



Citation: Metzner, C., Mäki-Marttunen, T., Zurowski, B., & Steuber, V. (2018). Modules for automated validation and comparison of models of neurophysiological and neurocognitive biomarkers of psychiatric disorders: ASSRUnit—A case study. *Computational Psychiatry*, 2, 74–91. https://doi.org/10.1162/cpsy_a_00015

DOI:
https://doi.org/10.1162/cpsy_a_00015

Received: 18 December 2017
Accepted: 1 April 2018

Competing Interests: The authors declare no conflict of interest.

Corresponding Author:
Christoph Metzner
c.metzner@herts.ac.uk

Copyright: © 2018
Massachusetts Institute of Technology
Published under a Creative Commons
Attribution 4.0 International
(CC BY 4.0) license



RESEARCH

Modules for Automated Validation and Comparison of Models of Neurophysiological and Neurocognitive Biomarkers of Psychiatric Disorders: ASSRUnit—A Case Study

Christoph Metzner¹, Tuomo Mäki-Marttunen², Bartosz Zurowski³, and Volker Steuber¹

¹Centre for Computer Science and Informatics Research, University of Hertfordshire, Hatfield, UK

²Simula Research Laboratory and Center for Cardiological Innovation, Oslo, Norway

³Center for Integrative Psychiatry, University of Lübeck, Lübeck, Germany

Keywords: biomarkers, endophenotypes, computational models, auditory steady-state responses, psychiatric disorders, schizophrenia

ABSTRACT

The characterization of biomarkers has been a central goal of research in psychiatry over the last years. While most of this research has focused on the identification of biomarkers, using various experimental approaches, it has been recognized that their instantiations, through computational models, have great potential to help us understand and interpret these experimental results. However, the enormous increase in available neurophysiological and neurocognitive as well as computational data also poses new challenges. How can a researcher stay on top of the experimental literature? How can computational modeling data be efficiently compared to experimental data? How can computational modeling most effectively inform experimentalists? Recently, a general scientific framework for the generation of executable tests that automatically compare model results to experimental observations, SciUnit, has been proposed. Here we exploit this framework for research in psychiatry to address the challenges mentioned. We extend the SciUnit framework by adding an experimental database, which contains a comprehensive collection of relevant experimental observations, and a prediction database, which contains a collection of predictions generated by computational models. Together with appropriately designed SciUnit tests and methods to mine and visualize the databases, model data, and test results, this extended framework has the potential to greatly facilitate the use of computational models in psychiatry. As an initial example, we present ASSRUnit, a module for auditory steady-state response deficits in psychiatric disorders.

INTRODUCTION

Psychiatric nosology, for centuries widely untouched by findings from clinical neuroscience, is at the beginning of a transformation process (Friston, Redish, & Gordon, 2017) toward an interactive evolution of diagnostic and biological categories. This change of focus stems from the hope that biomarkers and endophenotypic measures show a better correspondence with genetic alterations identified by large genome-wide association studies (Meyer-Lindenberg &

Weinberger, 2006) and promises to more readily shed light on the mechanisms underlying these disorders and to facilitate the discovery of novel therapeutic interventions (Siekmeier, 2015). Naturally, much effort has been put into translating these measures into practice using human studies (Perlis, 2011) and animal models (Markou, Chiamulera, Geyer, Tricklebank, & Steckler, 2009).

Computational approaches also have gained significantly more attention over the last years, and this has led to the emergence of computational psychiatry as a novel multidisciplinary and integrative discipline (see, e.g., Adams, Huys, & Roiser, 2016; Corlett & Fletcher, 2014; Friston, Stephan, Montague, & Dolan, 2014; Montague, Dolan, Friston, & Dayan, 2012; Stephan & Mathys, 2014; Wang & Krystal, 2014). This emergence can be attributed to three main factors: First, the earlier mentioned increase in experimental studies has provided a wealth of neuroscientific (including neurochemical, molecular, anatomic, and neurophysiological) data that are essential to building computational models; second, methodological and infrastructural advances, such as the various atlases, databases, and online tools from the Allen Brain Institute (<http://brain-map.org/>) or the BRAIN initiative (<https://www.braininitiative.nih.gov/>), have made it possible to analyze and process this enormous amount of data; third, the increase in computing power of high-performance computers as well as standard personal computers has made it possible (and affordable) to build and use models of increasingly high computational complexity. Therefore, the rapid growth of the field of computational psychiatry comes as no surprise. However, to fully exploit the potential that computational modeling offers, we have to identify systemic weaknesses in current approaches and take a look at other disciplines that use computational models (and have used them for much longer than psychiatry) and even look at disciplines, such as software development, that face similar challenges.

At the core of computational modeling lies the concept of validation, that is, the rigorous comparison of model predictions against experimental findings. Furthermore, for a model to be useful and provide a true contribution to knowledge, the validation has to use sound criteria and the experimental observations need to sufficiently characterize the phenomenon the model tries to reproduce. Hence, to develop a computational model, scientists need to have an in-depth understanding of the current, relevant experimental data; the current state of computational modeling in the given area; and the state of the art of statistical testing to choose the appropriate criteria with which the model predictions and experimental observations will be compared (Gerkin & Omar, 2013; Sarma et al., 2016). In a field where the number of both experimental and computational studies grows rapidly, as is the case for psychiatry, this becomes more and more impracticable. Furthermore, the increase in the number of modeling and experimental studies has made it harder for reviewers to judge not only whether a new model adequately replicates the full range of experimental observations but also how it compares to competing models. Again, reviewers need an in-depth knowledge of the modeling and experimental literature as well as profound statistical knowledge. Finally, because computational modeling aims to generate predictions that can be experimentally tested, experimental neuroscientists must be able to extract and assess predictions from a rapidly growing body of computational models, a task that is becoming more and more impracticable.

The problems described herein are not unique to the field of computational psychiatry but occur in all scientific areas that use computational models. Furthermore, building a computational model is in the end a software development project of sorts. Omar, Aldrich, & Gerkin (2014) have therefore proposed a framework for automated validation of scientific models, SciUnit, which is based on unit testing, a technique commonly used in software development. SciUnit addresses the problems mentioned earlier by making the scope (i.e., the set

of observable quantities about which it can generate predictions) of the model explicit and by allowing its validity (i.e., the extent to which its predictions agree with available experimental observations of those quantities) to be automatically tested (Omar et al., 2014).

In this article, we propose to adopt this framework for the computational psychiatry community and to collaboratively build common repositories of computational models, tests, test suites, and tools. As a case in point, we have implemented a Python module (ASSRUnit) for auditory steady-state response (ASSR) deficits in schizophrenic patients, which are based on observations from several experimental studies (Krishnan et al., 2009; Kwon et al., 1999; Vierling-Claassen, Siekmeier, Stufflebeam, & Kopell, 2008), and we demonstrate how existing computational models (Beeman, 2013; Metzner, 2017; Metzner, Schweikard, & Zurowski, 2016; Vierling-Claassen et al., 2008) can be validated against these observations and compared with each other.

THE SciUnit FRAMEWORK

The module we present here is based on the general SciUnit framework for the validation of scientific models against experimental observations (Omar et al., 2014; see Figure 1).

In SciUnit, models declare and implement so-called capabilities, which the validation tests then use to interact with those models. By a *capability* of the model, we mean the ability of the model to describe certain biological phenomena that are possible to assess using physical quantities. Furthermore, the declaration and implementation of capabilities are separated,

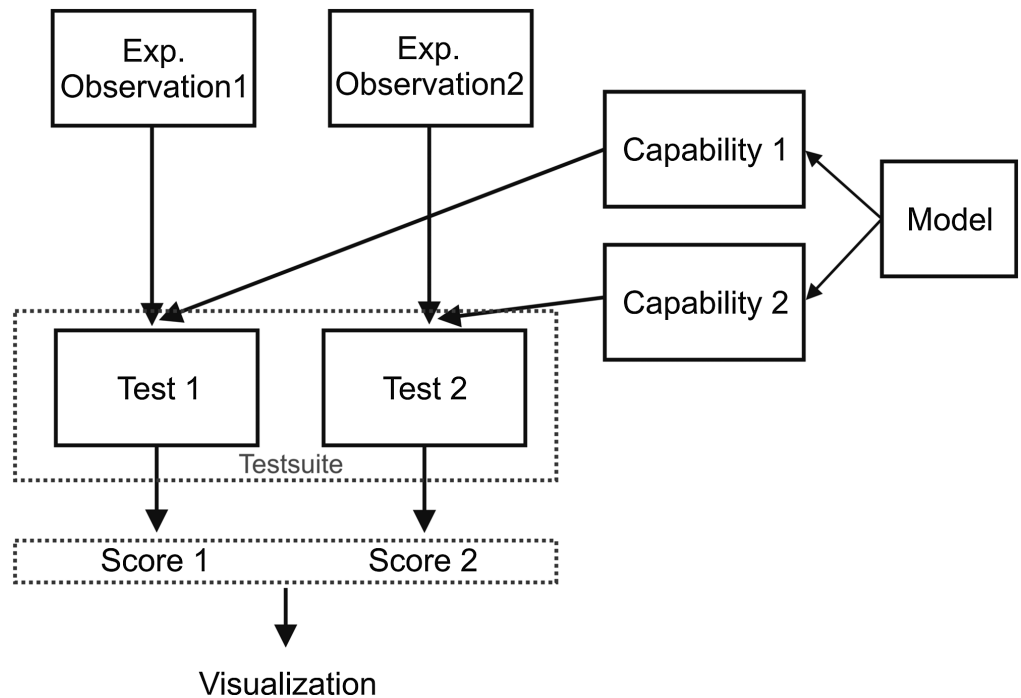


Figure 1. Schematic of the SciUnit framework. Models can be tested against experimental observations using specific tests. These tests incorporate an experimental observation and interface with the model through capabilities. Tests can be grouped into so-called test suites. The execution of a test produces a score, which describes how well the model captures the experimental observations. SciUnit also provides methods to visualize the resulting score(s), for example, in a table.

which allows for testing two different models that share the *same capabilities* on the *same experimental* observations using the *same test*. Tests then take the model, use its capabilities to generate data, compare these data to the experimental observations that are linked to the test, and create a score. This score, which can simply be a Boolean (pass/fail) or another more complex score type, describes if and to what extent the model data and the experimental observation(s) match.

Before we describe the actual implementations of capabilities, models, tests, and scores in our framework for ASSRs in schizophrenia, we first start with a summary of the experimental observations we included in the database, and then we describe the computational models that were realized.

THE ASSRUnit MODULE

The structure of the ASSRUnit module proposed here is shown schematically in Figure 2. As outlined earlier, the proposed module aims to provide three main functionalities: (a) a simple way of getting an overview of the experimental literature, (b) an easy and flexible way to automatically test computational models against experimental observations, and (c) an automated way of generating predictions from computational models. Functionality a is fully covered by the experimental database and its methods to query the database and visualize the results. Functionality b is provided by linking both the experimental database as well as the computational models to the SciUnit tests that cover the relevant experimental observations. The only action required from the user is, if the computational model has not yet been included in the model repository of the module, to provide an interfacing Python class

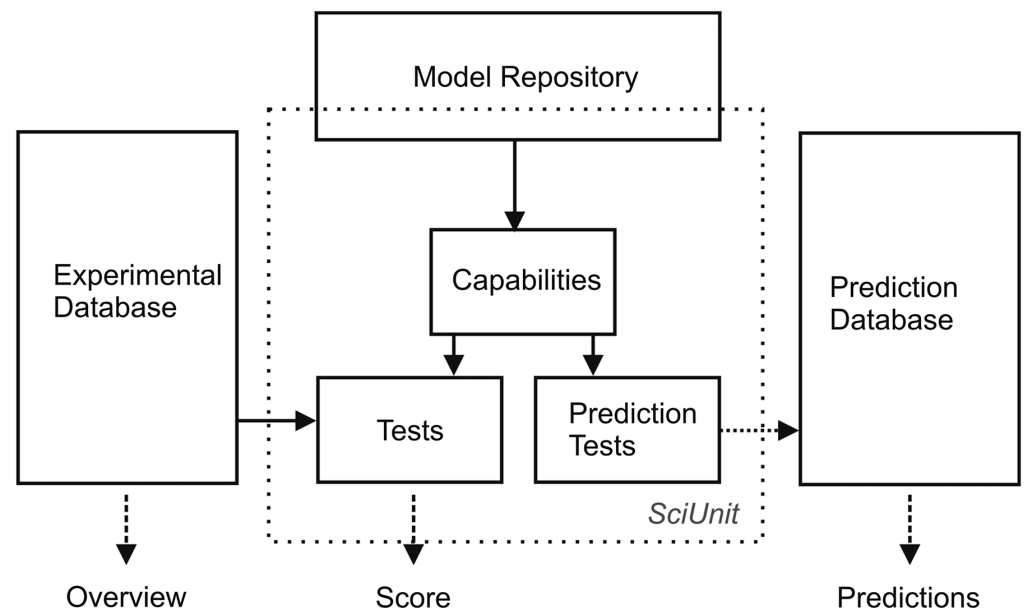


Figure 2. Schematic of the proposed framework highlighting the three main functions: (a) overview of experimental observations; (b) validation of computational models; (c) creation of a predictions database. At its core lies the SciUnit module, which provides the infrastructure for the automated validation of the computational models. In particular, through a set of suitable tests, the computational models can be compared against experimental observations queried from the experimental database. Another set of tests, the so-called prediction tests, are then employed to extract predictions from the computational models, thus populating the predictions database.

(i.e., a class that allows the original model to be run and analyzed from within Python) for the model that implements all the required capabilities. Note that the model itself does not have to be written in Python; it only has to be executable from a shell. Once the model is included, the SciUnit framework allows for automated testing, and the visualization methods provided in the proposed module allow for a comprehensive and clear presentation of the results. Functionality *c* can be achieved by a set of SciUnit tests and capabilities that, instead of covering experimental observations, cover experiments that have not yet been performed. By running the computational models with these tests, the module can be used to generate new predictions from the models, which can then be used to populate a prediction database similar to the experimental database. The module is available on GitHub (<https://github.com/ChristophMetzner/ASSRUnit/tree/CompPsychArticle>).

Experimental Observations Database

In patients suffering from schizophrenia, oscillatory deficits in general and ASSR deficits in particular have been extensively studied using electroencephalography (EEG) and magnetoencephalography (MEG; e.g., Brenner, Sporns, Lysaker, & O'Donnell, 2003; Hamm et al., 2015; Krishnan et al., 2009; Kwon et al., 1999; Light et al., 2006; Mulert, Kirsch, Pascual-Marqui, McCarley, & Spencer, 2011; O'Connell et al., 2015; Spencer, 2012; Spencer, Niznikiewicz, Nestor, Shenton, & McCarley, 2009; Spencer, Salisbury, Shenton, & McCarley, 2008; Vierling-Claassen et al., 2008; Zhang, Ma, Li, Yang, & Qin, 2016).

Neural oscillations have been hypothesized to subservise important functions in the brain and are critically involved in cognitive processes (see, e.g., the review of Başar, Başar-Eroglu, Karakaş, & Schürmann, 2001). Gamma oscillations, for example, have been demonstrated to underlie the formation of coherent percepts in different sensory modalities (e.g., Engel, Kreiter, König, & Singer, 1991; Jokeit & Makeig, 1994). Interestingly, abnormal power and synchrony in the gamma band have been found in schizophrenic patients in a number of different tasks and paradigms (e.g., Cho, Konecky, & Carter, 2006; Kwon et al., 1999; Spencer et al., 2003) and linked to the schizophrenic symptom profile (e.g., Gordon, Williams, Haig, Wright, & Mearns, 2001; K.-H. Lee, Williams, Haig, & Gordon, 2003). Although deficits in the generation and maintenance of gamma oscillations are not a classical symptom of schizophrenia, given the importance of gamma oscillations in sensory processing and cognition and the link between deficits and symptoms, these biomarkers might reflect a characteristic trait of the disorder.

Here we focus on three of these studies, looking at entrainment deficits in the gamma and beta ranges. Kwon et al. (1999) used a click train paradigm to study ASSRs at 20, 30, and 40 Hz in schizophrenic patients using EEG and found a prominent reduction of power at the driving frequency for 40 Hz drive but no changes of power at the driving frequency for 30 Hz and 20 Hz. Although Figure 3 in Kwon et al. (1999) seems to show an increase of the subharmonic 20 Hz component for 40 Hz drive, no statistical comparison is presented in the article. Vierling-Claassen et al. (2008) reproduced this reduction of power at the driving frequency for 40 Hz drive using the same paradigm with MEG. Additionally, they found an increase in power at the driving frequency during 20 Hz drive and changes of power at certain harmonic/subharmonic frequencies, namely, an increase of power at 20 Hz for 40 Hz drive and a decrease of power at 40 Hz for 20 Hz drive. Krishnan et al. (2009) used a slightly different paradigm, which employed amplitude-modulated tones instead of click trains, and tested a wide range of driving frequencies from 5 to 50 Hz. They found reduction of power at the driving frequency in the gamma range (i.e., at 40, 45, and 50 Hz) and no changes at other frequencies. Furthermore, they did not find any changes of power at harmonic or subharmonic frequencies.

The experimental database is realized as a nested Python dictionary, with an entry for each study included (a dictionary is a special data structure in which you can access data or values by a key[word]; in a nested dictionary, the values themselves can be dictionaries). Each study entry consists of two entries (i.e., two value-key pairs), which describe the study observations, one in a quantitative way and the other in a qualitative way. We have included the qualitative description because often either computational models do not allow for a strict quantitative comparison with experimental data or publications of experimental studies do not provide enough detail on the results, and in these cases, only a qualitative comparison is possible.

Together with the database, ASSRUnit provides basic methods to query and visualize the content of the database. These methods include commands to retrieve all studies or observations in the database and a method to display an overview of the results for the whole database or for certain studies or observations. Finally, the metadata associated with each study (e.g., the number of participants, the modality, the patient group) can also be retrieved and displayed.

Prediction Database

The prediction database is also implemented as a nested Python dictionary. Similar to the experimental observation database, methods that retrieve and visualize the content of the database are included in ASSRUnit.

Models, Capabilities, Tests, and More

Models To demonstrate the flexibility of the proposed framework, we included three different neural models of ASSR deficits.

The first model is based on a biophysically detailed model of primary auditory cortex by Beeman (2013). Our group has recently used it to study ASSR deficits (Metzner et al., 2016). The model was implemented using the neural simulator GENESIS (Bower, 1992; Bower & Beeman, 1993). Not only is this model a good example of a biophysically detailed model of ASSR deficits but its inclusion also demonstrates how models that are not written in Python can be used.

The second model is a reimplementations of the model of Beeman in NeuroML2, a simulator-independent markup language to describe neural network models developed by the NeuroML project (Cannon et al., 2014), which is featured in the open source brain model database (Gleeson et al., 2012). We included this model to demonstrate the ability of the proposed framework to incorporate state-of-the-art tools and databases for the design, implementation, and simulation of network models.

The last model we included is the simple model presented by Vierling-Claassen et al. (2008). The model is a simple network of two populations of theta neurons. The theta neuron model is a simple oscillator, where a single variable θ describes the phase angle of a point traveling around the unit circle. A detailed description of the model and its usefulness in the study of neural oscillations can be found in Börgers & Kopell (2005). We reimplemented the model in Python (for more details on the model and the replication, see Metzner, 2017). The model was included first of all to demonstrate that the framework is not limited to biophysically detailed models but can also be used with simpler, more abstract models. Additionally, the inclusion of the model demonstrates the simplest way of including a model, implementing the model in Python. This might not be the most common scenario, but because it is the simplest, we included it here.

We do not discuss the models in more detail here, because they have been described elsewhere (Beeman, 2013; Metzner, 2017; Metzner et al., 2016; Vierling-Claassen et al., 2008). Furthermore, our focus lies on the framework with which to use, validate, and compare models, not on the models themselves.

The three models mentioned herein are included in the SciUnit framework by wrapper classes (i.e., a class that encapsulates the functionality of the original model but is implemented in Python) that implement the necessary capabilities and make the models available to the tests. One important thing to note here is that, because we are dealing with models of neurofunctional deficits found in individuals with a particular disorder, a *model* as used in the module always means two configurations of a computational model, one representing the control configuration and one the disorder configuration. Therefore all wrapper classes take two sets of parameters as an argument describing the necessary parameters for the two configurations, respectively.

In addition to the standard model classes, we also implemented a second version of the model classes, which can simulate a certain number n of *subjects* and a certain number m of *trials* (realized by their `..._plus` methods). This allows for assessing the robustness of the results and can contribute in a major way to statistical rigor. The way in which these subjects and trials are implemented strongly depends on the model and its complexity. For example, for the simple model from Vierling-Claassen et al. (2008), which has all-to-all connectivity for all possible connection types, it is not possible to simulate different subjects (so $n = 1$), but different trials are simulated by changing the seed for the random number generator (RNG) that generates the background noise. In the case of the model from Beeman (2013), different subjects are realized by a change in seed for the RNG that is responsible for the formation of individual connections. This leads to each subject having a different connectivity on the level of individual connections, however, while preserving the connection probabilities for each connection type. Furthermore, by also changing the RNG seed that generates the background noise, several different trials for each subject can be realized.

Capabilities Table 1 summarizes the experimental observations included in the module at this stage. All observations are similar in nature: the power value of the EEG/MEG at a certain frequency in response to auditory entrainment at a certain frequency. Therefore the only capability necessary for a model to produce output that can be compared to these observations is a method that produces the power at a certain frequency X of a simulated EEG/MEG signal in response to drive at a frequency Y . This capability, Produce XY , is included in ASSRUnit, and all models must implement it.

Table 1. Summary of ASSR deficits in schizophrenic patients in the three studies considered here

Drive	Fundamental			Harmonic	Subharmonic
	40 Hz	30 Hz	20 Hz	20 Hz	40 Hz
Kwon et al.	↓	–	–	–	–
Vierling-Claassen et al.	↓	–	↑	↓	↑
Krishnan et al.	↓	–	–	–	–

Note. ↓ = significantly lower in patients; ↑ = significantly higher in patients; – = no significant difference between controls and patients. The tests included in the ASSRUnit module are based on this table. Krishnan et al. (2009) tested more driving frequencies than the ones shown in the table. The table only shows measures that are common to all three studies.

Tests and scores The five tests we implemented examine the five observations summarized in Table 1 individually. Furthermore, we implemented one prediction test, which tests 10 Hz power at 10 Hz drive. For the sake of simplicity, the test scores implemented so far are simple Boolean scores, indicating whether a model output fails or passes a test, that is, whether the difference between model output for the control and the schizophrenia-like network matches the experimental observation. In case of the model classes implementing sets of outputs, the mean difference is compared to the experimental observations. For the prediction test, we have chosen a RatioScore instead of a Boolean, which returns the ratio of the power for the schizophrenia-like configuration and the power for the control configuration.

Visualization, statistics, additional data In addition to the main features of the SciUnit framework for the analysis and comparison of the models, we use the fact that SciUnit allows for passing additional data, beyond the test scores, to provide a class that offers tools for the visualization of the results. This class includes functions to display the test results in a table, plot the results from a set of model outputs as a box plot, and perform and visualize a student's t test of the differences between control and schizophrenia-like networks.

Next, we describe three different use cases that show how the proposed module can be used for different purposes by experimentalists, modelers, and reviewers.

Use Case 1: Overview of the Experimental Literature

The first use case demonstrates how the experimental database can be used to get a comprehensive overview of the current experimental literature related to a neurophysiological or neurocognitive biomarker, in our case, ASSR deficits in patients suffering from schizophrenia. Figure 3 shows that with two simple commands, one can retrieve the names of all studies and all observations present in the database. These names will have to be used for all further queries of the database.

Figure 4 then shows how to get a complete overview of all observations of all studies in the database. As we can see in Figure 5, simply adding the parameter `meta=true`, to the command will additionally output the metadata associated with each study. This contains information on the subjects, modality, and so on. The overview command presents the data in a simple table and can be used to see which studies provided which observation and what the results were.

```
# List all studies in database
In [1]: s = get_studies(), print s

Out [1]:

['Kwon_1999', 'Vierling_2008', 'Krishnan_2009']

# List all observations in database
In [2]: o = get_observations(), print o

Out [2]:

['0505 :5Hz power at 5Hz drive', '1010 :10Hz power at 10Hz drive',
'1515 :15Hz power at 15Hz drive', '2020 :20Hz power at 20Hz drive',
'2525 :25Hz power at 25Hz drive', '3030 :30Hz power at 30Hz drive',
'3535 :35Hz power at 35Hz drive', '4040 :40Hz power at 40Hz drive',
'4545 :45Hz power at 45Hz drive', '5050 :50Hz power at 50Hz drive']
```

Figure 3. Display all studies and all observations included in the database.


```
In [1]: experimental_overview()
```

```
Out [1]:
```

	Kwon et al. 1999	Vierling-Claassen et al. 2008	Krishnan et al. 2009
5 Hz power at 5 Hz drive	not tested	not tested	equal
10 Hz power at 10 Hz drive	not tested	not tested	equal
15 Hz power at 15 Hz drive	not tested	not tested	equal
20 Hz power at 20 Hz drive	equal	higher	equal
25 Hz power at 25 Hz drive	not tested	not tested	equal
30 Hz power at 30 Hz drive	equal	equal	equal
35 Hz power at 35 Hz drive	not tested	not tested	equal
40 Hz power at 40 Hz drive	lower	lower	lower
45 Hz power at 45 Hz drive	not tested	not tested	lower
50 Hz power at 50 Hz drive	not tested	not tested	lower

Figure 4. Overview of the observations in the experimental literature. The command `experimental_overview` prints a table summarizing the results for all studies and all observations in the database. Note that by default, the qualitative study results are presented. This can be changed to the quantitative results setting the parameter `entrytype` to `Full`.

```
In [1]: experimental_overview(meta=True)
```

```
Out [1]:
```

```
...
```

```
Kwon_1999
```

```
{'Comments': 'Values estimated from figures, since values are not provided',
 'Measure': { 'Location': 'Midline frontal electrode',
              'Modality': 'EEG',
              'Processing': 'Butterworth band-pass filtered time averages
↪ followed by Fourier transform',
              'Value': 'Mean absolute power'},
 'Number of subjects': '15 (Ctrl) and 15 (Scz)',
 'Paradigm': 'Click train',
 'Subjects': 'Schizophrenia vs. control'}
```

```
Vierling_2008
```

```
{'Comments': 'Values estimated from figures, since values are not provided',
 'Measure': { 'Location': 'Left hemisphere',
              'Modality': 'MEG',
              'Processing': 'Time-averaging followed by PSD using Welch's
↪ method',
              'Value': 'Mean absolute power'},
 'Number of subjects': '12 (Ctrl) and 12 (Scz)',
 'Paradigm': 'Click train',
 'Subjects': 'Schizophrenia vs. control'}
```

```
Krishnan_2009
```

```
{'Measure': { 'Location': 'Cz in a 10-20 setting',
              'Modality': 'EEG',
              'Processing': 'Least square linear FIR filtered and Hilbert
↪ transformed',
              'Value': 'Mean baseline-corrected power'},
 'Number of subjects': '21 (Ctrl) and 21 (Scz)',
 'Paradigm': 'Amplitude-modulated tones; carrier frequency 1kHz',
 'Subjects': 'Schizophrenia vs. control'}
```

Figure 5. By setting the `meta` flag to `True`, additional information on the studies is displayed.

```
# Specific observations for specific studies
In [1]: observations = ['2020', '3030', '4040']
        studies = ['Kwon_1999', 'Krishnan_2009']
        experimental_overview(studies=studies, observations=observations)
```

Out [1]:

	Kwon et al. 1999	Krishnan et al. 2009
20 Hz power at 20 Hz drive	equal	equal
30 Hz power at 30 Hz drive	equal	equal
40 Hz power at 40 Hz drive	lower	lower

Figure 6. The `experimental_overview` command allows for querying for specific studies and observations using the names retrieved with the `get_studies` and `get_observations` commands.

However, as we can already see for our small demonstration database containing only three studies, a full overview is likely to become very large and therefore hard to grasp fully. By explicitly stating the studies and/or the observations in which one is interested, one can reduce the complexity of the table and get a clear and simple overview, as depicted in Figure 6. Note that in the examples, we have only used the qualitative description of the observations; the same functionality also applies to the quantitative descriptions. The functionality described here, along with more examples, can be explored in the accompanying Jupyter notebooks (https://github.com/ChristophMetzner/ASSRUnit/blob/CompPsychArticle/assrunit/Notebooks/Example_Experimental_Database.ipynb).

This simple querying functionality allows the user to get a quick, clean, and comprehensive overview of the experimental literature, to identify observations that are supported by many studies (see, in our case, the reduction of gamma power for stimulation at gamma frequency) but also to detect controversial findings. Furthermore, the display of the associated metadata allows for checking, for example, whether identified common observations extend over different modalities and postprocessing techniques and also whether controversial findings might be explained by differences in the experimental setup or other related aspects. In the future, it will also be possible to look at more than one database and compare the same observations across different patient groups to highlight commonalities and differences between disorders.

Use Case 2: Model Comparisons

While our first use case only exploited the experimental database, we now show the additional benefits of joining experimental and modeling data.

Simple model comparison By creating tests based on the model capabilities and grouping them into test suites, we can easily compare models against experimental data and against each other. Figure 7 demonstrates how we can use the module to create two different models along with several tests, run the models to produce the data relevant for the tests, and then judge the model outputs against experimental data and display the results together. Note that in this context, we use the term *model* as the *in silico* instantiation of a theoretical/conceptual model. Two different models may share the same code but differ only in parameter values. Again, the functionality described here, along with more examples, can be explored in an accompanying Jupyter notebook (https://github.com/ChristophMetzner/ASSRUnit/blob/CompPsychArticle/assrunit/Notebooks/Example_Model_Comparison.ipynb).

Advanced modeling data and visualization As already described in the Models subsection of Models, Capabilities, Tests, and More above, there is a second version of each model class

```

# Model instances
In [1]:
    conceptual_model_1 = VierlingSimpleModel(controlparams_model_1,
    ↪ schizparams_model_1,name='Conceptual_model_1')

    conceptual_model_2 = VierlingSimpleModel(controlparams_model_2,
    ↪ schizparams_model_2,name='Conceptual_model_2')

# Tests
In [2]: test_4040 = Test4040(observation={'ratio':0.5})
        test_3030 = Test3030(observation={'ratio':1.0})
        test_2020 = Test2020(observation={'ratio':1.2})
        test_4020 = Test4020(observation={'ratio':0.75})
        test_2040 = Test2040(observation={'ratio':1.2})

# A test suite
In [3]: vierling_main_suite = sciunit.TestSuite('vierling_main',
    ↪ [test4040,test3030,test2020,test2040,test4020])

        score_matrix = vierling_main_suite.judge([conceptual_model_1,
    ↪ conceptual_model_2])

        score_matrix.view()

Out [3]:

```

	Reduction of 40Hz power to 40Hz drive	No change of 30Hz power to 30Hz drive	Increase of 20Hz power to 20Hz drive	Decrease of 40Hz power to 20Hz drive	Increase of 20Hz power to 40Hz drive
Conceptual_model_1	Pass	Fail	Pass	Fail	Pass
Conceptual_model_2	Fail	Pass	Pass	Fail	Fail

Figure 7. Contrasting the results of comparing two models against experimental observations. First, the model instances are created and the parameters for the control network and the schizophrenia-like network are passed on together with a name. Then, appropriate tests are created and experimental observations are passed on. In this particular example, the observation is passed on as a “ratio,” which means that the value of the output of the schizophrenia-like simulation is divided by the value of the output of the control simulation. Afterward, the tests are grouped together to form a test suite, and the two example models are run against the test suite. The results of this run are stored in the matrix `score_matrix`, and by evoking the `view` method of the SciUnit score matrix, a comparison table is shown displaying the performance of each model against each test. Note that in this example, the two models and their resulting performance are purely hypothetical and do not reflect any actual model, and furthermore, the experimental observations do not reflect any actual findings.

that contains not only the standard methods that implement the necessary capabilities but also so-called `..._plus` methods, which can generate model data for different trials and/or subjects, depending on the type of model. Together with the methods from the visualization class, this additional model data can be used to better understand the model behavior, to judge the robustness of findings, and to statistically analyze model output. Figure 8 shows a simple example demonstrating the use of these classes/methods. When creating an instance of this class for the simple model from Vierling-Claassen et al. (2008), an additional parameter containing a number of RNG seeds is passed on. When the model is then run, a simulation is executed for each RNG seed, and the model output is a list containing the result for each simulation.

Use Case 3: Overview of Model Predictions

Finally, we show how predictions can be generated from existing models (see Figure 9). To generate the predictions, a set of prediction tests along with prediction capabilities, that is,

```

# Instantiating the model
In [1]: test_model = VierlingSimpleModelRobust(controlparams, schizparams,seeds)

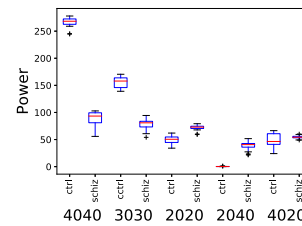
# Run simulations
In [2]: print 'Run simulations (this might take 15-20 minutes)'
        print '4040'
        mcontrol4040, mschiz4040, control4040, schiz4040 = test_model.produce_4040_plus(stimfrequency=40.0,powerfrequency=40.0)
        print '3030'
        mcontrol3030, mschiz3030, control3030, schiz3030 = test_model.produce_3030_plus(stimfrequency=30.0,powerfrequency=30.0)
        print '2020'
        mcontrol2020, mschiz2020, control2020, schiz2020 = test_model.produce_2020_plus(stimfrequency=20.0,powerfrequency=20.0)
        print '2040'
        mcontrol2040, mschiz2040, control2040, schiz2040 = test_model.produce_2040_plus(stimfrequency=40.0,powerfrequency=20.0)
        print '4020'
        mcontrol4020, mschiz4020, control4020, schiz4020 = test_model.produce_4020_plus(stimfrequency=20.0,powerfrequency=40.0)

# Display additional data as a box plot

labels = list(['ctrl','schiz','ctrl','schiz','ctrl','schiz','ctrl','schiz','ctrl','schiz'])
fig = plt.figure(1)
ax = fig.add_subplot(111)
bp=ax.boxplot([control4040,schiz4040,control3030,schiz3030,control2020,
↳ schiz2020,control2040,schiz2040,control4020,schiz4020],sym='+')
plt.setp(bp['boxes'], color='blue')
plt.setp(bp['whiskers'], color='blue')
plt.setp(bp['medians'], color='red')
ax.set_xticklabels(labels, rotation=90,fontsize=12)
for tick in ax.yaxis.get_major_ticks():
    tick.label.set_fontsize(12)
ax.set_ylabel('Power', fontsize=20)
a=ax.annotate('4040',xy=(0.1,0),xytext=(0,-65),xycoords='axes fraction',textcoords='offset
↳ points',size=20,ha='center',va='bottom')
a=ax.annotate('3030',xy=(0.3,0),xytext=(0,-65),xycoords='axes fraction',textcoords='offset
↳ points',size=20,ha='center',va='bottom')
a=ax.annotate('2020',xy=(0.5,0),xytext=(0,-65),xycoords='axes fraction',textcoords='offset
↳ points',size=20,ha='center',va='bottom')
a=ax.annotate('2040',xy=(0.7,0),xytext=(0,-65),xycoords='axes fraction',textcoords='offset
↳ points',size=20,ha='center',va='bottom')
a=ax.annotate('4020',xy=(0.9,0),xytext=(0,-65),xycoords='axes fraction', textcoords='offset
↳ points',size=20,ha='center',va='bottom')

```

Out [2]:



```

# Statistical analysis of some of the data
In [3]: t4040,p4040 = ttest_ind(control4040, schiz4040)
        t4020,p4020 = ttest_ind(control4020, schiz4020)
        print '40hz power at 40hz drive: t(df=19)=' ,t4040,'p=' ,p4040
        print '40hz power at 20hz drive: t(df=19)=' ,t4020,'p=' ,p4020

```

Out [3]:
40Hz power at 40Hz drive: t(df=19)= 51.8698283512 p= 7.00615973185e-37
40Hz power at 20Hz drive: t(df=19)= -2.08532975168 p= 0.0438080382856

Figure 8. Generating additional data. First, model instances are created and the `produce_XY_plus` method is used to run the simulation. The additional seed parameter contains a list of 20 RNG seeds, and a simulation is executed for each seed in that list. Thus, each simulation differs in background noise. The `produce_XY_plus` methods return the mean values of the outputs for the simulation runs (`mcontrol4040` and `mschiz4040` above), which can be used analogously to the output of the standard `produce_XY` methods. However, the values of the output of each single simulation run are returned for each run (`control4040` and `schiz4040` above) and can then be visualized or further analyzed statistically. Note that the model parameters used in this example are not based on any actual experimental findings in schizophrenic patients and that they do not aim to reproduce any experimental observations; they are only used for demonstration purposes (for model parameters of this model that reproduce experimental observations, see the original article by Vierling-Claassen et al. [2008]).

```

# Instantiate model
In [1]: conceptual_model = VierlingSimpleModel(controlparams, schizparams,
↪ name='Conceptual_model')

# Create prediction test
In [2]: prediction_test_1010 = PredictionTest1010()

# Simulate model and create prediction
In [3]: score = prediction_test_1010.judge(conceptual_model)

# Print result
In [4]: print score

Out [4]:

Ratio = 1.85

```

Figure 9. An overview of the workflow to generate predictions from a model. As before, a model is instantiated and the necessary parameters are passed on. Afterward, a prediction test is created in the same way as a standard test would be created, with the exception that prediction tests do not take experimental observations as arguments, because it is assumed that no experimental data exist yet. The test is then executed and returns a score. However, in the case of prediction tests, this score only contains the result of the model simulations (in this example, a ratio of the values of the output schizophrenia-like network and the control network). This score could, for example, be added to a prediction database.

capabilities the models must have for the models to generate the relevant data, needs to be created. For demonstration purposes, we have chosen to implement a single, simple prediction test. Because in ASSRUnit so far, we have only looked at experimental observations and computational models that cover gamma- and beta-range entrainment, the first test simply generates a prediction about how, in a given model, power in the alpha band (here at 10 Hz) differs between the control network and the schizophrenia-like network at 10 Hz drive. Note that this prediction test has been studied in the experimental literature, which means that it could have already been included in the experimental database and therefore does not represent a true prediction. However, we have chosen to include it for the purpose of demonstration. As before, more detailed information can be found in the accompanying Jupyter notebooks (https://github.com/ChristophMetzner/ASSRUnit/blob/CompPsychArticle/assrunit/Notebooks/Example_Prediction.ipynb).

DISCUSSION

The Potential Role of the Framework Within Computational Psychiatry

The use of computational approaches has seen a significant increase over the last decades in almost all areas of medicine and life sciences. Especially in psychiatry, it has become clear that the complex and often polygenic nature of psychiatric disorders might only be understood with the help of computational models (Adams et al., 2016; Corlett & Fletcher, 2014; Friston et al., 2014; Montague et al., 2012; Siekmeier, 2015; Stephan & Mathys, 2014; Wang & Krystal, 2014). Naturally, the number of computational models in the field of psychiatry has also increased significantly over the last years, and it has been argued that *in silico* instantiations of biomarkers are a crucial step toward understanding underlying disease mechanisms (Siekmeier, 2015). While this large increase in the number of modeling studies shows the importance of computational methods in the field, it also raises several issues that impede the community in exploiting these approaches to their full potential. For a computational model

to be a substantial contribution to knowledge, it has to adequately instantiate experimental observations, correctly implement the mathematical equations of the model, and generate experimentally testable predictions. The approach presented here addresses two of these three requirements, namely, the instantiation of experimental observations and the generation of testable predictions. While correctness of the code is an equally important requirement, it was out of the scope of the current work, because it very strongly depends on the type of computational model and on the programming language used to implement the model. Nevertheless, the approach presented here offers significant benefits for not only the computational psychiatry community but the psychiatry community as a whole, while imposing little additional effort on users and contributors. It gives modelers a tool to query experimental observations on neurophysiological and neurocognitive biomarkers and therefore helps them include current relevant experimental data in their modeling efforts. It further enables them to validate their modeling output against experimental observations during model construction and to demonstrate the performance of their models, both with respect to the experimental literature and with respect to other competing models. In addition to the benefits it offers modelers, it also enables experimentalists to quickly gain insight into the current state of modeling and to extract experimentally testable predictions from the models. Last, but not least, it offers a tool to reviewers that allows them to judge a newly proposed model by making explicit its performance against experimental data and competing models.

The concept of automated code testing and validation has been successfully applied in computer science for many years now; however, it is only slowly finding its way into the computational branches of scientific fields. SciUnit attempts to satisfy this demand by providing a simple, flexible, yet powerful framework to address the earlier mentioned issues. The computational neuroscience community has started to adopt this framework for the automatic validation of single neuron models (NeuronUnit; Gerkin & Omar, 2013). We are not aware of any similar efforts in the field of psychiatry.

Because schizophrenia is a polygenic, multifactorial, and very heterogeneous disorder, it has been argued that the usefulness of biomarkers lies in their potential to dissect the disorder into subtypes, which might even be linked more closely to findings on the genetic level (Markou et al., 2009; Meyer-Lindenberg & Weinberger, 2006; Perlis, 2011). The proposed ASSRUnit module together with computational models of biomarkers and specifically designed test suites could strongly facilitate this process by providing mechanistic links between neurophysiological or neurocognitive biomarkers and changes at the synaptic, cellular, and/or network level.

Future Directions for ASSRUnit

The presented ASSRUnit module can be easily extended and modified by others to fit their needs (e.g., to include more specialized visualization tools). Our efforts for establishing ASSRUnit as a widely used tool will focus on three main areas. (a) We aim to cover the majority of existing experimental studies with our experimental database in the future. Furthermore, we hope to convince experimentalists to provide more detailed experimental data or to ideally create database entries themselves. (b) We also aim to cover the majority of current computational models that describe the cortical circuitry responsible for the ASSR. Again, we hope to encourage modelers to contribute actively to ASSRUnit. (c) We aim to extend our set of prediction tests and thus our prediction database.

The most straightforward extension, in our view, is to include information on phase locking in addition to pure power in certain frequency bands. Several studies have reported,

additionally to a reduction in gamma power, a reduction in the phase-locking factor for patients suffering from schizophrenia (e.g., Brenner et al., 2003; Krishnan et al., 2009; Kwon et al., 1999; Light et al., 2006; Vierling-Claassen et al., 2008). These observations can very easily be incorporated into the existing module by including the experimental observations in the database, adding the necessary capabilities to the model classes, and adding the appropriate tests that link the experimental observations to the model capabilities.

Furthermore, the changes in oscillatory activity upon auditory stimulation are not limited to the gamma and the beta ranges for schizophrenic patients but also extend to lower-frequency bands, such as alpha, theta, and delta. For example, Brockhaus-Dumke, Mueller, Faigle, and Klosterkoetter (2008) found reduced phase locking in the alpha and theta bands for schizophrenic patients in an auditory paired-click paradigm, and Ford, Roach, Hoffman, & Mathalon (2008) found a reduction of phase locking in the delta and theta ranges for schizophrenic patients in an auditory oddball task. Abnormalities in these frequency bands have also been found in many other paradigms outside of the auditory system (see Basar & Guntekin [2013]). To the best of our knowledge, ASSRs to entrainment stimuli in the theta and delta ranges have not been looked at in schizophrenia. Therefore ASSRUnit could be used to generate predictions in these frequency ranges, as demonstrated in use case 3.

However, an inclusion of the earlier mentioned observations together with computational models explaining these deficits is not straightforward, because either the paradigms are different from the ones used to elicit ASSRs and/or the mechanisms underlying the effect are different, and therefore the computational models are substantially different to models of ASSRs. Therefore these deficits are better explored in separate modules solely focusing on each paradigm/deficit. However, it would be very interesting to co-explore computational models that have the capabilities to explain both ASSR gamma/beta band and delta/theta/alpha phase-locking deficits. Such an analysis could highlight interactions between different mechanisms underlying different symptoms or biomarkers.

Another very interesting and promising extension of the current module would be to include data and models from different psychiatric disorders, because schizophrenia is not the only disorder where patients show entrainment deficits. Wilson et al. (2007) explored gamma power in adolescents with psychosis and found reductions compared to normally developing controls. Their patient group consisted of patients suffering from schizophrenia and also from schizoaffective disorder and bipolar disorder. Interestingly, these disorders show overlapping symptoms, neurobiological substrates, and predisposing gene loci. Other studies have found reduced power and phase locking in the gamma range in patients with bipolar disorder (O'Donnell et al., 2004; Rass et al., 2010; Spencer et al., 2008). The presented module is perfectly suited to highlight commonalities and differences across disorders and to link those to mechanistic explanations via different theoretical and computational models.

Other Modules Beyond ASSRUnit

The approach presented here, combining an experimental database with a collection of models, tests, prediction tests, and a resulting predictions database, can be readily applied to a number of other neurophysiological biomarkers of schizophrenia as well as other psychiatric disorders.

In patients suffering from schizophrenia, a dysfunction of the auditory system has long been suspected. In fact, a large number of biomarkers for schizophrenia, other than ASSR

deficits, involve auditory processing. Several alterations of event-related potentials (ERPs), such as mismatch negativity (MMN), N100, and P50, have been described in the literature (see Shi, 2007; Siekmeier, 2015, for reviews of potential biomarkers and computational models thereof).

Naturally, our approach is well adaptable to brain circuits outside of the auditory system. Working memory deficits are probably one of the most robust and best described cognitive deficits in schizophrenic patients (reviewed in J. Lee & Park, 2005; Piskulic, Olver, Norman, & Maruff, 2007). Patients show a decrease in working memory capacity, that is, the capacity to maintain, manipulate, and use information online for a relatively short period of time, across a broad range of paradigms. Again, several theoretical and computational models have been proposed, aiming to provide mechanistic descriptions of the underlying mechanisms (e.g., Cano-Colino & Compte, 2012; Compte, Brunel, Goldman-Rakic, & Wang, 2000; Durstewitz, Seamans, & Sejnowski, 2000; Singh & Eliasmith, 2006; Wang, 2001; Wang, Tegnér, Constantinidis, & Goldman-Rakic, 2004).

All these deficits and alterations, along with relevant computational models, could be integrated into packages similar to the proposed ASSRUnit package. Such a unified framework would be of great benefit for the study of schizophrenia pathology due to the diversity of symptoms, biomarkers, and experimental observations linked to the mental disease.

CONCLUSION

We have proposed a framework for automated validation and comparison of computational models of neurophysiological and neurocognitive biomarkers of psychiatric disorders. The approach builds on SciUnit, a Python framework for scientific model comparison. As a case in point, we used this framework to develop ASSRUnit, a module comprising an experimental observations database, computational models, capabilities, tests/test suites, and visualization functions for ASSR response deficits in schizophrenia.

Our approach will facilitate the development, validation, and comparison of computational models of neurophysiological and neurocognitive biomarkers of psychiatric disorders by making the scope of models explicit and by making it easy for the user to assess a model's validity and to compare a model against competing models. Furthermore, it is easy to use; straightforward to extend to more experimental observations, computational models, and analyses; and ready to apply to other biomarkers. Therefore the adoption of the proposed framework could be of great use for modelers, reviewers, and experimentalists in the field of computational psychiatry.

AUTHOR CONTRIBUTIONS

Christoph Metzner: Conceptualization, Methodology, Software, Writing original draft, Writing review & editing. Tuomo Mäki-Marttunen: Software, Writing review & editing. Bartosz Zurowski: Conceptualization, Writing review & editing. Volker Steuber: Conceptualization, Writing review & editing.

FUNDING INFORMATION

Christoph Metzner, Deutsche Forschungsgemeinschaft (<http://dx.doi.org/10.13039/501100001659>), Award ID: ME 4391/1-1. Tuomo Mäki-Marttunen, Norges Forskningsråd (<http://dx.doi.org/10.13039/501100005416>), Award ID: 248828.

REFERENCES

- Adams, R. A., Huys, Q. J., & Roiser, J. P. (2016). Computational psychiatry: Towards a mathematically informed understanding of mental illness. *Journal of Neurology, Neurosurgery & Psychiatry*, *87*(1), 53–63.
- Başar, E., Başar-Eroglu, C., Karakaş, S., & Schürmann, M. (2001). Gamma, alpha, delta, and theta oscillations govern cognitive processes. *International Journal of Psychophysiology*, *39*(2–3), 241–248.
- Basar, E., & Guntekin, B. (2013). Review of delta, theta, alpha, beta, and gamma response oscillations in neuropsychiatric disorders. *Supplements to Clinical Neurophysiology*, *62*, 303–341.
- Beeman, D. (2013). A modeling study of cortical waves in primary auditory cortex. *BMC Neuroscience*, *14*(Suppl. 1), P23.
- Börgers, C., & Kopell, N. (2005). Effects of noisy drive on rhythms in networks of excitatory and inhibitory neurons. *Neural Computation*, *17*(3), 557–608.
- Bower, J. M. (1992). Modeling the nervous system. *Trends in Neuroscience*, *15*(11), 411–412.
- Bower, J. M., & Beeman, D. (1993). *The book of genesis: Exploring realistic neural models with the general neural simulation system*. New York, NY: Springer.
- Brenner, C. A., Sporns, O., Lysaker, P. H., & O'Donnell, B. F. (2003). Eeg synchronization to modulated auditory tones in schizophrenia, schizoaffective disorder, and schizotypal personality disorder. *American Journal of Psychiatry*, *160*(12), 2238–2240.
- Brockhaus-Dumke, A., Mueller, R., Faigle, U., & Klosterkoetter, J. (2008). Sensory gating revisited: Relation between brain oscillations and auditory evoked potentials in schizophrenia. *Schizophrenia Research*, *99*(1), 238–249.
- Cannon, R. C., Gleeson, P., Crook, S., Ganapathy, G., Marin, B., Piasini, E., & Silver, R. A. (2014). Lems: A language for expressing complex biological models in concise and hierarchical form and its use in underpinning neuroml 2. *Frontiers in Neuroinformatics*, *8*, Article 79.
- Cano-Colino, M., & Compte, A. (2012). A computational model for spatial working memory deficits in schizophrenia. *Pharmacopsychiatry*, *45*(S 01), S49–S56.
- Cho, R., Konecky, R., & Carter, C. (2006). Impairments in frontal cortical γ synchrony and cognitive control in schizophrenia. *Proceedings of the National Academy of Sciences of the United States of America*, *103*(52), 19878–19883.
- Compte, A., Brunel, N., Goldman-Rakic, P. S., & Wang, X.-J. (2000). Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model. *Cerebral Cortex*, *10*(9), 910–923.
- Corlett, P. R., & Fletcher, P. C. (2014). Computational psychiatry: A rosetta stone linking the brain to mental illness. *The Lancet Psychiatry*, *1*(5), 399–402.
- Durstewitz, D., Seamans, J. K., & Sejnowski, T. J. (2000). Dopamine-mediated stabilization of delay-period activity in a network model of prefrontal cortex. *Journal of Neurophysiology*, *83*(3), 1733–1750.
- Engel, A. K., Kreiter, A. K., König, P., & Singer, W. (1991). Synchronization of oscillatory neuronal responses between striate and extrastriate visual cortical areas of the cat. *Proceedings of the National Academy of Sciences of The United States of America*, *88*(14), 6048–6052.
- Ford, J. M., Roach, B. J., Hoffman, R. S., & Mathalon, D. H. (2008). The dependence of p300 amplitude on gamma synchrony breaks down in schizophrenia. *Brain Research*, *1235*, 133–142.
- Friston, K. J., Redish, A. D., & Gordon, J. A. (2017). Computational nosology and precision psychiatry. *Computational Psychiatry*, *1*, 2–23.
- Friston, K. J., Stephan, K. E., Montague, R., & Dolan, R. J. (2014). Computational psychiatry: The brain as a phantastic organ. *The Lancet Psychiatry*, *1*(2), 148–158.
- Gerkin, R., & Omar, C. (2013). *Neurounit: Validation tests for neuroscience models*. Paper presented at Neuroinformatics, Stockholm, Sweden.
- Gleeson, P., Piasini, E., Crook, S., Cannon, R., Steuber, V., Jaeger, D., . . . Silver, R. A. (2012). The open source brain initiative: Enabling collaborative modelling in computational neuroscience. *BMC Neuroscience*, *13*(1), O7.
- Gordon, E., Williams, L., Haig, A. R., Wright, J., & Meares, R. A. (2001). Symptom profile and gamma processing in schizophrenia. *Cognitive Neuropsychiatry*, *6*(1), 7–19.
- Hamm, J. P., Bobilev, A. M., Hayrynen, L. K., Hudgens-Haney, M. E., Oliver, W. T., Parker, D. A., . . . Clementz, B. A. (2015). Stimulus train duration but not attention moderates γ -band entrainment abnormalities in schizophrenia. *Schizophrenia Research*, *165*(1), 97–102.
- Jokeit, H., & Makeig, S. (1994). Different event-related patterns of gamma-band power in brain waves of fast- and slow-reacting subjects. *Proceedings of the National Academy of Sciences*, *91*(14), 6339–6343.
- Krishnan, G., Hetrick, W. P., Brenner, C., Shekhar, A., Steffen, A., & O'Donnell, B. F. (2009). Steady state and induced auditory gamma deficits in schizophrenia. *Neuroimage*, *47*(4), 1711–1719.
- Kwon, J. S., O'Donnell, B. F., Wallenstein, G. V., Greene, R. W., Hirayasu, Y., Nestor, P. G., . . . McCarley, R. W. (1999). Gamma frequency-range abnormalities to auditory stimulation in schizophrenia. *Archives of General Psychiatry*, *56*(11), 1001–1005.
- Lee, J., & Park, S. (2005). Working memory impairments in schizophrenia: A meta-analysis. *Journal of Abnormal Psychology*, *114*(4), 599–611.
- Lee, K.-H., Williams, L., Haig, A., & Gordon, E. (2003). “Gamma (40 Hz) phase synchronicity” and symptom dimensions in schizophrenia. *Cognitive Neuropsychiatry*, *8*(1), 57–71.
- Light, G. A., Hsu, J. L., Hsieh, M. H., Meyer-Gomes, K., Sprock, J., Swerdlow, N. R., & Braff, D. L. (2006). Gamma band oscillations reveal neural network cortical coherence dysfunction in schizophrenia patients. *Biological Psychiatry*, *60*(11), 1231–1240.
- Markou, A., Chiamulera, C., Geyer, M. A., Tricklebank, M., & Steckler, T. (2009). Removing obstacles in neuroscience drug discovery: The future path for animal models. *Neuropsychopharmacology*, *34*(1), 74–89.
- Metzner, C. (2017). [Re] modeling gaba alterations in schizophrenia: A link between impaired inhibition and gamma and beta auditory entrainment. *ReScience*, *3*(1).

- Metzner, C., Schweikard, A., & Zurowski, B. (2016). Multifactorial modeling of impairment of evoked gamma range oscillations in schizophrenia. *Frontiers in Computational Neuroscience, 10*, Article 89.
- Meyer-Lindenberg, A., & Weinberger, D. R. (2006). Intermediate phenotypes and genetic mechanisms of psychiatric disorders. *Nature Reviews Neuroscience, 7*(10), 818–827.
- Montague, P. R., Dolan, R. J., Friston, K. J., & Dayan, P. (2012). Computational psychiatry. *Trends in Cognitive Sciences, 16*(1), 72–80.
- Mulert, C., Kirsch, V., Pascual-Marqui, R., McCarley, R. W., & Spencer, K. M. (2011). Long-range synchrony of gamma oscillations and auditory hallucination symptoms in schizophrenia. *International Journal of Psychophysiology, 79*(1), 55–63.
- O’Connell, M., Barczak, A., Ross, D., McGinnis, T., Schroeder, C., & Lakatos, P. (2015). Multi-scale entrainment of coupled neuronal oscillations in primary auditory cortex. *Frontiers in Human Neuroscience, 9*, Article 655.
- O’Donnell, B. F., Hetrick, W. P., Vohs, J. L., Krishnan, G. P., Carroll, C. A., & Shekhar, A. (2004). Neural synchronization deficits to auditory stimulation in bipolar disorder. *Neuroreport, 15*(8), 1369–1372.
- Omar, C., Aldrich, J., & Gerkin, R. C. (2014). Collaborative infrastructure for test-driven scientific model validation. In *Companion proceedings of the 36th International Conference on Software Engineering* (pp. 524–527). New York, NY: ACM.
- Perlis, R. (2011). Translating biomarkers to clinical practice. *Molecular Psychiatry, 16*(11), 1076–1087.
- Piskulic, D., Olver, J. S., Norman, T. R., & Maruff, P. (2007). Behavioural studies of spatial working memory dysfunction in schizophrenia: A quantitative literature review. *Psychiatry Research, 150*(2), 111–121.
- Rass, O., Krishnan, G., Brenner, C. A., Hetrick, W. P., Merrill, C. C., Shekhar, A., & O’Donnell, B. F. (2010). Auditory steady state response in bipolar disorder: Relation to clinical state, cognitive performance, medication status, and substance disorders. *Bipolar Disorders, 12*(8), 793–803.
- Sarma, G. P., Jacobs, T. W., Watts, M. D., Ghayoomie, S. V., Larson, S. D., & Gerkin, R. C. (2016). Unit testing, model validation, and biological simulation. *F1000Research, 5*, Article 1946.
- Shi, W.-X. (2007). The auditory cortex in schizophrenia. *Biological Psychiatry, 61*(7), 829–830.
- Siekmeier, P. J. (2015). Computational modeling of psychiatric illnesses via well-defined neurophysiological and neurocognitive biomarkers. *Neuroscience & Biobehavioral Reviews, 57*, 365–380.
- Singh, R., & Eliasmith, C. (2006). Higher-dimensional neurons explain the tuning and dynamics of working memory cells. *Journal of Neuroscience, 26*(14), 3667–3678.
- Spencer, K. M. (2012). Baseline gamma power during auditory steady-state stimulation in schizophrenia. *Frontiers in Human Neuroscience, 5*, Article 190.
- Spencer, K. M., Nestor, P. G., Niznikiewicz, M. A., Salisbury, D. F., Shenton, M. E., & McCarley, R. W. (2003). Abnormal neural synchrony in schizophrenia. *Journal of Neuroscience, 23*(19), 7407–7411.
- Spencer, K. M., Niznikiewicz, M. A., Nestor, P. G., Shenton, M. E., & McCarley, R. W. (2009). Left auditory cortex gamma synchronization and auditory hallucination symptoms in schizophrenia. *BMC Neuroscience, 10*(1), Article 85.
- Spencer, K. M., Salisbury, D. F., Shenton, M. E., & McCarley, R. W. (2008). γ -band auditory steady-state responses are impaired in first episode psychosis. *Biological Psychiatry, 64*(5), 369–375.
- Stephan, K. E., & Mathys, C. (2014). Computational approaches to psychiatry. *Current Opinion in Neurobiology, 25*, 85–92.
- Vierling-Claassen, D., Siekmeier, P., Stufflebeam, S., & Kopell, N. (2008). Modeling gaba alterations in schizophrenia: A link between impaired inhibition and altered gamma and beta range auditory entrainment. *Journal of Neurophysiology, 99*(5), 2656–2671.
- Wang, X.-J. (2001). Synaptic reverberation underlying mnemonic persistent activity. *Trends in Neurosciences, 24*(8), 455–463.
- Wang, X.-J., & Krystal, J. H. (2014). Computational psychiatry. *Neuron, 84*(3), 638–654.
- Wang, X.-J., Tegnér, J., Constantinidis, C., & Goldman-Rakic, P. (2004). Division of labor among distinct subtypes of inhibitory neurons in a cortical microcircuit of working memory. *Proceedings of the National Academy of Sciences of the United States of America, 101*(5), 1368–1373.
- Wilson, T. W., Hernandez, O. O., Asherin, R. M., Teale, P. D., Reite, M. L., & Rojas, D. C. (2007). Cortical gamma generators suggest abnormal auditory circuitry in early-onset psychosis. *Cerebral Cortex, 18*(2), 371–378.
- Zhang, J., Ma, L., Li, W., Yang, P., & Qin, L. (2016). Cholinergic modulation of auditory steady-state response in the auditory cortex of the freely moving rat. *Neuroscience, 324*, 29–39.