Instructions to generate figures from the article *Herfurth, Tim, and Tatjana Tchumatchenko. "Information transmission of mean and variance coding in integrate-and-fire neurons." Physical Review E 99.3 (2019): 032420.*

Here, we provide instructions on how the figures that are shown in the manuscript and supplemental information can be generated. We have included all necessary files in the folder "source_code". The general procedure involves two steps: first, the data for all sampled parameters are created. Second, these data are used to create the respective figures. We used Python 2.7 and jupyter-notebooks for final figure generation.

1 Data generation

The scripts for data generation are to be found in the subfolder "generate_data" and have to be run first before the figures can be created. All scripts automatically save the results as compressed numpy arrays.

1.1 generateAnalyticData.py

This script generates the analytic data from linear response theory (only LIF). It can be run (after changing to "generate_data") by

```
python generateAnalyticData.py
```

The variables defined in the script can be changed as desired and are:

`mu`	constant current μ
`snr`	list of values of signal strength σ_s to be sampled
`sigN_a`	lists of values of noise strength σ_n
`tau_a`	list of values of signal correlation time τ_s
`w0_a`	list of values of central signal frequency Ω_0
`n`	the number of evaluated frequencies ω between 0 and `omega_end`

1.2 generateSimulationData.py

This script generates the simulation data. It can be run (after changing to "generate_data") by

```
python generateSimulationData.py
```

The variables defined in the script can be changed as desired and are:

`snr`	list of values of σ_s to be sampled
`tau_s`	list of values of τ_s
`w0_s`	list of values of Ω_0
`mu`	list of values of μ (separate for LIF and EIF)
`tau_n`	list of values of τ_n corresponding to each μ
`sigN_s`	list of lists of values of σ_n corresponding to each μ
`n`	the number of trials to be generated on each core (we used 32 cores)

1.3 generateGaussianityTestData.py

This script generates the data and Gaussianity test results as shown in the supplemental material. It can be run (after changing to "generate_data") by

```
python generateGaussianityTestData.py
```

The variables defined in the script can be changed as desired and are:

`snr`	value of σ_s to be tested
`tau_s`	value of τ_s
`w0_s`	value of Ω_0
`mu`	value of μ
`tau_n`	value of τ_n
`sigN_s`	value of σ_n
`n`	the number of trials to be generated on each core (we used 32 cores)
`lim1`	upper limit of range of frequencies from which frequencies are randomly drawn
`lim2`	the number of frequencies that are included in the test

1.4 generateMultivariateGaussianityTestData.py

This script generates the data and multivariate Gaussianity test results as shown in the supplemental material. It can be run (after changing to "generate_data") by

```
python generateMultivariateGaussianityTestData.py
```

The variables defined in the script can be changed as desired and are:

`snr`	value of σ_s to be tested
`tau_s`	value of τ_s
`w0_s`	value of Ω_0
`mu`	value of μ
`tau_n`	value of τ_n
`sigN_s`	value of σ_n
`n`	the number of trials to be generated on each core (we used 32 cores)
`lim1`	the range of frequencies from which frequencies are randomly drawn
`lim2`	the number of frequencies that are included in the test

1.5 generateFiringStatsData.py

This script generates the data that are used to determine the steady state firing statistics (no signal) as shown in the supplemental material. It can be run (after changing to "generate_data") by

```
python generateFiringStatsData.py
```

The variables defined in the script can be changed as desired and are:

`tau_s`	value of τ_s
`w0_s`	value of Ω_0
`mu`	value of μ
`tau_n`	value of τ_n
`sigN_s`	value of σ_n
`n`	the number of trials to be generated on each core (we used 32 cores)

2 Figure generation

The folder "generate_figures" contains IPython notebooks in which the figures shown in the manuscript and supplemental material can be created. The notebook `figs78.ipynb` provides figures 7 and 8. In the notebook `figsS4-S5-S10.ipynb` figures S4, S5, and S10 can be created, and `figsS6-S9.ipynb` generates figures S6–S9. All other figures are generated from `figs.ipynb`.
The notebooks should be run from top to bottom in order to avoid confusing data for LIF and EIF. The function of the cells in the notebook is indicated by the section names they belong to.
The notebooks are designed to work for the data as generated for the manuscript. For changes in parameter ranges smaller modifications in the notebook may be necessary.