CODECHECK certificate 2021-001

doi.org/10.5281/zenodo.4720843

CODECHECK summary

Item	Value
Title	*Causality indices for bivariate time series data: a comparative review of performance*
Authors	Tom Edinburgh (ORCID: 0000-0002-3599-7133), Stephen J. Eglen (ORCID: 0000-0001-8607-8025), Ari Ercole (ORCID: 0000-0001-8350-8093)
Reference	arxiv.org/abs/2104.00718
Repository	github.com/codecheckers/causality-review
Codechecker	Marcel Stimberg (ORCID: 0000-0002-2648-4790)
Date of check	2021-04-27
Summary	The authors provided all material and documented the process well, the check was therefore straightforward. Due to long computation times, only a subset of the results could be checked. Reproducing the results in the repository was successful, with non-significant numerical discrepancies. However, a small number of minor differences with the results in the arXiv preprint merit clarification.

Summary of output files generated

File	Comment	Size
`lp_values.csv`	Results of running linear process (LP) simulations	421066
`lp_time.csv`	Computational times for each linear process (LP) simulations	155149
`corr_plots.pdf`		33795
`corr_plots.eps`	Figure 2 in paper (pdf/eps)	9353954
`lp_figure.pdf`		80108

File	Comment	Size
lp_figure.eps	Figure 3 in paper (pdf/eps)	322102
ul_figure.pdf		113914
ul_figure.eps	Figure 4a in paper (pdf/eps)	550104
hu_figure.pdf		89964
hu_figure.eps	Figure 4b in paper (pdf/eps)	455639
hb_figure1.pdf		48009
hb_figure1.eps	Figure 5a in paper (pdf/eps)	8113334
hb_figure2.pdf		71634
hb_figure2.eps	Figure 5b in paper (pdf/eps)	8117588
corr_transforms_plots.pdf		20007
corr_transforms_plots.eps	Figure S1 in paper (pdf/eps)	3486705
ul_scaling_figure.pdf		140747
ul_scaling_figure.eps	Figure S2a in paper (pdf/eps)	1046568
ul_rounding_figure.pdf		203919
ul_rounding_figure.eps	Figure S2b in paper (pdf/eps)	1056125
ul_missing_figure.pdf		161463
ul_missing_figure.eps	Figure S3a in paper (pdf/eps)	804604
ul_gaussian_figure.pdf		207213
ul_gaussian_figure.eps	Figure S3b in paper (pdf/eps)	1071981
ul-transforms.txt	Table III in paper (LaTex code). Note: small error in paper in Baseline column, TE (H)/ETE (H) to be correct in arXiv. This is already corrected in journal submission.	3214
computational-times.txt	Table S.II in paper (LaTex code)	1180

Summary

The authors provided all material and documented the process well, the check was therefore straightforward. Due to long computation times, only a subset of the results could be checked. Reproducing the results in the repository was successful, with non-significant numerical discrepancies. However, a small number of minor differences with the results in the arXiv preprint merit clarification.

CODECHECKER notes

Workflow

The original repository for the code was located at github.com/tedinburgh/causality-review, and an earlier version had been archived at zenodo.org/record/4657015. I forked the repository at commit `010aa51a80d91857bea4f0aa33885183022ce59d` to github.com/codecheckers/causality-review and started the CODECHECK. The original repository already contained a `codecheck.yml` MANIFEST, as well as a `README.md` file detailing the steps to run the code, a `requirements.txt` file stating the dependencies (with minimal versions), and a `codecheck-instructions.sh` script to automatically execute the steps detailed in the README file. The script can be downloaded individually; executing it will download the GitHub repository, set up a conda environment and run all the steps to reproduce the results. Since I already had cloned the full repository, I did not execute the script but instead only run the steps following the cloning of the repository. As suggested by the authors, I only reproduced one part of the simulation results (linear processes), since re-running all the simulations would have taken too long. All other figures and data tables were regenerated from stored results also present in the repository (`simulation-data/`).

Execution of the workflow

I ran everything on a somewhat outdated workstation (Intel(R) Xeon(R) CPU E5-1630 v3 @ 3.70GHz, 16GB RAM) on Ubuntu Linux 18.04. The simulation time was 4 hours, comparable to the 3 hours stated by the authors. Regenerating the figures took only about 1 minute, significantly shorter than the "up to 15 minutes" suggested by the authors. Creating the figures emitted a number of warnings (see below), but none of them seemed to affect the output and all figures were created successfully.

Output from running `python causality-review-code/misc_ci.py`

```
causality-review-code/misc_ci.py:729: RuntimeWarning: Mean of empty slice
  percentiles = np.nanpercentile(np.nanmean(z, axis = 0), [5, 95])
causality-review-code/misc_ci.py:735: RuntimeWarning: Mean of empty slice
  percentiles = np.nanpercentile(np.nanmean(z, axis = 0), [5, 95])
The PostScript backend does not support transparency; partially transparent artists will be
↪  rendered opaque.
The PostScript backend does not support transparency; partially transparent artists will be
↪  rendered opaque.
causality-review-code/misc_ci.py:820: RuntimeWarning: Mean of empty slice
  hu_means = np.nanmean(hu_results, axis = 0).reshape(hu_reshape, order = 'F')
/mnt/data/anaconda2/envs/causality_test/lib/python3.8/site-packages/numpy/lib/nanfunctions.py:1664:
↪  RuntimeWarning: Degrees of freedom <= 0 for slice.
  var = nanvar(a, axis=axis, dtype=dtype, out=out, ddof=ddof,
The PostScript backend does not support transparency; partially transparent artists will be
↪  rendered opaque.
The PostScript backend does not support transparency; partially transparent artists will be
↪  rendered opaque.
causality-review-code/misc_ci.py:852: RuntimeWarning: Mean of empty slice
  hb_means = np.nanmean(hb_results, axis = 0)
causality-review-code/misc_ci.py:929: RuntimeWarning: Mean of empty slice
  ul_means = np.nanmean(ul_results, axis = 0).reshape(n_lambda, 2 * n_inds, 2)
/mnt/data/anaconda2/envs/causality_test/lib/python3.8/site-packages/numpy/lib/nanfunctions.py:1664:
↪  RuntimeWarning: Degrees of freedom <= 0 for slice.
  var = nanvar(a, axis=axis, dtype=dtype, out=out, ddof=ddof,
The PostScript backend does not support transparency; partially transparent artists will be
↪  rendered opaque.
The PostScript backend does not support transparency; partially transparent artists will be
↪  rendered opaque.
causality-review-code/misc_ci.py:972: RuntimeWarning: Mean of empty slice
  ul_means = np.nanmean(ul_results, axis = 0).reshape(n_lambda, 2 * n_inds, 2)
/mnt/data/anaconda2/envs/causality_test/lib/python3.8/site-packages/numpy/lib/nanfunctions.py:1664:
↪  RuntimeWarning: Degrees of freedom <= 0 for slice.
```

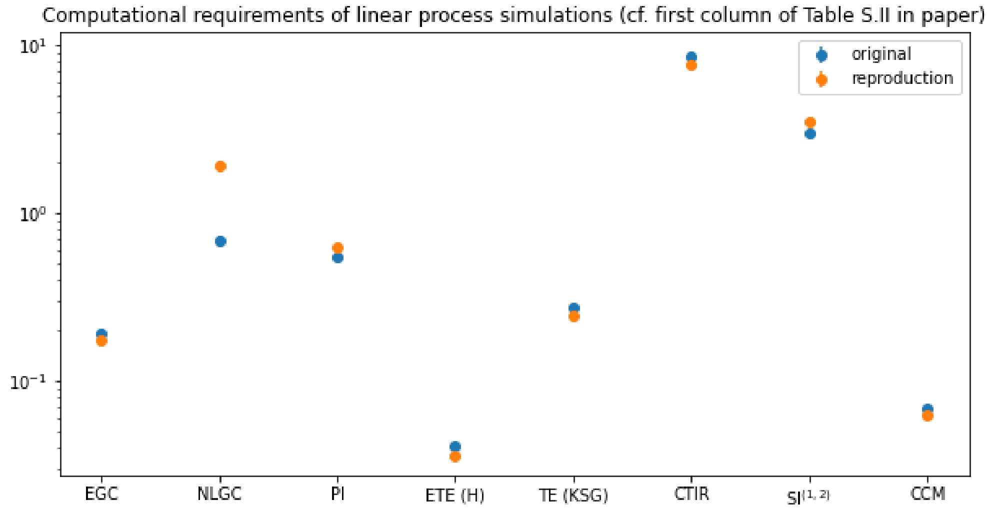
```
  var = nanvar(a, axis=axis, dtype=dtype, out=out, ddof=ddof,
causality-review-code/misc_ci.py:979: RuntimeWarning: Mean of empty slice
  ult_means = np.nanmean(ult_results, axis = 0)
The PostScript backend does not support transparency; partially transparent artists will be
↪  rendered opaque.
The PostScript backend does not support transparency; partially transparent artists will be
↪  rendered opaque.
The PostScript backend does not support transparency; partially transparent artists will be
↪  rendered opaque.
The PostScript backend does not support transparency; partially transparent artists will be
↪  rendered opaque.
The PostScript backend does not support transparency; partially transparent artists will be
↪  rendered opaque.
The PostScript backend does not support transparency; partially transparent artists will be
↪  rendered opaque.
The PostScript backend does not support transparency; partially transparent artists will be
↪  rendered opaque.
The PostScript backend does not support transparency; partially transparent artists will be
↪  rendered opaque.
/mnt/data/anaconda2/envs/causality_test/lib/python3.8/site-packages/numpy/lib/nanfunctions.py:1664:
↪  RuntimeWarning: Degrees of freedom <= 0 for slice.
  var = nanvar(a, axis=axis, dtype=dtype, out=out, ddof=ddof,
```

Comparison of results with author repository

By visual inspection, all regenerated figures are identical to the figures present in the repository. Comparison with `git diff-image` (github.com/ewanmellor/git-diff-image) showed minimal differences in some regions of the color plots of `hb_figure1.{pdf,eps}` and `hb_figure2.{pdf,eps}`, but these differences were not discernible by naked eye and seem to reflect very minor numerical differences. Given that my figures were generated with matplotlib 3.3.4 (see package versions at the end of this document) and the authors generated figures with 3.3.2, I suspected this version difference to be the reason, but a cursory check with a downgraded matplotlib did not change the result. The generated file `ul-transforms.txt` (underlying Table III in the paper) is identical to the file in the repository, except for some irrelevant differences between `0.000` and `-0.000`.

The simulation results for the linear process simulations stored in `lp_values.csv` differ slightly in columns 9–12 and 19–20, reflecting very minor numerical differences. After rounding all values to 10 decimal digits, the results were exactly identical.

Since the file `lp_times.csv` contains execution times measured during the run of this CODECHECK, it differs from the files provided by the authors. This also holds for the column representing the values for the linear processes simulations in file `computational-times.txt`. The results do seem comparable to the authors' results, though, and the order of the methods is preserved. See below for a graphical comparison:



Comparison of results with arXiv preprint

Table III The `codecheck.yml` manifest notes that the arXiv preprint has a small error in Table III in the baseline column for methods "TE (H)" and "ETE (H)", which I can confirm: the results in the repository state $\langle\mu\rangle = 0.675$ ("TE (H)") and $\langle\mu\rangle = 0.674$ ("ETE (H)"), wheras the paper states $\langle\mu\rangle = 0.673$ for both. However, I identified additional differences in the Gaussian noise column for the "NLGC" and "CCM" methods:

Paper

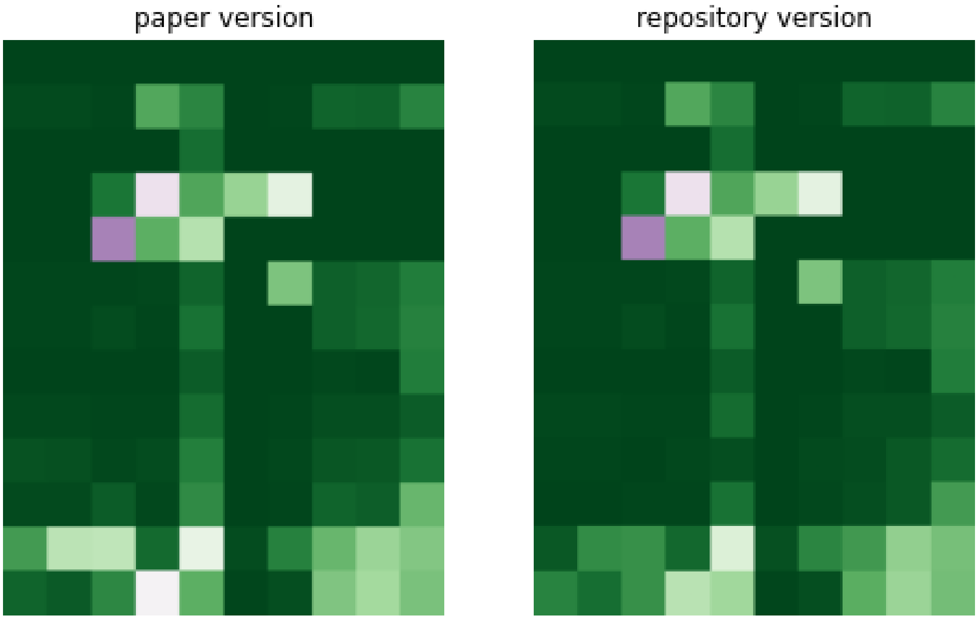
Method	$\sigma_G^2 = 0.1$	$\sigma_G^2 = 1$	$\sigma_G^2 = 1$
NLGC	0.030	0.741	-0.003
	0.972	1.335	2.313
CCM	0.005	0.176	-0.136
	0.981	0.986	0.951

Repository (file `ul-transforms.txt`):

Method	$\sigma_G^2 = 0.1$	$\sigma_G^2 = 1$	$\sigma_G^2 = 1$
NLGC	0.031	0.740	-0.007
	1.023	1.345	2.325
CCM	0.013	0.151	-0.075
	1.010	0.944	0.959

Figure 4 The top part of Figure 4 (file `ul_figure.{pdf,eps}`) uses a different y axis scale for the EGC method in the repository file compared to the one included in the paper. As far as I can tell, the plotted values appear to be the same, i.e. it is just a question of "zoom level".

Figure S1 There appears to be a small difference between the Figure S1 used in the arXiv preprint and the one in the repository (file `corr_transforms_plots.{pdf,eps}`). To confirm, I used the `pdfimages` tool to extract a png of the color plot from both the paper PDF and from the repository version, and plot them side by side:



While the difference is small, it seems to be too big to be simply explained by e.g. a color conversion process (note the differences in the lower left corner).

Recommendation to the authors

Overall, the authors provide very thorough and easy-to-follow steps for reproduction, and make it conveniently possible to only reproduce parts of their study by calling the respective scripts with command line arguments. Apart from clearing up the minor discrepancies detailed in the report above, I only have a few minor recommendations:

- It would be preferrable to have only one file for each figure instead of one PDF and one EPS version. Automatic treatement of the manifest file is also slightly impaired by the fact that the file comment is formally only attached to the EPS file entry but refers to both files.

- It would be helpful to clearly state if files are not expected to be reproduced exactly, e.g. if they represent measured execution times instead of calculated values (`lp_times.csv`, `computation-times.txt`).
- Long simulation runs (in this CODECHECK, the linear process simulations) would benefit from some indication of how much time (or how many iterations) is still needed to complete the run.
- The bold formatting in the tables (indicating e.g. minimum values per column) seems to have been added manually after the automatic generation of the tables. To avoid errors, it might make sense to have the code also take care of this highlighting.
- A very minor point: the `codechecker-instructions.sh` script contained in the repository is meant to be independent of the repository and starts by cloning it. It is unclear in what situation someone would have access to this script file but not have already cloned the repository. It might have been more straightforward to state in the README file to clone the repository, and then ask the user to execute the script file.

Citing this document

Marcel Stimberg (2021). CODECHECK Certificate 2021-001. Zenodo. doi.org/10.5281/zenodo.4720843

About CODECHECK

This certificate confirms that the codechecker could independently reproduce the results of a computational analysis given the data and code from a third party. A CODECHECK does not check whether the original computation analysis is correct. However, as all materials required for the reproduction are freely availableby following the links in this document, the reader can then study for themselves the code and data.

About this document

This document was created using a jupyter notebook and converted into PDF via nbconvert, pandoc, and xelatex. The command `make codecheck.pdf` will regenerate the report file.

License

Package versions

```
# packages in environment at /mnt/data/anaconda2/envs/causality_test:
#
# Name                    Version                   Build  Channel
_libgcc_mutex             0.1                        main
blas                      2.11                   openblas    conda-forge
ca-certificates           2021.4.13            h06a4308_1
certifi                   2020.12.5        py38h06a4308_0
cycler                    0.10.0                   py38_0
dbus                      1.13.18              hb2f20db_0
expat                     2.3.0                h2531618_2
fontconfig                2.13.1               h6c09931_0
freetype                  2.10.4               h5ab3b9f_0
glib                      2.68.1               h36276a3_0
gst-plugins-base          1.14.0               h8213a91_2
gstreamer                 1.14.0               h28cd5cc_2
icu                       58.2                 he6710b0_3
joblib                    1.0.1              pyhd3eb1b0_0
jpeg                      9b                   h024ee3a_2
kiwisolver                1.3.1            py38h2531618_0
lcms2                     2.12                 h3be6417_0
ld_impl_linux-64          2.33.1               h53a641e_7
libblas                   3.8.0               11_openblas    conda-forge
libcblas                  3.8.0               11_openblas    conda-forge
libffi                    3.3                  he6710b0_2
libgcc-ng                 9.1.0                hdf63c60_0
libgfortran-ng            7.3.0                hdf63c60_0
liblapack                 3.8.0               11_openblas    conda-forge
liblapacke                3.8.0               11_openblas    conda-forge
libopenblas               0.3.6                h5a2b251_2
libpng                    1.6.37               hbc83047_0
libstdcxx-ng              9.1.0                hdf63c60_0
libtiff                   4.1.0                h2733197_1
libuuid                   1.0.3                h1bed415_2
libxcb                    1.14                 h7b6447c_0
libxml2                   2.9.10               hb55368b_3
lz4-c                     1.9.3                h2531618_0
matplotlib                3.3.4            py38h06a4308_0
matplotlib-base           3.3.4            py38h62a2d02_0
ncurses                   6.2                  he6710b0_1
numpy                     1.20.1           py38h5a90a98_0
numpy-base                1.20.1           py38h34387ca_0
olefile                   0.46                     py_0
openssl                   1.1.1k               h27cfd23_0
palettable                3.3.0                    py_0
pandas                    1.2.4            py38h2531618_0
patsy                     0.5.1                  py38_0
pcre                      8.44                 he6710b0_0
pillow                    8.2.0            py38he98fc37_0
pip                       21.0.1           py38h06a4308_0
pyparsing                 2.4.7              pyhd3eb1b0_0
pyqt                      5.9.2            py38h05f1152_4
python                    3.8.8                hdb3f193_5
python-dateutil           2.8.1              pyhd3eb1b0_0
pytz                      2021.1             pyhd3eb1b0_0
qt                        5.9.7                h5867ecd_1
readline                  8.1                  h27cfd23_0
scikit-learn              0.23.1           py38h7ea95a0_0
scipy                     1.6.2            py38hf56f3a7_1
setuptools                52.0.0           py38h06a4308_0
```

```
sip                  4.19.13       py38he6710b0_0
six                  1.15.0        py38h06a4308_0
sqlite               3.35.4            hdfb4753_0
statsmodels          0.11.1        py38h7b6447c_0
threadpoolctl        2.1.0           pyh5ca1d4c_0
tk                   8.6.10            hbc83047_0
tornado              6.1           py38h27cfd23_0
wheel                0.36.2          pyhd3eb1b0_0
xz                   5.2.5             h7b6447c_0
zlib                 1.2.11            h7b6447c_3
zstd                 1.4.9             haebb681_0
```

Manifest files

CSV files

`simulation-data/lp_values.csv`

Author comment: *Results of running linear process (LP) simulations*

Column summary statistics:

	count	mean	std	min	25%	50%	75%	max
0	1010	0.0118	0.0013	0.0080	0.0111	0.0118	0.0125	0.0162
1	1010	0.0983	0.0671	0.0069	0.0336	0.0936	0.1584	0.2242
2	1010	-0.0007	0.0010	-0.0030	-0.0014	-0.0008	-0.0001	0.0028
3	1010	0.0904	0.0674	-0.0017	0.0254	0.0860	0.1510	0.2166
4	1010	-0.0012	0.0060	-0.0211	-0.0050	-0.0012	0.0027	0.0167
5	1010	0.1524	0.1221	-0.0152	0.0389	0.1312	0.2582	0.4026
6	1010	-0.0098	0.0069	-0.0228	-0.0158	-0.0098	-0.0035	0.0047
7	1010	0.0118	0.0092	-0.0035	0.0035	0.0113	0.0196	0.0320
8	1010	0.0001	0.0068	-0.0194	-0.0044	-0.0004	0.0039	0.0409
9	1010	0.0555	0.0496	-0.0105	0.0110	0.0437	0.0925	0.1890
10	1010	0.0000	0.0001	-0.0002	-0.0001	0.0000	0.0001	0.0005
11	1010	0.0183	0.0165	-0.0003	0.0036	0.0137	0.0302	0.0644
12	1010	-0.0001	0.0015	-0.0045	-0.0011	0.0001	0.0009	0.0046
13	1010	0.0697	0.0623	-0.0029	0.0132	0.0525	0.1176	0.2143
14	1010	0.1578	0.0646	0.0486	0.0956	0.1594	0.2174	0.2706
15	1010	0.2096	0.1249	0.0388	0.0889	0.1960	0.3221	0.4462
16	1010	0.0058	0.0002	0.0054	0.0057	0.0058	0.0060	0.0061
17	1010	0.0100	0.0016	0.0079	0.0084	0.0097	0.0114	0.0132
18	1010	0.0573	0.0411	0.0000	0.0188	0.0548	0.0944	0.1528
19	1010	0.1892	0.1190	0.0000	0.0750	0.2023	0.2996	0.3771

`simulation-data/lp_time.csv`

Author comment: *Computational times for each linear process (LP) simulations*

Column summary statistics:

	count	mean	std	min	25%	50%	75%	max
0	1010	0.0363	0.0003	0.0357	0.0362	0.0363	0.0364	0.0384
1	1010	0.2466	0.0026	0.2399	0.2448	0.2464	0.2482	0.2594
2	1010	7.7283	0.0607	7.5954	7.6871	7.7202	7.7674	7.9358
3	1010	0.1746	0.0086	0.1571	0.1679	0.1762	0.1821	0.2489
4	1010	1.9358	0.1149	1.8422	1.9066	1.9264	1.9517	5.1241
5	1010	0.6345	0.0040	0.6100	0.6320	0.6345	0.6370	0.6581
6	1010	3.4981	0.0163	3.4641	3.4856	3.4982	3.5089	3.6858
7	1010	0.0630	0.0007	0.0614	0.0626	0.0629	0.0633	0.0709

LaTeX tables

`figures/ul-transforms.txt`

Author comment: *Table III in paper (LaTex code). Note: small error in paper in Baseline column, TE (H)/ETE (H) to be correct in arXiv. This is already corrected in journal submission.*

EGC	$\langle\mu\rangle = 0.840$	$f(\mu,\hat{\mu})$	-0.064	0.036	0.207	-0.071	0.031	0.112	0.004	-0.027	-0.040	0.533	0.981	0.950
	$\langle\sigma\rangle = 0.021$	$g(\sigma,\hat{\sigma})$	0.660	1.004	1.425	0.691	0.959	0.946	0.970	1.023	1.033	1.025	0.473	0.598
NLGC	$\langle\mu\rangle = 0.610$	$f(\mu,\hat{\mu})$	0.013	0.299	2.905	-101.849	0.000	-0.003	0.001	-0.008	-0.020	0.031	0.740	-0.007
	$\langle\sigma\rangle = 0.023$	$g(\sigma,\hat{\sigma})$	0.089	0.608	64.469	126.734	1.000	0.954	0.994	1.353	1.906	1.023	1.345	2.325
PI	$\langle\mu\rangle = 0.380$	$f(\mu,\hat{\mu})$	0.002	0.293	1.576	-98.727	0.950	-0.951	-0.016	0.001	0.005	0.011	0.617	0.019
	$\langle\sigma\rangle = 0.032$	$g(\sigma,\hat{\sigma})$	0.094	0.681	69.340	66.364	0.918	1.051	1.001	1.214	1.511	1.007	2.041	2.120
TE (H)	$\langle\mu\rangle = 0.675$	$f(\mu,\hat{\mu})$	-0.158	0.000	0.000	0.000	0.011	0.030	0.000	0.071	0.159	0.026	0.786	0.731
	$\langle\sigma\rangle = 0.019$	$g(\sigma,\hat{\sigma})$	0.085	1.000	1.000	1.000	1.004	0.994	1.013	1.313	1.633	1.112	1.015	0.920
ETE (H)	$\langle\mu\rangle = 0.674$	$f(\mu,\hat{\mu})$	-0.158	0.000	0.000	0.000	0.014	0.026	0.000	0.075	0.155	0.026	0.748	0.774
	$\langle\sigma\rangle = 0.019$	$g(\sigma,\hat{\sigma})$	0.083	1.000	1.000	1.000	0.992	0.994	1.009	1.296	1.634	1.109	0.947	0.854
TE (KSG)	$\langle\mu\rangle = 1.509$	$f(\mu,\hat{\mu})$	-1.348	0.000	0.269	0.609	0.095	-0.134	-0.029	0.111	0.216	0.306	0.841	0.863
	$\langle\sigma\rangle = 0.025$	$g(\sigma,\hat{\sigma})$	0.120	1.071	0.869	1.026	1.232	1.320	1.042	1.197	1.430	1.028	1.017	0.962
CTIR	$\langle\mu\rangle = 0.462$	$f(\mu,\hat{\mu})$	-1.226	0.000	0.128	0.713	0.299	-0.355	-0.026	0.083	0.161	0.273	0.826	0.848
	$\langle\sigma\rangle = 0.014$	$g(\sigma,\hat{\sigma})$	0.110	1.016	0.898	0.920	1.159	1.209	1.042	1.076	1.258	0.958	0.942	0.872
$\text{SI}^{(1)}$	$\langle\mu\rangle = 0.001$	$f(\mu,\hat{\mu})$	0.015	-0.000	-0.000	0.000	0.549	-0.556	0.005	-0.013	0.018	-0.007	-0.061	0.066
	$\langle\sigma\rangle = 0.029$	$g(\sigma,\hat{\sigma})$	0.097	1.000	1.000	1.000	1.363	1.355	1.053	1.084	1.219	0.895	0.705	0.693
$\text{SI}^{(2)}$	$\langle\mu\rangle = 0.000$	$f(\mu,\hat{\mu})$	0.029	-0.000	-0.000	0.000	-3.105	3.121	-0.001	-0.037	0.017	0.000	-8.256	7.736
	$\langle\sigma\rangle = 0.000$	$g(\sigma,\hat{\sigma})$	0.000	1.000	1.000	1.000	1.394	1.399	1.074	1.666	2.630	0.968	2.334	2.018
CCM	$\langle\mu\rangle = 0.001$	$f(\mu,\hat{\mu})$	0.031	-0.000	-0.000	0.000	-1.249	1.289	-0.005	-0.009	-0.025	0.013	0.151	-0.075
	$\langle\sigma\rangle = 0.047$	$g(\sigma,\hat{\sigma})$	0.103	1.000	1.000	1.000	1.115	1.105	0.740	1.090	1.250	1.010	0.944	0.959

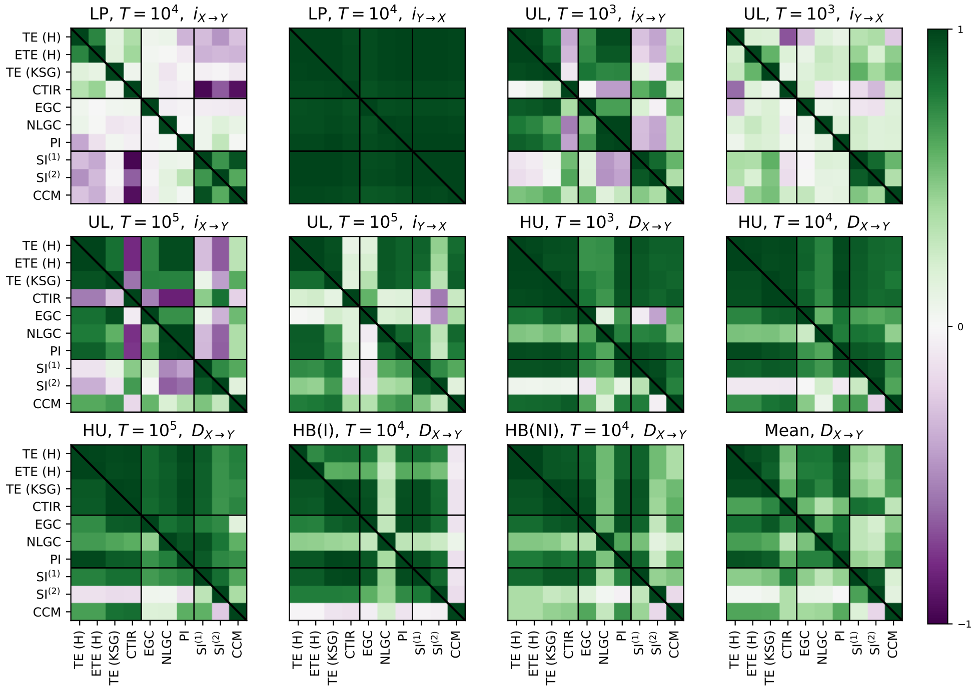
`figures/computational-times.txt`

Author comment: *Table S.II in paper (LaTex code)*

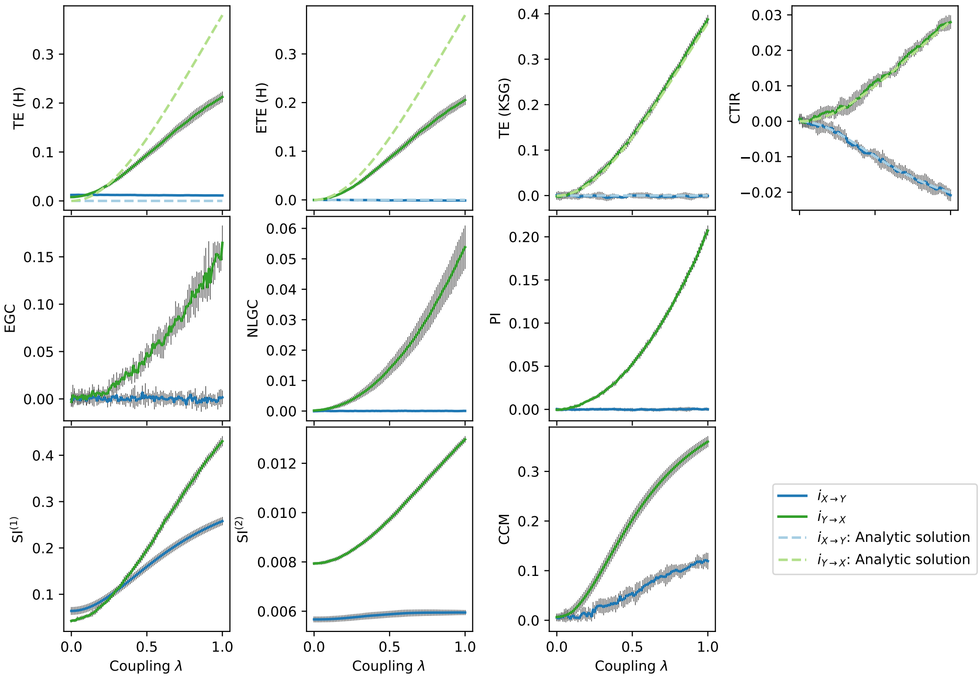
EGC	0.175 (0.009)	0.131 (0.016)	4.095 (12.783)	0.070 (0.076)	0.227 (0.010)	1.216 (0.452)	0.250 (0.032)	0.291 (0.114)
NLGC	1.936 (0.115)	0.314 (0.091)	7.488 (2.106)	0.462 (0.025)	1.536 (0.087)	27.718 (4.105)	0.319 (0.049)	0.320 (0.124)
PI	0.635 (0.004)	0.047 (0.006)	21.323 (61.590)	0.051 (0.002)	0.535 (0.025)	6.166 (1.105)	0.535 (0.056)	0.675 (0.393)
ETE (H)	0.036 (0.000)	0.032 (0.026)	1.060 (0.738)	0.013 (0.001)	0.039 (0.002)	0.370 (0.031)	0.039 (0.002)	0.040 (0.002)
TE (KSG)	0.247 (0.003)	0.021 (0.005)	12.461 (34.498)	0.018 (0.001)	0.216 (0.015)	3.097 (0.680)	0.200 (0.019)	0.264 (0.176)
CTIR	7.728 (0.061)	0.181 (0.029)	140.876 (417.764)	0.156 (0.011)	1.860 (0.104)	25.541 (2.283)	1.714 (0.173)	2.317 (1.624)
$\text{SI}^{(1,2)}$	3.498 (0.016)	0.086 (0.017)	108.029 (22.085)	0.118 (0.005)	2.856 (0.120)	88.200 (50.279)	3.301 (0.201)	3.463 (0.199)
CCM	0.063 (0.001)	0.028 (0.004)	4.500 (15.039)	0.029 (0.001)	0.065 (0.003)	0.740 (0.171)	0.068 (0.009)	0.101 (0.110)

Figures

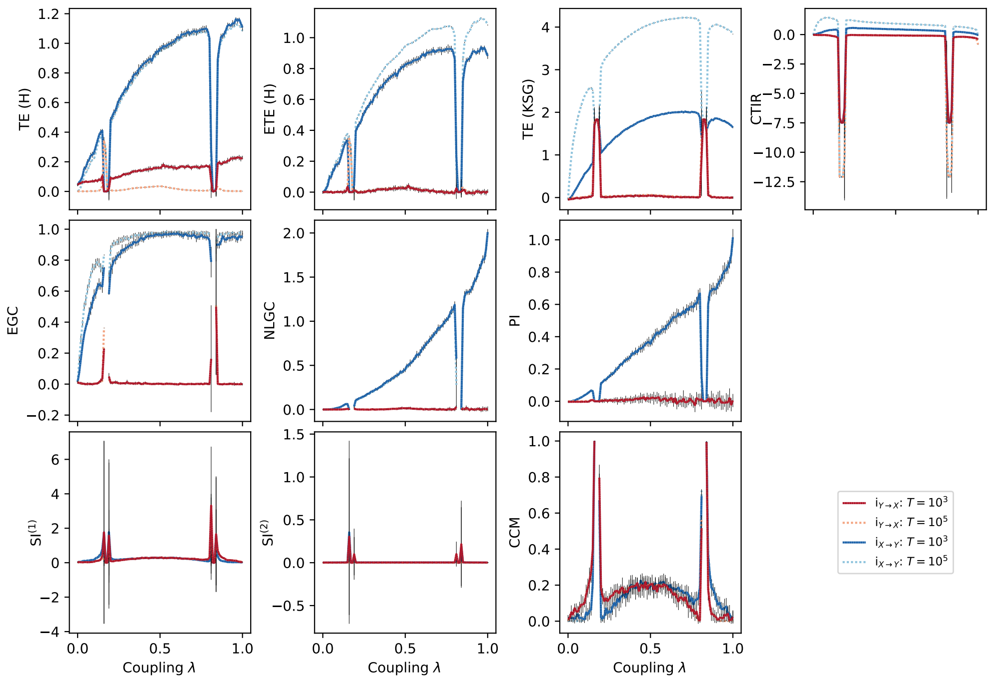
`figures/corr_plots.pdf`.
Author comment: *Figure 2 in paper (pdf/eps)*



`figures/lp_figure.pdf`.
Author comment: *Figure 3 in paper (pdf/eps)*

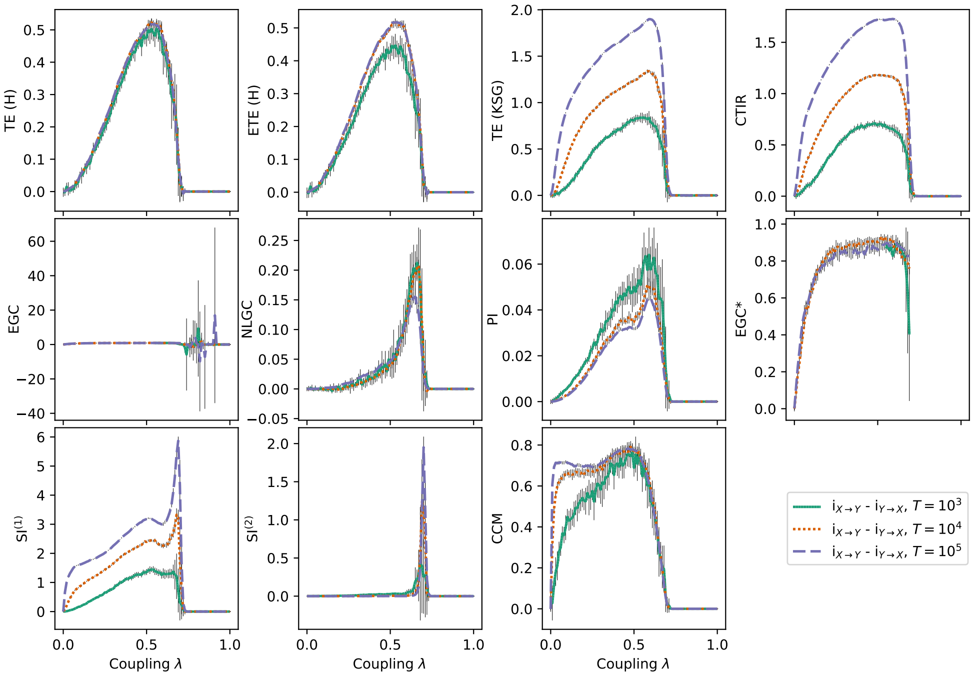


figures/ul_figure.pdf.
Author comment: *Figure 4a in paper (pdf/eps)*



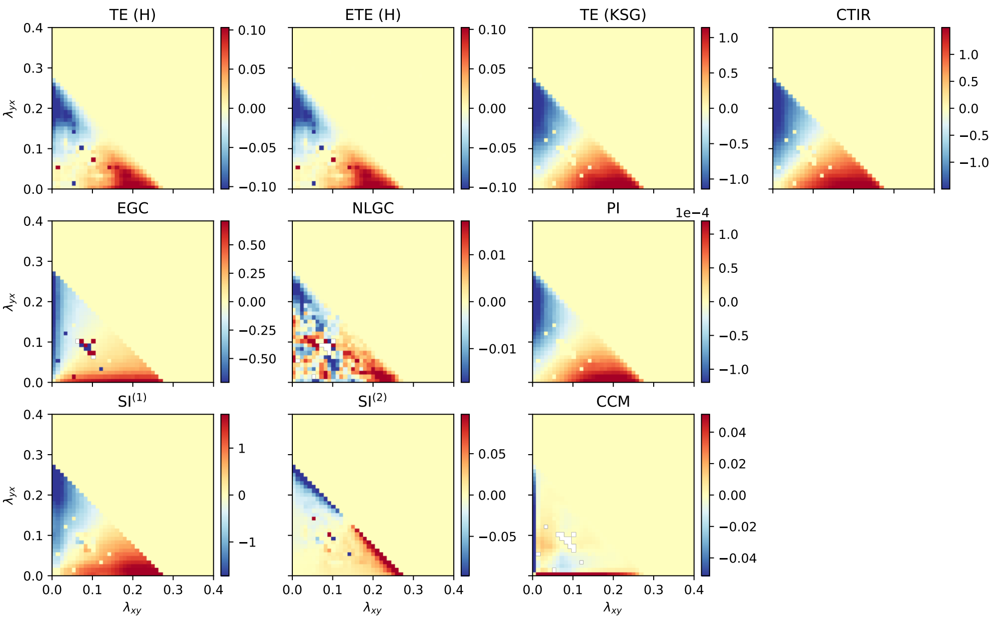
figures/hu_figure.pdf.

Author comment: *Figure 4b in paper (pdf/eps)*



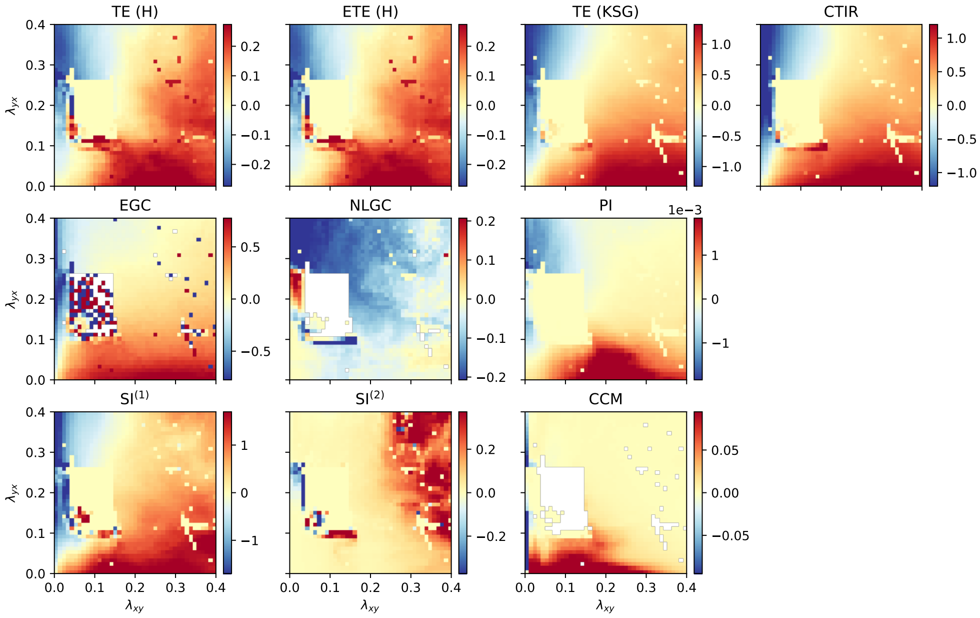
`figures/hb_figure1.pdf`.

Author comment: *Figure 5a in paper (pdf/eps)*

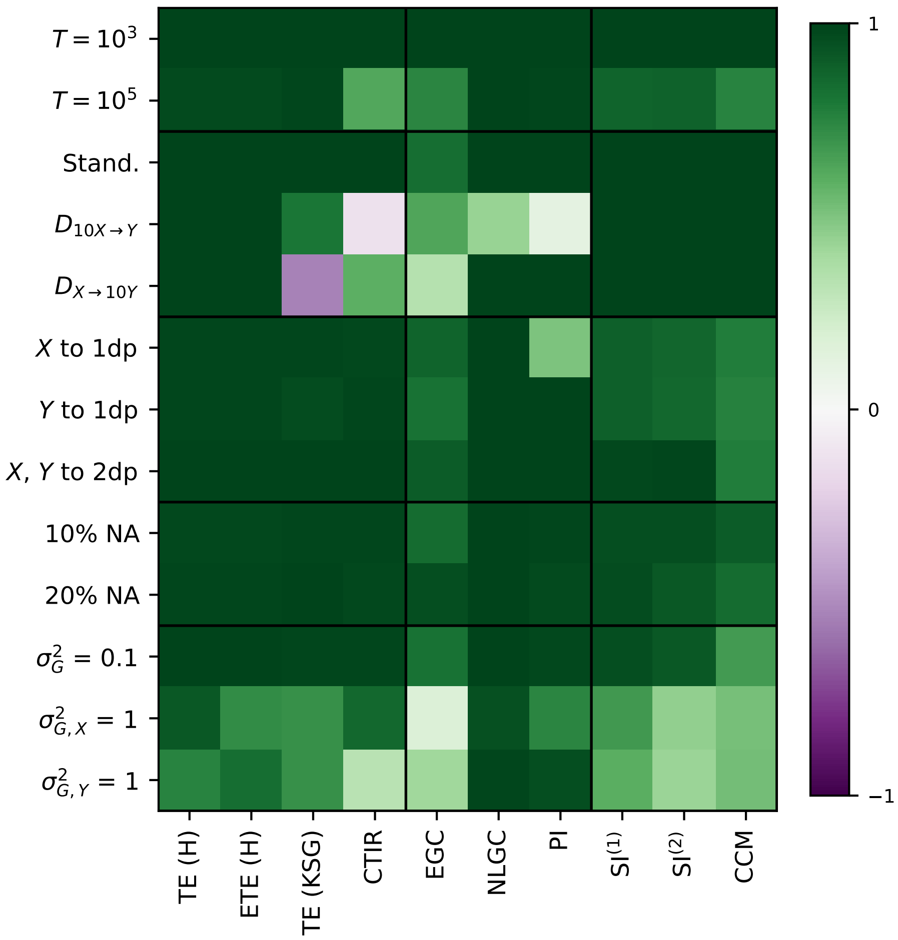


`figures/hb_figure2.pdf`.

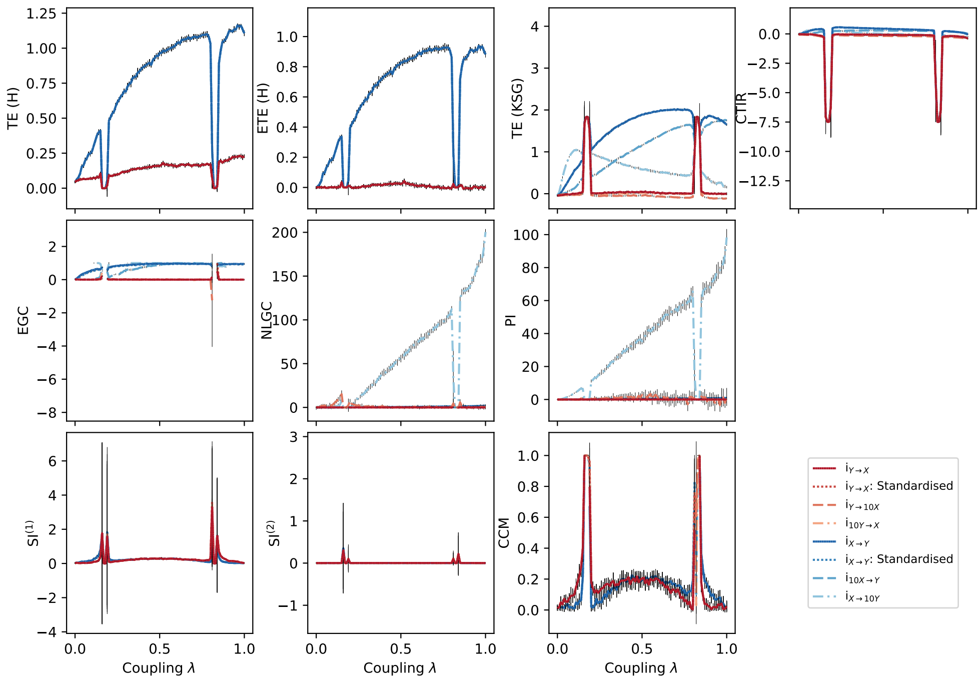
Author comment: *Figure 5b in paper (pdf/eps)*



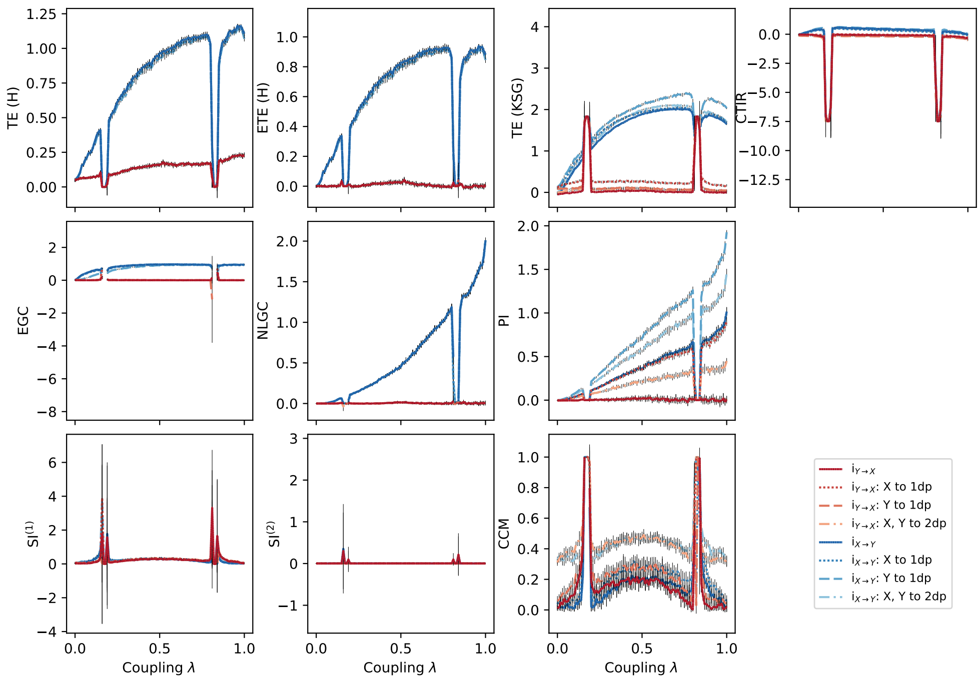
figures/corr_transforms_plots.pdf.
Author comment: *Figure S1 in paper (pdf/eps)*



`figures/ul_scaling_figure.pdf`.
Author comment: *Figure S2a in paper (pdf/eps)*

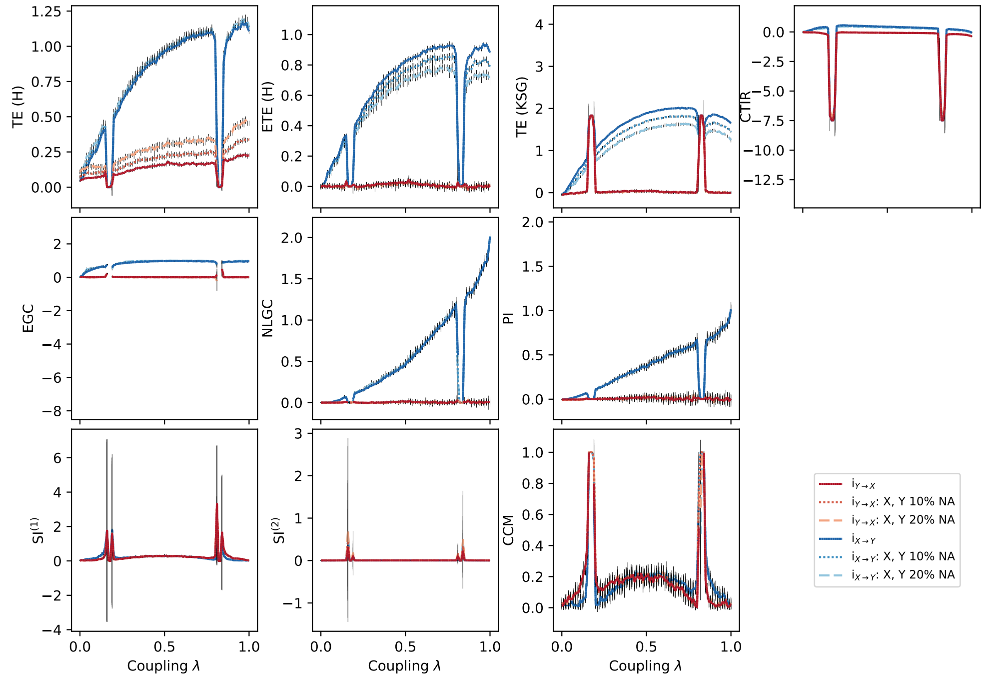


`figures/ul_rounding_figure.pdf`.
Author comment: *Figure S2b in paper (pdf/eps)*



`figures/ul_missing_figure.pdf`.

Author comment: *Figure S3a in paper (pdf/eps)*



`figures/ul_gaussian_figure.pdf`.

Author comment: *Figure S3b in paper (pdf/eps)*