additionally, when members engaged in different types of coordination within the same time segments, we took the sum of their frequencies. For example, Member 2 on MTS 1 engaged in two different types of sequences in the eighth segment: (H→J) – (J→C) and (F1→E), and each sequence occurred only once. The reason

**Table 6.4** Code for CDIST

18	<code>if(variability&gt;1){</code>
19	<code>  datsize&lt;-ncol(subdat)</code>
20	<code>  eve.seq&lt;-seqdef(subdat)</code>
21	<code>  eve.seqe&lt;-seqcreate(eve.seq)</code>
22	<code>  fsubseq&lt;-seqfsub(eve.seqe,minSupport=1,maxK=3,</code>
23	<code>  constraint = seqeconstraint (maxGap = datsize,</code>
24	<code>  windowSize = datsize,countMethod='CDIST'))</code>
25	<code>  evecount&lt;-matrix(seqeapplysub(fsubseq,method="count"),</code>
26	<code>  ncol = 1, nrow = ncol(seqeapplysub(fsubseq,method="count")))</code>

Line18: The if function checks whether the segment contains different action elements

Line20: The seqdef function converts a character string into a vector of events

Line21: The seqcreate function prepares data for sequence analysis from the event sequence object created in Line 20

Line22: The seqfsub function mines the data to produce sequences with maxK=3 meaning the function creates sequences of up to three shifts. CDIST is selected as counting operation for this analysis

Line25: The seqeapplysub function produces data containing frequency counts for the sequences found in Line 22. This line prepares the frequency count data in the matrix format

for this approach being adequate is that on average the members engaged in coordination sequences only once in each segment. Thus, summing frequencies of different types of sequences did not distort team coordination information. However, this approach could produce distorted information if frequencies for one type of sequences were exceedingly larger than those for the other types of sequences, but all the types of sequences were considered equally important. For example, in some data teams typically engage in implicit coordination about 100 times with standard deviation (SD) of 20 while engaging in explicit coordination 10 times with SD of 2. Additionally, we assume the researchers consider these two types of coordination equally important. However, if the frequency counts of these two types are summed across, the aggregate score that is supposed to represent the coordination construct is over-represented by implicit coordination, which is not aligned with how this construct is conceptualized. In that case, researchers could convert frequencies into z-scores first and take sum of them. Fortunately, in the current data, this was not a concern.

Table 6.5 summarizes the frequency counts of sequences that met the aforementioned criteria. Member 1 and 4 on Team 1 did not engage in activities as much as the other members while all the members on Team 2 were active throughout all the time segments.

In the last step, calculation was conducted on the extent to which members' activities over time were phase-locked. Using R package Synchrony (Gouhier & Guichard, 2014), phase-lock scores were calculated for every pair of members within each team, and then kurtosis scores were derived to evaluate the degree of peakness (Table 6.6).

**Table 6.5** Frequency counts of coordination sequences over Time

Team	Member	Time 1	Time 2	Time 3	Time 4	Time 5	Time 6	Time 7	Time 8	Time 9	Time 10
1	1	0	1	0	0	0	1	0	1	0	1
1	2	1	1	1	1	0	0	1	2	1	1
1	3	0	1	0	2	1	0	1	2	0	0
1	4	0	0	1	0	0	0	0	0	0	1
2	1	1	1	0	1	0	2	1	2	1	1
2	2	2	1	1	1	1	2	2	1	2	0
2	3	0	1	0	1	0	2	2	1	0	1
2	4	0	1	1	1	0	2	1	1	1	0

**Table 6.6** Synchrony commands

23	<code>for(mem_i in unique(mtsdat[,"member_i"]))</code>
24	<code>[-1*which(unique(mtsdat[,"member_i"])==members))]</code>
25	<code>{</code>
26	<code>run&lt;-run+1</code>
27	<code>t1&lt;-as.numeric(mtsdat[mtsdat[,"member_i"]==mem_i,-1:-2])</code>
28	<code>t2&lt;-as.numeric(mtsdat[mtsdat[,"member_i"]==mem_ii,-1:-2])</code>
29	<code>div_i&lt;-length(unique(t1))</code>
30	<code>div_ii&lt;-length(unique(t2))</code>
31	<code>if((div_i&gt;=3) (div_ii&gt;=3)){</code>
32	<code>sync.maxs&lt;-phase.sync(t1,t2,mins=TRUE)</code>
33	<code>k&lt;-0</code>
34	<code>s&lt;-NA</code>
35	<code>ave&lt;-NA</code>
36	<code>sds&lt;-NA</code>
37	<code>k&lt;-kurtosis(sync.maxs\$deltaphase\$mod_phase_diff_2pi,na.rm.=TRUE)</code>
38	<code>s&lt;-skewness(sync.maxs\$deltaphase\$mod_phase_diff_2pi,na.rm.=TRUE)</code>
39	<code>ave&lt;-mean(sync.maxs\$deltaphase\$mod_phase_diff_2pi,na.rm.=TRUE)</code>
40	<code>sds&lt;-sd(sync.maxs\$deltaphase\$mod_phase_diff_2pi,na.rm.=TRUE)</code>

Lines 27–28: This line selects a pair of members (mem\_i and mem\_ii) from the team data (mtsdat)

Lines 29–30: This line calculates the numbers of actions the members performed, and then calculates synchrony scores for the pairs whose numbers of actions are equal to or more than 3

Line32: The phase.sync function calculates the synchrony scores between two members' time-series data and automatically creates a distribution of cycle differences of the synchrony scores

Line37: The kurtosis function calculates the degree of peakedness from the distribution of the cycle differences

Line38: The skewness function calculates how skewed the distribution is

**Table 6.7** Kurtosis scores used to evaluate synchronization

Team	Pair	Kurtosis
1	1 and 2	2.73
1	1 and 3	1.84
1	1 and 4	NA
1	2 and 3	1.70
1	2 and 4	1.63
1	3 and 4	NA
2	1 and 2	1.50
2	1 and 3	2.23
2	1 and 4	NA
2	2 and 3	2.72
2	2 and 4	1.50
2	3 and 4	1.50

Table 6.7 summarizes kurtosis scores across all the pairs of members among the two teams, with higher scores indicating the more peaked the cycle difference distribution becomes (Cazelles & Stone, 2003). Values closer or larger than 3 indicate that the distribution has a higher peak than the normal distribution, which indicates that two members are entrained to each other. From this table, the 1–2 pair on Team 1, and the pairs of 1–3 and 2–3 on Team 2 have values closer to 3, indicating that their distributions have a higher peak than the normal distribution (DeCarlo, 1997). Interestingly, the kurtosis value between Member 1 and 2 was higher than that between Member 2 and 3. Although Member 2 and 3 were more active than the other members, Member 1 and 2 had more synchronization on their activities than did the other pair. Another notable point is that the phase-lock calculation produced NAs for the pairs involving Member 4. Member 4 was inactive as evidenced in that this member engaged in coordination only twice. Calculating phase-lock values requires enough fluctuation in data so it may not be useful if one’s data contain many members being inactive throughout.

## 6.6 Discussion

In this chapter, we have provided a step-by-step guide to perform sequence synchrony analysis to investigate the degree to which team members are socially entrained. Specifically, there are two objectives of the chapter. The first objective is not simply to explain how to use specific R functions from the R packages “synchrony” and “TraMineR”, but how to evaluate the theoretical relevance of behavioral elements that should be part of subsequences. The hybrid method of data mining and theory-based thinking provides a solid foundation on which subsequences mined from data acquire substantive meaning and relevance to one’s study. The second objective is to provide a further guidance on how to obtain unique team property “social entrainment” from subsequence data rather than simply calculating

average scores across members. By combining these two methods, sequential synchrony analysis enables researchers to capture compilational forms of emergence.

Group properties emerge in compilational and compositional forms as individuals become cohesive functioning teams (Chan, 1998; Kozlowski & Klein, 2000). Although researchers have argued importance for compilational forms, they have mainly relied on compositional forms or taking average scores to capture team properties. This practice suggests that the current state of science on group and team process is limited because the most preferred analytical approaches are designed to capture only compositional forms. We argue that a reason for the lack of utilizing compilational forms is that there is no theoretical as well as analytical guide to capture them. To spur the use of compilational forms, we have attempted to develop a solution to both of the problems.

Past studies have effectively demonstrated sequence analysis as a powerful technique in preserving contextual meanings of team processes. Sequence analysis can capture compilational forms of emergence especially when researchers directly conduct sequence analysis on data at the team level to obtain patterns of interactions in the team (Lehmann-Willenbrock et al., 2011; Poole & Roth, 1989; Tschan, 1995). However, this technique alone is not sufficient to capture compilational forms when it is conducted on individual-level time-series data because it simply converts the meaning of data from the raw information to subsequences. As a result, the converted data still require aggregation to be elevated to the team level. This is the situation we have illustrated in the example, where researchers must have a specific theoretical and analytical guide to obtain compilational forms.

Social entrainment (McGrath & Kelly, 1986) is a theoretical framework that serves a guide when researchers wonder what team property emerges at the team level in a compilational form. Social entrainment takes on a compilational form when it emerges because each member's behavioral rhythm does not accurately depict how synchronized members' behaviors are. One useful way to observe this phenomenon is to conduct synchronization analysis on members' time-series data.

Like all sequential process analysis, sequence synchronization analysis is a "work in progress." Currently, there are no definitive, canonical techniques for process analysis as there for analysis of experimental designs. While these are emerging, at this point sequence analysis requires improvisation and ingenuity. We encourage readers to build on what we have described as they pursue their own projects.

## References

- Abbott, A., & Tsay, A. (2000). Sequence analysis and optimal matching methods in sociology review and prospect. *Sociological Methods & Research*, 29, 3–33.
- Aisenbrey, S., & Fasang, A. E. (2010). New life for old ideas: The "second wave" of sequence analysis bringing the "course" back into the life course. *Sociological Methods & Research*, 38, 420–462.

- Ancona, D. G., & Chong, C. (1996). Entrainment: Pace, cycle, and rhythm in organizational behavior. In L. L. Cummings, & B. M. Staw (Eds.), *Research in Organizational Behavior* (vol. 18, pp. 251–284). Greenwich, CT: JAI Press.
- Ancona, D. G., & Chong, C. L. (1992, August). Timing is everything: Entrainment and performance in organization theory. In *Academy of Management Proceedings* (Vol. 1992, No. 1, pp. 166–169). Academy of Management.
- Axelrod, R. (1976). *Structure of decision*. Princeton, NJ: Princeton University Press.
- Axelrod, R. (Ed.) (2015). *Structure of decision: The cognitive maps of political elites*. Princeton, NJ: Princeton University Press.
- Bales, R. F. (1950). *Interaction process analysis; a method for the study of small groups*. Oxford, England: Addison-Wesley Press.
- Bales, R. F., & Strodtbeck, F. L. (1951). Phases in group problem-solving. *The Journal of Abnormal and Social Psychology*, *46*, 485–495.
- Barrick, M. R., Stewart, G. L., Neubert, M. J., & Mount, M. K. (1998). Relating member ability and personality to work-team processes and team effectiveness. *Journal of Applied Psychology*, *83*, 377–391.
- Brzinsky-Fay, C., Kohler, U., & Luniak, M. (2006). Sequence analysis with Stata. *Stata Journal*, *6*, 435–460.
- Campion, M. A., Medsker, G. J., & Higgs, A. C. (1993). Relations between work group characteristics and effectiveness: Implications for designing effective work groups. *Personnel Psychology*, *46*, 823–847.
- Cappella, J. N. (1991). The biological origins of automated patterns of human interaction. *Communication Theory*, *1*(1), 4–35.
- Cazelles, B., & Stone, L. (2003). Detection of imperfect population synchrony in an uncertain world. *Journal of Animal Ecology*, *72*, 953–968.
- Chan, D. (1998). Functional relations among constructs in the same content domain at different levels of analysis: A typology of composition models. *Journal of Applied Psychology*, *83*, 234–246.
- Cornwell, B. (2015). *Social sequence analysis: Methods and applications* (vol. 37). Cambridge: Cambridge University Press.
- Davis, J. H. (1969). *Group performance*. Reading, MA: Addison-Wesley.
- DeCarlo, L. T. (1997). On the meaning and use of kurtosis. *Psychological Methods*, *2*, 292–307.
- DeChurch, L. A., & Marks, M. A. (2006). Leadership in multiteam systems. *Journal of Applied Psychology*, *91*, 311–329.
- Fisher, B. A. (1970). Decision emergence: Phases in group decision-making. *Communications Monographs*, *37*, 53–66.
- Gabardinho, A., Ritschard, G., Mueller, N. S., & Studer, M. (2011). Analyzing and visualizing state sequences in R with TraMineR. *Journal of Statistical Software*, *40*(4), 1–37.
- Gersick, C. J. (1988). Time and transition in work teams: Toward a new model of group development. *Academy of Management Journal*, *31*(1), 9–41.
- Gouhier, T. C., Guichard, F., & Gonzalez, A. (2010). Synchrony and stability of food webs in meta-communities. *The American Naturalist*, *175*, 16–34.
- Gouhier, T. C., & Guichard, F. (2014). Synchrony: Quantifying variability in space and time. *Methods in Ecology and Evolution*, *5*, 524–533.
- Halpin, B. (2014). Three narratives of sequence analysis. In *Advances in sequence analysis: Theory, method, applications* (pp. 75–103). Springer International Publishing.
- Hare, A. P. (1976). *Handbook of small group research* (2nd ed., ). New York: Free Press.
- Hare, A. P. (2010). Theories of group development and categories for interaction analysis. *Small Group Research*, *41*, 106–140.
- Harrison, D. A., Mohammed, S., McGrath, J. E., Florey, A. T., & Vanderstoep, S. W. (2003). Time matters in team performance: Effects of member familiarity, entrainment, and task discontinuity on speed and quality. *Personnel Psychology*, *56*, 633–669.



- Herndon, B., & Lewis, K. (2015). Applying sequence methods to the study of team temporal dynamics. *Organizational Psychology Review*, 5, 318–332.
- Hewes, D. E. (1996). Small group communication may not influence decision-making: An amplification of socio-egocentric theory. In R. Y. Hirokawa, & M. S. Poole (Eds.), *Communication and group decision-making* (2nd ed., pp. 179–212). Thousand Oaks, CA: Sage.
- Hollister, M. (2009). Is optimal matching suboptimal? *Sociological Methods & Research*, 38, 235–264.
- Holmes, M. E., & Poole, M. S. (1991). Longitudinal analysis. In B. M. Montgomery, & S. Duck (Eds.), *Studying interpersonal interaction* (pp. 286–302). New York: Guilford Press.
- Ishak, A. W., & Ballard, D. I. (2012). Time to re-group: A typology and nested phase model for action teams. *Small Group Research*, 43, 3–29.
- Joshi, M. V., Karypis, G., & Kumar, V. (1999, May). A universal formulation of sequential patterns. In *Proceedings of the KDD'2001 workshop on Temporal Data Mining*.
- Koonin, E. V., & Galperin, M. Y. (2003). Principles and methods of sequence analysis. In *Sequence—Evolution—Function*, Springer US, pp. 111–192.
- Kozlowski, S. W. J., & Klein, K. J. (2000). A multilevel approach to theory and research in organizations: Contextual, temporal, and emergent processes. In K. J. Klein, & S. W. J. Kozlowski (Eds.), *Multilevel theory, research, and methods in organizations: Foundations, extensions, and new directions* (pp. 3–90). San Francisco, CA: Jossey-Bass.
- Kozlowski, S. W., Chao, G. T., Chang, C. H., & Fernandez, R. (2015). Team dynamics: Using “big data” to advance the science of team effectiveness. In S. Tonidandel, E. King, & J. Cortina (Eds.), *Big data at work: The data science revolution and organizational psychology*. New York, NY: Routledge.
- Kozlowski, S. W., Chao, G. T., Grand, J. A., Braun, M. T., & Kuljanin, G. (2013). Advancing multilevel research design capturing the dynamics of emergence. *Organizational Research Methods*, 16, 581–615.
- Lacoursiere, R. B. (1980). *The life cycle of groups*. New York: Human Sciences Press.
- Lehmann-Willenbrock, N., Meyers, R. A., Kauffeld, S., Neining, A., & Henschel, A. (2011). Verbal interaction sequences and group mood exploring the role of team planning communication. *Small Group Research*, 42, 639–668.
- Loreau, M., & de Mazancourt, C. (2008). Species synchrony and its drivers: Neutral and nonneutral community dynamics in fluctuating environments. *The American Naturalist*, 172, 48–66S.
- Marks, M. A., Mathieu, J. E., & Zaccaro, S. J. (2001). A temporally based framework and taxonomy of team processes. *Academy of Management Review*, 26, 356–376.
- McGrath, J. E. (1990). Time matters in groups. In J. Galegher, R. E. Kraut, & C. Egidio (Eds.), *Intellectual teamwork: Social and technological foundations of cooperative work* (pp. 23–61). New York: Lawrence Erlbaum Associates, Inc..
- McGrath, J. E., & Kelly, J. R. (1986). *Time and human interaction: Toward a social psychology of time*. New York: Guilford Press.
- Murase, T., Asencio, R., McDonald, J., Poole M.S., DeChurch, L. A., & Contractor, N. (2015). The effect of synchronization of group processes on multiteam system effectiveness. Presented at the annual meeting of National Communication Association, Las Vegas, NV.
- Murase, T., Doty, D., Wax, A. M. Y., DeChurch, L. A., & Contractor, N. S. (2012). Teams are changing: Time to “think networks”. *Industrial and Organizational Psychology*, 5, 41–44.
- Needleman, S. B., & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48, 443–453.
- Nunnally, J., & Bernstein, I. H. (1994). *Psychometric therapy*. New York, NY: McGraw-Hill.
- Pondy, L. R. (1967). Organizational conflict: Concepts and models. *Administrative Science Quarterly*, 12, 296–320.
- Poole, M. S. (1981). Decision development in small groups I: A comparison of two models. *Communication Monographs*, 48(1), 1–24.
- Poole, M. S., & Holmes, M. E. (1995). Decision development in computer-assisted group decision making. *Human Communication Research*, 22, 90–127.

- Poole, M. S., & Roth, J. (1989). Decision development in small groups IV a typology of group decision paths. *Human Communication Research, 15*, 323–356.
- Poole, M. S., Lambert, N. J., Murase, T., Asencio, R., & McDonald, J. (2017). Sequential analysis of processes. In H. Tsoukas, & A. Langley (Eds.), *The SAGE handbook of organizational process studies*. Thousand Oaks, CA: Sage.
- Rico, R., Sánchez-Manzanares, M., Gil, F., & Gibson, C. (2008). Team implicit coordination processes: A team knowledge-based approach. *Academy of Management Review, 33*, 163–184.
- Ritschard, G., Bürgin, R., & Studer, M. (2013). Exploratory mining of life event histories. In J. J. McArdle, & G. Ritschard (Eds.), *Contemporary issues in exploratory data mining in the behavioral sciences* (pp. 221–253). New York: Routledge.
- Roberts, K. H., Hulin, C. L., & Rousseau, D. M. (1978). *Developing an interdisciplinary science of organizations*. San Francisco: Jossey-Bass.
- Sambamurthy, V., & Poole, M. S. (1992). The effects of variations in capabilities of GDSS designs on management of cognitive conflict in groups. *Information Systems Research, 3*, 224–251.
- Sankoff, D., & Kruskal, J. B. (Eds.) (1983). *Time warps, string edits, and macromolecules: The theory and practice of sequence comparison*. Reading, MA: Addison-Wesley.
- Schiflett, S. G., Elliott, L. R., Salas, E., & Covert, M. D. (Eds.) (2004). *Scaled worlds: Development, validation, and applications*. Hants: Ashgate.
- Stachowski, A. A., Kaplan, S. A., & Waller, M. J. (2009). The benefits of flexible team interaction during crises. *Journal of Applied Psychology, 94*, 1536–1543.
- Stewart, G. L., Fulmer, I. S., & Barrick, M. R. (2005). An exploration of member roles as a multi-level linking mechanism for individual traits and team outcomes. *Personnel Psychology, 58*, 343–365.
- Tschan, F. (1995). Communication enhances small group performance if it conforms to task requirements: The concept of ideal communication cycles. *Basic and Applied Social Psychology, 17*, 371–393.
- Tuckman, B. W. (1965). Developmental sequence in small groups. *Psychological Bulletin, 63*, 384–399.
- Wu, L. (2000). Some comments on sequence analysis and optimal matching methods in sociology: Review and prospect. *Sociological Methods & Research, 29*, 41–64.