MIDERv2 USER'S GUIDE

Alejandro F. Villaverde (afvillaverde@iim.csic.es)

Julio R. Banga (julio@iim.csic.es)

With the collaboration of:
John Ross (Stanford)
Federico Morán (UCM)
Abel Folch Fortuny (UPV)
Alberto Ferrer (UPV)

January 29, 2015

Contents

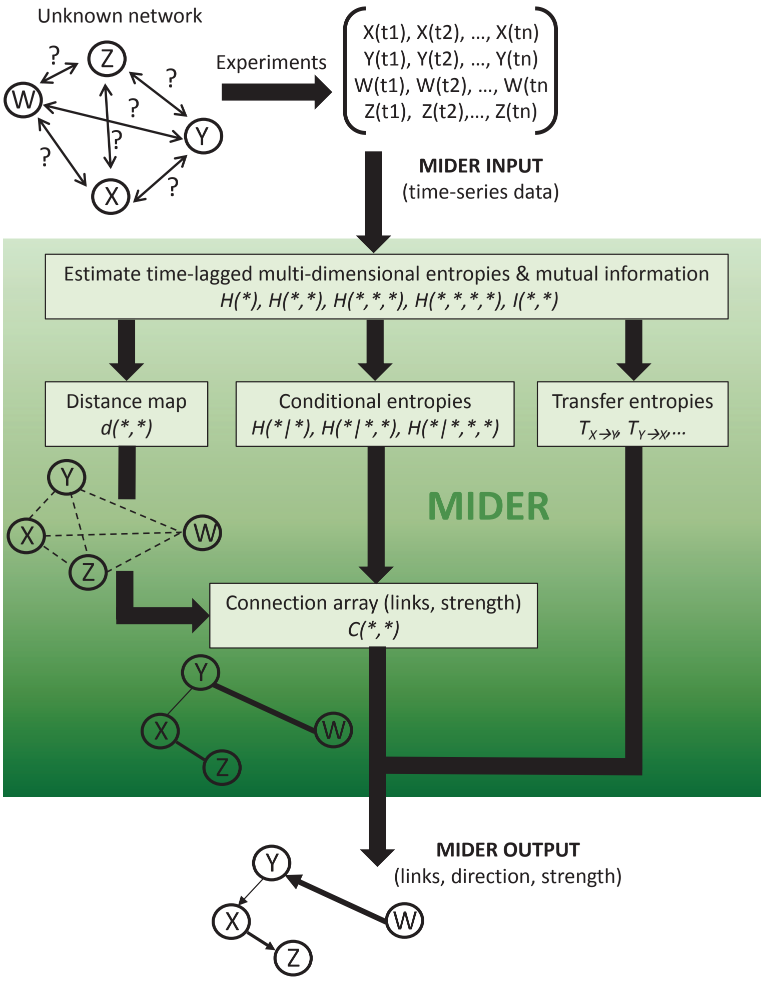


Figure 1: Workflow of the MIDER algorithm

1 Introduction

MIDER (Mutual Information Distance and Entropy Reduction) is a general purpose software tool for inferring network structures. It calculates distances among variables using an entropic measure based on mutual information, which takes into account time delays. For this purpose the user can choose between several definitions and normalizations of mutual information. After obtaining the distance map, conditional entropies calculated from joint entropies of multiple variables are used to distinguish between direct and indirect interactions and to assign directionality. A detailed description can be found in Villaverde et al. (2014); a diagram is shown in Fig. 1.

The MIDERv2 package[*] is implemented in Matlab/Octave, and it can run on any operating system compatible with those environments. The only additional requisite is the Statistics package (in Octave[†]) or the Statistics toolbox (in Matlab). However, the use of the Statistics toolbox/package is optional, since the core methodology does not require it.

The MIDERv2 code consists of a total of 10 Matlab/Octave scripts. The main script is `runMIDER.m`,

[*] In this document we will use the denominations MIDER and MIDERv2 somewhat interchangeably. In general, “MIDER” will be used when referring to the methodology, and “MIDERv2” will be preferred when mentioning some characteristic which is specific of the v2 version.

[†] To load the package, run “pkg load statistics” from the Octave prompt before using MIDER

which can be edited by the user in order to tune the algorithm's settings, as explained in section 3. `runMIDER.m` calls the function that implements the core of the method, `mider.m`, with the appropriate settings. In turn, `mider.m` calls functions `estimateH2.m` and, optionally, `estimateH3.m` and `estimateH4.m`, all of which perform adaptive estimation of mutual information and multi-dimensional joint entropies (of 2, 3, and 4 variables). The estimation is based on the algorithm presented in Cellucci et al. (2005), which was kindly provided by Dr. Alfonso Albano (aalbano@brynmawr.edu).

Functions `plotResults.m`, `projectVars.m` and `arrow.m` are used for visualization purposes. The latter was developed by Dr. Erik A. Johnson (johnsone@usc.edu), who kindly agreed to its distribution with the MIDER package.

The main novelty of MIDERv2 with respect to the original MIDER method is the capability to handle missing data and outliers. This has been enabled by two new functions, `OUTLIERS.m` and `TSR.m`, developed by Abel Folch Fortuny (abfolfor@upv.es) and Alberto Ferrer (aferrer@eio.upv.es), from the Universitat Politècnica de València (UPV), Spain. The rationale behind these functions is briefly explained as follows. If the input dataset is incomplete, that is, if some data points are missing for a certain variable or set of variables, `TSR.m` (Trimmed Scores Regression) generates artificial data points and fills the gaps in the data, in a way that is coherent with the latent structure of the dataset. Additionally, `OUTLIERS.m` detects if the dataset contains abnormal (faulty) values. If this is the case, the user can choose to use the original dataset, or alternatively to remove the outliers and impute new data with the TSR procedure. These additions increase the number of datasets on which the method can be run. They are described in Folch-Fortuny et al. (2015).

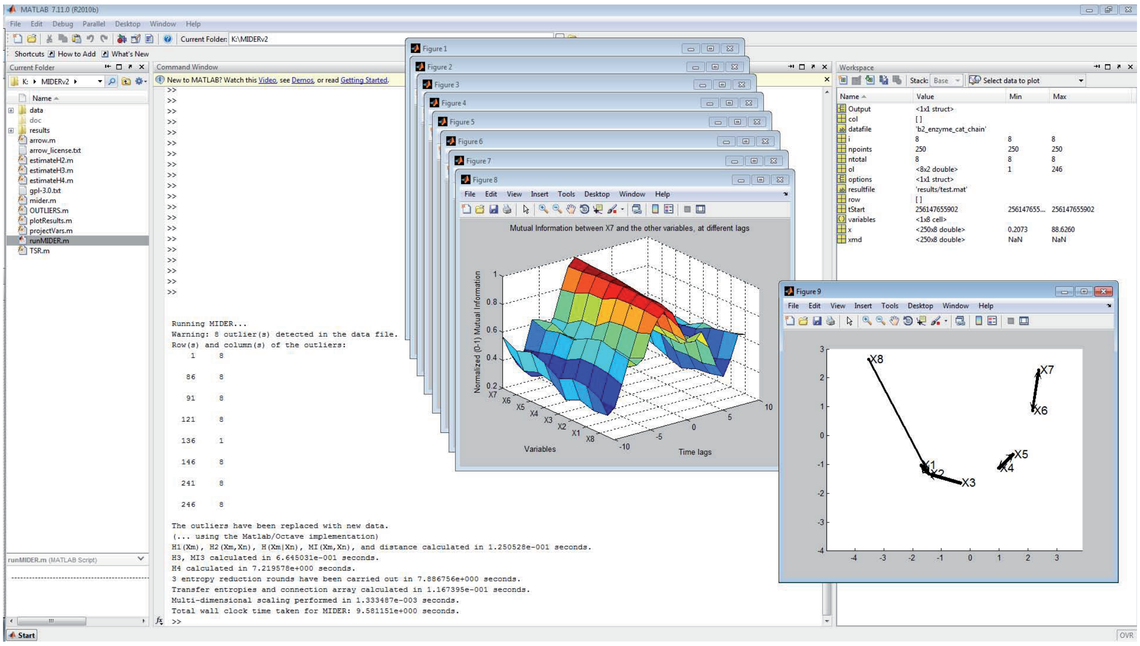
2 Quick start: How to infer a network with MIDER

The MIDERv2 package can be downloaded from `http://www.iim.csic.es/~gingproc/mider.html`. After downloading it, unzip it and save it in your computer. To use MIDER you only need to follow these four steps:

1. Open a Matlab or Octave session, and go to the MIDER root directory ("mider").
2. If you are using Matlab, you must have installed the Statistics toolbox[‡]. if you are using Octave, simply run "pkg load statistics" from the Octave prompt.
3. Define the problem and options by editing the script `runMIDER.m` (for details, see section 3). EXAMPLE (DEMO): If you are running MIDER for the first time and/or just want to see how it works, you can skip this step and leave `runMIDER.m` unedited. This will solve the benchmark problem B2 with default options.
4. Run `runMIDER.m` (to do this you can either type "runMIDER" in the command window, or right-click runMIDER.m in the "Current Directory" tab and select "run").

Done! Results should be obtained in a few seconds. A screenshot is shown in Figure 2. MIDER outputs two types of figures: (1) a 2D map of the distances among variables and the predicted links ("Figure 9" in the screenshot), and (2) for every variable, a plot of the mutual information between that variable and the rest, for all the time lags considered ("Figure 1"−"Figure 8" in the screenshot). Additionally, the results of the calculations are stored in the workspace and saved in a MAT-file. MIDER outputs are described in more detail in section 3.2. Further details about the use of MIDER are given in the next section, starting with the problem definition in 3.1.

[‡] If your Matlab doesn't have the Statistics toolbox, you can still use MIDER, since most of its features will still work. In this case you must change the default options, setting "options.useStatistics = 1", as explained in section 3.1.



```
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>


Running MIDER...
Warning: 8 outlier(s) detected in the data file.
Row(s) and column(s) of the outliers:
     1     8

    86     8

    91     8

   121     8

   136     1

   146     8

   241     8

   246     8

The outliers have been replaced with new data.
(... using the Matlab/Octave implementation)
H1(Xm), H2(Xm,Xn), H(Xm|Xn), MI(Xm,Xn), and distance calculated in 1.250528e-001 seconds.
H3, MI3 calculated in 6.645031e-001 seconds.
H4 calculated in 7.219578e+000 seconds.
3 entropy reduction rounds have been carried out in 7.886756e+000 seconds.
Transfer entropies and connection array calculated in 1.167395e-001 seconds.
Multi-dimensional scaling performed in 1.333487e-003 seconds.
Total wall clock time taken for MIDER: 9.581151e+000 seconds.
>>
```

Stack: Base Select data to plot

Name	Value	Min	Max
Output	<1x1 struct>		
col	[ ]		
datafile	'b2_enzyme_cat_chain'		
i	8	8	8
npoints	250	250	250
ntotal	8	8	8
ol	<8x2 double>	1	246
options	<1x1 struct>		
resultfile	'results/test.mat'		
row	[ ]		
tStart	256147655902	256147655...	256147655902
variables	<1x8 cell>		
x	<250x8 double>	0.2073	88.6260
xmd	<250x8 double>	NaN	NaN

Figure 2: Screenshot of an execution of MIDER

3 MIDER usage

3.1 Problem definition

The script `runMIDER.m` is the main file, and the only one that the user has to modify. It defines a Matlab structure called **options** containing the following fields:

- **options.useStatistics**: in MIDER, some calculations require the use of the Statistics toolbox (in Matlab) or the Statistics package (in Octave[§]). Set it to 1 if this toolbox/package is available in your system, set it to 0 otherwise. Note that, if it is set to 0, the visual output will be disabled, and it won't be possible to correct outliers or missing data.
  Default: `options.useStatistics = 1`.

- **options.correctOutliers**: chooses whether to replace the outliers in the dataset or not. Set it to 1 if you want to replace any existing outliers with new data, or to 0 if you want to use the original dataset even if it has outliers.
  Default: `options.correctOutliers = 1`.

- **options.q**: value of the entropic parameter. Choose $q = 1$ for the classic Shannon entropy (also known as Boltzmann-Gibbs), or $q > 1$ for the generalized Tsallis entropy (Tsallis (1988)). If you want to try Tsallis entropy, typical values are $1.5 < q < 3.5$.
  Default: `options.q = 1`.

- **options.MItype**: selects the type of normalization of mutual information used to create the distance map. Choose 'MI' for the classic, not normalized value; 'MImichaels' for the normalization presented in Michaels et al. (1998); 'MIlinfoot' for the one in Linfoot (1957); or 'MIstudholme' for the one in Studholme et al. (1999). Note that MIDER always calculates (and outputs) all the normalizations; this option is to select the one used in the distance map.
  Default: `options.MItype = 'MI'`.

- **options.fraction**: this parameter is involved in the adaptive estimation of mutual information of a pair of variables (X,Y). It is the minimum fraction of occupied bins in the (X,Y)

[§] To load it, run "pkg load statistics" from the Octave prompt

space with at least 5 points. It should be between 0.01 and 0.5.
Default: `options.fraction = 0.1*(log10(npoints)-1)`
(where `npoints` is a variable measuring the number of data points).

- **options.taumax**: the maximum time lag between two variables X and Y considered in the calculation of mutual information.
Default: `options.taumax = 10`.

- **options.ert_crit**: number of entropy reduction rounds to carry out (0, 1, 2, or 3).
Default: `options.ert_crit = 2`.

- **options.threshold**: entropy reduction threshold. Enter a number between 0.0 and 0.2 to fix it manually, or 1 to use a value obtained from the data.
Default: `options.threshold = 1`.

- **options.plotMI**: plot mutual information arrays (=1) or not (=0).
Default: `options.plotMI = 1`.

Additionally, a MAT-file containing the input data must exist in the "data" folder. The MIDER distribution includes 7 datafiles, one for each of the benchmark problems studied in Villaverde et al. (2014). The input MAT-file is specified in the first lines of the `runMIDER.m` script; by default the B2 benchmark data is chosen (`datafile = 'b2_enzyme_cat_chain';`). The MAT-file must contain at least two variables:

- A m*n data array named "x", with m data points (rows) and n variables (columns).
- A vector named "variables", containing the names of the variables as strings.

Finally, it is recommended to save the results in another MAT-file. To do this, specify a name of the results file (default: `resultfile = 'results/results_mider_b2.mat';`).

3.2 Output

MIDER produces a structure called **Output** containing the following fields:

- **Output.H1** = n-vector of entropies.
- **Output.MI** = n*n*(nlags+1) array, mutual information (several lags).
- **Output.MIl** = mutual information normalized as in Linfoot (1957).
- **Output.MIm** = mutual information normalized as in Michaels et al. (1998).
- **Output.MIs** = mutual information normalized as in Studholme et al. (1999).
- **Output.H2** = n*n*(nlags+1) array, joint entropy of 2 variables.
- **Output.H3** = n*n*n array of joint entropy of 3 variables (calculated only if ert_crit >= 2).
- **Output.MI3** = n*n*n array of three-way mutual information (calculated only if if ert_crit >= 2).
- **Output.H4** = n*n*n*n array of joint entropy of 4 variables (calculated only if if ert_crit >= 3).
- **Output.dist** = n*n*(nlags+1) array of distance between variables.
- **Output.taumin** = n*n array of the time lags that minimize the entropic distance.
- **Output.cond_entr2** = n*n array of conditional entropies of 2 variables.

- **Output.cond_entr3** = n*n*n array of conditional entropies of 3 variables (calculated only if if ert_crit >= 2).
- **Output.cond_entr4** = n*n*n*n array of conditional entropies of 4 variables (calculated only if if ert_crit >= 3).
- **Output.con_array** = n*n array of connections between variables
- **Output.adaptThres** = adaptive threshold value
- **Output.T** = n*n array of transfer entropies
- **Output.Y** = coordinates of the points from multidimensional scaling

References

Cellucci, C., Albano, A., and Rapp, P. (2005). Statistical validation of mutual information calculations: Comparison of alternative numerical algorithms. *Phys. Rev. E*, 71(6):066208.

Folch-Fortuny, A., Villaverde, A., Ferrer, A., and Banga, J. (2015). Enabling network inference methods to handle missing data and outliers. *BMC Bioinform. (submitted).*

Linfoot, E. (1957). An informational measure of correlation. *Inf. Control*, 1:85–89.

Michaels, G., Carr, D., Askenazi, M., Fuhrman, S., Wen, X., and Somogyi, R. (1998). Cluster analysis and data visualization of large scale gene expression data. In *Pac. Symp. Biocomp.*, volume 3, pages 42–53.

Studholme, C., Hill, D., and Hawkes, D. (1999). An overlap invariant entropy measure of 3d medical image alignment. *Pattern Recogn.*, 32:71–86.

Tsallis, C. (1988). Possible generalization of boltzmann-gibbs statistics. *J. Stat. Phys.*, 52(1):479–487.

Villaverde, A., Ross, J., Morán, F., and Banga, J. (2014). MIDER: network inference with mutual information distance and entropy reduction. *PLOS ONE*, 9(5):e96732.