CODECHECK certificate 2020-007

https://doi.org/10.53962/nsys-9a40

Item	Value
Title	[Re] Spike Timing Dependent Plasticity Finds the Start of Repeating Patterns in Continuous Spike Trains
Authors	Pamela Hathway, Dan F. M. Goodman
Reference	https://doi.org/10.5281/zenodo.1327348
Codechecker	Daniel Nüst , Stephen J. Eglen
Date of check	2025-05-19 10:00:00
Summary	The code was straightforward to codecheck. All figures could be recreated with the original code and data as described in the README file. Only the computing environment could not be recreated exactly, as a lot of time has passed since the publication of the original code. Both a quick (saved data) execution and a longer (seemingly new data) execution were successful and yielded figures who by visual inspection matched the figures in the original article. Errors and warnings were mostly related to deprecation of libraries and missing libraries.
Repository	https://github.com/codecheckers/Hathway-Goodman-2018

Table 1: CODECHECK summary

Output	Comment	Size (b)
figure_1_created_250519.pdf	manuscript Figure 1 created in the first code run	11710
figure_2_created_250519.pdf	manuscript Figure 2 created in the first code run	10955
figure_3_seed1_created_250519.pdf	manuscript Figure 3 created in the first code run	19299
figure_4_seed1_created_250519.pdf	manuscript Figure 4 created in the first code run	156363
figure_5_from_saved_created_250519.pdf	manuscript Figure 5 created in the first code run based on saved data	24821
figure_6_from_saved_created_250519.pdf	manuscript Figure 6 created in the first code run based on saved data	12294
figure_7AB_seed1_created_250519.pdf	manuscript Figure 7AB created in the first code run	40297
figure_7CD_seed28_created_250519.pdf	manuscript Figure 7CD created in the first code run	21465
figure_8_created_250519.pdf	manuscript Figure 8 created in the first code run	13687
figure_9sup_seed1_created_250519.pdf	manuscript Figure 9 created in the first code run	37132
figure_10sup_seed1_created_250519.pdf	manuscript Figure 10 created in the first code run	31361
figure_5_created_250520.pdf	manuscript Figure 5 created in the second code run with new data	22648
figure_6_created_250520.pdf	manuscript Figure 6 created in the second code run with new data	12103
figure_7CD_seed28_created_250520.pdf	manuscript Figure 7CD created in the second code run	21465
figure_8_created_250520.pdf	manuscript Figure 8 created in the second code run	13687

Table 2: Summary of output files generated

CODECHECKER notes

This check is an execution of a computational workflow created in the context of a ReScience article (https://rescience.github.io/bibliography/hathway_2018.html, https://doi.org/10.5281/zenodo.1327348). It adds another layer of confirmation to the applaudable efforts of a reference impleementation for the original research article (https://doi.org/10.1371/journal.pone.0001377). The GitHub repo https://github.com/codecheckers/Hathway-Goodman-2018 was forked from the original repository a while back and the check has been taken up by a new codechecker now. This check is based on the commit `af972c52a433fe6be681e169e3e836797f68eeaf`.

The repository contains a short README following the ReScience template. Code was written in Python. I went through the following steps based on the extensive instructions in `code/README.md` based on the section "Quick running of the code".

```
conda update conda

cd code
conda env create -f environment.yml
```

```
LibMambaUnsatisfiableError: Encountered problems while solving:
  - nothing provides bsddb needed by brian2-2.1.2-np113py27h568d706_0
```

I'm now trying brian2 without version `2.2.1` because it is pretty close and available at https://anaconda.org/conda-forge/brian2/labels. However, the channel `brian-team` does not exist anymore: https://conda.anaconda.org/brian-team linked from https://brian2.readthedocs.io/en/2.1.1/introduction/install.html gives a error message "page does not exist". Therefore, I removed all pinned versions except for Python and the main library, which could finally be resolved by conda:

```
dependencies:
  - brian2=2.*
  - matplotlib
  - numba
  - numpy
  - python=3.6.5
```

```
conda activate HathwayGoodman
```

As suggested in the README, I ran the following command to generate most figures but reusing some data:

```
python main.py
```

The code completed without any errors but included a few (deprecation) warnings and library errors, the log output was as below:

```
14:43 Preparing Figure 1: Potentials
WARNING    /home/daniel/miniconda3/envs/HathwayGoodman/lib/python3.6/site-packages/sympy/matrices/matrices.py:1391: SymPyDeprecationWarning:

Dot product of non row/column vectors has been deprecated since SymPy
1.2. Use * to take matrix products instead. See
https://github.com/sympy/sympy/issues/13815 for more info.

  useinstead="* to take matrix products").warn()
 [py.warnings]
14:44 Preparing Figure 2: STDP rules
WARNING    /home/daniel/miniconda3/envs/HathwayGoodman/lib/python3.6/site-packages/sympy/matrices/matrices.py:1391: SymPyDeprecationWarning:

Dot product of non row/column vectors has been deprecated since SymPy
1.2. Use * to take matrix products instead. See
https://github.com/sympy/sympy/issues/13815 for more info.

  useinstead="* to take matrix products").warn()
 [py.warnings]
14:44 Preparing Figures 3, 4, 7AB
14:44 Preparing Figure 3 and 4 and 7AB: latency and convergence and weights
    #### Simulation parameters:
    random seed =             1
    dt =                      100. us
    initial weight =          0.475
    jitter (SD) =             1
    % of neurons in pattern = 50.0
    pattern freq =            0.25
    % spikes deleted =        0
    #### Creating input
```

```
    #### Simulation (ca. 150s)
WARNING    /home/daniel/miniconda3/envs/HathwayGoodman/lib/python3.6/site-packages/sympy/matrices/matrices.py:1391: SymPyDeprecationWarning:

Dot product of non row/column vectors has been deprecated since SymPy
1.2. Use * to take matrix products instead. See
https://github.com/sympy/sympy/issues/13815 for more info.

  useinstead="* to take matrix products").warn()
 [py.warnings]
Number of spikes: 57643428
Number of spikes: 3318
Number of spikes: 21
Number of spikes: 10
Number of synapses: 2000
Number of synapses: 2000
Number of synapses: 2000
Number of synapses: 2000
Number of synapses: 2000
Number of synapses: 2000
Number of synapses: 2000
    #### Results
    Avg latency               =  3.757
    Hit rate (>98)            =  1.0
    Number false alarms (!=0) =  0
    Success                   =  1
    find_t                    =  23.26
    find_spike                =  1176
    #### Make figures
14:49 Preparing Figure 5: Robustness
WARNING    /home/daniel/git/codecheck/Hathway-Goodman-2018/code/figure_5.py:65: MatplotlibDeprecationWarning: Adding an axes using the same arguments as a previous axes currently reuses the earlier instan
  subplot(1, 5, 1)
 [py.warnings]
WARNING    /home/daniel/git/codecheck/Hathway-Goodman-2018/code/figure_5.py:68: MatplotlibDeprecationWarning: Adding an axes using the same arguments as a previous axes currently reuses the earlier instan
  subplot(1, 5, 2)
 [py.warnings]
WARNING    /home/daniel/git/codecheck/Hathway-Goodman-2018/code/figure_5.py:71: MatplotlibDeprecationWarning: Adding an axes using the same arguments as a previous axes currently reuses the earlier instan
  subplot(1, 5, 3)
 [py.warnings]
WARNING    /home/daniel/git/codecheck/Hathway-Goodman-2018/code/figure_5.py:74: MatplotlibDeprecationWarning: Adding an axes using the same arguments as a previous axes currently reuses the earlier instan
  subplot(1, 5, 4)
 [py.warnings]
WARNING    /home/daniel/git/codecheck/Hathway-Goodman-2018/code/figure_5.py:77: MatplotlibDeprecationWarning: Adding an axes using the same arguments as a previous axes currently reuses the earlier instan
  subplot(1, 5, 5)
 [py.warnings]
WARNING    /home/daniel/git/codecheck/Hathway-Goodman-2018/code/figure_5.py:80: MatplotlibDeprecationWarning: Adding an axes using the same arguments as a previous axes currently reuses the earlier instan
  subplot(1, 5, 1)
 [py.warnings]
WARNING    /home/daniel/git/codecheck/Hathway-Goodman-2018/code/figure_5.py:89: MatplotlibDeprecationWarning: Adding an axes using the same arguments as a previous axes currently reuses the earlier instan
  subplot(1, 5, 2)
 [py.warnings]
WARNING    /home/daniel/git/codecheck/Hathway-Goodman-2018/code/figure_5.py:98: MatplotlibDeprecationWarning: Adding an axes using the same arguments as a previous axes currently reuses the earlier instan
  subplot(1, 5, 3)
 [py.warnings]
WARNING    /home/daniel/git/codecheck/Hathway-Goodman-2018/code/figure_5.py:107: MatplotlibDeprecationWarning: Adding an axes using the same arguments as a previous axes currently reuses the earlier insta
  subplot(1, 5, 4)
 [py.warnings]
WARNING    /home/daniel/git/codecheck/Hathway-Goodman-2018/code/figure_5.py:118: MatplotlibDeprecationWarning: Adding an axes using the same arguments as a previous axes currently reuses the earlier insta
  subplot(1, 5, 5)
 [py.warnings]
14:49 Preparing Figure 6: Time of finding pattern
14:49 Preparing Figure 7 C and D: success with ATA rule
WARNING    /home/daniel/miniconda3/envs/HathwayGoodman/lib/python3.6/site-packages/sympy/matrices/matrices.py:1391: SymPyDeprecationWarning:

Dot product of non row/column vectors has been deprecated since SymPy
1.2. Use * to take matrix products instead. See
https://github.com/sympy/sympy/issues/13815 for more info.

  useinstead="* to take matrix products").warn()
 [py.warnings]
Number of spikes: 1032
Number of synapses: 2000
Number of synapses: 2000
Number of synapses: 2000
14:50 Preparing Figure 8: EPSP shape
WARNING    /home/daniel/miniconda3/envs/HathwayGoodman/lib/python3.6/site-packages/sympy/matrices/matrices.py:1391: SymPyDeprecationWarning:

Dot product of non row/column vectors has been deprecated since SymPy
1.2. Use * to take matrix products instead. See
https://github.com/sympy/sympy/issues/13815 for more info.

  useinstead="* to take matrix products").warn()
 [py.warnings]
Number of spikes: 1
Number of spikes: 1
Number of synapses: 1
Number of synapses: 1
libGL error: MESA-LOADER: failed to open radeonsi: /home/daniel/miniconda3/envs/HathwayGoodman/lib/python3.6/site-packages/matplotlib/../../../libstdc++.so.6: version `GLIBCXX_3.4.30' not found (required
libGL error: failed to load driver: radeonsi
libGL error: MESA-LOADER: failed to open swrast: /home/daniel/miniconda3/envs/HathwayGoodman/lib/python3.6/site-packages/matplotlib/../../../libstdc++.so.6: version `GLIBCXX_3.4.30' not found (required by
libGL error: failed to load driver: swrast
```

This finished in a few minutes. The command also opened 11 windows with figures, see screenshot:

From a visual inspection, the figures seem to match the ones from the ReScience reproduction paper, some of which having small visualisation details (missing x at data points, dot sizes of data points, missing annotations), some of which having slightly different graph curves in details (spikes etc.), but all of which match overall setup (scale, axes labels).

As the readme notes, the figures are also saved in `article/figures/` and are included in the check materials and embedded below.

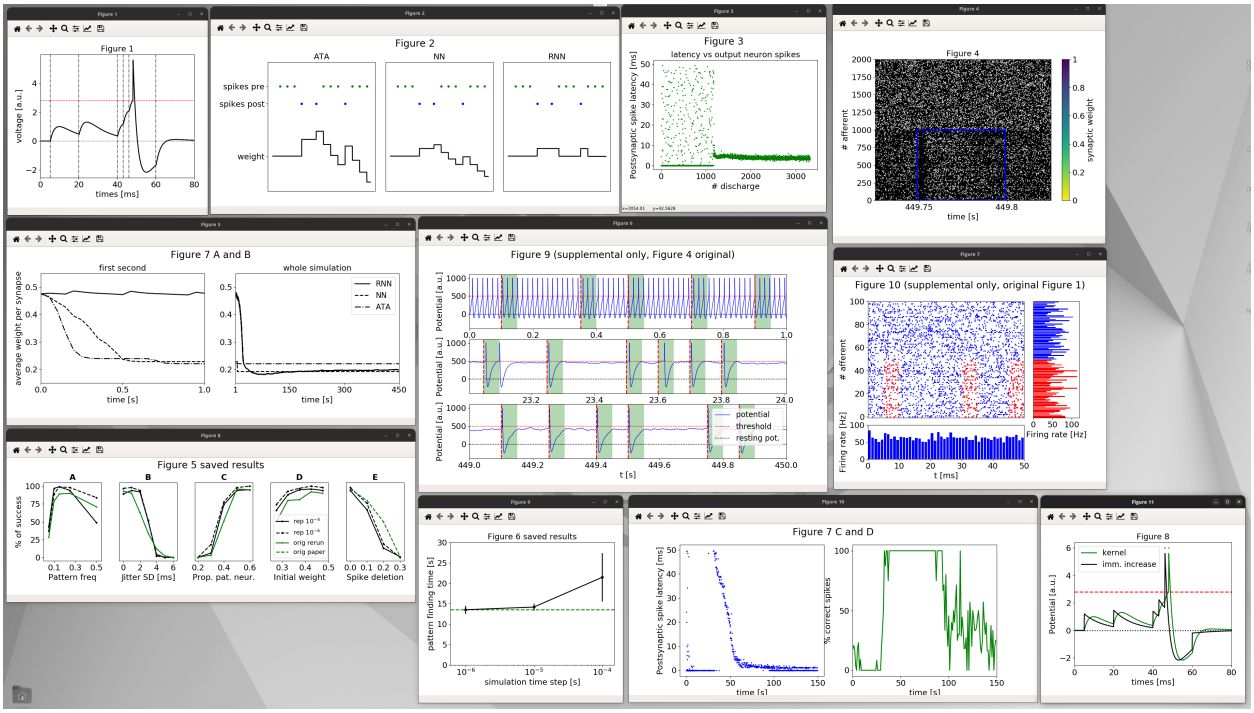


Figure 1: Screenshot of windows for figures 1 to 11

- `codecheck/figure_1_created_250519.pdf`
- `codecheck/figure_2_created_250519.pdf`
- `codecheck/figure_3_seed1_created_250519.pdf`
- `codecheck/figure_4_seed1_created_250519.pdf`
- `codecheck/figure_5_from_saved_created_250519.pdf`
- `codecheck/figure_6_from_saved_created_250519.pdf`
- `codecheck/figure_7AB_seed1_created_250519.pdf`
- `codecheck/figure_7CD_seed28_created_250519.pdf`
- `codecheck/figure_8_created_250519.pdf`
- `codecheck/figure_9sup_seed1_created_250519.pdf`
- `codecheck/figure_10sup_seed1_created_250519.pdf`

Because it was the end of my workday, I run the next command with "less repetitions" and see if it actually completes within 8-24 hours on my laptop:

```
python main.py --new True
```

At the next morning, the code completed with similar warnigns and errors as above, but also with a lot of log statements on the computations - see file `log-new-true.txt`. From what I cen tell from the file dates and file names, files for the figures 5, 6, 7CD, and 8 were newly created, all included in the check material and embedded below.

- `codecheck/figure_5_created_250520.pdf`
- `codecheck/figure_6_created_250520.pdf`
- `codecheck/figure_7CD_seed28_created_250520.pdf`
- `codecheck/figure_8_created_250520.pdf`

Figure 5 shows an additinal data line in each subfigure compared to "Figure 5 saved results" in the newly created version, which makes sense because the README points out a different parametrisation.

Figure 6 seems to have a smaller error bar (?) compared to "Figure 6 saved results" but largely matches the

figure created first.

Figure 7CD and 8 match precisely.

The documentation in the README on the details seems extensive and helpful for further investigations. However, with respect to this CODECHECK, I stop at this point of a successful reproduction of all figures in the article. It is noteworthy to point out that this community check was conduceted several years after the original workflow was created. The successful reproduction is a testament to the efforts of the authors as well as the contributors to the libraries and software infrastructure.

Recommendations

I suggest to the authors to consider the following suggestions for their next publication or workflow:

- Add a license (or several because of the data) to the repository to make it legally (re)usable

Manifest files

figure_1_created_250519.pdf

Comment: manuscript Figure 1 created in the first code run

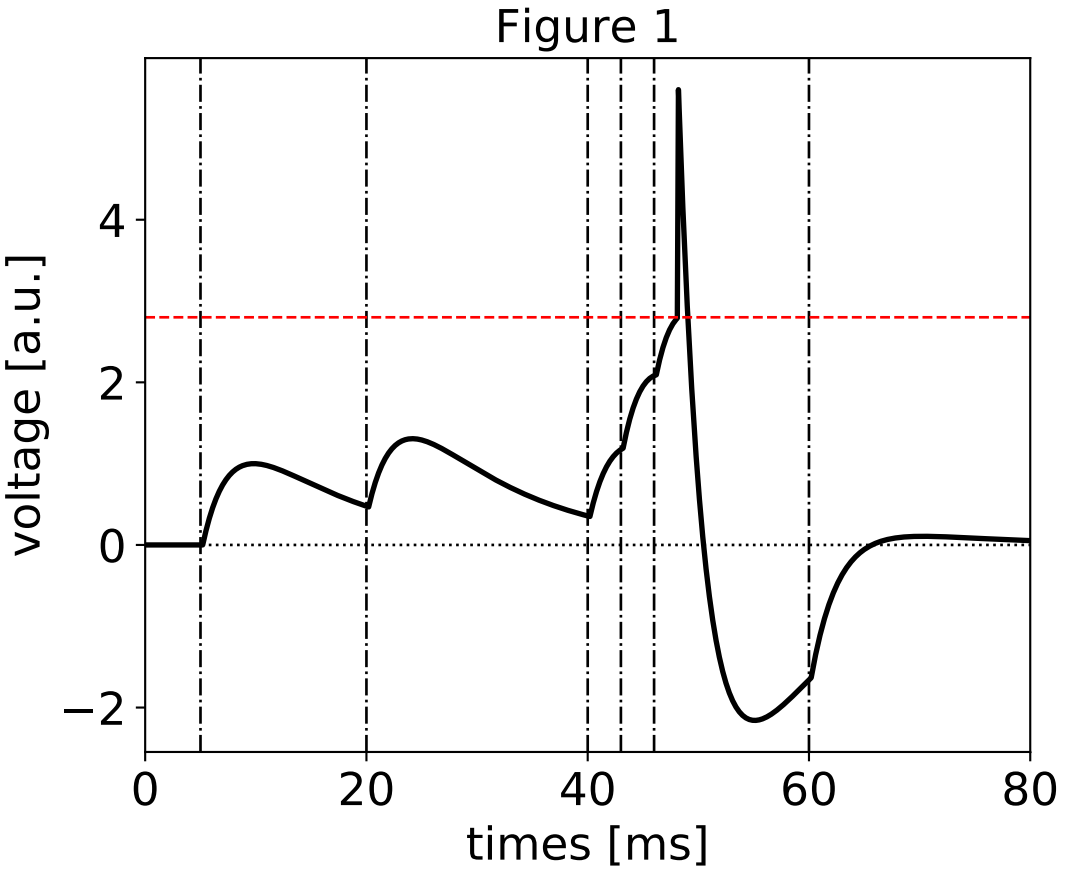


figure_2_created_250519.pdf

Comment: manuscript Figure 2 created in the first code run

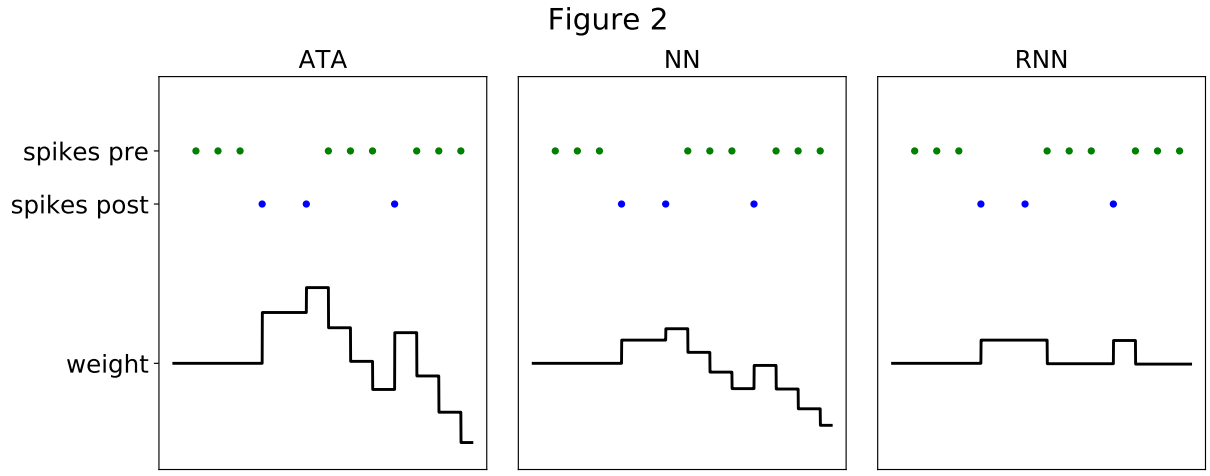


figure_3_seed1_created_250519.pdf

Comment: manuscript Figure 3 created in the first code run

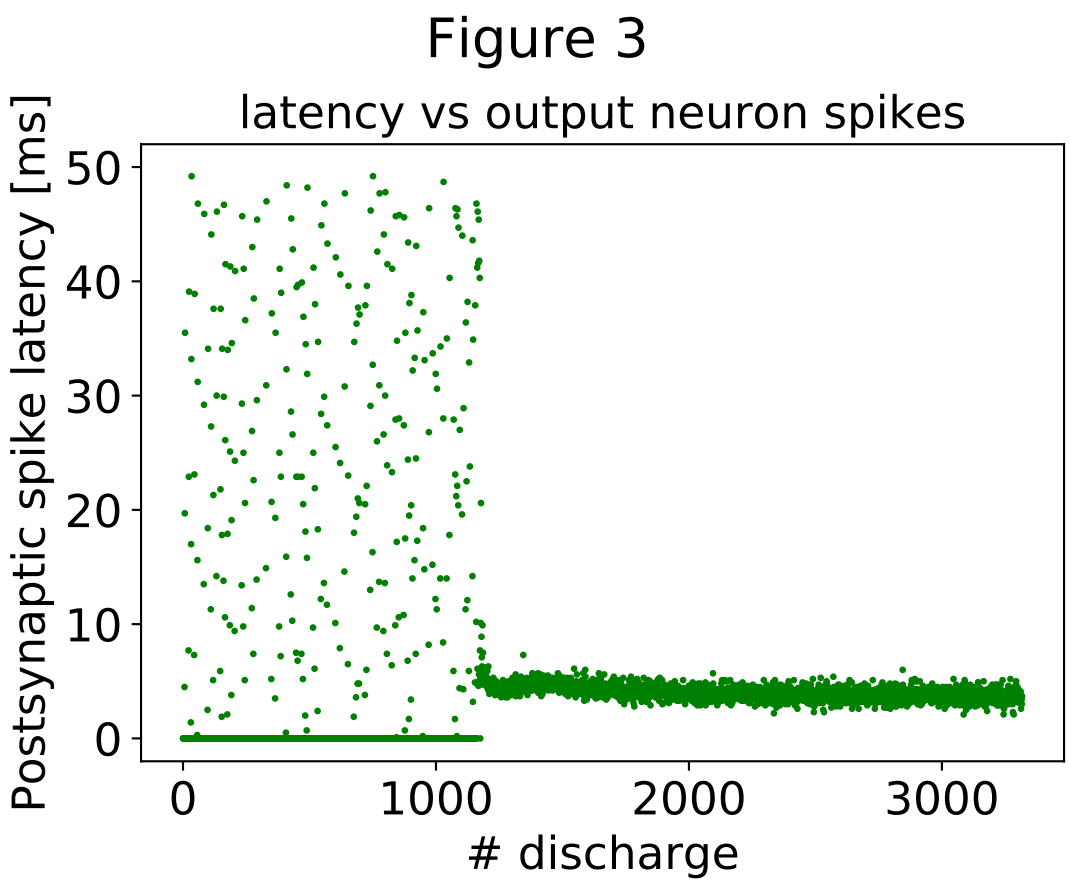


figure_4_seed1_created_250519.pdf

Comment: manuscript Figure 4 created in the first code run

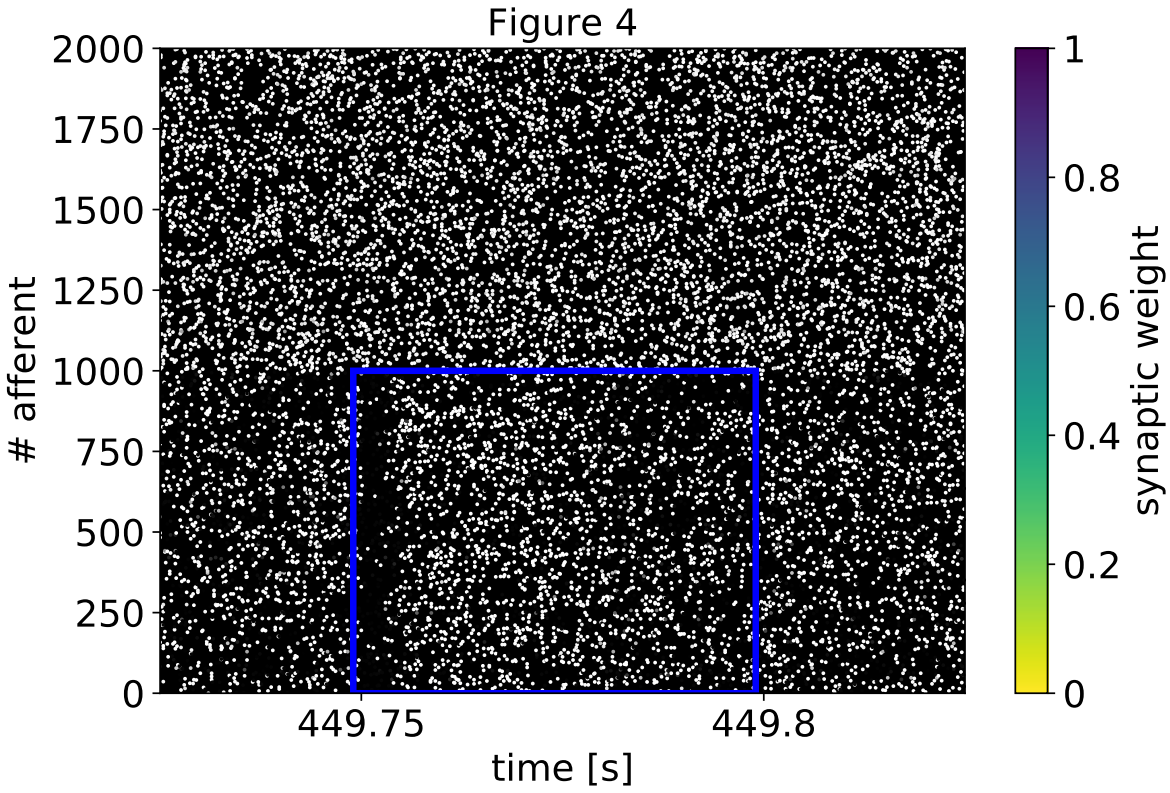


figure_5_from_saved_created_250519.pdf

Comment: manuscript Figure 5 created in the first code run based on saved data

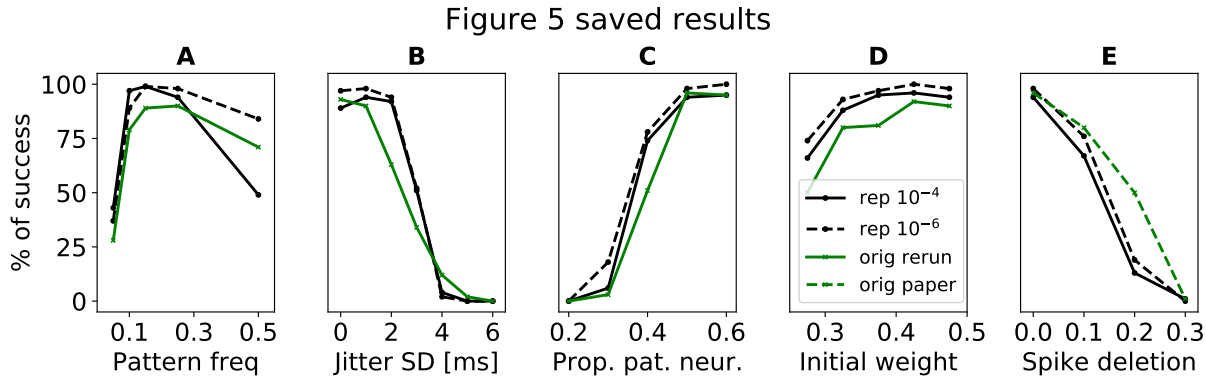


figure_6_from_saved_created_250519.pdf

Comment: manuscript Figure 6 created in the first code run based on saved data

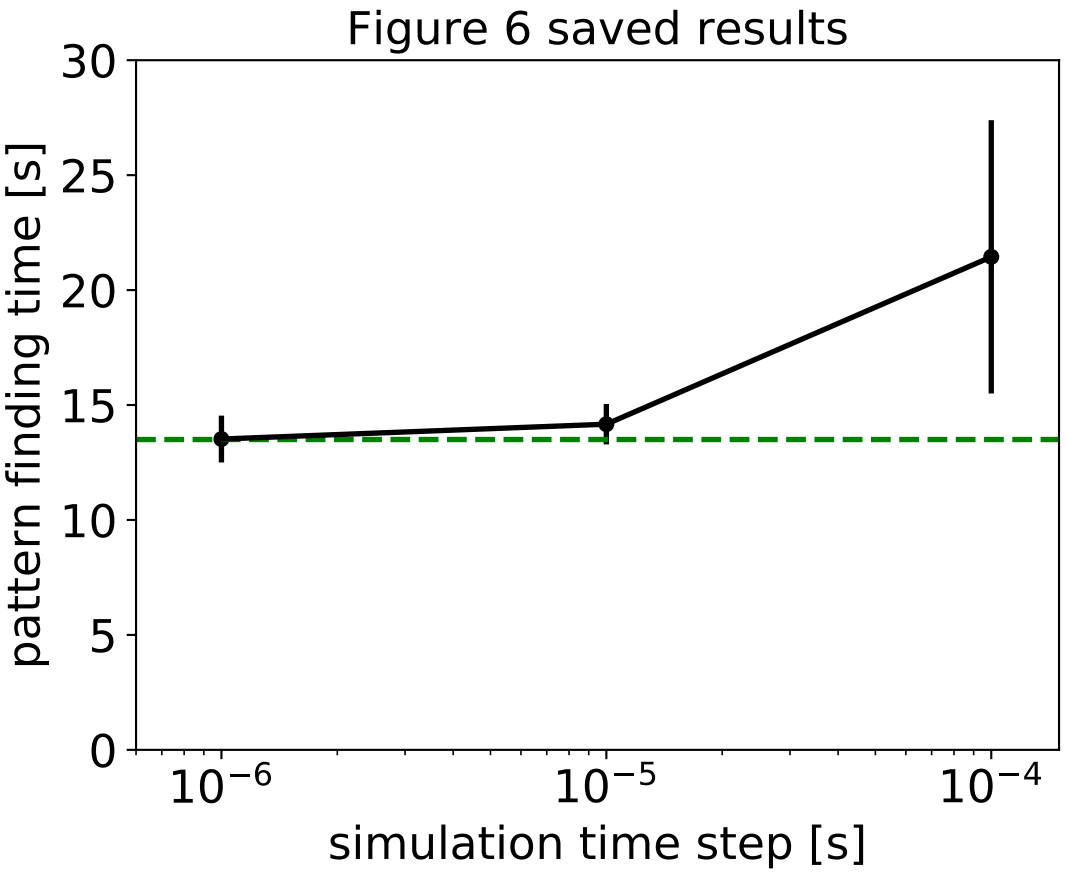


figure_7AB_seed1_created_250519.pdf

Comment: manuscript Figure 7AB created in the first code run

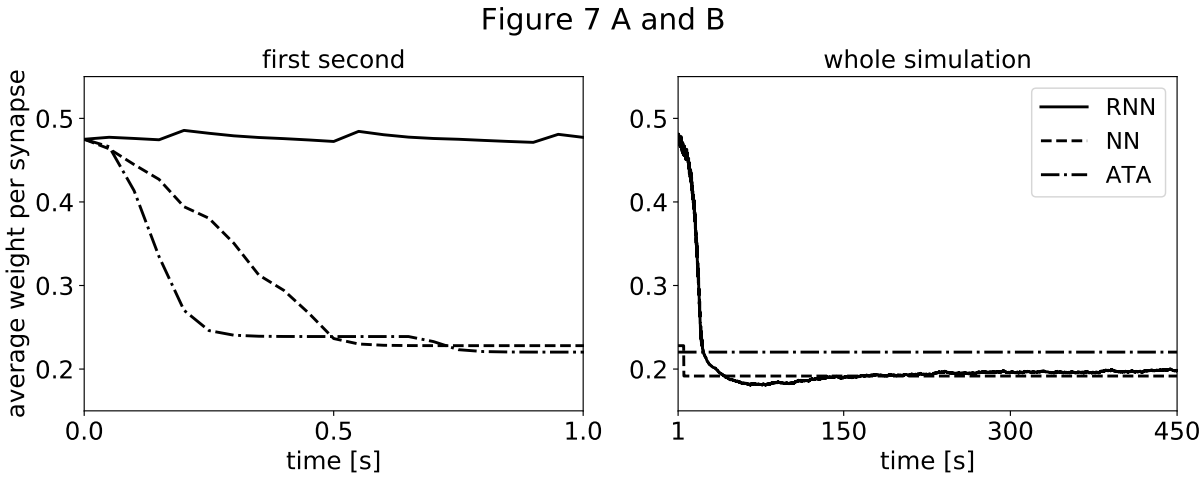


figure_7CD_seed28_created_250519.pdf

Comment: manuscript Figure 7CD created in the first code run

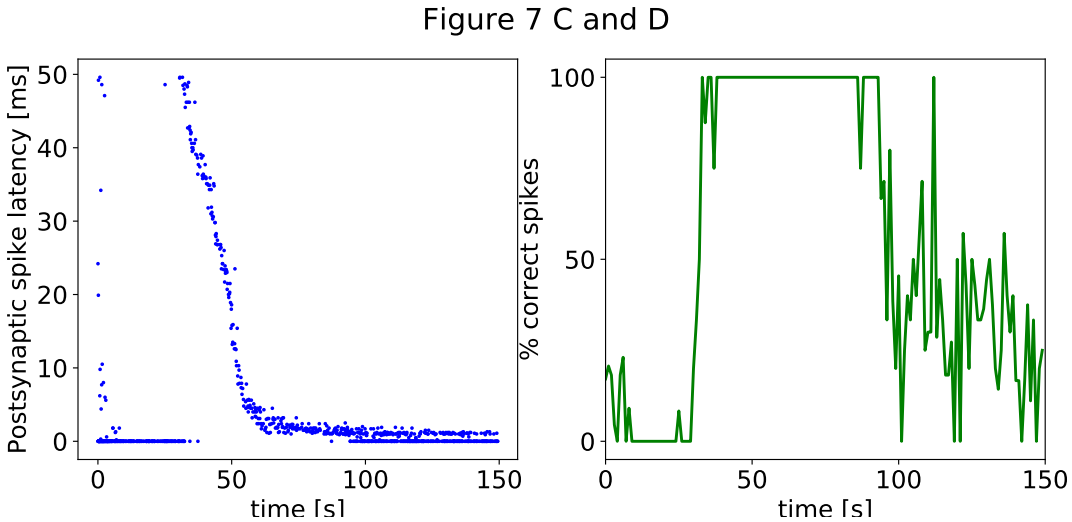


figure_8_created_250519.pdf

Comment: manuscript Figure 8 created in the first code run

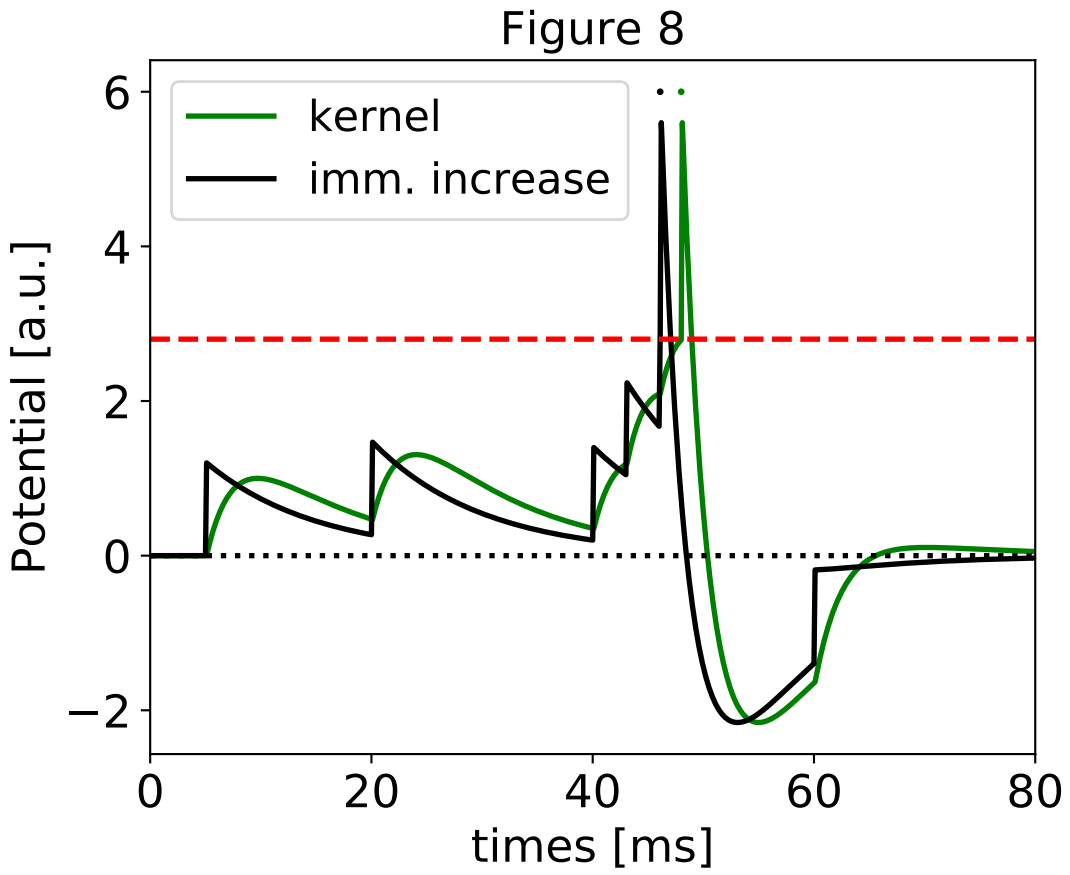


figure_9sup_seed1_created_250519.pdf

Comment: manuscript Figure 9 created in the first code run

Figure 9 (supplemental only, Figure 4 original)

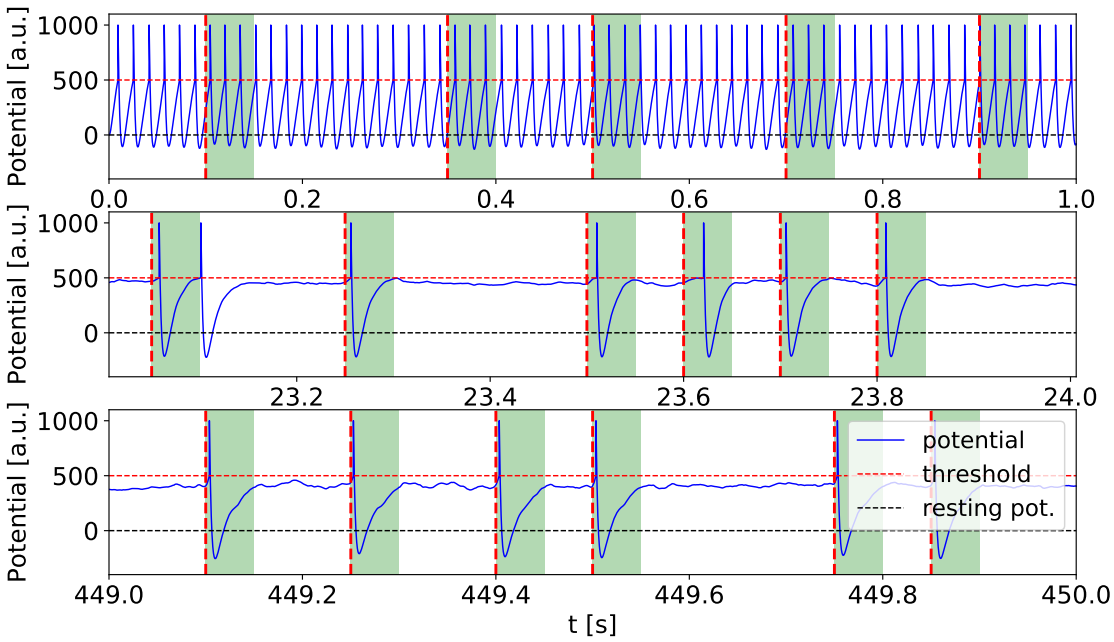


figure_10sup_seed1_created_250519.pdf

Comment: manuscript Figure 10 created in the first code run

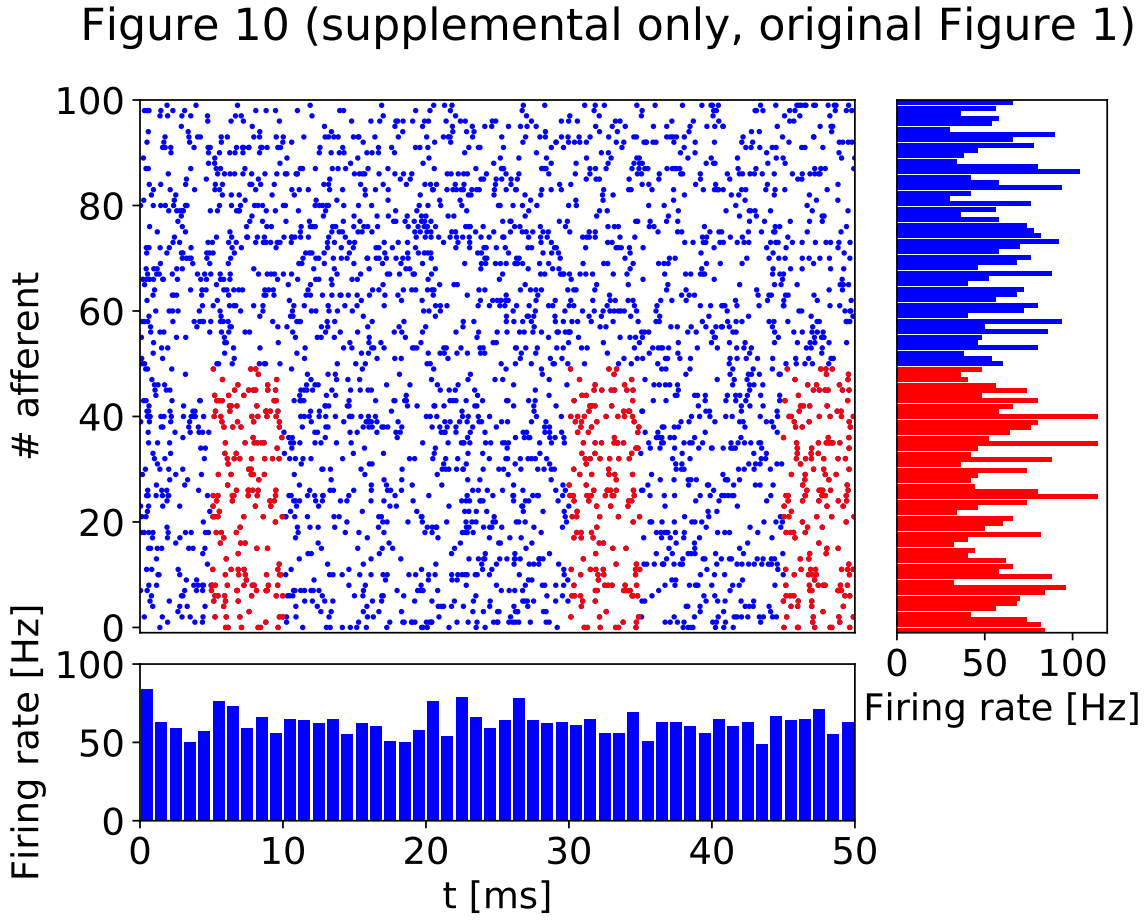


figure_5_created_250520.pdf

Comment: manuscript Figure 5 created in the second code run with new data

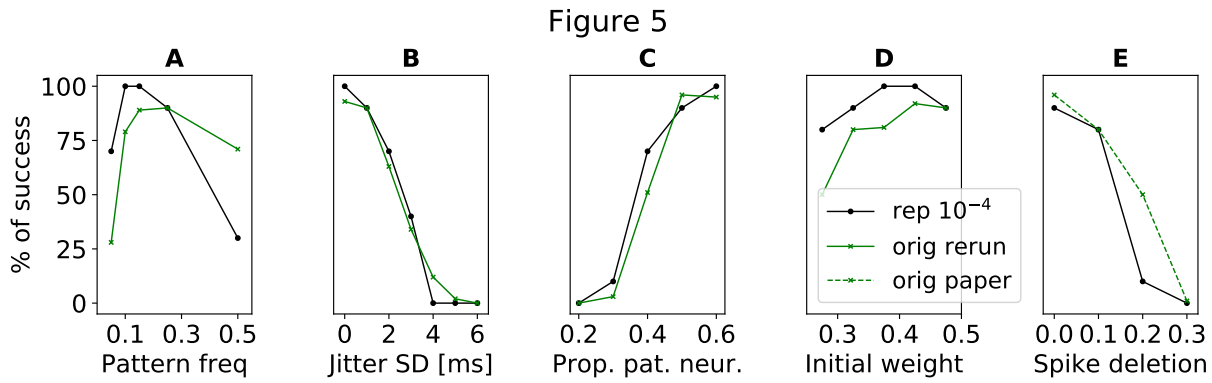


figure_6_created_250520.pdf

Comment: manuscript Figure 6 created in the second code run with new data

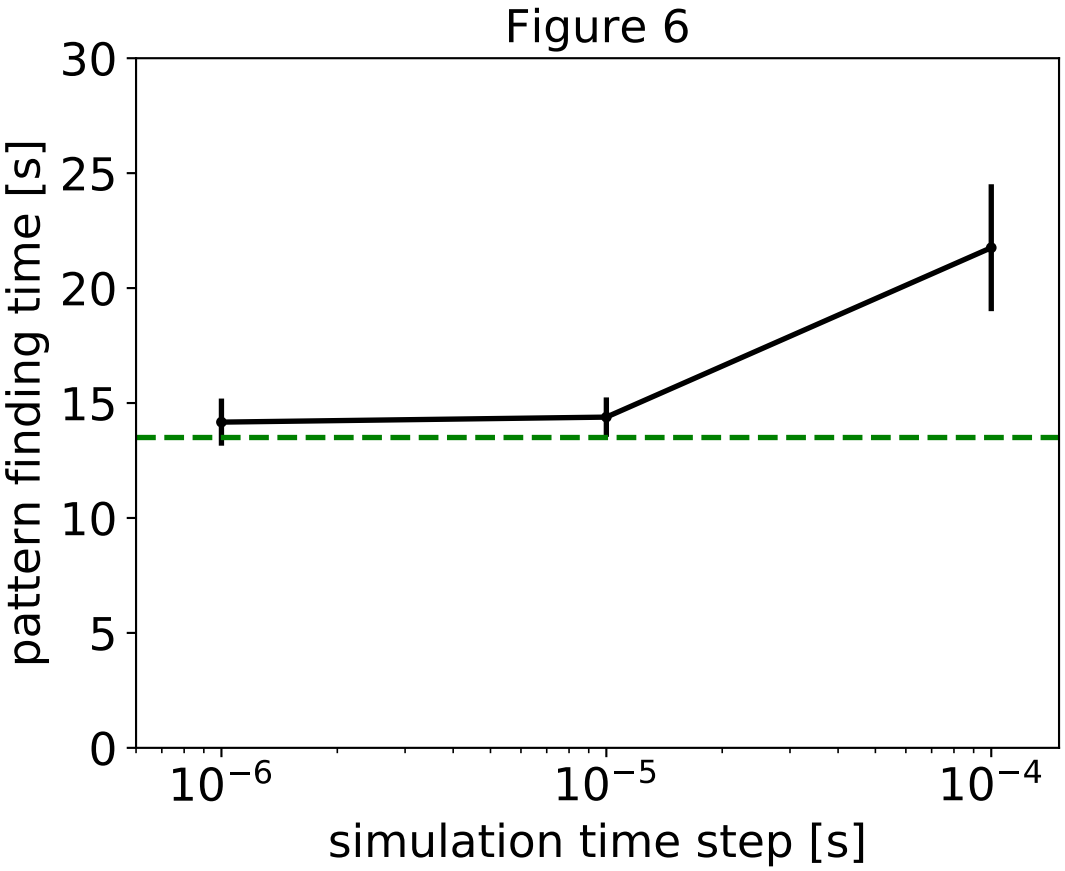


figure_7CD_seed28_created_250520.pdf

Comment: manuscript Figure 7CD created in the second code run

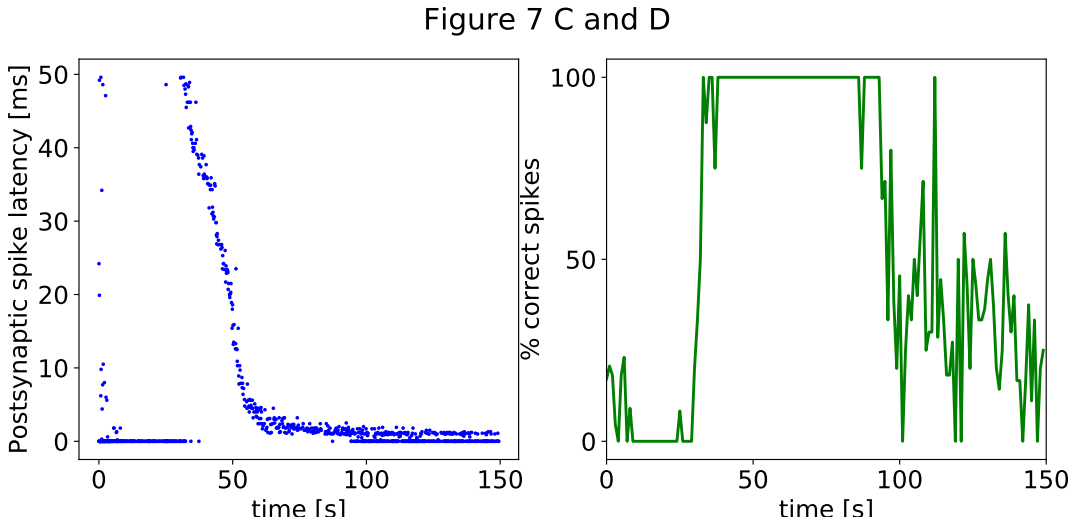
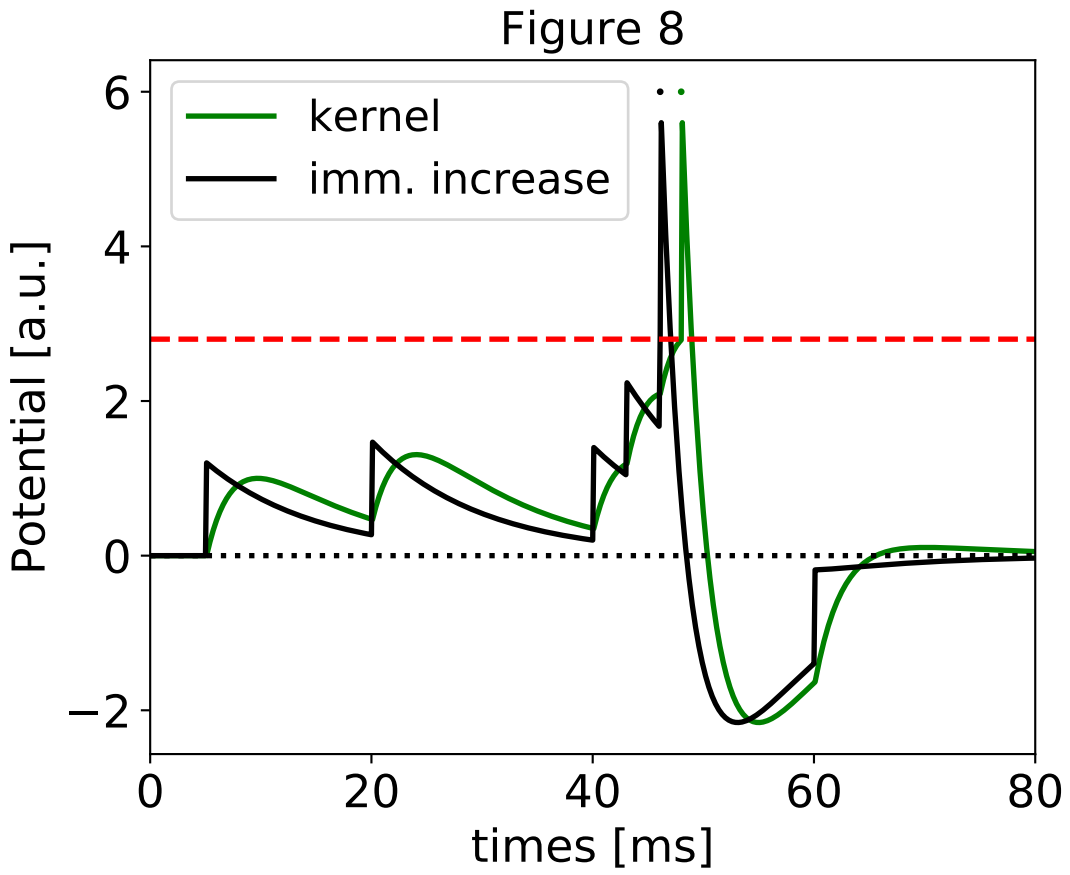


figure_8_created_250520.pdf

Comment: manuscript Figure 8 created in the second code run



Acknowledgements

I would like to thank Dr Bhatt and his team for promptly answering any queries I had with this reproduction. CODECHECK is financially supported by the Mozilla foundation.

Citing this document

Daniel Nüst, Stephen J. Eglen (2025). CODECHECK Certificate 2020-007. Zenodo. https://doi.org/10.5281/zenodo.15173962

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 available by 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 R Markdown using the `codecheck` R package. `make codecheck.pdf` will regenerate the report file.

```
sessionInfo()
```

```
## R version 4.5.0 (2025-04-11)
## Platform: x86_64-pc-linux-gnu
## Running under: Ubuntu 22.04.5 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.10.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.10.0  LAPACK version 3.10.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=de_DE.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=de_DE.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=de_DE.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=de_DE.UTF-8 LC_IDENTIFICATION=C
##
## time zone: Europe/Berlin
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats     graphics  grDevices datasets  utils
## [6] methods   base
##
## other attached packages:
##  [1] readr_2.1.5      tibble_3.2.1     xtable_1.8-4
##  [4] yaml_2.3.10      rprojroot_2.0.4  knitr_1.50
##  [7] codecheck_0.15.0 parsedate_1.3.2  R.cache_0.16.0
## [10] gh_1.4.1
##
## loaded via a namespace (and not attached):
```

```
##  [1] xfun_0.52         rdflib_0.2.9      bspm_0.5.7
##  [4] tzdb_0.5.0        vctrs_0.6.5       tools_4.5.0
##  [7] generics_0.1.3    parallel_4.5.0    curl_6.2.2
## [10] pkgconfig_2.0.3   pdftools_3.5.0    R.oo_1.27.0
## [13] redland_1.0.17-18 assertthat_0.2.1  lifecycle_1.0.4
## [16] git2r_0.36.2      compiler_4.5.0    atom4R_0.3-3
## [19] stringr_1.5.1     keyring_1.3.2     htmltools_0.5.8.1
## [22] crayon_1.5.3      pillar_1.10.2     whisker_0.4.1
## [25] tidyr_1.3.1       R.utils_2.13.0    cachem_1.1.0
## [28] zen4R_0.10        tidyselect_1.2.1  zip_2.3.2
## [31] digest_0.6.37     stringi_1.8.7     dplyr_1.1.4
## [34] purrr_1.0.4       fastmap_1.2.0     cli_3.6.5
## [37] magrittr_2.0.3    utf8_1.2.4        XML_3.99-0.18
## [40] crul_1.5.0        withr_3.0.2       osfr_0.2.9
## [43] bit64_4.6.0-1     roxygen2_7.3.2    rmarkdown_2.29
## [46] httr_1.4.7        bit_4.6.0         qpdf_1.3.5
## [49] askpass_1.2.1     R.methodsS3_1.8.2 hms_1.1.3
## [52] memoise_2.0.1     evaluate_1.0.3    rlang_1.1.6
## [55] Rcpp_1.0.14       glue_1.8.0        httpcode_0.3.0
## [58] xml2_1.3.8        fauxpas_0.5.2     rorcid_0.7.0
## [61] vroom_1.6.5       jsonlite_2.0.0    plyr_1.8.9
## [64] R6_2.6.1          fs_1.6.6
```