# WHOLODANCE

## Whole-Body Interaction Learning for Dance Education

**Call identifier:** H2020-ICT-2015 - **Grant agreement no**: 688865

**Topic**: ICT-20-2015 - Technologies for better human learning and teaching

# Deliverable 3.5

# Report on data-driven and model-driven analysis methodologies

Due date of delivery: December 31st, 2017

Actual submission date: December 31st, 2017

**Start of the project:** 1st January 2016

**Ending Date**: 31st December 2018

Partner responsible for this deliverable: UNIGE

Version: 0.8

**Dissemination Level:** Public

**Document Classification**

| | |
| --- | --- |
| **Title** | Report on data-driven and model-driven analysis methodologies |
| **Deliverable** | D3.5 |
| **Reporting Period** | M1-M24 |
| **Authors** | Antonio Camurri, Stefano Piana, Paolo Alborno, Ksenia Kolykhalova, Nikolas de Giorgis, Michele Buccoli, Massimiliano Zanoni. |
| **Work Package** | WP3 |
| **Security** | Public |
| **Nature** | Report |
| **Keyword(s)** | Multimodal signals, signal modelling, dance |

**Document History**

| Name | Remark | Version | Date |
| --- | --- | --- | --- |
| Stefano Piana | Table of content, introduction | 0.1 | 25 Nov 2017 |
| Stefano Piana | Model driven methods | 0.2 | 5 Dec 2017 |
| Ksenia Kolykhalova | Graph theory related methods | 0.3 | 5 Dec 2017 |
| Paolo Alborno | Intra and inter Network related method | 0.4 | 5 Dec 2017 |
| Nikolas De Giorgis | Movement Segmentation | 0.5 | 19 Dec 2017 |
| Michele Buccoli | Data-driven methodologies | 0.6 | 19 Dec 2017 |
| Massimiliano Zanoni | Data-driven methodologies | 0.7 | 19 Dec 2017 |
| Anna Rizzo | Final reviewed version | 0.8 | 29 Dec 2017 |

**List of Contributors**

| Name | Affiliation |
| --- | --- |
| Stefano Piana | UNIGE |
| Ksenia Kolykhalova | UNIGE |
| Paolo Alborno | UNIGE |
| Nikolas De Giorgis | UNIGE |
| Michele Buccoli | POLIMI |
| Massimiliano Zanoni | POLIMI |

**List of reviewers**

| Name | Affiliation |
| --- | --- |
| Oshri Even Zohar | Motek |
| Antonella Trezzani | Lynkeus |
| Anna Rizzo | Lynkeus |

## Executive Summary

This deliverable summarizes the description of the development of techniques adopted for multimodal analysis of dance at both individual and group levels, data-driven, and model-driven analysis.

Section 1 introduces the report and lists its objectives whereas Section 2 refers to the methodology employed in the data-driven approach.

Section 3 provides an overview of developed model-driven approaches to extract movement dimensions related to the dance-learning scenario: from low-level model-based movement dimension to more complex intra- and inter- network related methodologies, including a technique to automatically segment dance sequences in meaningful chunks.

## Table of contents

## 1. Introduction

This deliverable serves to summarize the advancements in the development of movement analysis techniques in the context of the WhoLoDancE project. This document focuses on the automatic analysis of movement principles, movement dimensions and qualities identified in D1.6. In particular, we mainly focus on the description of methodologies and algorithms developed to automatically extract such qualities from recorded dance sequences.

## 2. Data Driven Methodologies

The data-driven methodologies exploit statistical models to infer the distribution of the input data and the correlation of such distribution with a desired output. More specifically, given a numerical representation of the input and a set of desired output values for each input sample, a machine learning algorithm is trained in order to compute those parameters that model the relationship between the input domain and the output domain. Once the algorithm has been trained, its parameters can be used to analyse unseen and unlabelled data in order to predict a corresponding output. The block diagram of a generic data-driven pipeline is shown in Figure 1.

With the generic schema in mind, in the specific context of the movement analysis, the requirements to develop data-driven methodologies are:

1. the input data, which includes:
    - the motion capture recordings described in D3.4 and D2.3;
    - a numerical representation of such data, such as the features extracted with model-driven methodologies (see also D3.4 and D3.3);
    - pre-processing techniques for de-noising the data, and post-processing techniques for the automatic selection or aggregation of features;
2. the output data, which is composed of:
    - the semantic model that describes such output;
    - a set of labelled samples, i.e., a set of input data with the corresponding output;
3. the training/prediction algorithm, which is designed in order to properly link the input and the output domains.

The collection of data and the development of algorithms for the three main requirements are discussed in detail in the next subsections.

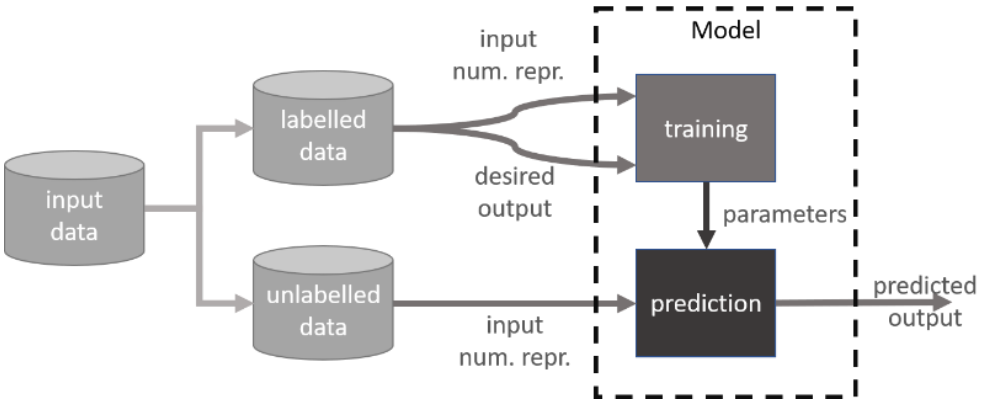



*Figure 1. Pipeline of a generic data-driven training/prediction*

### Output data: the collection of annotations

The movement qualities described in D1.6 represent the semantics of the output. We then started the collection of annotations as described in D5.2 and 5.3. From a preliminary analysis of the collected annotations, however, we discovered a set of issues:

1) **small amount of data**: during the first round, the dance partners annotated 65 performances over a total of 485 annotations. Among these, unfortunately, only 189 concerned the movement qualities. Once divided by the 65 performances, this number was equal to less than 3 annotations per performance;
2) **analysis of consensus**: it is impossible to model a system for the automatic prediction of movement qualities if the semantics of the qualities has a high degree of subjectivity. We aimed at estimating the subjectivity of movement qualities by analysing the degree of consensus of the annotations, i.e., whether the semantics of the movement qualities was shared among partners. Unfortunately, too few segments within the performances were annotated by more than one partner. This makes the computing of the consensus not a feasible task;
3) **low variety of annotations**: the annotation tool, described in D5.3, allowed partners to annotate segments of performances by assigning the degree of a movement quality in a range from 0 to 10. However, most partners only used the central value of 5 to highlight the presence of that movement quality (weakly-labelled data). Without a dual annotation of the absence of the movement quality, it is impossible to train a proper model.

The aforementioned issues raised the need to run the collection of annotations a second time. To develop data-driven methodologies, the amount of collected data, as well as its correctness, are crucial. With weakly-labelled output data, or noisy annotations, the parameters learned to build the model would lead to not meaningful predictions, down to the case that the model prediction is not better than a random prediction.

The collection of new data, however, required to shift the scheduled time for the development of the data-driven techniques ahead of time. Without data, it was impossible to develop techniques, as well as to evaluate the effectiveness of the different numerical representation or machine learning models. What follows is a preliminary review of the state of the art for the data-driven methodologies for motion capture signals.

### Input data: numerical representation and processing techniques

In the first year of the project we acquired multimodal recordings of the dance performances, meaning high-fidelity motion capture, video, audio, background music, and low-fidelity motion capture. Here we focus on the analysis of high-fidelity motion capture recordings.

The motion capture system extracts a skeleton representation of the performers. This kind of representation is extremely helpful because it preserves and exposes the bio-mechanical relationships across the different limbs, and it allows to represent the different joints of the skeleton as a combination of the transformation of the whole chain. It is therefore possible to define a local and a global representation of the signal. The local representation stores the information of each joint (rotation, translation, scaling) with regard to the transformation of the parent node: for example, the movement of the left wrist of the performer is described with regard to the movement of the left elbow. The global representation, instead, stores the information of each joint with regard to the world: for example, the position of each joint, measured in meters, and the orientation of each joint measured in degrees with regard to an absolute system of reference that can be the centre of the room. The selection of the raw data representation is one of the pre-processing stages required for the computation of numerical representation.

It is possible to consider further numerical representations computed from such raw data, which include the kinematics of the movement, the properties of balance, equilibrium, coordination, etc. (see Section Model driven qualities). We will investigate which representations (i.e., which properties) are more effective in the prediction of a certain movement quality. For instance, for the modelling of the movement quality *fluidity*, we assume to consider the frequency analysis of the *acceleration* of the movement, in order to measure the smoothness (low-frequency content) or abruptness (high-frequency content) of the movements.

A motion capture signal can be represented as a time-series of poses, i.e., the evolution over time of the position of the performer. With a framerate of 60 frames per seconds, this means obtaining a huge number of frames for even small movements. The processing of such a fine representation may raise issues related to computational time and lack of generalization. For this reason, we employ a widely used post-processing technique for aggregating data over fixed-width overlapping windows, with the goal of obtaining one input sample for each window. In our work, we use two of the most common aggregation techniques: the average and the median.

The number of dimensions can easily become high, with several dimensions not adding much information to the scope or negatively affect the learning task. As a matter of fact, the well-known *dimensionality curse* makes machine learning techniques to worsen their prediction ability when too many dimensions are considered, due to possible over-fitting. For this reason, we consider the **Principal Component Analysis (PCA)**, which aims to statistically identify the most relevant components of the input data, as a linear combination of the dimensions. The PCA allows us to scale down the dimensionality of the input data to a few tens of dimensions while keeping more than the 90% of the information.

Finally, dimensions commonly range into different intervals. For instance, angles range between -180° and 180°, while the height position spans about 3 meters (from floor to jumps). The process of making the ranges uniform is called **normalization**. We use two well-known normalization techniques: the min-

max normalization (the range is scaled so to range between 0 and 1 or between -1 and 1) and the z-score normalization (the dimensions are scaled so to have 0 mean and 1 standard deviation).

### Training of machine learning techniques

With regard to the machine learning techniques, there are several techniques that can be employed, which are usually divided into two main groups: *classifiers* and *regressors*. The former predicts an output value into a discrete set of categories; the latter assume that the output spans a continuous range of values. The movement qualities are annotated into 11 values (0 to 10): such values should be considered as a quantized version of the continuous range from 0 to 10. In this section, we provide a brief overview of the regressors we aim to use once the collection of annotations will be completed.

Firstly, we can model a regressor as a linear combination of the input components. The **linear regression** model aims at learning the manifold that minimizes the squared distance between the manifold at a given point (i.e., the predicted value) and the correspondent expected value. The linear approach is simple but often effective to model the relationship between input data and desired output. However, it may also lead to overfitting, i.e., to learn a model that fits too well the data used for training and it is not general for the real-case scenario.

The **Ridge Regressor** addresses the over-fitting issue by setting a constraint on the sums of the parameters of the linear combination (*weight decay*). The constraint forces the model to share the predictive power of the weights of the linear combination, hence to focus on the more informative components. This helps the model to improve the regressor performance in the real-world scenario.

The **Support Vector Regressor (SVR)**, instead, aims to learn a manifold by using *support vectors* as the vectors that connect the samples with the higher error. To minimize the distance between the manifold and the support vectors means to minimize the error, therefore, to achieve higher prediction performance. A graphic representation of the SVR is shown in Figure 2.

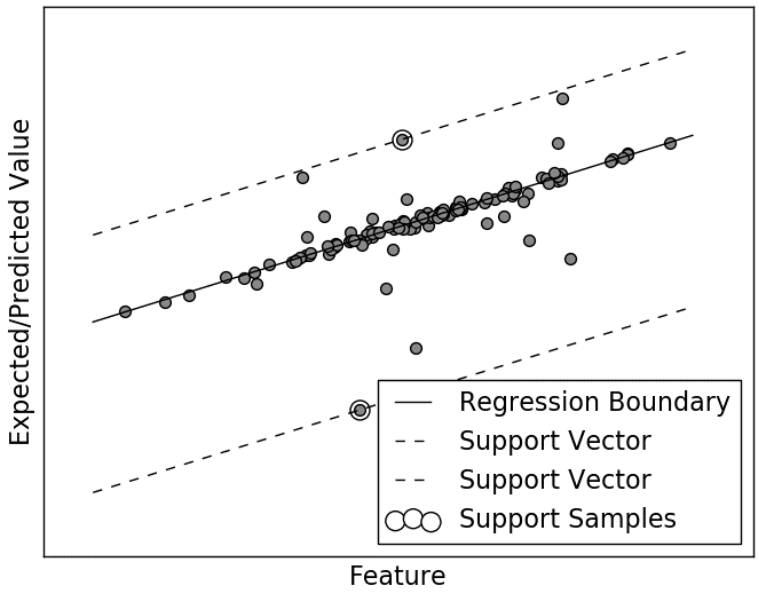



*Figure 2. A graphical representation of the SVR*

The linear and ridge regression models are based on the assumption that the output value can be predicted as a (possibly constrained) linear combination of the components. In order to overcome the limitations of the linearity assumption, we can make use of a *kernel expansion*. By applying a non-linear *kernel*, i.e., a non-linear transformation of the input data, we can apply linear operation in the transformed space, which correspond to applying non-linear operation in the original space.

As an example, we can expand the dimensions of the input with a polynomial kernel (i.e., to compute a polynomial of a certain degree with the factors being the original dimensions). Performing a linear regression on polynomial-expanded data is referred to as **polynomial regression**.

We intend to test these and further techniques as soon as the second round of annotation of movement qualities will be completed.

#### Deep learning techniques

The aforementioned machine learning techniques aims to learn the relationship between a set of features extracted from the input data and the desired output represented by the annotations. For some movement qualities, however, it is hard to understand which are the relevant properties, or how to capture them. To overcome this issue, we can rely on deep learning techniques, that aim at automatically extract a salient representation from raw data to predict a desired output.

On the one hand, deep learning networks have received a great deal of attention due to the performances they are able to achieve in the prediction. They can be seen as a way to collapse the step of feature design and the machine learning technique in a unique model, by using a layered representation and processing from raw data to the desired output. The first layer extracts low-level salient information by looking for patterns in the input data; the second layer examines the output of the first level, looking for further patterns within it, extracting a set of higher-level properties that are, in fact, patterns of patterns. By iterating this process - i.e., adding more and more layers - the network finally trains the highest-level layer to learn the desired output. This kind of process corresponds to the act of decomposing the problem of movement quality prediction in a number of sub-problems that estimate the properties that compose the movement qualities.

On the other hand, deep learning techniques require a huge amount of labelled data to train the network parameters for each layer of the network. We intend to train deep learning networks over the collected data, but the predictive performance of the network will ultimately depend on the amount of annotations we will gather.

## 3. Model-Driven Methodologies

This section describes algorithms and methodologies to extract various model-based movement qualities, first some low- and mid- level qualities are presented (e.g., balance, coordination) then in subsequent sections some higher-level qualities are defined (e.g., analysis of the origin of movement).

## Model Driven Movement Qualities

### Kinematics

Kinematics includes physical qualities such as velocities, kinetic energy, accelerations, that are computed on single joints of the body.

### Balance/equilibrium

According to the official guidelines, any loss of equilibrium is a severe error that influences the evaluation of the performance. To detect equilibrium loss, we measure the projection of the barycentre of the body on the rectangular area defined by the performer's feet. The algorithm first detects, for each frame, whether both feet are on the ground by checking the values of the vertical component of feet joints l (Xl, Yl, Zl), R (Xr, Yr, Zr). For each frame, it then defines a rectangle Z with four corners: (Xl; Zl), (Xl; Zr), (Xr; Zr), and (Xr; Zl), and measures the distance between the barycentre B=(Xb; Yb; Zb) projected on the 2D plane defined by the feet positions and the centre of the Z rectangle. The smaller the distance, the better the equilibrium:

$$F_1 = \sqrt{\left(\frac{x_r + x_l}{2} - x_b\right)^2 + \left(\frac{z_r + z_l}{2} - z_b\right)^2}$$

This algorithm assumes that the dancer is standing on her feet, but in some dance genres (i.e., in contemporary dance) the dancer often touches the ground with other body parts. To address these different situations, we are developing an extended version of the algorithm where the area of the rectangle used in the computation is based on the detection of which body part is touching the ground.

### Coordination between limbs

Intra-personal coordination, i.e., coordination of relevant pairs of joints such as the two wrists, or wrist and knee is important for assessing movement quality. We measure coordination of pairs of joints using the Event Synchronization (ES) algorithm [Quian Quiroga et al. 2002]. This technique compares the timing of occurrence of discrete events in two time-series. In our case, events are defined as local maxima of energy. Essentially, the output of ES is a Synchronization index (SI), computed on the whole data segment, representing the fraction of event pairs matching in time over the 2 time-series. SI ranges in [0; 1]: the larger SI, the more coordinated the movements. We compute SI for the following relevant joints: wrists, feet, shoulders. For example, synchronization between right wrist (RW) and left wrist (LW) joints is computed as follows: the coordinates of the RW and of the LW markers of each participant are taken as input data set; after computing the time-series of velocities for both markers, peaks of velocities, i.e., local maxima are retrieved; two new time-series of the same length as the input ones are created, in which local maxima are coded with 1, while the all other values are set to 0; Event Synchronization is finally applied to these time-series to compute coordination.

## Automatic Movement Segmentation based on Space-Scale and Persistence Approach

Segmenting movement is the process of starting from a whole recording of a motion capture system (which can vary in length), which might contain movements of arbitrary complexity/length/nature, and obtaining a set of basic movements which, combined, span the whole original recording (i.e., every instant in the original recording belongs to one and only one segment in the output).

The very definition of what constitutes a segment depends on the semantic problem addressed and on particular features used; in the literature it has been widely studied for various kind of signals, such as audio, images and video, and recently (approximately since the 2000s) the interest has also been put into the segmentation of motion capture.

The approach followed by UNIGE tries to be as generic and flexible as possible, making no assumptions on the particular movements present in the recording: a flamenco dance will have very different basic movements with respect to a Greek dance or a contemporary one, and our method aims at being able to extract them with no assumption.

Also, our algorithms are able to extract different segmentations from the same recordings: as stated before, the specific segmentation is dependent on what definition of segment is given; in our case, we can see different granularity of segmentation, both in the spatial domain (i.e., are we considering each joint as *relevant* for the segmentation or just the whole body considered as a unique object?) and in the temporal one (i.e., when considering the legs, is a step a segment? Or should we consider a segment just as a series of steps starting and ending in a still position?). Our techniques associate each segment with a quantitative measure that tells us the *scale* of the segment, making it possible to have different granularities just by thresholding differently on this measure.

**Feature selection**

Feature selection is the first crucial step to segmentation; the team at UNIGE decided to use only kinetic features, as they are both good in discriminating relevant movements and generic enough to make the system not dependant on assumptions. According to literature, a good kinetic feature to identify relevant instants in the motion is *acceleration*.

Starting from the recordings and the whole set of markers placed on the dancer's body and tracked, UNIGE developed a hierarchical clustering of them into sets with different granularity (e.g., starting from a single cluster containing the whole body to a fine clustering where, for example, each hand is a single cluster); then, the barycentre is extracted from each cluster, and thereafter used to represent the whole cluster.

Since measurement noise can be present in motion capture systems (e.g., as gaps in the tracking of the markers) as well as in the motion itself, we employ some filtering techniques to obtain a cleaned-up version of acceleration, computed as the derivative of a moving average of the velocity (i.e., a window of appropriate size slides through the velocity and computes the average each time, which is then derived to obtain the *acceleration*).

We study this signal reducing our problem to the analysis of its critical points (i.e., maxima and minima); this method is well rooted in math and computer science literature (e.g. Morse theory, Scale-space) and it has been shown that studying them gives us most of the information carried by the original signal: furthermore, it is intuitive (and empirically verified) that maxima and minima of acceleration (the latter being what is common referred to as a maximum of *deceleration*) correspond to relevant instants in the motion, i.e., the abrupt change of speed when a movement starts and ends.

**Multi-space analysis**

In order to classify and give quantitative measures to the critical points of the input signal, we employ techniques coming from the scale-space analysis, a well-known field developed in the image processing area and then exported to the analysis of various kind of signals.

The analysis of a signal can be carried out at different level of detail: in the case of motion capture data, there are two different domains in which we extract features at different levels.

**Space**: we developed a hierarchical clustering of the tracked markers: the whole set is at each level subdivided in different subsets, as follows:

- a single set containing the whole body; this is the coarsest level of detail, whose feature capture the movements done by the body as a whole;
- a set for the upper body (above the hips) and one for the lower part (hips and below); this subdivision can distinguish between movements done with the legs and those done with the rest of the body.
- a subdivision in the following sets: Head, Left Arm, Right Arm, Torso, Left Leg, Right Leg;
- the finest subdivision is analogue to the previous one, but limbs are further subdivided separating from them hands and feet.

**Time**: even when fixing the granularity of the clustering process (let us say, for example, that we analyse the right arm), we have different ways to segment a recording:

- coarse: a segment is considered such as a whole, from a still position to another one;
- medium: a *coarse* segment can be further divided into smaller movements composing it;
- fine: each significant change in velocity constitutes the end of a segment and the beginning a new one.

To achieve this last one different granularity, we make use of the *scale-space* and the *persistence* analysis.

#### Scale-space

We take the input signal and generate a set of derived signals, every of which is obtained from the original by applying a Gaussian filter of increasing size; the practical effect is that each layer is a smoothed version of the previous one, with the number of critical points decreasing at each level.

We analyse the disappearing of critical points of the *acceleration*, and we assign to each one of them a value that represent how long it survives the smoothing process (i.e., its *life* in the scale-space filtering). Normalizing this value in the [0,1] range and thresholding low values (e.g. <0.3) we get rid of noisy points and identify only instants that are relevant in the motion (as shown in a work by Alborno et al., MOCO 2017).

#### Persistence

Persistence Analysis can be seen as another multiscale approach in which, instead of filtering in the frequency domain (i.e., filtering the input signal), filters in the *amplitude* domain. It works by progressively removing pairs of critical points (one maximum and one minimum at a time), based on the difference in value between them and the neighbouring critical points.

In this case we have an output that is analogue, conceptually, to the one of the *scale-space*, where at each pair of critical points is associated a value of importance (normalized between 0 and 1). Note that, while both analysis are related to the concepts of *scale* and *importance*, they measure different aspects of those, as they work on different domains.

#### Combining the multi-scale approaches

Since both techniques give us signal that are non-zero only in the presence of critical points of the input signal, we can combine them linearly, multiplying one by *alpha* (between 0 and 1) and the other by 1-*alpha* before summing them. For the first experiment we worked with a direct sum between the two (i.e., *alpha*=0.5).

#### Extracting the segmentation

From the output signal described in the previous section, we then extract the desired segmentation with a thresholding operation applied to it. The threshold parameter gives us control on the *granularity* of the segmentation (in the *temporal* domain, as discussed previously): each instant that is over the selected threshold is considered a point that divides two segments; this gives us a full segmentation (i.e., every frame is part of a segment) with no overlap. Lowering the threshold creates more divisions, thus computing a finer segmentation. Note that we define segments as sequence of frames between two relevant instants, and not with a semantic definition on the segment itself; this would make the creation of a ground truth easier, as the process should rely only on the identification of single frames instead that on the analysis of whole periods of time.

#### Preliminary Results

To the best of the knowledge of the UNIGE team, there are currently no adequate available datasets to test the results obtained by our algorithm, so we ran an empirical validation done visually by non-expert in the dancing field; given the fact that preliminary results seems promising, we decided, together with the team at PoliMi, to set up an annotation tool to be submitted to the dance partners together with a set of recordings, in order to obtain a qualified ground truth to test our algorithm.

### Graph Theory Approach for the Evaluation of the Origin of Movement

The main purpose of this work is the development of a novel approach and the associated computational method for the analysis of expressive full-body movement qualities, combining tools and methods from graph theory and cooperative game theory.

We apply the method of graph-restricted game approach in order to investigate the origin and target of the movement, as well as how the movement propagates through time. We evaluate the propagation of movement by computing mathematical properties based of the proposed approach and investigate their relevance to the joints involved in the movement.

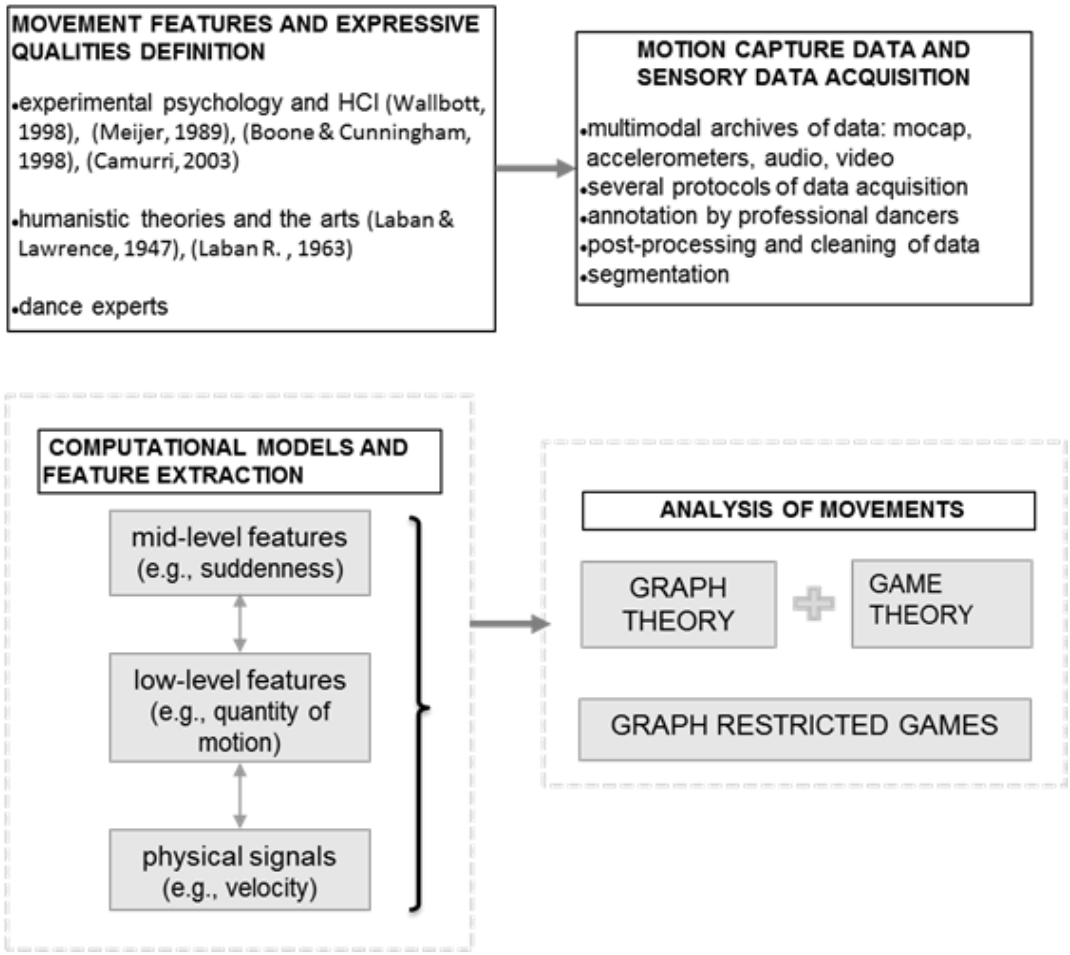



*Figure 3. Framework of graph-restricted game approach application*

**Proposed method description**

Our method is based on the idea that "important" joints during a specific movement are those that separate parts of the body characterized by different motion behaviours. For instance, if an arm is moving and the corresponding shoulder and the other parts of the body are at rest, then that shoulder may be considered as an "important" joint because, in a certain sense, though being at rest, it "controls" the motion of the arm. In summary, our method tends to consider as the most important vertex one that connects different clusters (here identified by spectral clustering), where each cluster corresponds to connected joints with similar movement qualities. We expect the method to be relevant for the analysis of movement qualities, because, differently from other approaches: 1) its game-theoretical component makes it possible to identify the most important joint; 2) its graph-theoretical component allows one to take into account the skeletal structure of the body in the analysis, and also the possible presence of "bridge" edges.

To implement the ideas above, we first define a graph G representing the skeletal structure of the body, having its joints as vertices. The edges belonging to this skeletal structure are called physical edges. For each frame of a recording session, the similarity in the current values of a specific motion-related feature is used to assign positive weights to these physical edges. In more detail, the edge weight is inversely proportional to the sum of a small positive constant (to prevent division by 0) and the absolute value of the difference of the feature values associated with the two vertices joined by the edge. Additional nonphysical edges (or bridge edges) are inserted between vertices not directly connected by physical

edges. Positive weights are assigned to these nonphysical edges, too, which are proportional to the current similarity in feature values of the associated vertices, with a constant of proportionality chosen to be much smaller than for the physical edges (5 times smaller, in the present implementation of the method). Indeed, compared with physical edges, nonphysical edges should have a large weight only in case of a very large similarity of feature values.

Then, we cluster the resulting weighted graph applying spectral clustering to its weighted edge set. In such a way, vertices that belong to the same cluster are expected to have similar feature values, whereas edges between different clusters should be associated with vertices having significantly different feature values.

At this point, for each frame we construct an auxiliary graph, whose edges form the subset of the physical edges of the original graph that connect joints belonging to different clusters (i.e., in this phase, bridge edges are not considered any more). Then, we attribute weights to these physical edges, which are proportional to the dissimilarity of the feature values of the associated joints (in the specific case, they are equal to the absolute value of the difference of the feature values associated with the two joints).

As a further step, we construct a cooperative game on the auxiliary graph. Its characteristic function is defined as follows: for each coalition V', the value $c(V')$ is defined as the sum of the weights (in the auxiliary graph) of all the physical edges contained in the subgraph induced by $V'$.

For the above-defined game we compute the Shapley value and we use it to rank the joints. The "most important" joint in each specific frame is defined as the one with the largest rank, if there is only one joint with that property. In case of more than one joint with the largest rank, the one with the smallest index (according to an a-priori given labelling of the joints) is considered (a random selection is not used, to avoid introducing noise in the model).

Interestingly, it can be shown that this approach is equivalent to ranking the joints according to their weighted degree centrality on the auxiliary graph.

Finally, a filtering step is applied to the computed Shapley values, in such a way to keep only the vertices that were automatically evaluated to be the most important ones for some number of consecutive frames (5, in the current implementation of the method).

**Experiments**

We tested our method using a data set extracted from a larger data set of Motion Capture data of subjects performing expressive movements, recorded in the framework of our EU project. The main topics of the recordings were fluidity, weight, external and internal propagation of movement, balance, equilibrium, coordination, and suddenness. In order to create the data set, first we started acquiring data on volunteer dancers, with the aim of defining and building a reliable and effective Motion Capture setup. Then, we proceeded capturing data on professional dancers. In summary, the multimodal data set of the project consists of 127 trials (recordings), acquired with the aim of investigating movement, defining movement features, and developing techniques for their computation.

Two professional dancers took part in the recordings, using a Motion Capture system with 2 video cameras. The dancers were equipped with 64 infra-red reflective markers, 5 accelerometers, and a microphone. After the recording sessions, the data were post-processed and cleaned, and the origin, path, and destination of the movement were annotated by the experts.

In our initial implementation of the proposed method, the following steps were performed:

1. The initial motion capture data set - x; y; z positions of the 64 reflective markers for each frame of the recordings, see Fig. 4 for a sample frame) was transformed into a reduced data set associated with a smaller number (20) of joints: more precisely, head, hip centre, left ankle, left elbow, left foot, left hand, left hip, left knee, left shoulder, left wrist, right ankle, right elbow, right foot, right hand, right hip, right knee, right shoulder, right wrist, shoulder centre, spine. Compared with the initial 64-joints model, using this 20-joints skeletal structure allowed a faster implementation, useful for testing purposes.

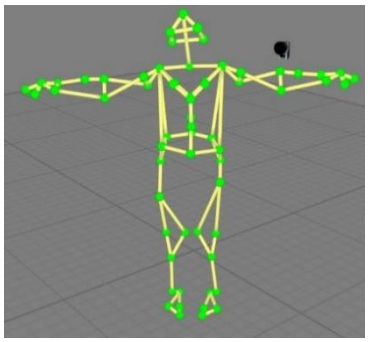


*Figure 4. Example of motion capture data*

2. Then, for each frame, starting from the position of each one among these 20 joints, we computed its speed as the motion-related feature needed by the method. This choice was motivated by simplicity reasons as well as the ease of visualization of such a feature.
3. For each frame, we computed the Shapley values of all these 20 joints, using the method detailed in Section 4. Then, we extracted the "most important" joint, according to the proposed method. In order to conduct our preliminary tests, the whole method was implemented in MATLAB. Due to the reduced number of joints, we fixed 4 as the maximum number of clusters to be detected by spectral clustering (such a small number of clusters was chosen due to the small number of vertices of the skeletal structure).

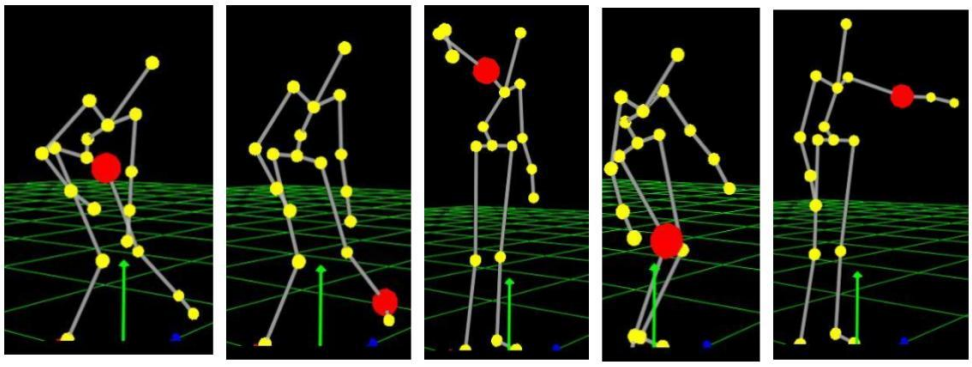


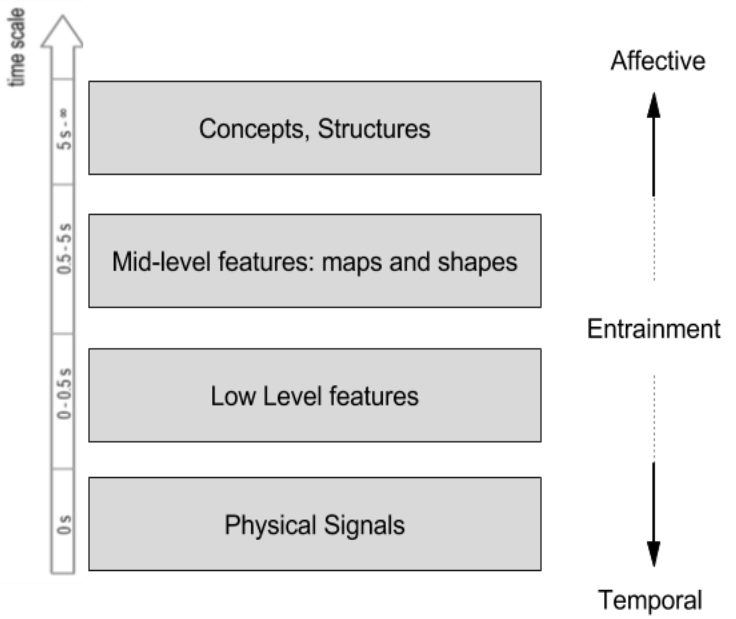
*Figure 5. Sequence showing the origin of movement moving across joints*

4. To ease the evaluation of the results, we visualized them, highlighting in red, frame-by-frame, the most important joint extracted by the method inside the 20-markers skeletal graph (see Fig. 2 for an example of visualization of the results). The EyesWeb XMI1 open platform was used to implement this visualization step.

The preliminary results obtained so far (see Fig. 5 for a sample) showed that, on the data set used for the evaluation, the method was able to extract meaningful joints, i.e., it identified, as important joints, the ones already identified in preliminary annotations. Additional and more quantitative validations of the method will be considered in future works.

### Algorithms for the Evaluation of the Entrainment and Intra- and Inter-Personal Synchronization

Referring to the conceptual multi-level framework explained in (Camurri, 2016), synchronization and entrainment are considered *analyses primitives*, which can be applied at various levels of the framework. Synchronization (not to be confused with synchronization at the software/hardware level), is a high-level feature that deals with other user's movement features and that can be computed both on a single user (intra-personal features) and on multiple users (inter-personal features). For example, synchronization can be used to measure the degree of coordination between the joints' velocities of a dancer or to detect the level of entrainment between multiple users to measure collaboration and coalition between them.




Studies on the **synchronization** of expressive human behaviours have a long tradition (Glowinski, Gnecco, Piana, & Camurri, 2013) (Gnecco, et al., 2013) (Varni, Mancini, Volpe, & Camurri, 2009): **intra-personal synchronization** between sequences of expressive behaviours of one or more body modalities (i.e., within one's body) is an important cue of several emotion displays, synchronization between physiological signals and movement kinematics allows one to distinguish between different qualities of human full-body movement. At the same time, **inter-personal synchronization** of expressive behaviours in a group of people is an important cue of group cohesion and soft-entrainment. In dance we can investigate how individuals coordinate their behaviours in space and time to communicate.

**Entrainment can be thought of as a process of interaction, between two or more individuals, that results in movement (or movement features)** shared **dynamics.** Entrainment between dancers emerges if their interactions (e.g. their **movement or movement features)** are characterized by the alignment of period and phase. Entrainment is made of two components (Phillips-Silver, 2012): a *temporal* component i.e., a measure of rhythmic behaviours and exchanges between partners and related to low level features, and an *affective* component, instead, assessed at higher levels in the framework and therefore related to more complex concepts such as sharing an emotional state.

It is necessary to emphasize that these two components are not mutually independent, on the contrary there is a very precise relationship between them: entrainment temporal component can be considered as a necessary condition for the affective one to emerge.

To objectively measure such high-level features (i.e., synchronization and entrainment) we need to consider dancers as interacting systems.

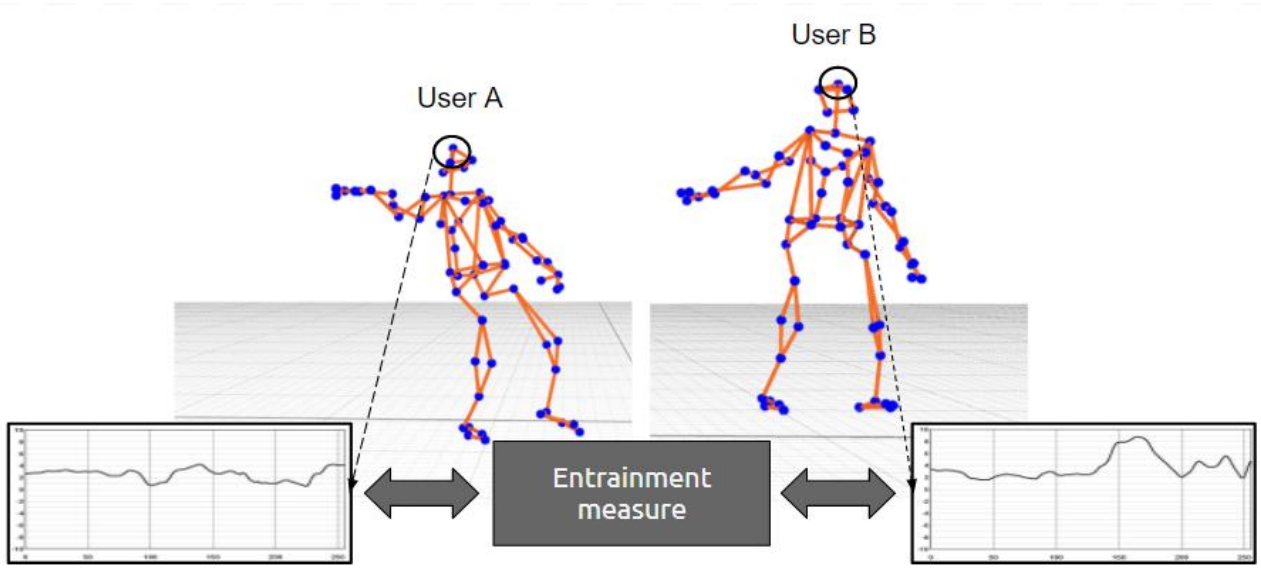



*Figure 6. Example of inter user entrainment extraction procedure*

Non-linear methods can be employed to investigate these concepts, including cross-recurrence quantification and event synchronization (e.g., as an example of how such methods were applied to analysis of entrainment in music performance).

The following section describes some of the relevant techniques and algorithm to investigate synchronization and entrainment and their applicability.

**Multi Event Synchronization Analysis**

Event synchronization analysis was originally conceived with the aim of providing a simple and robust method to measure synchronization between two time series.

The presented synchronization technique is inspired and extends the Event Synchronization (ES) algorithm developed in (Quiroga, 2002). The algorithm is named Multi Event Class Synchronization [Alborno et al. 2017, in preparation], and evaluates the synchronization degree between relevant events belonging to different classes that are detected in multiple time series.

Synchronization can be computed between events belonging to the same class (intra-class synchronization) or between events belonging to different classes (inter-class synchronization).

**Recurrence Quantification Analysis (RQA) as simple entrainment measure**

Recurrences are trajectories outlined and covered by a system (in the space of the system states) that has already been previously visited. In entrainment and synchronization analysis in dance, dancers are modelled as systems and a recurrence is a movement, patterns of movements or temporal evolution of movement qualities reworked and repeated during the performance.

Recurrence Quantification Analysis (RQA) is a method to evaluate the presence of recurrent pattern in time series data. RQA allows to identify the degree to which a considered time series repeats itself and provides a measure of reliability i.e., quantifies how much the recurrent patterns reflect the presence of predictable system dynamics.

**Cross Recurrence Quantification Analysis (CRQA)**

Cross-recurrence quantification analysis extends RQA and it is used to determine the presence and duration of overlap between the dynamics of two different time series by quantifying the regularity, predictability, and stability of two concurrent behavioural performances in reconstructed state space.

CRQA, for example, measures if two dancers come to exhibit similar patterns, dynamics and behaviour in time, and which is the lag time for one individual to maximally match the other, or whether there is a leader–follower type of relationship.

**Recurrence Plots (RP)**

Recurrence plots (RP) are a two-dimensional analytical tool that visualizes recurrences of data.

RP are binary matrices, which depicts the pairs of times at which the trajectory of the system recurs. Each point of the system trajectory at the instant $n$ is represented by a feature vector $\mathbf{v}(\mathbf{n})$. A point (or pixel) $RP_{ij}$ in the plot is recurrent if the distance $D_{ij} = ||\mathbf{v}(\mathbf{n}_i) - \mathbf{v}(\mathbf{n}_j)||$ between two state vectors $\mathbf{v}(\mathbf{n}_i)$ (at time $n_i$) and $\mathbf{v}(\mathbf{n}_j)$ (at time $n_j$) is smaller than the threshold $\boldsymbol{T}$.

This is expressed by:

- $RP_{ij} = \theta(T - ||\mathbf{v}(\mathbf{n}_i) - \mathbf{v}(\mathbf{n}_j)||)$

where $\theta(x)$ is the Heaviside step function, which has the value 0 for $x < 0$ and 1 for $x \geq 0$. If $D_{ij}$ is $< T$ a recurrence point is identified and a value one (or graphically a black pixel) is inserted in the N×N matrix (N is the time interval length), otherwise a value zero (or graphically a white pixel) will be inserted in the RP matrix.

**Software prototype for Recurrence Analysis in dance**

This software module, implemented in EyesWeb, measures the 3D position of different body parts (arms, trunk and head) and shows in real-time the visualisation of dancer's movement analysis of recurrences as Recurrence Plots.

In this preliminary version, recurrences are analysed between movements velocity of different parts of the body of the same dancer. In particular, for the submitted instance, only on the movements of the hands are considered.

Figure 7 shows two recurrence plots, i.e., a graphical representation of the recurrence matrices computed on the movement's velocities of the two hands of the dancer.

In particular, a noisy speed movement will generate a disordered and unregularly binary matrix (see signal and plot on the left of the figure), while a regular and quasi-periodic, movement (again in terms of speed) will generate a graph with a visually more ordered pattern.

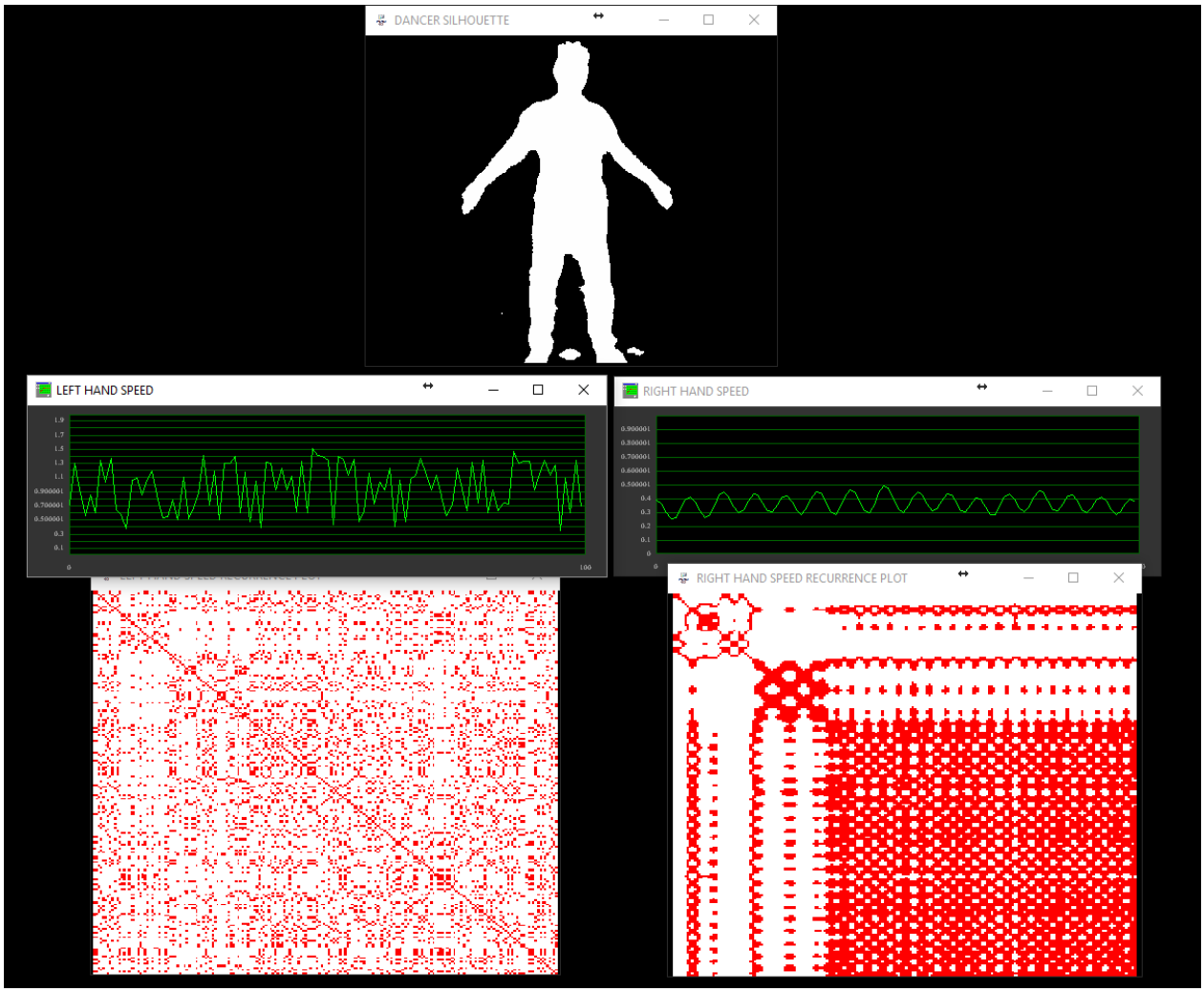



*Figure 7. Example of the recurrence plot extraction prototype application*

## Bibliography

Camurri, A. G. (2016). The dancer in the eye: towards a multi-layered computational framework of qualities in movement. *Proceedings of the 3rd International Symposium on Movement and Computing.* ACM.

Phillips-Silver, J. a. (2012). Searching for roots of entrainment and joint action in early musical interactions. *Frontiers in human neuroscience*.

Quiroga, R. Q. (2002). Event synchronization: a simple and fast method to measure synchronicity and time delay patterns. *Physical review*.