

# TIES basic complete run (rev A)

## TIES basic complete run (rev A)

Community rating: ?????

This trains and tests the full TIES coupled-oscillator model of emotional interaction. Using cross-validation, this learns shared parameters for the oscillator and tests those models. Documentation for the underlying tool is at <http://www.compties.org/>

## Quick Start

- To use **TIES basic complete run (rev A)**, import your data in **CSV** format. Upload your file to the Data Store and browse to its name as the input data.
- Choose an "observable" category name, a "distinguisher" category name, and an optional "moderator" category name.
- Click "Launch Analysis" and let the tool go to work!
- **Resources:** <http://www.compties.org/>

## Input Data

The input data file contains measurements and other information about participants, who are paired into dyads (pairs, couples). The file contains both time-varying and time-invariant data.

The input data file is organized as a table of numbers, stored in **comma-separated-variable (CSV)** format. (Spreadsheets such as Excel and OpenOffice can export data in this form.) The first row stores text labels, and rows 2 and below hold numeric data for the participants. Each row stores information about one participant at one moment in time. Each column stores one category of data.

The text labels in the first row describe the data categories. Each entry in rows 2 and below should either be a number, or the special string "NA" to indicate a missing datum. Two column labels are mandatory: **Dyad**, and **time**.

## Required column labels (and meanings)

Two of the columns must have the exact labels below. The labels are case-sensitive.

- **Dyad** -- a unique dyad number, e.g., 1, 2, 3, which is an identifier shared by exactly two participants. This number is used extensively in the output.
- **time** -- an index of the time measurement, e.g., 1, 2, 3, . . .

## Other required columns

At least two more columns are required, but the labels are up to you:

- (a *distinguisher* category) -- one column must contain just 0 or 1 and differentiate between the two members of the dyad. For example, in a mother-daughter study, this column might have row-1 label "is\_mother" and rows 2 and below could store 1 for the mother, and 0 for the daughter. You must specify this label name when you run the app.
- (an *observable* category) -- at least one column contains time-varying data for the participant. For example, this might be respiration rate. It could have row-1 label "resp\_rate" and rows 2 and below would store numeric measurements, changing on each row.
- (optional: a *moderator* category) -- if you wish to investigate the explanatory power of a time-invariant factor, then your file should include one or more columns of such data. When they are continuously-varying numerical values, we call them "moderators." For example, this might be age. It could have row-1 label "age\_years" and constant values (perhaps rounded) per participant. The hypothesis is that the oscillator parameters are a linear function of the moderator (or, that they tend to be, if the parameters are stochastic).
- (optional: a *grouping variable*) -- if the explanatory factor takes the form of discrete, categorical values, then it is not a moderator but a "grouping variable." For example, an individual's writing-hand preference, possibly encoded as 0=right hand, 1=left hand. Two independent oscillators will then be inferred -- one for each group. The column needs a row-1 label and a value for each participant.

For more discussion about the differences between moderator and grouping-variable factors, please see below, [Moderator or Grouping Variable?](#)

Extra columns are ignored, and thus you may store all your measurements in one file. Unreferenced columns have no effect on the model.

Every column in the file should have a row-1 label composed of solely of letters, numbers, and underline characters. Do not use spaces, punctuation, or other characters. Each label must be unique.

Test data for this app appears directly in the Discovery Environment in the Data window under *Community Data* -> *iplantcollaborative* -> *example\_data* -> *ties\_basic\_complete\_run*

### Input File(s)

The fictional example below shows the format.

---

Dyad	is_mother	age_years	conflict	resp_rate	dial	time
2	0	17.3	4.2	0.84	0	1
2	0	17.3	4.2	0.72	0.2	2
2	0	17.3	4.2	0.70	-0.4	3
2	0	17.3	4.2	0.83	-0.1	4
2	0	17.3	4.2	0.87	0.3	5
3	0	19.4	1.1	0.50	0.3	1
3	0	19.4	1.1	0.55	0.3	2
3	0	19.4	1.1	0.57	0.4	3
3	0	19.4	1.1	0.55	0.2	4
3	0	19.4	1.1	0.56	-0.1	5
3	0	19.4	1.1	0.55	-0.1	6
7	0	16.7	2.3	0.70	-0.3	1
7	0	16.7	2.3	0.77	-0.2	2
7	0	16.7	2.3	0.80	-0.1	3
7	0	16.7	2.3	0.78	0.2	4
2	1	39.2	4.2	0.73	0.1	1
2	1	39.2	4.2	0.74	0.2	2
2	1	39.2	4.2	0.73	0.2	3
2	1	39.2	4.2	0.72	0.1	4
2	1	39.2	4.2	0.73	0.0	5
3	1	44.6	1.1	0.68	-0.4	1
3	1	44.6	1.1	0.69	-0.5	2
3	1	44.6	1.1	0.65	-0.4	3
3	1	44.6	1.1	0.60	0.2	4
3	1	44.6	1.1	0.58	0.3	5
3	1	44.6	1.1	0.62	0.1	6
7	1	41.1	2.3	0.77	-0.2	1
7	1	41.1	2.3	0.80	-0.1	2
7	1	41.1	2.3	0.78	0.2	3
7	1	41.1	2.3	0.75	0.1	4

If you open the CSV file in a text editor, the first few lines of the above example would look something like this:

```
"Dyad","is_mother","age_years","conflict","resp_rate","dial","time"
2,0,17.3,4.2,0.84,0,1
2,0,17.3,4.2,0.72,0.2,2]
```

## App Parameters

Before you launch the application, you will be prompted for several items:

- **Analysis Name** – this becomes part of the name of the folder to be created to store your output. You can accept the default value or

modify it, however best fits your personal style for organizing your files.

- **Comments** – this is an optional, free-form text field for remarks about this experiment.
- **Select output folder** – this should be the name of one of your Data Store folders. You can accept the default value or modify it.
- **Retain inputs?** This checkbox lets you make a local copy of the input datafile into the output folder. This could be useful if the input file is subject to change.

Section 1 contains the prompts below:

- **Data file input (CSV)** – enter the name of the [input file](#)
- **Moderator category name** – this is an optional category name for moderator data, which corresponds to a dyad and does not change with time. If you use this field, you must enter the exact name of the column label for the moderator column, including lower-case or capital letters. This basic TIES app only supports one moderator (or none) – a future app will support multiple moderators.
- **Observable category name** – this is a category name for data that changes with time and corresponds to an individual participant. Enter the exact name of the column label for the observable data column. This basic TIES app only supports one observable variable – a future app will support multiple observables.
- **Dyad distinguisher category name** – this is a category name for data that corresponds to an individual participant, does not change with time, and is 0 or 1 to differentiate between two members of the dyad.

## Moderator or Grouping Variable?

If you are testing the hypothesis that some known per-individual time-invariant factor can help explain the observed oscillations in your data, then you should use either a *moderator* or a *grouping variable*. But which one?

A **grouping variable** is intended for discrete values that describe categories lacking a natural order. For example, if you hypothesize that an individual's first language helps explain your data, you might record categorical values for each individual indicating 0=Cantonese, 1=Catalan, 2=Korean, 3=Urdu, etc. The order of these numbers is meaningless: whatever the "average" of Catalan and Urdu might be, it is not Korean, regardless of the fact that  $\text{Average}\{1, 3\} = 2$ . Although this fact is obvious to a human interpreter, it would not be obvious to the computer -- it must be told. By indicating that `1st_language` is a grouping variable, you explicitly tell the TIES modeler not to rely on order properties.

The TIES modeler uses grouping variables to segregate the data, and then it infers independent oscillator models for each group. All else being equal, data with fewer groups or more individuals per group will yield results with better significance. Of course a grouping variable must assume at least two values to have any explanatory power.

A **moderator** is intended for numerical values that have a meaningful natural order. Examples: age, body-mass index, number of siblings. A moderator category might assume discrete values, but the order of the values naturally has meaning. For example, if body-mass index truly helps predict good oscillator parameters, then two individuals with BMIs of 30 and 31 (*ceteris paribus*) will have oscillator parameters more similar to each other than to those of an individual with BMI of 20.

The TIES modeler uses moderators in a linear regression model, either to determine oscillator parameters, or (if the parameters are stochastic) to determine the distributions of the oscillator parameters.

## Output

The analysis creates an output folder, using the name specified at launch-time. Inside there are subfolders for the inference results, and with baseline models for comparison.

### Results from the TIES model

There is a subdirectory named **shared-param-CLO** which stores all the results of the training and testing. The **errors** subfolder shows fitting error.

#### File `err-couples.txt`

This file, in the **errors** subfolder, contains the RMS fit error between the data and the oscillator outputs, for each dyad in the input, when it is used for testing (not training).

This file can be useful for diagnosing problems. One can see if there are dyads with data that never fit well. This might mean the data are outliers, or corrupted somehow. Person-0 represents the dyad member with zero-value distinguisher, and Person-1 represents the dyad member with one-valued distinguisher. This file shows errors during the fitting (the early 80% of the data) in two columns, and during the prediction (the late 20% of the data). A quality fit will have low prediction error.

#### File `err-summary.txt`

This file represents the average of the columns of **err-couples.txt** – that is, it shows the RMS fitting error averaged across time and across

couples.

## Results from baseline models

There are more subdirectories containing similarly-organized results for the three baseline models (flat average value, straight line fit, and independent coupled oscillator). The error results are found in **error/err-couples.txt** and **error/err-summary.txt** with the same interpretation as the results in **shared-param-CLO** (see above).

## Interpretation of results

(fill in more later.) Basic story: as the baseline models get more sophisticated (average is simplest, line-fit is intermediate, independent-CLO is the most sophisticated), the fit gets better but the predictions get worse. By taking a Bayesian approach and introducing (and learning) a prior distribution over oscillator characteristics, the TIES model predicts better than any of the baseline models.