



HAL
open science

HippoUnit: A software tool for the automated testing and systematic comparison of detailed models of hippocampal neurons based on electrophysiological data

Sára Sáray, Christian A Rössert, Shailesh Appukuttan, Rosanna Migliore, Paola Vitale, Carmen A Lupascu, Luca L Bologna, Werner Van Geit, Armando Romani, Andrew P. Davison, et al.

► To cite this version:

Sára Sáray, Christian A Rössert, Shailesh Appukuttan, Rosanna Migliore, Paola Vitale, et al.. HippoUnit: A software tool for the automated testing and systematic comparison of detailed models of hippocampal neurons based on electrophysiological data. PLoS Computational Biology, 2021, 17 (1), pp.e1008114. 10.1371/journal.pcbi.1008114 . hal-03063383

HAL Id: hal-03063383

<https://hal.science/hal-03063383v1>

Submitted on 14 Dec 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

1 HippoUnit: A software tool for the automated
2 testing and systematic comparison of detailed
3 models of hippocampal neurons based on
4 electrophysiological data

5

6

7 Sára Sáráy^{1,2*}, Christian A. Rössert³, Shailesh Appukuttan⁴, Rosanna Migliore⁵, Paola Vitale⁵,
8 Carmen A. Lupascu⁵, Luca L. Bologna⁵, Werner Van Geit³, Armando Romani³, Andrew P.
9 Davison⁴, Eilif Muller³, Tamás F. Freund^{1,2}, Szabolcs Káli^{1,2*}

10

11

12 ¹Faculty of Information Technology and Bionics, Pázmány Péter Catholic University,
13 Budapest, Hungary

14 ² Institute of Experimental Medicine, Budapest, Hungary,

15 ³ Blue Brain Project, École Polytechnique Fédérale de Lausanne, Geneva, Switzerland,

16 ⁴Paris-Saclay Institute of Neuroscience, Centre National de la Recherche Scientifique/
17 Université Paris-Saclay, Gif-sur-Yvette, France,

18 ⁵Institute of Biophysics, National Research Council, Palermo, Italy.

19

20 * Corresponding author: saray.sara@koki.hu (SS), kali@koki.hu (SK)

21 **Abstract**

22

23 Anatomically and biophysically detailed data-driven neuronal models have become
24 widely used tools for understanding and predicting the behavior and function of neurons. Due
25 to the increasing availability of experimental data from anatomical and electrophysiological
26 measurements as well as the growing number of computational and software tools that enable
27 accurate neuronal modeling, there are now a large number of different models of many cell
28 types available in the literature. These models were usually built to capture a few important or
29 interesting properties of the given neuron type, and it is often unknown how they would
30 behave outside their original context. In addition, there is currently no simple way of
31 quantitatively comparing different models regarding how closely they match specific
32 experimental observations. This limits the evaluation, re-use and further development of the
33 existing models. Further, the development of new models could also be significantly
34 facilitated by the ability to rapidly test the behavior of model candidates against the relevant
35 collection of experimental data. We address these problems for the representative case of the
36 CA1 pyramidal cell of the rat hippocampus by developing an open-source Python test suite,
37 which makes it possible to automatically and systematically test multiple properties of models
38 by making quantitative comparisons between the models and electrophysiological data. The
39 tests cover various aspects of somatic behavior, and signal propagation and integration in
40 apical dendrites. To demonstrate the utility of our approach, we applied our tests to compare
41 the behavior of several different hippocampal CA1 pyramidal cell models from the ModelDB
42 database against electrophysiological data available in the literature, and concluded that each
43 of these models provides a good match to experimental results in some domains but not in
44 others. We also show how we employed the test suite to aid the development of models within

45 the European Human Brain Project (HBP), and describe the integration of the tests into the
46 validation framework developed in the HBP, with the aim of facilitating more reproducible
47 and transparent model building in the neuroscience community.

48 **Author summary**

49

50 Anatomically and biophysically detailed neuronal models are useful tools in
51 neuroscience because they allow the prediction of the behavior and the function of the studied
52 cell type under circumstances that are hard to investigate experimentally. However, most
53 detailed biophysical models have been built to capture a few selected properties of the real
54 neuron, and it is often unknown how they would behave under different circumstances, or
55 whether they can be used to successfully answer different scientific questions. To help the
56 modeling community develop better neural models, and make the process of model building
57 more reproducible and transparent, we developed a test suite that enables the comparison of
58 the behavior of models of neurons in the rat hippocampus and their evaluation against
59 experimental data. Applying our tests to several models available in the literature, we show
60 that each model is able to capture some of the important properties of the real neuron but fails
61 to match experimental data in other domains. We also use the test suite in the model
62 development workflow of the European Human Brain Project to aid the construction of better
63 models of hippocampal neurons and networks.

64 **Introduction**

65

66 The construction and simulation of anatomically and biophysically detailed models is
67 becoming a standard tool in neuroscience [1]. Such models, which typically employ the
68 compartmental modeling approach and a Hodgkin-Huxley-type description of voltage-gated
69 ion channels, are capable of providing fairly accurate models of single neurons [2–10] and
70 (when complemented by appropriate models of synaptic interactions) even large-scale circuits
71 [11–14]. However, building such detailed multi-compartmental models of neurons requires
72 setting a large number of parameters (such as the densities of various ion channels in multiple
73 neuronal compartments) that are often not directly constrained by the available experimental
74 data. These parameters are typically tuned (either manually or using automated parameter-
75 search methods [9,15–17]) until the simulated physiological behavior of the model matches
76 some pre-defined set of experimental observations.

77 For an increasing number of cell types, the available experimental data already provide
78 diverse constraints on the expected physiological behavior of the neuron under a variety of
79 conditions. Based on various (typically small) subsets of the available constraints, a large
80 number of different models of several cell types have been developed to investigate diverse
81 aspects of single-cell behavior, and for inclusion in realistic circuit models. As an example,
82 there are currently 131 different models related to the hippocampal CA1 pyramidal cell (PC)
83 in the ModelDB database [18]. However, even though these models are publicly available, it
84 is still technically challenging to verify their behavior beyond the examples explicitly
85 included with the model, and especially to test their behavior outside the context of the
86 original study, or to compare it with the behavior of other models. This sparsity of
87 information about the performance of detailed models may also be one reason why model re-

88 use in the community is relatively limited, which decreases the chance of spotting errors in
89 modeling studies, and may lead to an unnecessary replication of effort.

90 A systematic comparison of existing models built in different laboratories requires the
91 development of a comprehensive validation suite, a set of automated tests that quantitatively
92 compare various aspects of model behavior with the corresponding experimental data. Such
93 validation suites enable all modeling groups to evaluate their existing and newly developed
94 models according to common, standardized criteria, thus facilitating model comparison and
95 providing an objective measure of progress in matching relevant experimental observations.
96 Applying automated tests also allows researchers to learn more about models published by
97 other groups (beyond the results included in the papers) with relatively little effort, thus
98 facilitating optimal model re-use and co-operative model development. In addition,
99 systematic, automated testing is expected to speed up model development in general by
100 allowing researchers to easily evaluate models in relation to the relevant experimental data
101 after every iteration of model adjustment. Finally, a comprehensive evaluation of model
102 behavior appears to be critical for models that are then expected to provide useful predictions
103 in a new context. A prime example of this is detailed single cell models included in network
104 models, where diverse aspects of cellular function such as synaptic integration, intracellular
105 signal propagation, spike generation and adaptation mechanisms all contribute to the input-
106 output function of the neuron in the context of an active network. By comparing multiple
107 different aspects of the behavior of the single cell model with experimental data, one can
108 increase the chance of having a model that also behaves correctly within the network. The
109 technical framework for developing automated test suites for models already exists [19], and
110 is currently used by several groups to create a variety of tests for models of neural structure
111 and function at different scales [20–24]. In the current study, our goal was to develop a

112 validation suite for the physiological behavior of one of the most studied cell types of the
113 mammalian brain, the pyramidal cell in area CA1 of the rat hippocampus.

114 CA1 pyramidal neurons display a large repertoire of nonlinear responses in all of their
115 compartments (including the soma, axon, and various functionally distinct parts of the
116 dendritic tree), which are experimentally well-characterized. In particular, there are detailed
117 quantitative results available on the subthreshold and spiking voltage response to somatic
118 current injections [3,25]; on the properties of the action potentials back-propagating from the
119 soma into the dendrites [26–28], which is a basic measure of dendritic excitability; and on the
120 characteristics of the spread [29] and non-linear integration of synaptically evoked signals in
121 the dendrites, including the conditions necessary for the generation of dendritic spikes [30–
122 33].

123 The test suite that we have developed allows the quantitative comparison of the
124 behavior of anatomically and biophysically detailed models of CA1 pyramidal neurons with
125 experimental data in all of these domains. In this paper, we first describe the implementation
126 of the HippoUnit validation suite. Next, we show how we used this test suite to systematically
127 compare existing models from six prominent publications from different laboratories. We
128 then show an example of how the tests have been applied to aid the development of new
129 models in the context of the European Human Brain Project (HBP). Finally, we describe the
130 integration of our test suite into the general validation framework developed in the HBP.

131

132 **Methods**

133 **Implementation of HippoUnit**

134

135 HippoUnit is a Python test suite based on the SciUnit [19] framework, which is a
136 Python package for testing scientific models, and during its implementation the NeuronUnit
137 package [20] was taken into account as an example of how to use the SciUnit framework for
138 testing neuronal models. In SciUnit tests usually four main classes are implemented: the test
139 class, the model class, the capabilities class and the score class. HippoUnit is built in a way
140 that keeps this structure. The key idea behind this structure is the decoupling of the model
141 implementation from the test implementation by defining standardized interfaces
142 (capabilities) between them, so that tests can easily be used with different models without
143 being rewritten, and models can easily be adapted to fit the framework.

144 Each test of HippoUnit is a separate Python class that, similarly to other SciUnit
145 packages, can run simulations on the models to generate model *predictions*, which can be
146 compared with experimental *observations* to yield the final score, provided that the model has
147 the required capabilities implemented to mimic the appropriate experimental protocol and
148 produce the same type of measurable output. All measured or calculated data that contribute
149 to the final score (including the recorded voltage traces, the extracted features and the
150 calculated feature scores) are saved in JSON or pickle files (or, in many cases, in both types
151 of files). JSON files are human readable, and can be easily loaded into Python dictionaries.
152 Data with a more complex structure are saved into pickle files. This makes it possible to
153 easily write and read the data (for further processing or analysis) without changing its Python
154 structure, no matter what type of object or variable it is. In addition to the JSON files a text
155 file (log file) is also saved, that contains the final score and some useful information or notes
156 specific to the given test and model. Furthermore, the recorded voltage traces, the extracted
157 features and the calculated feature scores are also plotted for visualization.

158 Similarly to many of the existing SciUnit packages the implementations of specific
159 models are not part of the HippoUnit package itself. Instead, HippoUnit contains a general

160 `ModelLoader` class. This class is implemented in a way that it is able to load and deal with
161 most types of models defined in the HOC language of the NEURON simulator (either as
162 standalone HOC models or as HOC templates) [34]. It implements all model-related methods
163 (capabilities) that are needed to simulate these kinds of neural models in order to generate the
164 prediction without any further coding required from the user.

165 For the smooth validation of the models developed using parameter optimization
166 within the HBP there is a child class of the `ModelLoader` available in `HippoUnit` that is
167 called `ModelLoader_BPO`. This class inherits most of the functions (especially the capability
168 functions) from the `ModelLoader` class, but it implements additional functions that are able
169 to automatically deal with the specific way in which information is represented and stored in
170 these optimized models. The role of these functions is to gather all the information from the
171 metadata and configuration files of the models that are needed to set the parameters required
172 to load the models and run the simulations on them (such as path to the model files, name of
173 the model template or the simulation temperature (the `celsius` variable of `Neuron`)). This
174 enables the validation of these models without any manual intervention needed from the user.
175 The section lists required by the tests of `HippoUnit` are also created automatically using the
176 morphology files of these models (for details see the “Classify apical sections of pyramidal
177 cells” subsection). For neural models developed using other software and methods, the user
178 needs to implement the capabilities through which the tests of `HippoUnit` perform the
179 simulations and recordings on the model.

180 The capabilities are the interface between the tests and the models. The `ModelLoader`
181 class inherits from the capabilities and must implement the methods of the capability. The test
182 can only be run on a model if the necessary capability methods are implemented in the

183 ModelLoader. All communication between the test and the model happens through the
184 capabilities.

185 The methods of the score classes perform the quantitative comparison between the
186 *prediction* and the *observation*, and return the score object containing the final score and some
187 related data, such as the paths to the saved figure and data (JSON) files and the prediction and
188 observation data. Although SciUnit and NeuronUnit have a number of different score types
189 implemented, those typically compare a single *prediction* value to a single *observation* value,
190 while the tests of HippoUnit typically extract several features from the model's response to be
191 compared with experimental data. Therefore, each test of HippoUnit has its own score class
192 implemented that is designed to deal with the specific structure of the output *prediction* data
193 and the corresponding *observation* data. For simplicity, we refer to the discrepancy between
194 the target experimental data (*observation*) and the models' behavior (*prediction*) with respect
195 to a studied feature using the term feature score. In most cases, when the basic statistics (mean
196 and standard deviation) of the experimental features (typically measured in several different
197 cells of the same cell type) are available, feature scores are computed as the absolute
198 difference between the feature value of the model and the experimental mean feature value,
199 divided by the experimental standard deviation (Z-score) [35]. The final score of a given test
200 achieved by a given model is given by the average (or, in some cases, the sum) of the feature
201 scores for all the features evaluated by the test.

202

203 **Implementation of the tests of HippoUnit**

204 **The Somatic Features Test**

205

206 The Somatic Features Test uses the Electrophys Feature Extraction Library (eFEL)
207 [36] to extract and evaluate the values of both subthreshold and suprathreshold (spiking)
208 features from voltage traces that represent the response of the model to somatic current
209 injections of different positive (depolarizing) and negative (hyperpolarizing) current
210 amplitudes. Spiking features describe action potential shape (like AP width, AP rise/fall rate,
211 AP amplitude, etc.) and timing (frequency, inter-spike intervals, time to first/last spike, etc.),
212 while some passive features (such as the voltage base or the steady state voltage), and
213 subthreshold features for negative current stimuli (voltage deflection, sag amplitude, etc.) are
214 also examined.

215 In this test step currents of varying amplitudes are injected into the soma of the model
216 and the voltage response is recorded. The simulation protocol is set according to an input
217 configuration JSON file, which contains all the current amplitudes, the delay and the duration
218 of the stimuli, and the stimulation and recording positions. Simulations using different current
219 amplitudes are run in parallel if this is supported by the computing environment.

220 As the voltage responses of neurons to somatic current injections can strongly depend
221 on the experimental method, and especially on the type of electrode used, target values for
222 these features were extracted from two different datasets. One dataset was obtained from
223 sharp electrode recordings from adult rat CA1 neurons (sharp electrode data set) [3], and the
224 other dataset is from patch clamp recordings in rat CA1 pyramidal cells (data provided by
225 Judit Makara (patch clamp dataset)). For both of these datasets we had access to the recorded
226 voltage traces from multiple neurons, which made it possible to perform our own feature
227 extraction using eFEL. This ensures that the features are interpreted and calculated the same
228 way for both the experimental data and the models' voltage response during the simulation.
229 Furthermore, it allows a more thorough comparison against a large number of features
230 extracted from experimental recordings yielded using the exact same protocol, which is

231 unlikely to be found in any paper of the available literature. However, to see how
232 representative these datasets are of the literature as a whole we first compared some of the
233 features extracted from these datasets to data available on Neuroelectro.org [37] and on
234 Hippocampome.org [38]. The features we compared were the following: resting potential,
235 voltage threshold, after-hyperpolarization (AHP) amplitudes (fast, slow), action potential
236 width and sag ratio. Although these databases have mean and standard deviation values for
237 these features that are calculated from measurements using different methods, protocols and
238 from different animals, we found that most of the feature values for our two experimental
239 datasets fall into the ranges declared as typical for CA1 PCs in the online databases. The only
240 conspicuous exception is the fast AHP amplitude of the patch clamp dataset used in this
241 study, which is 1.7 ± 1.5 mV, while the databases cite values between 6.8 and 11.64 mV. This
242 deviation could possibly stem from a difference in the way that the fast AHP is measured.

243 We also performed a more specific review of the relevant literature to compare the
244 most important somatic features of the patch clamp dataset to results from available patch
245 clamp recordings (Table 1). Our analysis confirmed that the values of several basic
246 electrophysiological features such as the AP voltage threshold, the AP amplitude, the AP
247 width, and the amplitude of the hyperpolarizing sag extracted from our patch clamp dataset
248 fall into the range observed experimentally. We conclude that the patch clamp dataset is in
249 good agreement with experimental observations available in the literature, and will be used as
250 a representative example in this study.

251

Feature <i>(eFEL feature name)</i>	Value in literature	Value in patch clamp dataset
AP voltage threshold	-46 - -53 mV [39–42]	-51.13±0.97 mV (0.15 nA current step)

<i>(AP_begin_voltage)</i>		-50.14±1.97 mV (0.2 nA current step) -49.36±2.02 mV (0.25 nA current step)
AP amplitude <i>(AP_amplitude_from_voltagebase)</i>	71 - 112 mV [39,42,43]	98.36±5.82 mV (0.15 nA current step) 96.83±5.66 mV (0.2 nA current step) 95.99±5.22 mV (0.25 nA current step)
AP width at half amplitude <i>(AP_duration_half_width)</i>	0.8 - 1.29 ms [39,41–43]	1.23±0.096 ms (0.15 nA current step) 1.25±0.11 ms (0.2 nA current step) 1.32±0.086 ms (0.25 nA current step)
sag ratio <i>(sag_ratio2)</i>	0.84±0.02 [43], 0.83±0.01 [44]	0.79±0.023 (-0.05 nA current step) 0.81±0.03 (-0.1 nA current step) 0.81±0.027 (-0.15 nA current step) 0.81±0.03 (-0.2 nA current step) 0.80±0.03 (-0.25 nA current step)

252 Table 1: Comparison of the most important somatic features extracted using eFEL from the patch clamp dataset

253 (used as target data in the Somatic Features Test) to results from patch clamp recordings available in the

254 literature.

255

256 The *observation* data are loaded from a JSON file of a given format which contains

257 the names of the features to be evaluated, the current amplitude for which the given feature is

258 evaluated and the corresponding experimental mean and standard deviation values. The

259 feature means and standard deviations are extracted using BluePyEfe [45] from a number of

260 voltage traces recorded from several different cells. Its output can be converted to stimulus

261 and feature JSON files used by HippoUnit using the script available here:

262 https://github.com/sasaray/HippoUnit_demo/blob/master/target_features/Examples_on_creating_JSON_files/Somatic_Features/convert_new_output_feature_data_for_valid.py.

263 Setting the

264 `specify_data_set` parameter it can be ensured that the test results against different

265 experimental data sets are saved into different folders.

266 For certain features eFEL returns a vector as a result; in these cases, the feature value
267 used by HippoUnit is the average of the elements of the vector. These are typically spiking
268 features for which eFEL extracts a value corresponding to each spike fired. For features that
269 use the ‘AP_begin_time’ or ‘AP_begin_voltage’ feature values for further calculations, we
270 exclude the first element of the vector output before averaging because we discovered that
271 these features are often incorrectly detected for the first action potential of a train.

272 The score class of this test returns as the final score the average of *Z-scores* for the
273 evaluated eFEL features achieved by the model. Those features that could not be evaluated
274 (e.g., spiking features from voltage responses without any spikes) are listed in a log file to
275 inform the user, and the number of successfully evaluated features out of the number of
276 features attempted to be evaluated is also reported.

277

278 **The Depolarization Block Test**

279

280 This test aims to determine whether the model enters depolarization block in response
281 to a prolonged, high intensity somatic current stimulus. For CA1 pyramidal cells, the test
282 relies on experimental data from Bianchi et al. [25]. According to these data, CA1 PCs
283 respond to somatic current injections of increasing intensity with an increasing number of
284 action potentials until a certain threshold current intensity is reached. For current intensities
285 higher than the threshold, the cell does not fire over the whole period of the stimulus; instead,
286 firing stops after some action potentials, and the membrane potential is sustained at some
287 constant depolarized level for the rest of the stimulus. This phenomenon is termed
288 depolarization block [25].

289 This test uses the same capability class as the Somatic Features Test for injecting
290 current and recording the somatic membrane potential (see the description above). Using this
291 capability, the model is stimulated with 1000 ms long square current pulses increasing in
292 amplitude from 0 to 1.6 nA in 0.05 nA steps, analogous to the experimental protocol. The
293 stimuli of different amplitudes are run in parallel. Somatic spikes are detected and counted
294 using eFEL [36].

295 From the somatic voltage responses of the model, the following features are evaluated.
296 I_{th} is the threshold current to reach depolarization block; experimentally, this is both the
297 amplitude of the current injection at which the cell exhibits the maximum number of spikes,
298 and the highest stimulus amplitude that does not elicit depolarization block. In the test two
299 separate features are evaluated for the model and compared to the experimental I_{th} : the current
300 intensity for which the model fires the maximum number of action potentials ($I_{maxNumAP}$),
301 and the current intensity one step before the model enters depolarization block
302 ($I_{below_depol_block}$). If these two feature values are not equal, a penalty is added to the
303 score. The model is defined to exhibit depolarization block if $I_{maxNumAP}$ is not the highest
304 amplitude tested, and if there exists a current intensity higher than $I_{maxNumAP}$, for which
305 the model does not fire action potentials during the last 100 ms of its voltage response.

306 In the experiment the V_{eq} feature is extracted from the voltage response of the
307 pyramidal cells to the current injection one step above I_{th} (or $I_{max_num_AP}$ in the test).
308 Both in the experiment and in this test this is calculated as the mean voltage over the last 100
309 ms of the voltage trace. However, in the test, before calculating this value it is examined
310 whether there are any action potentials during this period. The presence of spikes here means
311 that the model did not enter depolarization block prior to this period. In these cases the test
312 iterates further on the voltage traces corresponding to larger current steps to find if there is
313 any where the model actually entered depolarization block; if an appropriate trace is found,

314 the value of V_{eq} is extracted there. This trace is the response to the current intensity one step
315 above $I_{below_depol_block}$.

316 If the model does not enter depolarization block, a penalty is applied, and the final
317 score gets the value of 100. Otherwise, the final score achieved by the model on this test is the
318 average of the feature scores (Z-scores) for the features described above, plus an additional
319 penalty if $I_{maxNumAP}$ and $I_{below_depol_block}$ differ. This penalty is 200 times the
320 difference between the two current amplitude values (in pA – which in this case is 10 times
321 the number of examined steps between them).

322

323 **The Back-propagating AP Test**

324

325 This test evaluates the strength of action potential back-propagation in the apical trunk
326 at locations of different distances from the soma. The observation data for this test were
327 yielded by the digitization of Figure 1B of [27], using the DigitizeIt software [46]. The values
328 were then averaged over distances of 50, 150, 250, 350 ± 20 μm from the soma to get the
329 mean and standard deviation of the features. The features tested here are the amplitudes of the
330 first and last action potentials of a 15 Hz spike train, measured at the 4 different dendritic
331 locations.

332 The test automatically finds current amplitudes for which the soma fires, on average,
333 between 10-20 Hz and chooses the amplitude that leads to firing nearest to 15 Hz. For this
334 task, the following algorithm was implemented. Increasing current step stimuli of 0.0 - 1.0 nA
335 amplitude with a step size of 0.1 nA are applied to the model and the number of spikes is
336 counted for each resulting voltage trace. If spontaneous spiking occurs (i.e., if there are spikes
337 even when no current is injected) or if the spiking rate does not reach 10 Hz even for the

338 highest amplitude, the test quits with an error message. Otherwise the amplitudes for which
339 the soma fires between 10 and 20 Hz are appended to a list and (if the list is not empty) the
340 one providing the spiking rate nearest to 15 Hz is chosen. If the list is empty because the
341 spiking rate is smaller than 10 Hz for a step amplitude but higher than 20 Hz for the next step,
342 a binary search method is used to find an appropriate amplitude in this range.

343 This test uses a trunk section list (or generates one if the `find_section_lists`
344 variable of the `ModelLoader` is set to `True` – see the section ‘Classifying the apical sections
345 of pyramidal cells’ below) to automatically find the dendritic locations for the measurements.
346 The desired distances of the locations from the soma and the distance tolerance are read from
347 the input configuration file, and must agree with the distances and the tolerance over which
348 the experimental data were averaged. All the trunk dendritic segments whose distance from
349 the soma falls into one of the distance ranges are selected. The locations and also their
350 distances are then returned in separate dictionaries.

351 Then the soma is stimulated with a current injection of the previously chosen
352 amplitude and the voltage response of the soma and the selected dendritic locations are
353 recorded and returned.

354 The test implements its own function to extract the amplitudes of back-propagating
355 action potentials, but the method is based on eFEL features. This is needed because eFEL’s
356 spike detection is based on a given threshold value for spike initiation, which may not be
357 reached by the back-propagating signal at more distant regions. First the maximum
358 depolarization of the first and the last action potentials are calculated. This is the maximum
359 value of the voltage trace in a time interval around the somatic action potential, based on the
360 start time of the spike (using the `AP_begin_time` feature of eFEL) and the inter-spike interval
361 to the next spike recorded at the soma. Then the amplitudes are calculated as the difference
362 between this maximum value and the voltage at the begin time of the spike (on the soma)

363 minus 1 ms (which is early enough not to include the rising phase of the spike, and late
364 enough in the case of the last action potential not to include the afterhyperpolarization of the
365 previous spike).

366 To calculate the feature scores the amplitude values are first averaged over the
367 distance ranges to be compared to the experimental data and get the feature Z-scores. The
368 final score here is the average of the Z-scores achieved for the features of first and last action
369 potential amplitudes at different dendritic distances. In the result it is also stated whether the
370 model is more like a strongly or a weakly propagating cell in the experiment, where they
371 found examples of both types [27].

372

373 **The PSP Attenuation Test**

374

375 The PSP Attenuation test evaluates how much the post-synaptic potential attenuates as it
376 propagates from different dendritic locations to the soma in CA1 pyramidal cell models. The
377 *observation* data for this test were yielded by the digitization of Figure 1E and Figure 2B of
378 Magee and Cook, 2000 [29] using the DigitizeIt software [46]. The somatic and dendritic
379 depolarization values were then averaged over distances of 100, 200, 300 ± 50 μm from the
380 soma and the soma/dendrite attenuation was calculated to get the mean and standard deviation
381 of the attenuation features at the three different input distances. The digitized data and the
382 script that calculates the feature means and standard deviations, and creates the JSON file are
383 available here:
384 https://github.com/sasaray/HippoUnit_demo/tree/master/target_features/Examples_on_creatin
385 [g_JSON_files/Magee2000-PSP_att/](#).

386 In this test the apical trunk receives excitatory post-synaptic current (EPSC)-shaped
387 current stimuli at locations of different distances from the soma. The maximum depolarization
388 caused by the input is extracted at the soma and divided by the maximum depolarization at the
389 location of the stimulus to get the soma/dendrite attenuation values that are then averaged in
390 distance ranges of 100, 200, 300 \pm 50 μ m and compared to the experimental data. The
391 distances and tolerance are defined in the configuration file and must agree with how the
392 *observation* data were generated.

393 The test uses a trunk section list, which needs to be specified in the NEURON HOC
394 model (or the test generates one if the `find_section_lists` variable of the `ModelLoader`
395 is set to True – see the section ‘Classify apical sections of pyramidal cells’ below) to find the
396 dendritic locations to be stimulated. Randomly selected dendritic locations are used because
397 the distance ranges that are evaluated cover almost the whole length of the trunk of a
398 pyramidal cell. The probability of selecting a given dendritic segment is set to be proportional
399 to its length. The number of dendritic segments examined can be chosen by the user by setting
400 the `num_of_dend_locations` argument of the test. The random seed (also an argument of
401 the test) must be kept constant to make the selection reproducible. If a given segment is
402 selected multiple times (or it is closer than 50 μ m or further than 350 μ m), a new random
403 number is generated. If the number of locations to be selected is more than the number of
404 trunk segments available in the model, all the segments are selected.

405 The *Exp2Syn* synaptic model of NEURON with a previously calculated weight is used to
406 stimulate the dendrite. The desired EPSC amplitude and time constants are given in the input
407 configuration file according to the experimental protocol. To get the proper synaptic weight,
408 first the stimulus is run with weight = 0. The last 10% of the trace is averaged to get the
409 resting membrane potential (V_m). Then the synaptic weight required to induce EPSCs with
410 the experimentally determined amplitude is calculated according to Equation 1:

411 (1) $\text{weight} = -\text{EPSC_amp} / V_m$

412 where EPSC_amp is read from the `config` dictionary, and the synaptic reversal potential is
413 assumed to be 0 mV.

414 To get the somatic and dendritic maximum depolarization from the voltage traces, the
415 baseline trace (weight = 0) is subtracted from the trace recorded in the presence of the input.
416 To get the attenuation ratio the maximum value of the somatic depolarization is divided by the
417 maximum value of the dendritic depolarization.

418 To calculate the feature scores the soma/dendrite attenuation values are first averaged
419 over the distance ranges to be compared to the experimental data to get the feature Z-scores.
420 The final score is the average of the feature scores calculated at the different dendritic
421 locations.

422

423 **The Oblique Integration Test**

424

425 This test evaluates the signal integration properties of radial oblique dendrites,
426 determined by providing an increasing number of synchronous (0.1 ms between inputs) or
427 asynchronous (2 ms between inputs) clustered synaptic inputs. The experimental mean and
428 standard error (SE) of the features examined are available in the paper of Losonczy and
429 Magee [33] and are read from a JSON file into the *observation* dictionary of the test. The SE
430 values are then converted to standard deviation values. The following features are tested:
431 voltage threshold for dendritic spike initiation (defined as the expected somatic depolarization at
432 which a step-like increase in peak dV/dt occurs); proximal threshold (defined the same way as
433 above, but including only those results in the statistics where the proximal part of the examined
434 dendrite was stimulated); distal threshold; degree of nonlinearity at threshold; suprathreshold

435 degree of nonlinearity; peak derivative of somatic voltage at threshold; peak amplitude of somatic
436 EPSP; time to peak of somatic EPSP; degree of nonlinearity in the case of asynchronous inputs.

437 The test automatically selects a list of oblique dendrites that meet the criteria of the
438 experimental protocol, based on a section list containing the oblique dendritic sections (this
439 can either be provided by the HOC model, or generated automatically if the
440 `find_section_lists` variable of the `ModelLoader` is set to `True` – see the section
441 ‘Classify apical sections of pyramidal cells’ below). For each selected oblique dendrite a
442 proximal and a distal location is examined. The criteria for the selection of dendrites, which
443 were also applied in the experiments, are the following. The selected oblique dendrites should
444 be terminal dendrites (they have no child sections) and they should be at most 120 μm from
445 the soma. This latter criterion can be changed by the user by changing the value of the
446 `ModelLoader`’s `max_dist_from_soma` variable, and it can also increase automatically if
447 needed. In particular, if no appropriate oblique is found up to the upper bound provided, the
448 distance is increased iteratively by 15 μm , but not further than 190 μm .

449 Then an increasing number of synaptic inputs are activated at the selected dendritic
450 locations separately, while recording the local and somatic voltage response. `HippoUnit`
451 provides a default synapse model to be used in the `ObliqueIntegrationTest`. If the
452 `AMPA_name`, and `NMDA_name` variables are not set by the user, the default synapse is used. In
453 this case the AMPA component of the synapse is given by the built-in `Exp2Syn` synapse of
454 `NEURON`, while the NMDA component is defined in an NMODL (.mod) file which is part of
455 the `HippoUnit` package. This NMDA receptor model uses a Jahr-Stevens voltage dependence
456 [47] and rise and decay time constants of 3.3 and 102.38 ms, respectively. The time constant
457 values used here are temperature- (Q10-) corrected values from [42]. Q10 values for the rise
458 and decay time constants were 2.2 [48] and 1.7 [49], respectively. The model’s own AMPA
459 and NMDA receptor models can also be used in this test if their NMODL files are available

460 and compiled among the other mechanisms of the model. In this case the `AMPA_name`, and
461 `NMDA_name` variables need to be provided by the user. The time constants of the built-in
462 `Exp2Syn` AMPA component and the AMPA/NMDA ratio can be adjusted by the user by
463 setting the `AMPA_tau1`, `AMPA_tau2` and `AMPA_NMDA_ratio` parameter of the
464 `ModelLoader`. The default AMPA/NMDA ratio is 2.0 from [42], and the default
465 `AMPA_tau1` and `AMPA_tau2` are 0.1 ms and 2.0 ms, respectively [29,30].

466 To test the Poirazi et al. 2003 model using its own receptor models, we also had to
467 implement a modified version of the synapse functions of the `ModelLoader` that can deal
468 with the different (pointer-based) implementation of synaptic activation in this model. For this
469 purpose, a child class was implemented that inherits from the `ModelLoader` class. This
470 modified version is not part of the official `HippoUnit` version, because this older, more
471 complicated implementation of synaptic models is not generally used anymore; however, this
472 is a good example on how one can modify the capability methods of `HippoUnit` to match their
473 own models or purposes. The code for this modified `ModelLoader` is available here:
474 [https://github.com/KaliLab/HippoUnit_demo/blob/master/ModelLoader_Poirazi_2003_CA1.](https://github.com/KaliLab/HippoUnit_demo/blob/master/ModelLoader_Poirazi_2003_CA1.py)
475 [py](https://github.com/KaliLab/HippoUnit_demo/blob/master/ModelLoader_Poirazi_2003_CA1.py).

476 The synaptic weights for each selected dendritic location are automatically adjusted by
477 the test using a binary search algorithm so that the threshold for dendritic spike generation is 5
478 synchronous inputs – which was the average number of inputs that had to be activated by
479 glutamate uncaging to evoke a dendritic spike in the experiments [33]. This search runs in
480 parallel for all selected dendritic locations. The search interval of the binary search and the
481 initial step size of the searching range can be adjusted by the user through the `c_minmax` and
482 `c_step_start` variables of the `ModelLoader`. During the iterations of the algorithm the
483 step size may decrease if needed; a lower threshold for the step size (`c_step_stop` variable

484 of the `ModelLoader`) must be set to avoid infinite looping. Those dendritic locations where
485 this first dendritic spike generates a somatic action potential, or where no dendritic spike can
486 be evoked, are excluded from further analysis. To let the user know, this information is
487 displayed on the output and also printed into the log file saved by the test. Most of the features
488 above are extracted at the threshold input level (5 inputs).

489 The final score of this test is the average of the feature scores achieved by the model
490 for the different features; however, a T-test analysis is also available as a separate score type
491 for this test.

492

493 **Parallel computing**

494

495 Most of the tests of HippoUnit require multiple simulations of the same model, either
496 using stimuli of different intensities or at different locations in the cell. To run these
497 simulations in parallel and save time, the Python `multiprocessing.Pool` module is used.
498 The size of the pool can be set by the user. Moreover, all NEURON simulations are
499 performed in multiprocessing pools to ensure that they run independently of each other, and to
500 make it easy to erase the models after the process has finished. This is especially important in
501 the case of HOC templates in order to avoid previously loaded templates running in the
502 background and the occurrence of ‘Template cannot be redefined’ errors when the same
503 model template is loaded again.

504

505 **Classifying the apical sections of pyramidal cells**

506

507 Some of the validation tests of HippoUnit require lists of sections belonging to the
508 different dendritic types of the apical tree (main apical trunk, apical tuft dendrites, radial
509 oblique dendrites). To classify the dendrites NeuroM [50] is used as a base package. NeuroM
510 contains a script that, starting from the tuft (uppermost dendritic branches in Fig 1) endpoints,
511 iterates down the tree to find a single common ancestor. This is considered as the apical point.
512 The apical point is the upper end of the main apical dendrite (trunk), from where the tuft
513 region arises. Every dendrite branching from the trunk below this point is considered an
514 oblique dendrite.

515 However, there are many CA1 pyramidal cell morphologies where the trunk bifurcates
516 close to the soma to form two or even more branches. In these cases the method described

517 above finds this proximal bifurcation point as the apical point (see Fig 1A). To overcome this
518 issue, we worked out and implemented a method to find multiple apical points by iterating the
519 function provided by NeuroM. In particular, if the initial apical point is closer to the soma
520 than a pre-defined threshold, the function is run again on subtrees of the apical tree where the
521 root node of the subtree is the previously found apical point, to find apical points on those
522 subtrees (see Fig 1B). When (possibly after multiple iterations) apical points that are far
523 enough from the soma are found, NeuroM is used to iterate down from them on the parent
524 sections, which will be the trunk sections (blue dots in Fig 1C). Iterating up, the tuft sections
525 are found (green dots in Fig 1C), and the other descendants of the trunk sections are
526 considered to be oblique dendrites (yellow dots in Fig 1C). Once all the sections are
527 classified, their NeuroM coordinates are converted to NEURON section information for
528 further use.

529

530

531 Fig 1: Classifying the apical dendrites of pyramidal cells. Morphological reconstruction made within the HBP at
532 University College London (UCL). The soma is marked in black, the red dendrites underneath are the basal
533 dendrites, apical dendrites are colored purple. (A) The original method of NeuroM finds a single apical point
534 which is actually a bifurcation of the trunk. (B) Further developing the method, multiple apical points can be
535 found. (C) The apical dendritic sections are classified. Blue: trunk, yellow: oblique dendrites, green: tuft
536 sections.

537

538 We note that this function can only be used for hoc models that load their
539 morphologies from a separate morphology file (e.g., ASC, SWC) as NeuroM can only deal
540 with morphologies provided in these standard formats. For models with NEURON

541 morphologies implemented directly in the hoc language, the SectionLists required by a given
542 test should be implemented within the model.

543

544 **Models from literature**

545

546 In this paper we demonstrate the utility of the HippoUnit validation test suite by
547 applying its tests to validate and compare the behavior of several different detailed
548 hippocampal CA1 pyramidal cell models available on ModelDB [18]. For this initial
549 comparison we chose models published by several modeling groups worldwide that were
550 originally developed for various purposes. The models compared were the following: the
551 Golding et al., 2001 model [27] (ModelDB accession number: 64167), the Katz et al., 2009
552 model [51] (ModelDB accession number: 127351), the Migliore et al., 2011 model [52]
553 (ModelDB accession number: 138205), the Poirazi et al., 2003 model [6,53] (ModelDB
554 accession number: 20212), the Bianchi et al., 2012 model [25] (ModelDB accession number:
555 143719), and the Gómez González et al., 2011 [54] model (ModelDB accession number:
556 144450).

557 Models from literature that are published on ModelDB typically implement their own
558 simulations and plots to make it easier for users and readers to reproduce and visualize the
559 results shown in the corresponding paper. Therefore, to be able to test the models described
560 above using our test suite, we needed to create standalone versions of them. These standalone
561 versions do not display any GUI, or contain any built-in simulations and run-time
562 modifications, but otherwise their behavior should be identical to the published version of the
563 models. We also added section lists of the radial oblique and the trunk dendritic sections to
564 those models where this was not done yet, as some of the tests require these lists. To ensure

565 that the standalone versions have the same properties as the original models, we checked their
566 parameters after running their built-in simulations (in case including any run-time
567 modifications), and made sure they match the parameters of the standalone version. The
568 modified models used for running validation tests are available in this GitHub repository:
569 https://github.com/KaliLab/HippoUnit_demo.

570 **Results**

571 **The HippoUnit validation suite**

572

573 HippoUnit (<https://github.com/KaliLab/hippounit>) is an open source test suite for the
574 automatic and quantitative evaluation of the behavior of neural single cell models. The tests of
575 HippoUnit automatically perform simulations that mimic common electrophysiological
576 protocols on neuronal models to compare their behavior with quantitative experimental data
577 using various feature-based error functions. Current validation tests cover somatic
578 (subthreshold and spiking) behavior as well as signal propagation and integration in the
579 dendrites. These tests were chosen because they collectively cover diverse functional aspects
580 of cellular behavior that have been thoroughly investigated in experimental and modeling
581 studies, and particularly because the necessary experimental data were available in sufficient
582 quality and quantity. However, we note that the currently implemented tests, even in
583 combination, probably do not fully constrain the behavior of the cell under all physiological
584 conditions, and thus the test suite can be further improved by including additional tests and
585 more experimental data. The tests were developed using data and models for rat hippocampal
586 CA1 pyramidal cells. However, most of the tests are directly applicable to or can be adapted

587 for other cell types if the necessary experimental data are available; examples of this will be
588 presented in later sections.

589 HippoUnit is implemented in the Python programming language, and is based on the
590 SciUnit [19] framework for testing scientific models. The current version of HippoUnit is
591 capable of handling single cell models implemented in the NEURON simulator, provided that
592 they do not apply any runtime modification, do not have a built-in graphical user interface,
593 and do not automatically perform simulations. Meeting these conditions may require some
594 modifications in the published code of the model. Once such a “standalone” version of the
595 model is available, the tests of HippoUnit can be run by adapting and using the example
596 Jupyter notebooks described in S1 Appendix, without any further coding required from the
597 user. In principle, neural models developed using other software tools can also be tested by
598 HippoUnit; however, this requires the re-implementation by the user of the interface functions
599 that allow HippoUnit to run the necessary simulations and record their output (see the
600 Methods section for more details).

601 In the current tests of HippoUnit, once all the necessary simulations have been
602 performed and the responses of the model have been recorded, electrophysiological features
603 are extracted from the voltage traces, and the discrepancy between the model’s behavior and
604 the experiment is computed by comparing the feature values with those extracted from the
605 experimental data (see Methods). Biological variability is taken into account by measuring the
606 difference between the feature value for the model and the mean of the feature in the
607 experiments in units of the standard deviation for that particular feature observed in the
608 experiments. For simplicity, we refer to the result of this comparison as the feature score;
609 however, we note that there are many possible sources of such discrepancy including, among
610 others, experimental artefacts and noise, shortcomings of the models, and differences between
611 the conditions assumed by the models and those in the actual experiments (see the Discussion

612 for more details). The final score of a given test achieved by a given model is given by the
613 average (or, in some cases, the sum) of the feature scores for all the features evaluated by the
614 test.

615 Besides the final score, which is the basic output of all the tests, the tests of HippoUnit
616 typically provide a number of other useful outputs (see Methods), including figures that
617 visualize the model's behavior through traces and plot the feature and feature score values
618 compared to the experimental data. It is always strongly recommended to look at the traces
619 and other figures to get a fuller picture of the model's response to the stimuli, which helps
620 with the correct interpretation of validation results. Such closer inspection also makes it
621 possible to detect possible test failures, when the extraction of certain features does not work
622 correctly for a given model.

623 HippoUnit can also take advantage of the parallel execution capabilities of modern
624 computers. When tests require multiple simulations of the same model using different settings
625 (e.g., different stimulation intensities or different stimulus locations in the cell), these
626 simulations are run in parallel, which can make the validation process substantially faster,
627 depending on the available computing resources.

628 One convenient way of running a test on a model is to use an interactive
629 computational notebook, such as the Jupyter Notebook [55], which enables the combination
630 of program codes to be run (we used Python code to access the functionality of HippoUnit),
631 the resulting outputs (e.g. figures, tables, text) and commentary or explanatory text in a single
632 document. Therefore, we demonstrate the usage of HippoUnit through this method (See S1
633 Appendix and https://github.com/KaliLab/HippoUnit_demo).

634

635 **Comparison of the behavior of rat hippocampal CA1 pyramidal cell models**
636 **selected from the literature**

637

638 We selected six different publications containing models of hippocampal CA1
639 pyramidal cells whose implementations for the NEURON simulator were available in the
640 ModelDB database. Our aim was to compare the behavior of every model to the experimental
641 target data using the tests of HippoUnit, which also allowed us to compare the models to each
642 other, and to test their generalization performance in paradigms that they were not originally
643 designed to capture. These models differ in their complexity regarding the number and types
644 of ion channels that they contain, and they were built for different purposes.

645 The Golding et al., 2001 model [27] was developed to show the dichotomy of the
646 back-propagation efficacy and the amplitudes of the back-propagating action potentials at
647 distal trunk regions in CA1 pyramidal cells and to make predictions on the possible causes of
648 this behavior. It contains only the most important ion channels (Na , K_{DR} , K_{A}) needed to
649 reproduce the generation and propagation of action potentials. [26]

650 The Katz et al., 2009 model [51] is based on the Golding et al. 2001 model and was
651 built to investigate the functional consequences of the distribution of strength and density of
652 synapses on the apical dendrites that they observed experimentally, for the mode of dendritic
653 integration.

654 The Migliore et al., 2011 model [52] was used to study schizophrenic behavior. It is
655 based on earlier models of the same modeling group, which were used to investigate the
656 initiation and propagation of action potentials in oblique dendrites, and have been validated
657 against different electrophysiological data.

658 The Poirazi et al., 2003 model [6,53] was designed to clarify the issues about the
659 integrative properties of thin apical dendrites that may arise from the different and sometimes
660 conflicting interpretations of available experimental data. This is a quite complex model in the
661 sense that it contains a large number of different types of ion channels, whose properties were
662 adjusted to fit in vitro experimental data, and it also contains four types of synaptic receptors.

663 The Bianchi et al., 2012 model [25] was designed to investigate the mechanisms
664 behind depolarization block observed experimentally in the somatic spiking behavior of CA1
665 pyramidal cells. It was developed by combining and modifying the Shah et al., 2008 [56] and
666 the Poirazi et al. 2003 models [6,53]. The former of these was developed to show the
667 significance of axonal M-type potassium channels.

668 The Gómez González et al., 2011 [54] model is based on the Poirazi et al. 2003 model
669 and it was modified to replicate the experimental data of [33] on the nonlinear signal
670 integration of radial oblique dendrites when the inputs arrive in a short time window.

671 A common property of these models is that their parameters were set using manual
672 procedures with the aim of reproducing the behavior of real CA1 PCs in one or a few specific
673 paradigms. As some of them were built by modifying and further developing previous
674 models, these share the same morphology (see Fig. 2). On the other hand, the model of
675 Gómez González et al. 2011 was adjusted to 5 different morphologies, which were all tested.
676 In the case of the Golding et al. 2001 model, we tested three different versions (shown in
677 Figures 8A, 8B and 9A of the corresponding paper [27]) that differ in the distribution of the
678 sodium and the A-type potassium channels, and therefore the back-propagation efficacy of the
679 action potentials. The morphologies and characteristic voltage responses of all the models
680 used in this comparison are displayed in Fig 2.

681

682

683 Fig 2: The morphologies of the different models tested and their voltage responses to a 400 ms somatic step
684 current injection of 0.6 nA amplitude. (Some of the models share the same morphology, while the Gómez
685 González et al. 2011 model was adjusted to five different morphologies.)

686

687 Running the tests of HippoUnit on these models we took into account the original
688 settings of the simulations of the models, and set the `v_init` (the initial voltage when the
689 simulation starts), and the `celsius` (the temperature at which the simulation is done)
690 variables accordingly. For the Bianchi et al 2012 model we used variable time step integration
691 during all the simulations, as it was done in the original modeling study. For the other models
692 a fixed time step were used (`dt=0.025 ms`).

693

694 **Somatic Features Test**

695

696 Using the Somatic Features Test of HippoUnit, we compared the behavior of the
697 models to both patch clamp recordings (patch clamp dataset) and sharp electrode recordings
698 (sharp electrode dataset). After performing a review of the relevant literature, we conclude
699 that the patch clamp dataset is in good agreement with experimental observations available in
700 the literature (see Table 1 in Methods), and will be used as a representative example in this
701 study.

702 The two datasets used in this study (sharp electrode dataset, patch clamp dataset) differ
703 not only in the recording technique, but also in the simulation protocol. In the sharp electrode
704 recordings, the cells received 400 ms-long depolarizing and hyperpolarizing current
705 injections, using amplitudes of 0.2, 0.4, 0.6, 0.8 and 1.0 nA in both directions. In the patch
706 clamp recordings, both the depolarizing and the hyperpolarizing current injections were 300
707 ms long and 0.05, 0.1, 0.15, 0.2, 0.25 nA in amplitude.

708 As each of the tested models apparently used experimental data obtained from patch
709 clamp recordings as a reference, here we show the detailed results of the test on the models
710 when their output was compared to the features extracted from the patch clamp data (we will
711 return to the comparison between the two datasets near the end of this section). During these
712 recordings the cells were stimulated with relatively low amplitude current injections. Some of
713 the examined models (Migliore et al. 2011, Gómez González et al. 2011 n125 morphology)
714 did not fire even for the highest amplitude tested. Some other models started to fire for higher
715 current intensities than it was observed experimentally. In these cases the features that
716 describe action potential shape or timing properties cannot be evaluated for the given model
717 (for the current amplitudes affected). Therefore, besides the final score achieved by the
718 models on this test (the average Z-score for the successfully evaluated features – see Methods
719 for details), we also consider the proportion of the successfully evaluated features as an
720 important measure of how closely the model matches this specific experimental dataset (Fig
721 4B).

722 Fig 3 shows how the extracted feature values of the somatic response traces of the
723 different models fit the experimental values. It is clear that the behavior of the different
724 models is very diverse. Each model captures some of the experimental features but shows a
725 larger discrepancy for others.

726

727

728 Fig 3: Feature values from the Somatic Features Test of HippoUnit applied to several published models.
729 Absolute feature values extracted (using the electrophys Feature Extraction Library (eFEL)) from the voltage
730 responses of the models to somatic current injections of varying amplitude, compared to mean experimental
731 values (black X) that were extracted from the patch clamp dataset. Black solid, horizontal lines indicate the
732 experimental standard deviation. Colored solid, horizontal lines typically show the standard deviation of spiking
733 features of models, where the feature value of each action potential in the voltage trace is extracted and

734 averaged. Feature names (y axis labels) are indicated as they are used in eFEL combined with the step current
735 injection amplitude. Not all the evaluated features are shown here. (The (s) and (w) notations of the Golding et
736 al. 2001 models in the legend indicate the strong and weak propagating versions of the model.)

737

738 The resting membrane potential (*voltage_base*) for all of the models was apparently
739 adjusted to a more hyperpolarized value than in the experimental recordings we used for our
740 comparison, and most of the models also return to a lower voltage value after the step stimuli
741 (*steady_state_voltage*). An exception is the Poirazi et al. 2003 model, where the decay time
742 constant after the stimulus is unusually high (this feature is not included in Fig 3, but the slow
743 decay can be seen in the example trace in Fig 2, and detailed data are available here:
744 https://github.com/KaliLab/HippoUnit_demo). The voltage threshold for action potential
745 generation (*AP_begin_voltage*) is lower than the experimental value for most of the models
746 (that were able to generate action potentials in response to the examined current intensities),
747 but it is higher than the experimental value for most versions of the Gómez González et al.
748 2011 model. For negative current steps most of the models gets more hyperpolarized
749 (*voltage_deflection*) (the most extreme is the Gómez González et al. 2011 model with the
750 n129 morphology), while the Gómez González et al. 2011 model with the n125 morphology
751 and the Migliore et al. 2011 model get less hyperpolarized than it was observed
752 experimentally. The sag amplitudes are also quite high for the Gómez González et al. 2011
753 n129, and n130 models, while the Katz et al. 2009, and all versions of the Golding et al. 2001
754 models basically have no hyperpolarizing sag.

755 It is quite conspicuous how much the amplitude of the action potentials (*APlast_amp*,
756 *AP_amplitude*, *AP2_amp*) differs in the Gómez González et al. 2011 models from the
757 experimental values and from the other models as well. The Katz et al. 2009 and one of the
758 versions (Fig 8A) of the Golding et al. 2001 model have slightly too high action potential

759 amplitudes, and these models have relatively small action potential width (*AP_width*). On the
760 other hand, the rising phase (*AP_rise_time*, *AP_rise_rate*) of the Katz et al. 2009 model
761 appears to be too slow.

762 Looking at the inverse interspike interval (*ISI*) values, it can be seen that the
763 experimental spike trains show adaptation in the ISIs, meaning that the first ISI is smaller (the
764 inverse ISI is higher) than the last ISI for the same current injection amplitude. This behavior
765 can be observed in the case of the Katz et al. 2009 model, three versions (n128, n129, n130
766 morphology) of the Gómez González et al. 2011 model, but cannot really be seen in the
767 Bianchi et al. 2011, the Poirazi et al. 2003 and the three versions of the Golding et al. 2001
768 models. At first look it may seem contradictory that in the case of the Gómez González et al.
769 2011 model version n129 morphology the spike counts are quite low, while the mean
770 frequency and the inverse ISI values are high. This is because the soma of this model does not
771 fire over the whole period of the stimulation, but starts firing at higher frequencies, then stops
772 firing for rest of the stimulus (see Fig 2). The Katz et al. 2009 model fires quite a high number
773 of action potentials (*Spikecount*) compared to the experimental data, at a high frequency.

774 In the experimental recordings there is a delay before the first action potential is
775 generated, which becomes shorter with increasing current intensity (indicated by the
776 *inv_time_to_first_spike* feature that becomes larger with increasing input intensity). In most
777 of the models this behavior can be observed, albeit to different degrees. The Katz et al. 2009
778 model has the shortest delays (highest *inv_time_to_first_spike* values), but the effect is still
779 visible.

780 To quantify the difference between the experimental dataset and the simulated output
781 of the models, these were compared using the feature-based error function (Z-Score)
782 described above to calculate the feature score. Fig 4A shows the mean scores of the model
783 features whose absolute values are illustrated in Fig 3 (averaged over the different current step

784 amplitudes examined). From this figure it is even more clearly visible that each model fits
785 some experimental features well but does not capture others. For example, it is quite
786 noticeable in Fig 4A that most of the versions of the Gómez González et al. 2011 model
787 (greenish dots) perform well for features describing action potential timing (upper part of the
788 figure, e.g., *ISIs*, *mean_frequency*, *spikecount*), but get higher feature scores for features of
789 action potential shape (lower part of the figure, e.g., *AP_rise_rate*, *AP_rise_time*,
790 *AP_fall_rate*, *AP_fall_time*, *AP amplitudes*). Conversely, the Katz et al. 2009 model achieved
791 better scores for AP shape features than for features describing AP timing. It is also worth
792 noting that none of the feature scores for the model of Migliore et al. 2011 was higher than 4;
793 however, looking at Fig 4B it can be seen that less than half of the experimental features were
794 successfully evaluated in this model, which is because it does not fire action potentials for the
795 current injection amplitudes examined here.

796

797

798 Fig 4: Evaluation of results from the Somatic Features Test of HippoUnit applied to published models. (A)
799 Mean feature scores (the difference between the model's feature value and the experimental mean in units of the
800 experimental SD) of the different models. Feature score values are averaged over the different input step
801 amplitudes. (B) The bars represent the number of features that were attempted to be evaluated for the models
802 (i.e., the number of features extracted from the experimental patch clamp dataset). The number of successfully
803 evaluated features for the various models is shown in green, and the number of features that could not be
804 evaluated for a particular model is shown in red. Features that are not evaluated successfully are most often
805 spiking features at step amplitudes for which the tested model does not fire action potentials.

806

807 Besides enabling the comparison of different models regarding how well they match a
808 particular dataset, the tests of HippoUnit also allow one to determine the match between a
809 particular model and several datasets of the same type. As experimental results can be heavily

810 influenced by recording conditions and protocols, and also depend on factors such as the
811 strain, age, and sex of the animal, it is important to find out whether the same model can
812 simultaneously capture the outcome of different experiments, and if not, how closely it is able
813 to match the different datasets. As a practically relevant example, we looked at how well the
814 various published models that we were testing captured a different experimental dataset that
815 also contained current clamp recordings from rat CA1 PCs, but which was obtained using
816 sharp electrodes rather than the whole-cell patch clamp technique [3]. We therefore evaluated
817 all the models with the Somatic Features Test of HippoUnit using both datasets, and then
818 compared the results.

819 When we simply compared the raw outputs of the test for each model evaluated using
820 the two different data sets (Fig 5A) we identified two factors that substantially bias the results.
821 First, we found that the standard deviation values for the features extracted from the two
822 datasets are very different in magnitude; more specifically, the patch clamp recording data set
823 had much lower standard deviation values for most of the features. This results in relatively
824 higher feature scores achieved by the models, as the difference of the model output from the
825 experimental features is given in the unit of the experimental standard deviation. The other
826 source of bias is that the two datasets were recorded not only using different recording
827 methods – patch clamp and sharp electrode – but (partly as a consequence) also using
828 different protocols (current amplitudes, current duration etc.), and therefore provide different
829 sets of features. As an important example, voltage traces with and without action potentials
830 clearly provide different types of features. Also note that the same electrophysiological
831 parameter can often be extracted from multiple voltage traces, and these are all treated as
832 separate features in the test, so a difference in the number of recorded traces automatically
833 leads to a difference in the set of features. Consequently, the models are not compared to
834 exactly the same set of features in the two cases. Mainly as a result of these two confounding

835 factors, comparison of the raw scores of the models for the two data sets (Fig 5A) appears to
836 indicate that most models fit the dataset obtained from sharp electrode recordings better, even
837 though these models were typically built mostly based on patch clamp data.

838

839

840 Fig 5: Comparison of the final scores achieved by the different models on the Somatic Features Test against
841 validation data from two different datasets (sharp electrode data, patch clamp data). Final scores are calculated as
842 the average of all the feature scores. In the upper panel (A) the raw output of the tests is shown, while in the
843 lower panel (B) the feature scores and the final scores have been recalculated using standardized standard
844 deviation values. Numbers above each data point show the proportion of the successfully evaluated features
845 compared to the number of features attempted to be evaluated (successfully extracted from the data set). Note
846 that while in the recalculated final scores (B) only those eFEL features were taken into account that could be
847 extracted from both datasets, they are extracted for different current step amplitudes, which accounts for the
848 difference in the number of observation features for the two datasets.

849

850 To overcome these issues and make unbiased comparisons of the models to the two
851 datasets, the feature scores and the final scores were recalculated in the following way (Fig
852 5B). The new feature scores for the two different data sets were calculated as the difference of
853 the model's feature value from the mean feature value of each dataset (as before), but divided
854 by a common standard deviation value. This standardized SD value for each eFEL feature was
855 the mean of the standard deviation values over the current steps in the patch clamp dataset
856 (the results were qualitatively similar if we used the SD values from the sharp electrode
857 dataset everywhere instead). Averaging the standard deviation values of the eFEL features
858 over the current steps was required because the current step amplitudes were not the same in
859 the two data sets, and we therefore needed to define SD values that were independent of the
860 amplitude. To get rid of the second bias, only those eFEL features were used in the final score

861 recalculation that are present in both observations (sharp electrode and patch clamp datasets)
862 for at least one current step amplitude. (This change had the side effect of significantly
863 decreasing the final score for the Poirazi et al. 2003 model because the feature
864 *decay_time_constant_after_stim* was excluded here, as it could not be extracted from the
865 sharp electrode data.) Now that the final scores are recalculated to get rid of most of the
866 biasing factors, it becomes clear that the somatic behavior of every model fits the patch clamp
867 data better (Fig 5B).

868 It is worth noting that one biasing factor still remains in the last comparison: as it has
869 already been mentioned, not all the observation features can be evaluated for each of the
870 models, especially when they are compared to the patch clamp data set, which uses smaller
871 currents. To allow the assessment of the potential effect of this issue, the proportion of the
872 successfully evaluated features relative to the number of features attempted to be evaluated
873 (successfully extracted from the data set) for each model is also shown in Fig 5 next to each
874 data point.

875

876 **Depolarization Block Test**

877

878 In the Depolarization Block Test three features are evaluated. Two of them examine
879 the threshold current intensity to reach depolarization block. The *I_maxNumAP* feature is the
880 current intensity at which the model fires the maximum number of action potentials, and the
881 *I_below_depol_block* feature is the current intensity one step before the model enters
882 depolarization block. Both are compared to the experimental I_{th} feature because, in the
883 experiment [25], the number of spikes increased monotonically with increasing current
884 intensity up to the current amplitude where the cell entered depolarization block during the

885 stimulus, which led to a drop in the number of action potentials. By contrast, we experienced
886 that some models started to fire fewer spikes for higher current intensities while still firing
887 over the whole period of the current step stimulus, i.e., without entering depolarization block.
888 Therefore, we introduced the two separate features for the threshold current. If these two
889 feature values are not equal, a penalty is added to the score. The third evaluated feature is V_{eq} ,
890 the equilibrium potential during the depolarization block, which is calculated as the average
891 of the membrane potential over the last 100 ms of a current pulse with amplitude 50 pA above
892 $I_{maxNumAP}$ (or 50 pA above $I_{below_depol_block}$ if its value is not equal to
893 $I_{maxNumAP}$). Each model has a value for the $I_{maxNumAP}$ feature, while those models that
894 do not enter depolarization block are not supposed to have a value for the
895 $I_{below_depol_block}$ feature and the V_{eq} feature.

896 The results from applying the Depolarization Block Test to the models from ModelDB
897 are shown in Fig 6. According to the test, four of the models entered depolarization block.
898 However, by looking at the actual voltage traces provided by the test, it becomes apparent that
899 only the Bianchi et al. 2011 model behaves correctly (which was developed to show this
900 behavior). The other three models actually managed to “cheat” the test.

901

902

903 Fig 6: Results from the Depolarization Block Test of HippoUnit applied to published models. (A) Number of
904 APs fired by the models in response to 1 sec long somatic current injections of increasing intensity. (B)
905 Depolarization block feature values extracted from the voltage responses of the models compared to the
906 experimental observations. $exp_{I_{th}}$ is the mean (SD is indicated with a solid line) of the experimentally observed
907 threshold current amplitude to reach depolarization block. In the test two separate features are compared to the
908 experimental threshold value: The $I_{maxNumAP}$ feature is the current intensity at which the model fires the
909 maximum number of action potentials, and the $I_{below_depol_block}$ feature is the current intensity one step
910 before the model enters depolarization block. According to the experimental observation, these two values are

911 supposed to be the same, but for models, they may differ, in which case a penalty is added to the final score (see
912 the text for more details). The V_{eq} is the equilibrium potential to which the somatic voltage settles after entering
913 depolarization block. (C) Voltage traces of different models that were recognized by the test as depolarization
914 block. Note that only the Bianchi et al. 2012 model actually entered depolarization block, the others “cheated”
915 the test (see the text for more details).

916

917 In the case of the Katz et al. 2009 and the Golding et al. 2001 Fig 9B models, the APs
918 get smaller and smaller with increasing stimulus amplitude until they get so small that they do
919 not reach the threshold for action potential detection; therefore, these APs are not counted by
920 the test and V_{eq} is also calculated. The Gómez González et al. 2011 model adjusted to the
921 n129 morphology does not fire during the whole period of the current stimulation for a wide
922 range of current amplitudes (see Fig 2). Increasing the intensity of the current injection it fires
923 an increasing number of spikes, but always stops after a while before the end of the stimulus.
924 On the other hand, there is a certain current intensity after which the model starts to fire fewer
925 action potentials, and which is thus detected as $I_{maxNumAP}$ by the test. Because no action
926 potentials can be detected during the last 100 ms of the somatic response one step above the
927 detected “threshold” current intensity, the model is declared to have entered depolarization
928 block, and a V_{eq} value is also extracted.

929 In principle, it would be desirable to modify the test so that it correctly rejects the
930 three models above. However, the models described above shows so similar behavior to
931 depolarization block that is hard to distinguish using automatic methods. Furthermore, we
932 have made substantial efforts to make the test more general and applicable to a wide variety
933 of models with different behavior, and we are concerned that defining and adding further
934 criteria to the test to deal with these specific cases would be an ad hoc solution, and would
935 possibly cause further ‘cheats’ when applied to other models with unexpected behavior. These

936 cases underline the importance of critically evaluating the full output (especially the figures of
937 the recorded voltage traces) of the tests rather than blindly accepting the final scores provided.
938

939 **Back-propagating Action Potential Test**

940

941 This test first finds all the dendritic segments that belong to the main apical dendrite of
942 the model and which are 50, 150, 250, 350 \pm 20 μ m from the soma, respectively. Then a train
943 of action potentials of frequency around 15 Hz is triggered in the soma by injecting a step
944 current of appropriate amplitude (as determined by the test), and the amplitudes of the first
945 and last action potentials in the train are measured at the selected locations. In the Bianchi et
946 al. 2012 and the Poirazi et al. 2003 models (which share the same morphology, see Fig 2) no
947 suitable trunk locations could be found in the most proximal (50 \pm 20 μ m) and most distal
948 (350 \pm 20 μ m) regions. This is because this morphology has quite long dendritic sections that
949 are divided into a small number of segments. In particular, the first trunk section
950 (apical_dendrite[0]) originates from the soma, is 102.66 μ m long, and has only two segments.
951 The center of one of them is 25.67 μ m far from the soma, while the other is already 77 μ m
952 away from the soma. None of these segments belongs to the 50 \pm 20 μ m range, and therefore
953 they are not selected by the test. The n123 morphology of the Gómez González et al. 2011
954 model has the same shape (Fig 2), but in this case the segments are different, and therefore it
955 does not share the same problem.

956 At the remaining, successfully evaluated distance ranges in the apical trunk of the
957 Bianchi et al. 2012 model, action potentials propagate very actively, barely attenuating. For
958 the *API_amp* and *APlast_amp* features at these distances, this model has the highest feature
959 score (Fig 7), while the Poirazi et al. 2003 model performs quite well.

960

961

962 Fig 7: Results from the Back-propagating Action Potential Test of HippoUnit applied to published models. (A)
963 The amplitudes of the first back-propagating action potentials (in a train of spikes with frequency around 15 Hz
964 evoked by somatic current injection) as a function of recording location distance from the soma. (B) Feature
965 scores achieved by the different models on the Back-propagating AP Test. The amplitudes of the first and last
966 back-propagating action potentials were averaged over the distance ranges of 50, 150, 250, 350 \pm 20 μ m and
967 compared to the experimental features (see Methods for more details).

968

969 The Golding et al. 2001 model was designed to investigate how the distribution of ion
970 channels can affect the back-propagation efficacy in the trunk. The two versions of the
971 Golding et al. 2001 model (“fig8A” and “fig9B” versions) which are supposed to be weakly
972 propagating according to the corresponding paper [27], are also weakly propagating according
973 to the test. However, the difference between their strongly and weakly propagating feature
974 scores is not too large (Fig 7), which is probably caused by the much smaller standard
975 deviation value of the experimental data for the weakly propagating case. Although the
976 amplitudes of the first action potentials of these two models fit the experimental data
977 relatively well, they start to decline slightly closer to the soma than it was observed
978 experimentally, as the amplitudes are already very small at 250 \pm 20 μ m (Fig 7). (In Fig 7 the
979 data corresponding to these two versions of the model are almost completely overlapping for
980 more distal regions.) The amplitudes for the last action potential fit the data well, except in the
981 most proximal regions (see the relatively high feature score in Fig 7 B or the detailed results
982 here: https://github.com/KaliLab/HippoUnit_demo). For all versions of the Golding et al.
983 2001 model, AP amplitudes are too high at the most proximal distance range. As for the
984 strongly propagating version of the Golding et al. 2001 model (“fig8B” version), the
985 amplitude of the first action potential is too high at the proximal locations, but further it fits

986 the data well. The amplitude of the last action potential remains too high even at more distal
987 locations. It is worth noting that, in the corresponding paper [27], they only examined a single
988 action potential triggered by a 5 ms long input in their simulations, and did not examine or
989 compare to their data the properties of the last action potential in a longer spike train. Finally,
990 we note that in all versions of the Golding et al. 2001 model a spike train with frequency
991 around 23 Hz was evoked and examined as it turned out to be difficult to set the frequency
992 closer to 15 Hz.

993 The different versions of the Gómez González et al. 2011 model behave qualitatively
994 similarly in this test, although there were smaller quantitative differences. In almost all
995 versions the amplitudes of the first action potential in the dendrites are slightly too low at the
996 most proximal locations but fit the experimental data better at further locations. The
997 exceptions are the versions with the n128 and n129 morphologies, which have lower first
998 action potential amplitudes at the furthest locations, but not low enough to be considered as
999 weak propagating. The amplitudes for the last action potential are too high at the distal
1000 regions but fit better at the proximal ones. The only exception is the one with morphology
1001 n129, where the last action potential attenuates more at further locations and fits the data
1002 better.

1003 In the case of the Katz et al. 2009 model, a spike train with frequency around 40 Hz
1004 was examined, as the firing frequency increases so suddenly with increasing current intensity
1005 in this model that no frequency closer to 15 Hz could be adjusted. In this model the last action
1006 potential propagates too strongly, while the dendritic amplitudes for the first action potential
1007 are close to the experimental values.

1008 In the Migliore et al. 2011 model the amplitudes for the last action potential are too
1009 high, while the amplitude of the first back-propagating action potential is too low at locations
1010 in the $250 \pm 20 \mu\text{m}$ and $350 \pm 20 \mu\text{m}$ distance ranges.

1011 Finally, all the models that we examined were found to be strongly propagating by the
1012 test, with the exception of those versions of the Golding et al. 2001 model that were explicitly
1013 developed to be weakly propagating.

1014

1015 **PSP Attenuation Test**

1016

1017 In this test the extent of the attenuation of the amplitude of an excitatory post-synaptic
1018 potential (EPSP) is examined as it propagates towards the soma from different input locations
1019 in the apical trunk. The Katz et al. 2009, the Bianchi et al. 2012, and all versions of the
1020 Golding et al. 2001 models perform quite well in this test. The various versions of the
1021 Golding et al. 2001 model are almost identical in this respect, which is not surprising as they
1022 differ only in the distribution of the sodium and A-type potassium channels. This shows that,
1023 as we would expect, these properties do not have much effect on the propagation of relatively
1024 low-amplitude signals such as unitary PSPs. Interestingly, the different versions of the Gómez
1025 González et al. 2011 model, with different morphologies, behave quite differently, which
1026 shows that this behavior can depend very much on the morphology of the dendritic tree.

1027

1028

1029 Fig 8: Results from the PSP Attenuation Test of HippoUnit applied to published models. Soma/dendrite EPSP
1030 attenuation as a function of the synaptic input distance from the soma in the different models.

1031

1032 **Oblique Integration Test**

1033

1034 This test probes the integration properties of the radial oblique dendrites of CA1
1035 pyramidal cell models. The test is based on the experimental results described in [33]. In this
1036 study, the somatic voltage response was recorded while synaptic inputs in single oblique dendrites
1037 were activated in different spatio-temporal combinations using glutamate uncaging. The main
1038 finding was that a sufficiently high number of synchronously activated and spatially clustered
1039 inputs produced a supralinear response consisting of a fast (Na) and a slow (NMDA) component,
1040 while asynchronously activated inputs summed linearly or sublinearly.

1041 This test selects all the radial oblique dendrites of the model that meet the
1042 experimental criteria: they are terminal dendrites (they have no child sections) and are at most
1043 120 μm from the soma. Then the selected dendrites are stimulated in a proximal and in a
1044 distal region (separately) using an increasing number of clustered, synchronous or
1045 asynchronous synaptic inputs to get the voltage responses of the model, and extract the
1046 features of dendritic integration. The synaptic inputs are not unitary inputs, i.e., their strength
1047 is not equivalent to the strength of one synapse in the real cell; instead, the strength is adjusted
1048 in a way that 5 synchronous inputs are needed to trigger a dendritic action potential. The
1049 intensity of the laser used for glutamate uncaging was set in a similar way in the experiments
1050 [33]. Most of the features were extracted at this just-suprathreshold level of input. We noticed
1051 that in some cases the strength of the synapse is not set correctly by the test; for example, it
1052 may happen that an actual dendritic spike does not reach the spike detection threshold in
1053 amplitude, or sometimes the EPSP may reach the threshold for spike detection without actual
1054 spike generation. The user has the ability to set the threshold used by eFEL for spike
1055 detection, but sometimes a single threshold may not work even for the different oblique
1056 dendrites (and proximal and distal locations in the same dendrites) of a single model. For
1057 consistency, we used the same spike detection threshold of -20 mV for all the models.

1058 The synaptic stimulus contains an AMPA and an NMDA receptor-mediated
1059 component. As the default synapse, HippoUnit uses the Exp2Syn double exponential synapse
1060 built into NEURON for the AMPA component, and its own built-in NMDA receptor model,
1061 whose parameters were set according to experimental data from the literature (see the
1062 Methods section for more details). In those models that originally do not have any synaptic
1063 component (the Bianchi et al 2011 model and all versions of the Golding et al. 2001 model)
1064 this default synapse was used. Both the Katz et al. 2009 and the Migliore et al. 2011 models
1065 used the Exp2Syn in their simulations, so in their case the time constants of this function were
1066 set to the values used in the original publications. As these models did not contain NMDA
1067 receptors, the default NMDA receptor model and the default AMPA/NMDA ratio of
1068 HippoUnit were used. The Gómez González et al 2011 and the Poirazi et al. 2003 models
1069 have their own AMPA and NMDA receptor models and their own AMPA/NMDA ratio
1070 values to be tested with.

1071 As shown by the averaged “measured EPSP vs expected EPSP” curves in Fig 9, all
1072 three versions of the Golding et al. 2001 model have a jump in the amplitude of the somatic
1073 response at the threshold input level, which is the result of the generation of dendritic spikes.
1074 However, even these larger average responses do not reach the supralinear region, as it would
1075 be expected according to the experimental observations [33]. The reason for this discrepancy
1076 is that a dendritic spike was generated in the simulations in only a subset of the stimulated
1077 dendrites; in the rest of the dendrites tested, the amplitude of the EPSPs went above the spike
1078 detection threshold during the adjustment of the synaptic weight without actually triggering a
1079 dendritic spike, which led to the corresponding synaptic strength being incorrectly set for that
1080 particular dendrite. Averaging over the results for locations with and without dendritic spikes
1081 led to an overall sublinear integration profile.

1082

1083

1084 Fig 9: Results from the Oblique Integration Test of HippoUnit applied to published models. (A) Comparison of
1085 the responses of the models to experimental results (black X) according to features of dendritic integration.
1086 Features values are given as mean and standard deviation, as several dendritic locations of each model are tested.
1087 (B) The averaged input – output curves of all the dendritic locations examined. EPSP amplitudes are measured at
1088 the soma. Dashed line shows linearity. In models whose curve goes above the dashed line, oblique dendrites
1089 integrate synaptic inputs that are spatially and temporally clustered supralinearly.

1090

1091 The Migliore et al. 2011 model performs quite well on this test. In this case, seven
1092 dendrites could be tested out of the ten dendrites within the correct distance range because, in
1093 the others, the dendritic spike at the threshold input level also elicited a somatic action
1094 potential, and therefore these dendrites were excluded from further testing.

1095 In the Katz et al. 2009 model all the selected dendritic locations could be tested, and in
1096 most of them the synaptic strength could be adjusted appropriately. For a few dendrites, some
1097 input levels higher than the threshold for dendritic spike generation also triggered somatic
1098 action potentials. This effect causes the high supralinearity in the “measured EPSP vs
1099 expected EPSP” curve in Fig 9, but has no effect on the extracted features.

1100 In the Bianchi et al. 2012 model only one dendrite could be selected, in which very
1101 high amplitude dendritic spikes were evoked by the synaptic inputs, making the signal
1102 integration highly supralinear.

1103 In the Poirazi et al. 2003 model also only one dendrite could be selected based on its
1104 distance from the soma; furthermore, only the distal location could be tested even in this
1105 dendrite, as at the proximal location the dendritic action potential at the threshold input level
1106 generated a somatic action potential. However, at the distal location, the synaptic strength
1107 could not be set correctly. For the synaptic strength chosen by the test, the actual threshold
1108 input level where a dendritic spike is first generated is at 4 inputs, but this dendritic AP is too

1109 small in amplitude to be detected, and the response to 5 inputs is recognized as the first
1110 dendritic spike instead. Therefore, the features that should be extracted at the threshold input
1111 level are instead extracted from the voltage response to 5 inputs. In this model this results in a
1112 reduced *supralinearity* value, as this feature is calculated one input level higher than the
1113 actual threshold. In addition, for even higher input levels dendritic bursts can be observed,
1114 which causes large *supralinearity* values in the “measured EPSP vs expected EPSP” curve in
1115 Fig 9, but this does not affect the feature values.

1116 Models from Gómez González et al. 2011 were expected to be particularly relevant for
1117 this test, as these models were tuned to fit the same data set on which this test is based.
1118 However, we encountered an important issue when comparing our test results for these
1119 models to the results shown in the paper [54]. In particular, the paper clearly indicates which
1120 dendrites were examined, and it is stated that those are at maximum 150 μm from the soma.
1121 However, when we measured the distance of these locations from the soma by following the
1122 path along the dendrites (as it is done by the test of HippoUnit), we often found it to be larger
1123 than 150 μm . We note that when the distance was measured in 3D coordinates rather than
1124 along the dendrites, all the dendrites used by Gómez González et al. 2011 appeared to be
1125 within 150 μm of the soma, so we assume that this definition was used in the paper. As we
1126 consider the path distance to be more meaningful than Euclidean distance in this context, and
1127 this was also the criterion used in the experimental study, we consistently use path distance in
1128 HippoUnit to find the relevant dendritic segments. Nevertheless, this difference in the
1129 selection of dendrites should be kept in mind when the results of this validation for models of
1130 Gómez González et al. 2011 are evaluated.

1131 In two versions of the Gómez González et al. 2011 model (those that were adjusted to
1132 the n123 and n125 morphologies) only one oblique dendrite matched the experimental criteria
1133 and could therefore be selected, and these are not among those that were studied by the

1134 developers of the model. In each of these cases the dendritic spike at the proximal location at
1135 the input threshold level triggered a somatic action potential, and therefore only the distal
1136 location could be tested. In the case of the n125 morphology, the dendritic spikes that appear
1137 first for just-suprathreshold input are so small in amplitude that they do not reach the spike
1138 detection threshold (-20 mV), and are thus not detected. Therefore, the automatically adjusted
1139 synaptic weight is larger than the appropriate value would be, which results in larger somatic
1140 EPSPs than expected (see Fig 9). With this synaptic weight, the first dendritic spike and
1141 therefore the jump to the supralinear region in the “measured EPSP vs expected EPSP” curve
1142 is for 4 synaptic inputs instead of 5. This is also the case in one of the two selected dendrites
1143 of the version of this model with the n128 morphology. Similarly to the Poirazi et al. 2003
1144 model, this results in a lower *degree of nonlinearity at threshold* feature value, than it would
1145 be if the feature were extracted at the actual threshold input level (4 inputs) instead of the one
1146 which the test attempted to adjust (5 inputs). The *suprathreshold nonlinearity* feature has a
1147 high value because at that input level (6 inputs), somatic action potentials are triggered.

1148 In the version of the Gómez González et al. 2011 model that uses the n129
1149 morphology, 10 oblique dendrites could be selected for testing (none of them is among those
1150 that its developers used) but only 4 could be tested because, for the rest, the dendritic spike at
1151 the threshold input level already elicits a somatic action potential. The synaptic weights
1152 required to set the threshold input level to 5 are not found correctly in most cases; the actual
1153 threshold input level is at 4 or 3. Suprathreshold nonlinearity is high, because at that input
1154 level (6 inputs) somatic action potentials are triggered for some of the examined dendritic
1155 locations.

1156 The version of the Gómez González et al. 2011 model that uses the n130 morphology
1157 achieves the best (lowest) final score on this test. In this model many oblique dendrites could
1158 be selected and tested, including two (179, 189) that the developers used in their simulations

1159 [54]. In most cases the synaptic weights are nicely found to set the threshold input level to 5
1160 synapses. For some dendrites there are somatic action potentials at higher input levels, but
1161 that does not affect the features.

1162 The value of the *time to peak* feature for each model is much smaller than the
1163 experimental value (Fig 9). This is because in each of the models the maximum amplitude of
1164 the somatic EPSP is determined by the fast component, caused by the appearance of the
1165 dendritic sodium spikes, while in the experimental observation this is rather shaped by the
1166 slow NMDA component following the sodium spike.

1167

1168 **Overall characterization and model comparison based on all tests of HippoUnit**

1169

1170 In summary, using HippoUnit, we compared the behavior of several hippocampal CA1
1171 pyramidal cell models available on ModelDB in several distinct domains, and found that all of
1172 these models match experimental results well in some domains (typically those that they were
1173 originally built to capture) but fit the experimental observations less precisely in others. Fig
1174 10 summarizes the final scores achieved by the different models on the various tests (lower
1175 scores indicate a better match in all cases).

1176

1177

1178 Fig 10: Normalized final scores achieved by the different published models on the various tests of HippoUnit.
1179 The final scores of each test are normalized by dividing the scores of each model by the best achieved score on
1180 the given test.

1181

1182 Perhaps a bit surprisingly, the different versions of the Golding et al. 2001 model
1183 showed a good match to the experimental data in all of the tests (except for the Depolarization

1184 Block Test), even though these are the simplest ones among the models in the sense that they
1185 contain the smallest number of different types of ion channels. On the other hand, these
1186 models do not perform outstandingly well on the Back-propagating Action Potential Test,
1187 although they were developed to study the mechanisms behind (the dichotomy of) action
1188 potential back-propagation, which is evaluated by this test based on the data that were
1189 published together with these models [27]. The most probable reason for this surprising
1190 observation is that, in the original study, only a few features of the model's response were
1191 compared with the experimental results. HippoUnit tested the behavior of the model based on
1192 a larger set of experimental features from the original study, and was therefore able to
1193 uncover differences between the model's response and the experimental data on features for
1194 which the model was not evaluated in the source publication.

1195 The Bianchi et al. 2012 model is the only one that can produce real depolarization
1196 block within the range of input strengths examined by the corresponding test. The success of
1197 this model in this test is not surprising because this is the only model that was tuned to
1198 reproduce this behavior; on the other hand, the failure of the other models in this respect
1199 clearly shows that proper depolarization block requires some combination of mechanisms that
1200 are at least partially distinct from those that allow good performance in the other tests. The
1201 Bianchi et al. 2012 model achieves a relatively high final score only on the Back-propagating
1202 Action Potential Test, as action potentials seem to propagate too actively in its dendrites,
1203 leading to high AP amplitudes even in more distal compartments.

1204 The Gómez González et al. 2011 models were developed to capture the same
1205 experimental observations on dendritic integration that are tested by the Oblique Integration
1206 Test of HippoUnit, but, somewhat surprisingly, some of its versions achieved quite high
1207 feature scores on this test, while others perform quite well. This is partly caused by the fact
1208 that HippoUnit often selects different dendritic sections for testing from those that were

1209 studied by the developers of these models (see above for details). The output of HippoUnit
1210 shows that the different oblique dendrites of these models can show quite diverse behavior,
1211 and beyond those studied in the corresponding paper [54], other oblique dendrites do not
1212 necessarily match the experimental observations. Some of its versions also perform
1213 relatively poorly on the PSP-Attenuation Test, similar to the Migliore et al. 2011 and the
1214 Poirazi et al. 2003 models. The Katz et al. 2009 model is not outstandingly good in any of the
1215 tests, but still achieves relatively good final scores everywhere (although its apparent good
1216 performance on the Depolarization Block Test is misleading - see detailed explanation above).

1217 The model files that were used to test the models described above, the detailed
1218 validation results (all the output files of HippoUnit), and the Jupyter Notebooks that show
1219 how to run the tests of HippoUnit on these models are available in the following Github
1220 repository: https://github.com/KaliLab/HippoUnit_demo.

1221

1222 **Application of HippoUnit to models built using automated parameter** 1223 **optimization within the Human Brain Project**

1224

1225 Besides enabling a detailed comparison of published models, HippoUnit can also be
1226 used to monitor the performance of new models at various stages of model development.
1227 Here, we illustrate this by showing how we have used HippoUnit within the HBP to
1228 systematically validate detailed multi-compartmental models of hippocampal neurons
1229 developed using multi-objective parameter optimization methods implemented by the open
1230 source Blue Brain Python Optimization Library (BluePyOpt [16]). To this end, we extended
1231 HippoUnit to allow it to handle the output of optimization performed by BluePyOpt (see
1232 Methods).

1233 Models of CA1 pyramidal cells were optimized using target feature data extracted
1234 from the same sharp electrode dataset [3] that was also one of the datasets used by the
1235 Somatic Features Test of HippoUnit. However, while during validation all the eFEL features
1236 that could be successfully extracted from the data are considered, only a subset of these
1237 features was used in the optimization (mostly those that describe the rate and timing of the
1238 spikes; e.g., the different inter-spike interval (ISI), time to last/first spike, mean frequency
1239 features).

1240 In addition, sharp electrode measurements were also available for several types of
1241 interneuron in the hippocampal CA1 region, and models of these interneurons were also
1242 constructed using similar automated methods [3]. Using the appropriate observation file and
1243 the stimulus file belonging to it, the Somatic Features Test of HippoUnit can also be applied
1244 to these models to evaluate their somatic spiking features. The other tests of HippoUnit are
1245 currently not applicable to interneurons, mostly due to the lack of appropriate target data.

1246 We applied the tests of HippoUnit to the version of the models published in [3], and
1247 to a later version (v4) described in Ecker et al. (2020)[57], which was intended to further
1248 improve the dendritic behavior of the models, as this is critical for their proper functioning in
1249 the network. The two sets of models were created using the same morphology files and
1250 similar optimization methods and protocols. These new optimizations differed mainly in the
1251 allowed range for the density of the sodium channels in the dendrites. For the pyramidal cell
1252 models a new feature was also introduced in the parameter optimization that constrains the
1253 amplitudes of back-propagating action potentials in the main apical dendrite. The new
1254 interneuron models also had an exponentially decreasing (rather than constant) density of Na
1255 channels, and A-type K channels with more hyperpolarized activation in their dendrites. For
1256 more details on the models, see the original publications ([3,57]).

1257 After running all the tests of HippoUnit on both sets of models generated by
1258 BluePyOpt, we performed a comparison of the old [3] and the new versions of the models by
1259 doing a statistical analysis of the final scores achieved by the models of the same cell type on
1260 the different tests. In Fig 11 the median, the interquartile range and the full range of the final
1261 scores achieved by the two versions of the model set are compared. According to the results
1262 of the Wilcoxon signed-rank test the new version of the models achieved significantly better
1263 scores on the Back-propagating Action Potential test ($p = 0.0046$), on the Oblique Integration
1264 Test ($p = 0.0033$), and on the PSP Attenuation Test ($p = 0.0107$), which is the result of
1265 reduced dendritic excitability. Moreover, in most of the other cases the behavior of the models
1266 improved slightly (but not significantly) with the new version. Only in the case of the Somatic
1267 Features test applied to bAC interneurons did the new models perform slightly worse (but still
1268 quite well), and this difference was not significant ($p = 0.75$).

1269 These results show the importance of model validation performed against
1270 experimental findings, especially those not considered when building the model, in every
1271 iteration during the process of model development. This approach can greatly facilitate the
1272 construction of models that perform well in a variety of contexts, help avoid model
1273 regression, and guide the model building process towards a more robust and general
1274 implementation.

1275

1276

1277 Fig 11: Employing the tests of HippoUnit to monitor the behavior of a set of detailed data-driven models of
1278 hippocampal neurons at different stages of model development. Models of four different cell types (pyramidal
1279 cells and continuous accommodating (int cAC), bursting accommodating (int bAC) and continuous non-
1280 accommodating (int cNAC) interneurons) of the hippocampal CA1 region were developed within the Human
1281 Brain Project by automated optimization using BluePyOpt. The tests of HippoUnit were used to evaluate and
1282 compare the behavior of the older (Migliore et al 2018) version and the new (v4) version of these models. The

1283 median, the interquartile range and the full range of the final scores achieved by the models of each cell type
1284 were calculated and the results of the two versions of the model set are compared. Asterisks indicate significant
1285 differences (*: $p < 0.05$; **: $p < 0.01$).

1286

1287 **Integration of HippoUnit into the Validation Framework and the Brain** 1288 **Simulation Platform of the Human Brain Project**

1289

1290 The HBP is developing scientific infrastructure to facilitate advances in neuroscience,
1291 medicine, and computing [58]. One component of this research infrastructure is the Brain
1292 Simulation Platform (BSP) (<https://bsp.humanbrainproject.eu>), an online collaborative
1293 platform that supports the construction and simulation of neural models at various scales. As
1294 we argued above, systematic, automated validation of models is a critical prerequisite of
1295 collaborative model development. Accordingly, the BSP includes a software framework for
1296 quantitative model validation and testing that explicitly supports applying a given validation
1297 test to different models and storing the results [59]. The framework consists of a web service,
1298 and a set of test suites, which are Python modules based on the SciUnit package. As we
1299 discussed earlier, SciUnit uses the concept of capabilities, which are standardized interfaces
1300 between the models to be tested and the validation tests. By defining the capabilities to which
1301 models must adhere, individual validation tests can be implemented independently of any
1302 specific model and used to validate any compatible model despite differences in their internal
1303 structures, the language and/or the simulator used. Each test must include a specification of
1304 the required model capabilities, the location of the reference (experimental) dataset, and data
1305 analysis code to transform the recorded variables (e.g., membrane potential) into feature
1306 values that allow the simulation results to be directly and quantitatively compared to the
1307 experimental data through statistical analysis. The web services framework [59] supports the

1308 management of models, tests, and validation results. It is accessible via web apps within the
1309 HBP Collaboratory, and also through a Python client. The framework makes it possible to
1310 permanently record, examine and reproduce validation results, and enables tracking the
1311 evolution of models over time, as well as comparison against other models in the domain.

1312 Every test of HippoUnit described in this paper has been individually registered in the
1313 Validation Framework. The JSON files containing the target experimental data for each test
1314 are stored (besides the HippoUnit_demo GitHub repository) in storage containers at the Swiss
1315 National Supercomputing Centre (CSCS), where they are publicly available. The location of
1316 the corresponding data file is associated with each registered test, so that the data are loaded
1317 automatically when the test is run on a model via the Validation Framework. As the Somatic
1318 Features Test of HippoUnit was used to compare models against five different data sets (data
1319 from sharp electrode measurements in pyramidal cells and interneurons belonging to three
1320 different electronic types, and data obtained from patch clamp recordings in pyramidal cells),
1321 these are considered to be and have been registered as five separate tests in the Validation
1322 Framework.

1323 All the models that were tested and compared in this study (including the CA1
1324 pyramidal cell models from the literature and the BluePyOpt optimized CA1 pyramidal cells
1325 and interneurons of the HBP) have been registered and are available in the Model Catalog of
1326 the Validation Framework with their locations in the CSCS storage linked to them. In addition
1327 to the modifications that were needed to make the models compatible with testing with
1328 HippoUnit (described in the section “Methods – Models from literature”), the versions of the
1329 models uploaded to the CSCS container also contain an `__init__.py` file. This file
1330 implements a python class that inherits all the functions of the `ModelLoader` class of
1331 HippoUnit without modification. Its role is to make the validation of these models via the
1332 Framework more straightforward by defining and setting the parameters of the `ModelLoader`

1333 class (such as the path to the HOC and NMODL files, the name of the section lists, etc.) that
1334 otherwise need to be set after instantiating the `ModelLoader` (see the `HippoUnit_demo`
1335 `GitHub` repository:
1336 https://github.com/KaliLab/HippoUnit_demo/tree/master/jupyter_notebooks).

1337 The validation results discussed in this paper have also been registered in the
1338 Validation Framework, with all their related files (output figures and JSON files) linked to
1339 them. These can be accessed using the Model Validation app of the framework.

1340 The Brain Simulation Platform of the HBP contains several online ‘Use Cases’, which
1341 are available on the platform and help the users to try and use the various established
1342 pipelines. The Use Case called ‘Hippocampus Single Cell Model Validation’ can be used to
1343 apply the tests of HippoUnit to models that were built using automated parameter
1344 optimization within the HBP.

1345 The Brain Simulation Platform also hosts interactive “Live Paper” documents that
1346 refer to published papers related to the models or software tools on the Platform. Live Papers
1347 provide links that make it possible to visualize or download results and data discussed in the
1348 respective paper, and even to run the associated simulations on the Platform. We have created
1349 a Live Paper ([https://humanbrainproject.github.io/hbp-bsp-live-](https://humanbrainproject.github.io/hbp-bsp-live-papers/2020/saray_et_al_2020/saray_et_al_2020.html)
1350 [papers/2020/saray_et_al_2020/saray_et_al_2020.html](https://humanbrainproject.github.io/hbp-bsp-live-papers/2020/saray_et_al_2020/saray_et_al_2020.html)) showing the results of the study
1351 presented in this paper in more detail. This interactive document provides links to all the
1352 output figures and data files resulting from the validation of the models from literature
1353 discussed here. This provides a more detailed insight into their behavior individually.
1354 Moreover, as part of this Live Paper a HippoUnit Use Case is also available in the form of a
1355 Jupyter Notebook, which guides the user through running the validation tests of HippoUnit on
1356 the models from literature that are already registered in the Framework, and makes it possible
1357 to reproduce the results presented here.

1358 **Discussion**

1359 **Applications of the HippoUnit test suite**

1360

1361 In this article, we have described the design, usage, and some initial applications of
1362 HippoUnit, a software tool that enables the automated comparison of the physiological
1363 properties of models of hippocampal neurons with the corresponding experimental results.
1364 HippoUnit, together with its possible extensions and other similar tools, allows the rapid,
1365 systematic evaluation and comparison of neuronal models in multiple domains. By providing
1366 the software tools and examples for effective model validation, we hope to encourage the
1367 modeling community to use more systematic testing during model development, with the aim
1368 of making the process of model building more efficient, reproducible and transparent.

1369 One important use case for the application of HippoUnit is the evaluation and
1370 comparison of existing models. We demonstrated this by using HippoUnit to test and compare
1371 the behavior of several models of CA1 pyramidal neurons available on ModelDB [18] in
1372 several distinct domains against electrophysiological data available in the literature (or shared
1373 by collaborators). Besides providing independent and standardized verification of the
1374 behavior of the models, the results also allow researchers to judge which existing models
1375 show a good match to the experimental data in the domains that they care about, and thus to
1376 decide whether they could re-use one of the existing models in their own research.

1377 HippoUnit is also a useful tool during model development. In a typical data-driven
1378 modeling scenario, researchers decide which aspects of model behavior are relevant for them,
1379 find experimental data that constrain these behaviors, then use some of these data to build the
1380 model, and use the rest of the data to validate the model. HippoUnit and similar test suites
1381 make it possible to define quantitative criteria for declaring a model valid (ideally before

1382 modeling starts), and to apply these criteria consistently throughout model development. We
1383 demonstrated this approach through the example of detailed single cell models of CA1
1384 pyramidal cells and interneurons optimized within the HBP.

1385 Furthermore, several authors have argued for the benefits of creating “community
1386 models” [7,60,61] through the iterative refinement of models in an open collaboration of
1387 multiple research teams. Such consensus models would aim to capture a wide range of
1388 experimental observations, and may be expected to generalize (within limits) to novel
1389 modeling scenarios. A prerequisite for this type of collaborative model development is an
1390 agreement on which experimental results will be used to constrain and validate the models.
1391 Automated test suites provide the means to systematically check models with respect to all the
1392 relevant experimental data, with the aim of tracking progress and avoiding “regression,”
1393 whereby previously correct model behavior is corrupted by further tuning.

1394 Finally, the tests of HippoUnit have been integrated into the recently developed
1395 Validation Framework of the HBP, which makes it possible to collect neural models and
1396 validation tests, and supports the application of the registered tests to the registered models.
1397 Most importantly, it makes it possible to save the validation results and link them to the
1398 models in the Model Catalog, making them publicly available and traceable for the modeling
1399 community.

1400

1401 **Interpreting the results of HippoUnit**

1402

1403 It is important to emphasize that a high final score on a given validation test using a
1404 particular experimental dataset does not mean that the model is not good enough or cannot be
1405 useful for a variety of purposes (including the ones it was originally developed for). The

1406 discrepancy between the target data and the model’s behavior, as quantified by the validation
1407 tests, may be due to several different reasons. First, all experimental data contain noise and
1408 may have systematic biases associated with the experimental methods employed. Sometimes
1409 the experimental protocol is not described in sufficient detail to allow its faithful reproduction
1410 in the simulations. It may also occur that a model is based on experimental data that were
1411 obtained under conditions that are substantially different from the conditions for the
1412 measurement of the validation target dataset. Using different recording techniques, such as
1413 sharp electrode or patch clamp recordings or the different circumstances of the experiments
1414 (e.g., the strain, age, and sex of the animal, or the temperature during measurement) can
1415 heavily affect the experimental results. Furthermore, the post-processing of the recorded
1416 electrophysiological data can also alter the results. For these reasons, probably no single
1417 model should be expected to achieve an arbitrarily low score on all of the validation tests
1418 developed for a particular cell type. Keeping this in mind, it is important that the modelers
1419 decide which properties of the cell type are relevant for them, and what experimental
1420 conditions they aim to mimic. Validation results should be interpreted or taken into account
1421 accordingly, and the tests themselves may need to be adapted.

1422 The issue of neuronal variability also deserves consideration in this context. The
1423 morphology, biophysical parameters, and physiological behavior of neurons is known to be
1424 non-uniform, even within a single cell type, and this variability may be important for the
1425 proper functioning and robustness of neural circuits[62–66]. Recent models of neuronal
1426 networks have also started to take into account this variability [11,67,68]. The tests of
1427 HippoUnit account for experimental variability by measuring the distance of the feature
1428 values of the model from the experimental mean (the feature score) in units of the
1429 experimental standard deviation. This means that any feature score less than about 1 actually
1430 corresponds to behavior which may be considered “typical” in the experiments (within one

1431 standard deviation of the mean), and a feature score of 2 or 3 may still be considered
1432 acceptable for any single model. In fact, even higher values of the feature score may
1433 sometimes be consistent with the data if the experimental distribution is long-tailed rather
1434 than normal. However, such high values of the feature score certainly deserve attention as
1435 they signal a large deviation from the typical behavior observed in the experiments.

1436 Furthermore, the acceptable feature score will generally depend on the goal of the
1437 modeling study. In particular, a study which intends to construct and examine a single model
1438 of typical experimental behavior should aim to keep all the relevant feature scores relatively
1439 low. On the other hand, when modeling entire populations of neurons, one should be prepared
1440 to accept a wider range of feature scores in some members of the model population, although
1441 the majority of the cells (corresponding to typical members of the experimental population)
1442 should still display relatively low scores. In fact, when modeling populations of neurons, one
1443 would ideally aim to match the actual distribution of neuronal features (including the mean,
1444 standard deviation, and possibly higher moments as well), and the distribution of feature
1445 scores (and actual feature values) from the relevant tests of HippoUnit actually provides the
1446 information that is necessary to compare the variability of the experimental and model cell
1447 populations.

1448

1449 **Uniform model formats reduce the costs of validation**

1450

1451 Although HippoUnit is built in a way that its tests are, in principle, model-agnostic, so
1452 that the implementation of the tests does not depend on model implementation, it still required
1453 a considerable effort to create the standalone versions of the models from literature to be
1454 tested, even though all of the selected models were developed for the NEURON simulator.

1455 This is because each model has a different file structure and internal logic that needs to be
1456 understood in order to create an equivalent standalone version. When the section lists of the
1457 main dendritic types do not exist, the user needs to create them by extensively analyzing the
1458 morphology and even doing some coding. In order to reduce the costs of systematic
1459 validation, models would need to be expressed in a format that is uniform and easy to test. As
1460 HippoUnit already has its capability functions implemented in a way that it is able to handle
1461 models developed in NEURON, the only requirement for such models is that they should
1462 contain a HOC file that describes the morphology (including the section lists for the main
1463 dendritic types of the dendritic tree) and all the biophysical parameters of the model, without
1464 any additional simulations, GUIs or run-time modifications. Currently, such a standalone
1465 version of the models is not made available routinely in publications or on-line databases, but
1466 could be added by the creators of the models with relatively little effort.

1467 On the other hand, applying the tests of HippoUnit to models built in other languages
1468 requires the re-implementation of the capability functions that are responsible for running the
1469 simulations on the model (see Methods). In order to save the user from this effort, it would be
1470 useful to publish neuronal models in a standard and uniform format that is simulator
1471 independent and allows general use in a variety of paradigms. This would allow an easier and
1472 more transparent process of community model development and validation, as it avoids the
1473 need of reimplementation of parts of software tools (such as validation suites), and the
1474 creation of new, (potentially) non-traced software versions. This approach is already initiated
1475 for neurons and neuronal networks by the developers of NeuroML [69], NineML [70], PyNN
1476 [71], Sonata [72], and Brian [73]. Once a large set of models becomes available in these
1477 standardized formats, it will be straightforward to extend HippoUnit (and other similar test
1478 suites) to handle these models.

1479

1480 **Extensibility of HippoUnit**

1481

1482 Although we were aiming to develop a test suite that is as comprehensive as possible,
1483 and that captures the most typical and basic properties of the hippocampal CA1 pyramidal
1484 cell, the list of features that can be tested by HippoUnit is far from complete. Upon
1485 availability of the appropriate quantitative experimental data, new tests addressing additional
1486 properties of the CA1 pyramidal cell could be included, for example, on the signal integration
1487 of the basal or the more distal apical dendrites, or on action potential initiation and
1488 propagation in the axon. Therefore, we implemented HippoUnit in a way that makes it
1489 possible to extend it by adding new tests.

1490 As HippoUnit is based on the SciUnit package [19] it inherits SciUnits's modular
1491 structure. This means that a test is usually composed of four main classes: the test class, the
1492 model class, the capabilities class and the score class (as described in more detail in the
1493 Methods section). Thanks to this structure it is easy to extend HippoUnit with new tests by
1494 implementing them in new test classes and adding the capabilities and scores needed. The
1495 methods of the new capabilities can be implemented in the `ModelLoader` class, which is a
1496 generalized `Model` class for models built in the NEURON simulator, or in a newly created
1497 `Model` class specific to the model to be tested.

1498 Adding new tests to HippoUnit requires adding the corresponding target experimental
1499 data as well in the form of a JSON file. The way the JSON files are created depends on the
1500 nature and source of the experimental data. In some cases the data may be explicitly provided
1501 in the text of the papers (as for the Oblique Integration and the Depolarization Block tests),
1502 therefore their JSON files are easy to make manually. Most typically, the data have to be
1503 processed to get the desired feature mean and standard deviation values and create the JSON
1504 file. In these cases it is worth writing a script that does this automatically. Some examples on

1505 how this was done for the current tests of HippoUnit are available here:
1506 https://github.com/sasaray/HippoUnit_demo/tree/master/target_features/Examples_on_creatin
1507 [g_JSON_files/](#).

1508 As HippoUnit is open-source and is shared on GitHub, it is possible for other
1509 developers, modelers or scientists to modify or extend the test suite working on their own
1510 forks of the repository. If they would like to directly contribute to HippoUnit, a ‘pull request’
1511 can be created to the main repository.

1512

1513 **Generalization possibilities of the tests of HippoUnit**

1514

1515 In the current version of HippoUnit most of the validation tests can only be used to test
1516 models of hippocampal CA1 pyramidal cells, as the observation data come from
1517 electrophysiological measurements of this cell type and the tests were designed to follow the
1518 experimental protocols of the papers from which these data derive. However, with small
1519 modifications most of the tests can be used for other cell types, or with slightly different
1520 stimulation protocols, if there are experimental data available for the features or properties
1521 tested.

1522 The Somatic Features Test can be used for any cell type and with any current step
1523 injection protocol even in its current form using the appropriate data and configuration files.
1524 These two files must be in agreement with each other; in particular, the configuration file
1525 should contain the parameters of the step current protocols (delay, duration, amplitude) used
1526 in the experiments from which the feature values in the data file derive. In this study this test
1527 was used with two different experimental protocols (sharp electrode measurements and patch

1528 clamp recordings that used different current step amplitudes and durations), and for testing
1529 four different cell types (hippocampal CA1 PC and interneurons).

1530 In the current version of the Depolarization Block Test the properties of the stimulus
1531 (delay, duration, amplitudes) are hard-coded to reproduce the experimental protocol used in a
1532 study of CA1 PCs [25]. However, the test could be easily modified to read these parameters
1533 from a configuration file like in the case of other tests, and then the test could be applied to
1534 other cell types if data from similar experimental measurements are available.

1535 As the Back-propagating AP Test examines the back-propagation efficacy of action
1536 potentials in the main apical dendrite (trunk), it is mainly suitable for testing pyramidal cell
1537 models; however, it can be used for PC models from other hippocampal or cortical regions,
1538 potentially using different distance ranges of the recording sites. If different distances are
1539 used, the feature names ('AP1_amp_X' and 'APlast_amp_X', where X is the recording
1540 distance) in the observation data file and the recording distances given in the stimuli file must
1541 be in agreement. Furthermore, it would also be possible to set a section list of other dendritic
1542 types instead of the trunk to be examined by the test. This way, models of other cell types
1543 (with dendritic trees qualitatively different from those of PCs) could also be tested. The
1544 frequency range of the spike train (10 – 20 Hz, preferring values closest to 15 Hz) is currently
1545 hard-coded in the function that automatically finds the appropriate current amplitude, but the
1546 implementation could be made more flexible in this case as well.

1547 The PSP Attenuation Test is quite general. Both the distances and tolerance values
1548 that determine the stimulation locations on the dendrites and the properties of the synaptic
1549 stimuli are given using the configuration file. Here again the feature names in the observation
1550 data file ('attenuation_soma/dend_x_um', where x is the distance from the soma) must fit the
1551 distances of the stimulation locations in the configuration file when one uses the tests with
1552 data from a different cell type or experimental protocol. Similarly to the Back-propagating AP

1553 Test the PSP Attenuation Test also examines the main apical dendrite (trunk), but could be
1554 altered to use section lists of other dendritic types.

1555 The Oblique Integration Test is very specific to the experimental protocol of [33].
1556 There is no configuration file used here, but the synaptic parameters (of the `ModelLoader`
1557 class) and the number of synapses to which the model should first generate a dendritic spike
1558 ('`threshold_index`' parameter of the test class) can be adjusted by the user after instantiating
1559 the `ModelLoader` and the test classes respectively. The time intervals between the inputs
1560 (synchronous (0.1 ms), asynchronous (2.0 ms)) are currently hard-coded in the test.

1561 HippoUnit has been used mainly to test models of rat hippocampal CA1 pyramidal
1562 cells as described above. However, having the appropriate observation data, most of its tests
1563 could easily be adapted to test models of different cell types, even in cases when the
1564 experimental protocol is slightly different from the currently implemented ones. The extent to
1565 which a test needs to be modified in order to test models of other cell types depends on how
1566 much the behavior of the new cell type differs from the behavior of CA1 pyramidal cells, and
1567 to what extent the protocol of the experiment differs from the ones we used as the bases of
1568 comparison in the current study.

1569

1570 **Acknowledgements**

1571 We thank Judit Makara and her group in the Laboratory of Neuronal Signaling,
1572 Institute of Experimental Medicine, Hungary for the patch clamp recording data used in this
1573 study. We also thank Luca Tar, a member of our group, for her help in testing the validation
1574 tests, and in literature review, and for useful discussions. We also would like to thank Michael
1575 Gevaert (Blue Brain Project) for providing the script that finds the apical point, and that were
1576 further developed for classifying apical sections.

1577

1578 **References**

- 1579 1. Einevoll GT, Destexhe A, Diesmann M, Grün S, Jirsa V, de Kamps M, et al. The
1580 Scientific Case for Brain Simulations. *Neuron*. 2019;102: 735–744.
1581 doi:10.1016/j.neuron.2019.03.027
- 1582 2. Káli S, Freund TF. Distinct properties of two major excitatory inputs to hippocampal
1583 pyramidal cells: A computational study. *European Journal of Neuroscience*. 2005;22:
1584 2027–2048. doi:10.1111/j.1460-9568.2005.04406.x
- 1585 3. Migliore R, Lupascu CA, Bologna LL, Romani A, Courcol JD, Antonel S, et al. The
1586 physiological variability of channel density in hippocampal CA1 pyramidal cells and
1587 interneurons explored using a unified data-driven modeling workflow. *PLoS*
1588 *Computational Biology*. 2018;14: 1–25. doi:10.1371/journal.pcbi.1006423
- 1589 4. Hay E, Hill S, Schürmann F, Markram H, Segev I. Models of neocortical layer 5b
1590 pyramidal cells capturing a wide range of dendritic and perisomatic active properties.
1591 *PLoS Computational Biology*. 2011;7. doi:10.1371/journal.pcbi.1002107
- 1592 5. Herz AVM, Gollisch T, Machens CK, Jaeger D. Modeling single-neuron dynamics
1593 and computations: A balance of detail and abstraction. *Science*. 2006;314: 80–85.
1594 doi:10.1126/science.1127240
- 1595 6. Poirazi P, Brannon T, Mel BW. Pyramidal neuron as two-layer neural network.
1596 *Neuron*. 2003;37: 989–999. doi:10.1016/S0896-6273(03)00149-1
- 1597 7. Bower JM. The 40-year history of modeling active dendrites in cerebellar purkinje
1598 cells: Emergence of the first single cell “community model.” *Frontiers in*
1599 *Computational Neuroscience*. 2015;9: 1–18. doi:10.3389/fncom.2015.00129

- 1600 8. Traub RD, Wong RKS, Miles R, Michelson H. A model of a CA3 hippocampal
1601 pyramidal neuron incorporating voltage-clamp data on intrinsic conductances. *Journal*
1602 *of Neurophysiology*. 1991;66: 635–650. doi:10.1152/jn.1991.66.2.635
- 1603 9. Almog M, Korngreen A. A quantitative description of dendritic conductances and its
1604 application to dendritic excitation in layer 5 pyramidal neurons. *Journal of*
1605 *Neuroscience*. 2014;34: 182–196. doi:10.1523/JNEUROSCI.2896-13.2014
- 1606 10. Almog M, Korngreen A. Is realistic neuronal modeling realistic? *Journal of*
1607 *Neurophysiology*. 2016;116: 2180–2209. doi:10.1152/jn.00360.2016
- 1608 11. Markram H, Muller E, Ramaswamy S, Reimann MW, Abdellah M, Sanchez CA, et
1609 al. Reconstruction and Simulation of Neocortical Microcircuitry. *Cell*. 2015;163: 456–
1610 492. doi:10.1016/j.cell.2015.09.029
- 1611 12. Traub RD, Contreras D, Cunningham MO, Murray H, LeBeau FEN, Roopun A, et al.
1612 Single-column thalamocortical network model exhibiting gamma oscillations, sleep
1613 spindles, and epileptogenic bursts. *Journal of Neurophysiology*. 2005;93: 2194–2232.
1614 doi:10.1152/jn.00983.2004
- 1615 13. Schneider CJ, Bezaire M, Soltesz I. Toward a full-scale computational model of the
1616 rat dentate gyrus. *Frontiers in Neural Circuits*. 2012;6: 1–8.
1617 doi:10.3389/fncir.2012.00083
- 1618 14. Bezaire MJ, Raikov I, Burk K, Vyas D, Soltesz I. Interneuronal mechanisms of
1619 hippocampal theta oscillations in a full-scale model of the rodent CA1 circuit. *eLife*.
1620 2016;5: 1–106. doi:10.7554/eLife.18566
- 1621 15. Friedrich P, Vella M, Gulyás AI, Freund TF, Káli S. A flexible, interactive software
1622 tool for fitting the parameters of neuronal models. *Frontiers in Neuroinformatics*.
1623 2014;8: 1–19. doi:10.3389/fninf.2014.00063

- 1624 16. van Geit W, Gevaert M, Chindemi G, Rössert C, Courcol JD, Muller EB, et al.
1625 BluePyOpt: Leveraging open source software and cloud infrastructure to optimise
1626 model parameters in neuroscience. *Frontiers in Neuroinformatics*. 2016;10: 1–18.
1627 doi:10.3389/fninf.2016.00017
- 1628 17. Vanier MC, Bower JM. A comparative survey of automated parameter-search
1629 methods for compartmental neural models. *Journal of Computational Neuroscience*.
1630 1999;7: 149–171. doi:10.1023/A:1008972005316
- 1631 18. McDougal RA, Morse TM, Carnevale T, Marenco L, Wang R, Migliore M, et al.
1632 Twenty years of ModelDB and beyond: building essential modeling tools for the future
1633 of neuroscience. *Journal of Computational Neuroscience*. 2017;42: 1–10.
1634 doi:10.1007/s10827-016-0623-7
- 1635 19. Omar C, Aldrich J, Gerkin RC. Collaborative infrastructure for test-driven scientific
1636 model validation. *36th International Conference on Software Engineering, ICSE
1637 Companion 2014 - Proceedings*. 2014; 524–527. doi:10.1145/2591062.2591129
- 1638 20. Gerkin R, Omar C. NeuroUnit: Validation Tests for Neuroscience Models. *Frontiers
1639 in Neuroinformatics*. 2013. doi:10.3389/conf.fninf.2013.09.00013
- 1640 21. Appukuttan S, Garcia PE, Davison AP. MorphoUnit. Zenodo; 2020.
1641 doi:<https://doi.org/10.5281/zenodo.3862936>
- 1642 22. Appukuttan S, Dainauskas J, Davison AP. SynapseUnit. Zenodo; 2020.
1643 doi:<http://doi.org/10.5281/zenodo.3862944>
- 1644 23. Garcia PE, Davison AP. HippoNetworkUnit. Zenodo; 2020.
1645 doi:<http://doi.org/10.5281/zenodo.3886484>
- 1646 24. Sharma BL, Davison AP. CerebUnit. Zenodo; 2020.
1647 doi:<http://doi.org/10.5281/zenodo.3885673>

- 1648 25. Bianchi D, Marasco A, Limongiello A, Marchetti C, Marie H, Tirozzi B, et al. On the
1649 mechanisms underlying the depolarization block in the spiking dynamics of CA1
1650 pyramidal neurons. *Journal of Computational Neuroscience*. 2012;33: 207–225.
1651 doi:10.1007/s10827-012-0383-y
- 1652 26. Spruston N, Schiller Y, Stuart G, Sakmann B. Activity-Dependent Action Potential
1653 Invasion and Calcium Influx into Hippocampal CA1 Dendrites. *Science*. 1995;268:
1654 297–300. doi:10.1126/science.7716524
- 1655 27. Golding NL, Kath WL, Spruston N. Dichotomy of action-potential backpropagation
1656 in CA1 pyramidal neuron dendrites. *Journal of Neurophysiology*. 2001;86: 2998–3010.
1657 doi:10.1152/jn.2001.86.6.2998
- 1658 28. Gasparini S, Losonczy A, Chen X, Johnston D, Magee JC. Associative pairing
1659 enhances action potential back-propagation in radial oblique branches of CA1
1660 pyramidal neurons. *Journal of Physiology*. 2007;580: 787–800.
1661 doi:10.1113/jphysiol.2006.121343
- 1662 29. Magee JC, Cook EP. Somatic EPSP amplitude is independent of synapse location in
1663 hippocampal pyramidal neurons. *Nature Neuroscience*. 2000;3: 895–903.
1664 doi:10.1038/78800
- 1665 30. Gasparini S, Magee JC. State-dependent dendritic computation in hippocampal CA1
1666 pyramidal neurons. *Journal of Neuroscience*. 2006;26: 2088–2100.
1667 doi:10.1523/JNEUROSCI.4428-05.2006
- 1668 31. Ariav G, Polsky A, Schiller J. Submillisecond precision of the input-output
1669 transformation function mediated by fast sodium dendritic spikes in basal dendrites of
1670 CA1 pyramidal neurons. *Journal of Neuroscience*. 2003;23: 7750–7758.
1671 doi:10.1523/jneurosci.23-21-07750.2003

- 1672 32. Takahashi H, Magee JC. Pathway Interactions and Synaptic Plasticity in the Dendritic
1673 Tuft Regions of CA1 Pyramidal Neurons. *Neuron*. 2009;62: 102–111.
1674 doi:10.1016/j.neuron.2009.03.007
- 1675 33. Losonczy A, Magee JC. Integrative Properties of Radial Oblique Dendrites in
1676 Hippocampal CA1 Pyramidal Neurons. *Neuron*. 2006;50: 291–307.
1677 doi:10.1016/j.neuron.2006.03.016
- 1678 34. Hines ML, Carnevale NT. The NEURON Simulation Environment. *Neural*
1679 *Computation*. 1997;9: 1179–1209. doi:<https://doi.org/10.1162/neco.1997.9.6.1179>
- 1680 35. Druckmann S, Banitt Y, Gidon A, Schrümann F, Markram H, Segev I. A novel
1681 multiple objective optimization framework for constraining conductance-based neuron
1682 models by experimental data. *Frontiers in Neuroscience*. 2007;1: 7–18.
1683 doi:10.3389/neuro.01.1.1.001.2007
- 1684 36. van Geit W, Moor R, Ranjan R, Roessert C, Riquelme L. Electrophys Feature
1685 Extraction Library. 2020. [cited 25. March 2020] GitHub repository [Internet]
1686 Available: <https://github.com/BlueBrain/eFEL>
- 1687 37. Tripathy SJ, Savitskaya J, Burton SD, Urban NN, Gerkin RC. NeuroElectro: A
1688 window to the world’s neuron electrophysiology data. *Frontiers in Neuroinformatics*.
1689 2014;8: 1–11. doi:10.3389/fninf.2014.00040
- 1690 38. Wheeler DW, White CM, Rees CL, Komendantov AO, Hamilton DJ, Ascoli GA.
1691 Hippocampome.org: A knowledge base of neuron types in the rodent hippocampus.
1692 *eLife*. 2015;4: 1–28. doi:10.7554/eLife.09960
- 1693 39. Staff NP, Jung HY, Thiagarajan T, Yao M, Spruston N. Resting and active properties
1694 of pyramidal neurons in subiculum and CA1 of rat hippocampus. *Journal of*
1695 *Neurophysiology*. 2000;84: 2398–2408. doi:10.1152/jn.2000.84.5.2398

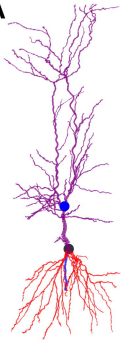
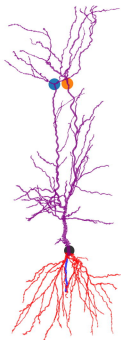

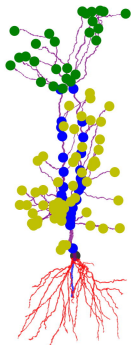
- 1696 40. Dougherty KA, Islam T, Johnston D. Intrinsic excitability of CA1 pyramidal neurones
1697 from the rat dorsal and ventral hippocampus. *Journal of Physiology*. 2012;590: 5707–
1698 5722. doi:10.1113/jphysiol.2012.242693
- 1699 41. Malik R, Dougherty KA, Parikh K, Byrne C, Johnston D. Mapping the
1700 electrophysiological and morphological properties of CA1 pyramidal neurons along the
1701 longitudinal hippocampal axis. *Hippocampus*. 2016;26: 341–361.
1702 doi:10.1002/hipo.22526
- 1703 42. McDermott CM, Hardy MN, Bazan NG, Magee JC. Sleep deprivation-induced
1704 alterations in excitatory synaptic transmission in the CA1 region of the rat
1705 hippocampus. *Journal of Physiology*. 2006;570: 553–565.
1706 doi:10.1113/jphysiol.2005.093781
- 1707 43. Graves AR, Moore SJ, Bloss EB, Mensh BD, Kath WL, Spruston N. Hippocampal
1708 Pyramidal Neurons Comprise Two Distinct Cell Types that Are Countermodulated by
1709 Metabotropic Receptors. *Neuron*. 2012;76: 776–789. doi:10.1016/j.neuron.2012.09.036
- 1710 44. Golding NL, Mickus TJ, Katz Y, Kath WL, Spruston N. Factors mediating powerful
1711 voltage attenuation along CA1 pyramidal neuron dendrites. *Journal of Physiology*.
1712 2005;568: 69–82. doi:10.1113/jphysiol.2005.086793
- 1713 45. Roessert C, Tanguy D, van Geit W. BluePyEfe: Blue Brain Python E-feature
1714 extraction. Zenodo; 2020. doi:10.5281/zenodo.3728192
- 1715 46. Bormann I. DigitizeIt: Digitizer software - digitize a scanned graph or chart into
1716 (x,y)-data. 2020. [cited 25. March 2020] [Internet] Available: <https://www.digitizeit.de/>
- 1717 47. Jahr CE, Stevens CF. Voltage dependence of NMDA-activated macroscopic
1718 conductances predicted by single-channel kinetics. *Journal of Neuroscience*. 1990;10:
1719 3178–3182. doi:10.1523/jneurosci.10-09-03178.1990

- 1720 48. Hestrin S, Nicoll RA, Perkel DJ, Sah P. Analysis of excitatory synaptic action in
1721 pyramidal cells using whole-cell recording from rat hippocampal slices. *Physiology*.
1722 1990; 203–225. doi:10.1113/jphysiol.1990.sp017980
- 1723 49. Korinek M, Sedlacek M, Cais O, Dittert I, Vyklicky L. Temperature dependence of
1724 N-methyl-d-aspartate receptor channels and N-methyl-d-aspartate receptor excitatory
1725 postsynaptic currents. *Neuroscience*. 2010;165: 736–748.
1726 doi:10.1016/j.neuroscience.2009.10.058
- 1727 50. Gevaert M, Kanari L, Palacios J, Zisis E, Coste B. NeuroM. 2020. [cited 25. March
1728 1579 2020] GitHub repository [Internet] Available:
1729 <https://github.com/BlueBrain/NeuroM>
- 1730 51. Katz Y, Menon V, Nicholson DA, Geinisman Y, Kath WL, Spruston N. Synapse
1731 Distribution Suggests a Two-Stage Model of Dendritic Integration in CA1 Pyramidal
1732 Neurons. *Neuron*. 2009;63: 171–177. doi:10.1016/j.neuron.2009.06.023
- 1733 52. Migliore M, de Blasi I, Tegolo D, Migliore R. A modeling study suggesting how a
1734 reduction in the context-dependent input on CA1 pyramidal neurons could generate
1735 schizophrenic behavior. *Neural Networks*. 2011;24: 552–559.
1736 doi:10.1016/j.neunet.2011.01.001
- 1737 53. Poirazi P, Brannon T, Mel BW. Arithmetic of subthreshold synaptic summation in a
1738 model CA1 pyramidal cell. *Neuron*. 2003;37: 977–987. doi:10.1016/S0896-
1739 6273(03)00148-X
- 1740 54. Gómez González JF, Mel BW, Poirazi P. Distinguishing linear vs. non-linear
1741 integration in CA1 radial oblique dendrites: It’s about time. *Frontiers in Computational*
1742 *Neuroscience*. 2011;5: 1–12. doi:10.3389/fncom.2011.00044
- 1743 55. Jeffrey M. Perkel. Why Jupyter is data scientists’ computational notebook of choice.
1744 *Nature*. 2018; 5–6. Available: <https://www.nature.com/articles/d41586-018-07196-1>

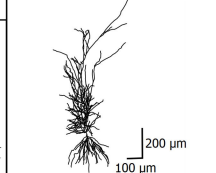
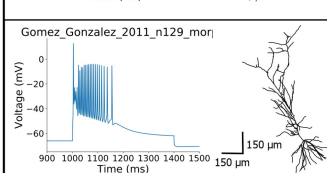
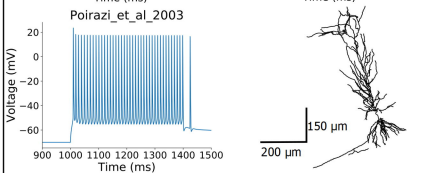
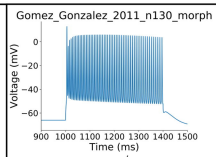
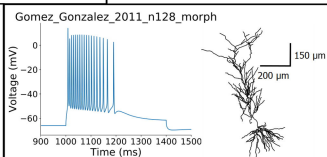
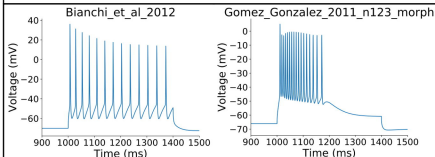
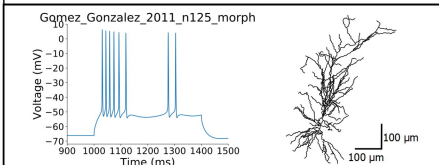
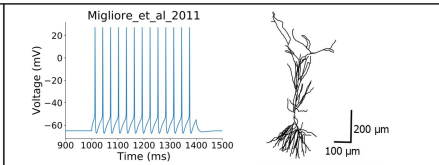
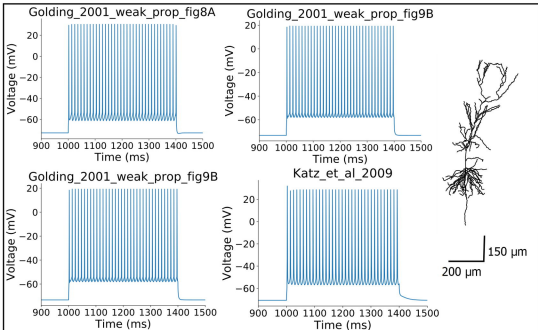
- 1745 56. Shah MM, Migliore M, Valencia I, Cooper EC, Brown DA. Functional significance
1746 of axonal Kv7 channels in hippocampal pyramidal neurons. *Proceedings of the*
1747 *National Academy of Sciences of the United States of America*. 2008;105: 7869–7874.
1748 doi:10.1073/pnas.0802805105
- 1749 57. Ecker A, Romani A, Sáray S, Káli S, Migliore M, Mercer A, et al. Data-driven
1750 integration of hippocampal CA1 synapse physiology in silico. *Hippocampus*. 2020.
1751 doi:10.1002/hipo.23220
- 1752 58. Amunts K, Ebell C, Muller J, Telefont M, Knoll A, Lippert T. The Human Brain
1753 Project: Creating a European Research Infrastructure to Decode the Human Brain.
1754 *Neuron*. 2016;92: 574–581. doi:10.1016/j.neuron.2016.10.046
- 1755 59. Fragnaud H, Gonin J, Duperrier J, Legouee E, Davison AP, Appukuttan S. hbp-
1756 validation-framework. Zenodo; 2020. doi:http://doi.org/10.5281/zenodo.3888123
- 1757 60. Ramaswamy S, Courcol JD, Abdellah M, Adaszewski SR, Antille N, Arsever S, et al.
1758 The neocortical microcircuit collaboration portal: A resource for rat somatosensory
1759 cortex. *Frontiers in Neural Circuits*. 2015;9. doi:10.3389/fncir.2015.00044
- 1760 61. D’Angelo E, Antonietti A, Casali S, Casellato C, Garrido JA, Luque NR, et al.
1761 Modeling the cerebellar microcircuit: New strategies for a long-standing issue.
1762 *Frontiers in Cellular Neuroscience*. 2016;10: 1–29. doi:10.3389/fncel.2016.00176
- 1763 62. Marder E, Taylor AL. Multiple models to capture the variability in biological neurons
1764 and networks. *Nature Neuroscience*. 2011;14: 133–138. doi:10.1038/nn.2735
- 1765 63. Marder E, Goeritz ML, Otopalik AG. Robust circuit rhythms in small circuits arise
1766 from variable circuit components and mechanisms. *Current Opinion in Neurobiology*.
1767 2015;31: 156–163. doi:10.1016/j.conb.2014.10.012
- 1768 64. Ransdel JL, Nair SS, Schulz DJ. Neurons within the same network independently
1769 achieve conserved output by differentially balancing variable conductance magnitudes.

- 1770 Journal of Neuroscience. 2013;33: 9950–9956. doi:10.1523/JNEUROSCI.1095-
1771 13.2013
- 1772 65. Marder E, Goaillard JM. Variability, compensation and homeostasis in neuron and
1773 network function. Nature Reviews Neuroscience. 2006;7: 563–574.
1774 doi:10.1038/nrn1949
- 1775 66. Golowasch J, Goldman MS, Abbott LF, Marder E. Failure of averaging in the
1776 construction of a conductance-based neuron model. Journal of Neurophysiology.
1777 2002;87: 1129–1131. doi:10.1152/jn.00412.2001
- 1778 67. Hjorth JJJ, Kozlov A, Carannante I, Nylén JF, Lindroos R, Johansson Y, et al. The
1779 microcircuits of striatum in silico. Proceedings of the National Academy of Sciences of
1780 the United States of America. 2020;117: 9554–9565. doi:10.1073/pnas.2000671117
- 1781 68. Aradi I, Soltesz I. Modulation of network behaviour by changes in variance in
1782 interneuronal properties. Journal of Physiology. 2002;538: 227–251.
1783 doi:10.1113/jphysiol.2001.013054
- 1784 69. Gleeson P, Crook S, Cannon RC, Hines ML, Billings GO, Farinella M, et al.
1785 NeuroML: A language for describing data driven models of neurons and networks with
1786 a high degree of biological detail. PLoS Computational Biology. 2010;6: 1–19.
1787 doi:10.1371/journal.pcbi.1000815
- 1788 70. Raikov I. NineML – a description language for spiking neuron network modeling: the
1789 abstraction layer. BMC Neuroscience. 2010;11: 2202. doi:10.1186/1471-2202-11-s1-
1790 p66
- 1791 71. Davison AP, Brüderle D, Eppler J, Kremkow J, Müller E, Pecevski D, et al. PyNN: A
1792 common interface for neuronal network simulators. Frontiers in Neuroinformatics.
1793 2009;2: 1–10. doi:10.3389/neuro.11.011.2008

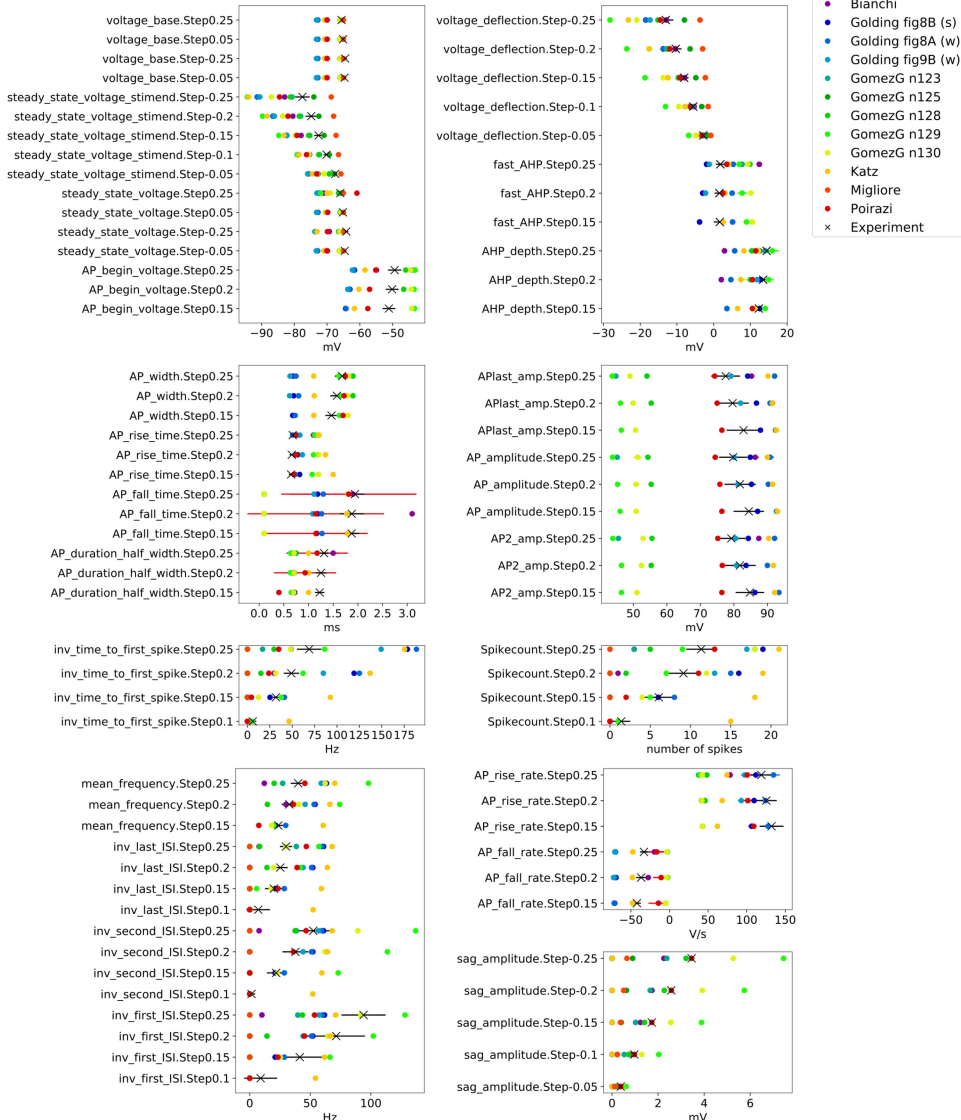
- 1794 72. Dai K, Hernando J, Billeh YN, Gratiy SL, Planas J, Davison A, et al. The SONATA
1795 Data Format for Efficient Description of Large-Scale Network Models. PLoS
1796 Computational Biology. 2019;16: 1–24. doi:10.2139/ssrn.3387685
- 1797 73. Stimberg M, Brette R, Goodman DFM. Brian 2, an intuitive and efficient neural
1798 simulator. eLife. 2019;8: 1–41. doi:10.7554/eLife.47314
- 1799
- 1800 S1 Appendix. Example of running the SomaticFeaturesTest of HippoUnit using a Jupyter
1801 notebook

A**B****C**

100 μm
100 μm

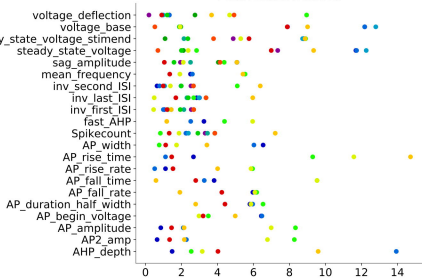


Absolute features

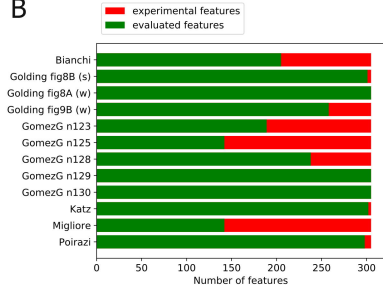


A

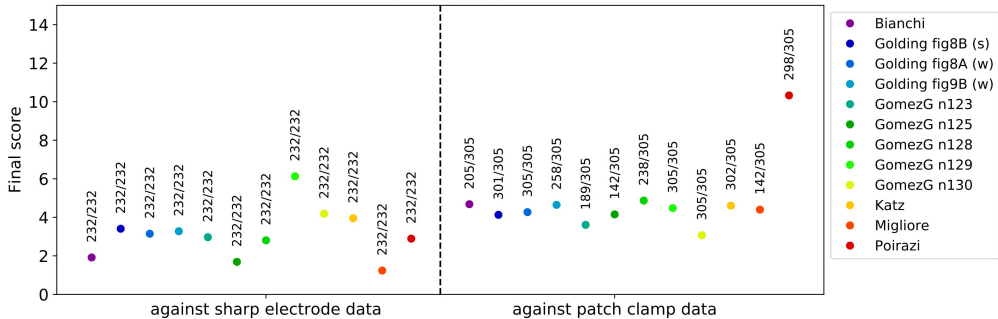
Mean feature scores



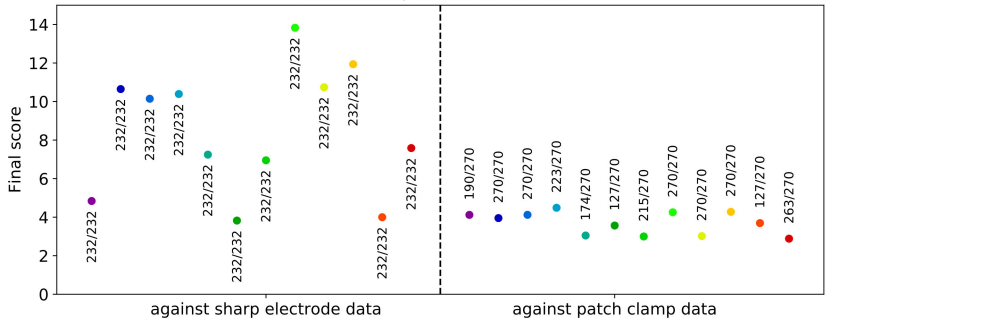
B

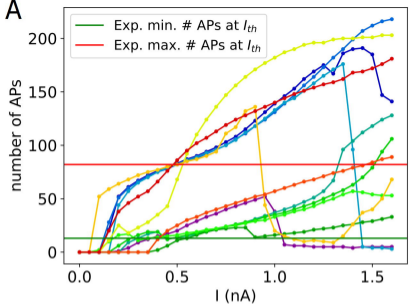
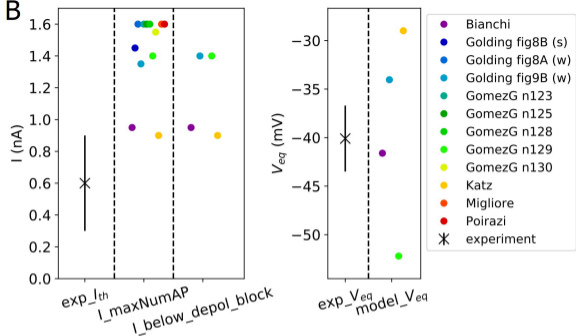
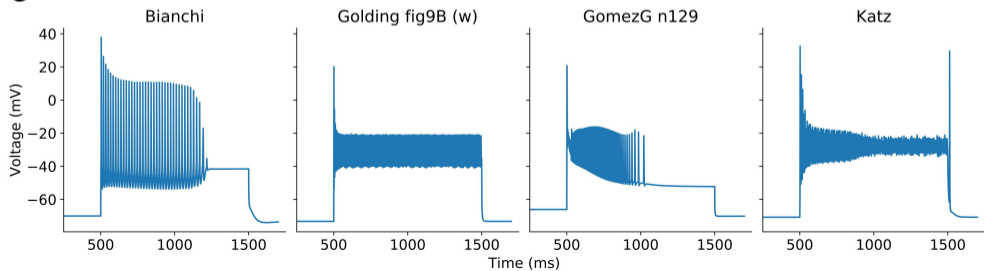


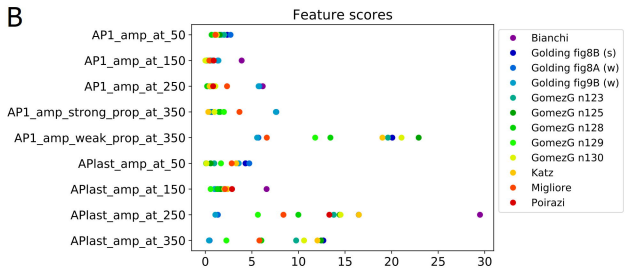
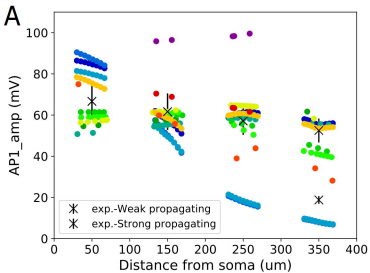
Raw final score

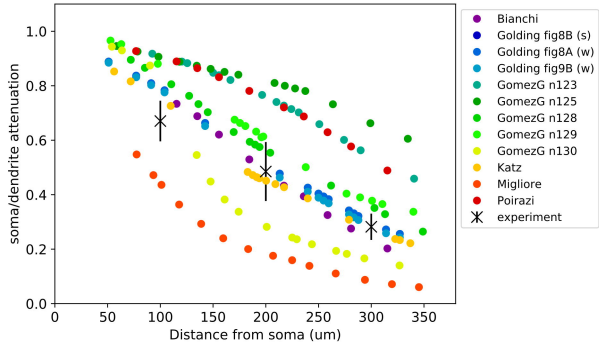


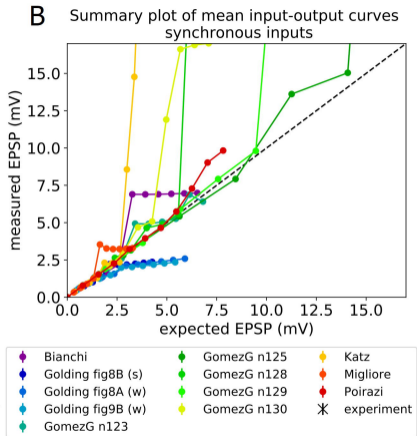
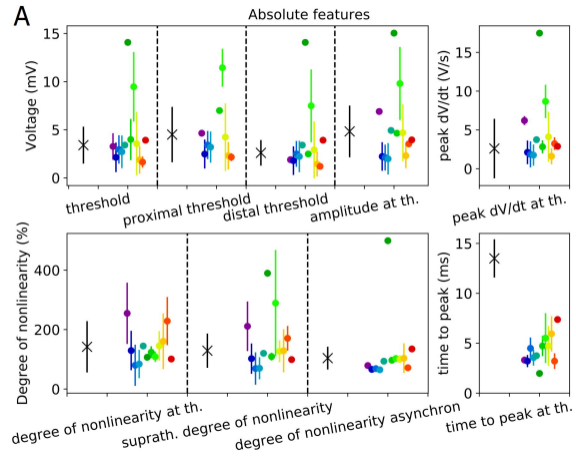
Recalculated, unbiased final score

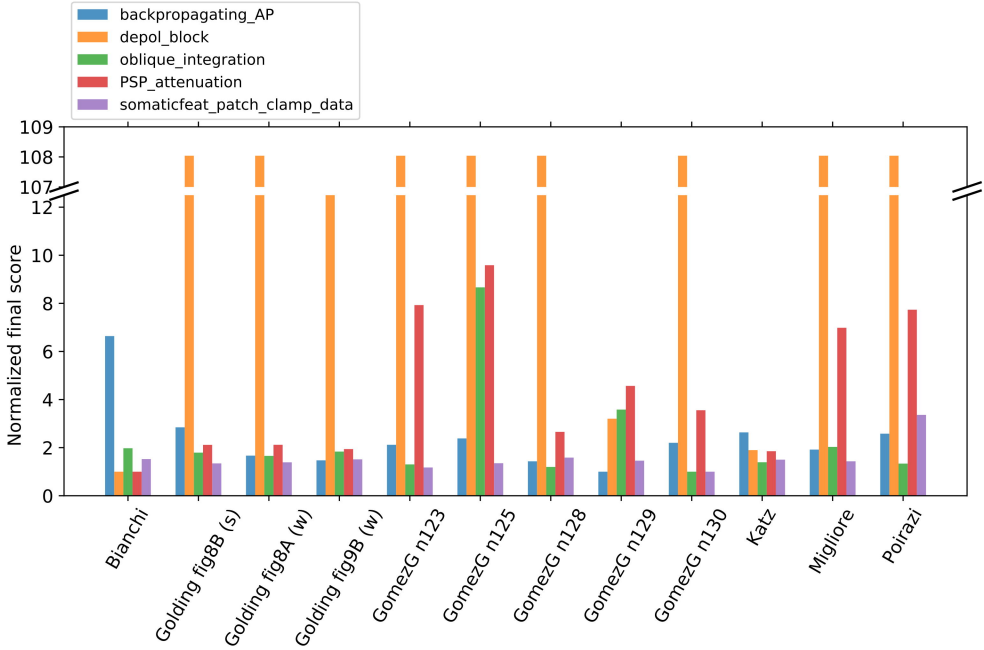


A**B****C**

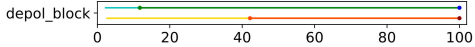
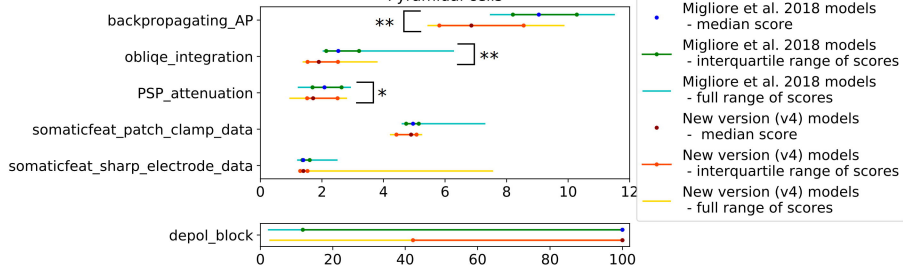








Pyramidal cells



Interneurons

