

神経細胞データベースモデルのインタフェース同定と再構成

大武 美保子^{†,††}

高木 利久^{†††,†}

† 東京大学総括プロジェクト機構領域創成学術統合化プロジェクト研究部門

†† 科学技術振興機構戦略的創造研究推進事業さきかけ

††† 東京大学大学院新領域創成科学研究科情報生命科学専攻

本研究の目的は、神経系のマイクロからマクロまでの状態を予測することが可能な神経系シミュレータを開発することである。そのようなシミュレータを実現するために、神経科学者により記述され、レビューされ、データベースに公開されている神経系の部分モデルを活用する。しかしながら、変数やパラメータには説明がなく、他のモデルと結合して使える状態にはないという問題がある。これは、研究者が個々の実験と対照する目的でモデルを構築しているからである。そこで本研究では、個々の部分モデルを統合するために、一部自動化してモデルを読み取ることによって効率よく入出力インタフェースを同定し、モデルを再構成する手法を提案する。

Identification of Interfaces and Reassembly of Neural Models Registered on Neural Databases

Mihoko Otake^{†,††}

Toshihisa Takagi^{†††,†}

† Science Integration Program - Humans, Department of Frontier Sciences and Science Integration, Division of Project Coordination, The University of Tokyo

†† PRESTO program, Japan Science and Technology Agency

††† Department of Computational Biology, Graduate School of Frontier Science, The University of Tokyo

The importance of modeling and simulation of biological process is growing for further understanding of living systems at all scales from molecular to cellular, organic, and individuals. In the field of neuroscience, there are so called platform simulators, the de-facto standard neural simulators. More than a hundred neural models are registered on the model database. These models are executable in corresponding simulation environments. But usability of the registered models is not sufficient. In this study, we propose a novel method of reassembly and interfacing models registered on neural model database in order to make use of the model.

1 序論

近年、一次データを共有し活用することは、生命科学の進歩のために非常に重要であることが、広く認識されてきている。ゲノミクスやプロテオミクスの分野において、研究者は論文を出版する際、集中管理されたデータベースにシーケンスデータを登録する。その結果、データベースは日々成長しており、分子レベルの生物学的データを統合する為の努力はより洗練された方法で行われるようになった [1, 2]。神経

科学の分野では、ヒューマンブレインプロジェクト (HBP) が、米国政府の主導により、1993年に設立された。複数の行政機関で構成されるプロジェクトで、ヒトゲノムプロジェクト (HGP) に平行して進められた。HBPは、神経科学者のためのデータベースそして関連するツールのための研究を支援してきた [3]。2005年には、科学者同士の国際的なコラボレーションを推進することを目的として、OECD Global Science Forumの7つの加盟国が、International Neuroinformatics Coordinating Facility (INCF) を

立ち上げた．ヒトの脳の働きに関する理解を深めるために，神経科学のデータを共有し解析する新しいやり方を作り出すことを目指している．神経科学者が生成するデータは，ゲノム配列データと比べて複雑である．扱わなければならないデータの種類の種類は，形態，機能的結合，神経生理，化学，分子生物学，脳機能イメージング，行動と多岐に渡る [4]．それぞれのスケールのデータを扱う多様なデータベースが開発されてきた [5]．従って，これらのデータや知識を統合する手法は必要不可欠のものであり，健康および病態の神経系がどのように働くかということについてさらなる理解を深めることができると期待される．

本研究の目的は，神経系のミクロからマクロまでの状態を計算できる神経系シミュレータを開発することである．神経系全体の状態をシミュレーションするためには，詳細な神経系モデルが必要になる．神経細胞の性質は部位により異なり，論文から一つずつモデリングし，検証するのでは，全体をシミュレーションするところまで到達するのが困難である．幸いなことに，神経科学者により記述された神経系の部分モデルが，レビューされ，データベースに多数登録されている．もちろんヒトの神経系全体をカバーするものではないが，これらを活用することができれば，信頼性の高い要素で構成されるシミュレータを構築することができる．そこで本研究では，本来単体で動いていた神経系の部分モデルの入出力を同定し，再構成する手法を提案する [6]．

2 神経細胞モデルとモデルデータベース

2.1 モデルデータベース

モデルは既存の仮説に新しい情報を統合したり，新しい仮説を作り出す為の強力な方法を提供する．そして，これらの仮説を検証する新しい実験を組み立てる基礎となる [7, 4]．モデルはデータベースから関連するデータを分類整理し，研究者が矛盾に気づくのを助ける．モデルは仮説の正式な表現であり，生命科学者同士がコミュニケーションし，考えを量的に比較する手段を提供する．モデルデータベースはそのようなモデルと，モデルに必要とされる多様なデータを扱う．

神経科学の分野では，Bower のグループが GENESIS[8] シミュレーションの環境のための ChannelDB[9] というイオンチャンネルデータベース

を開発した．白井のグループは，視覚系に関する研究分野で用いられる再利用可能なリソースを提供する Visiome Platform を開発した [10]．ここには数学モデル，実験データ，解析のためのライブラリ，関連情報が含まれている．Shepherd らは，ModelDB[11] と名づけられたモデルデータベースを開発し，運用している．神経細胞モデルを効率よく登録し，取り出せるようによく構成されており，これらの情報にアクセスするための場を提供している．ModelDB のユーザは，様々な角度からモデルを検索することができる．検索項目には，作者の名前，モデル特性（神経細胞の種類，電流，レセプタ，トランスミッタなど），扱われているトピック（シナプス可塑性やパターン認識など），シミュレーション環境の種類（NEURON や GENESIS など）等がある．モデルを構成するコードは，モデルをダウンロードする前に見ることができ，モデルを自動で起動するように設定することもできる．それぞれ，基本文献へのポインタも提供されている．モデルはどのようなシミュレーション環境でも記述することができるが，ModelDB は NEURON[12] と GENESIS シミュレーション環境で実行可能なモデルを扱うために作られたので，これらのモデルが多く登録されている．NEURON 環境で動くモデルが 100 個以上登録されている．

本研究では，ModelDB に登録されている神経細胞モデルに焦点を合わせ，これを再構成し，モデル同士をインタフェースするための方法を提案する．その目的は，モデルとその背後にある知識をマイニングし，解析し，統合することにある．

2.2 神経細胞モデルとそのシミュレーション環境

モデル統合の難しさを指摘するために，神経細胞シミュレーションのプラットフォームおよび神経細胞モデルについて簡単に整理する．

神経科学の分野では，プラットフォームシミュレータと呼ばれる業界標準的な神経細胞シミュレータがある．NEURON[12] 及び GENESIS[8] は，実験データと密接に関連付けられた個々の神経細胞および神経細胞ネットワークモデルのための典型的なプラットフォームシミュレータである．これらは，複雑な分岐形状，イオンチャンネルタイプ，一様でないイオンチャンネル分布，イオン拡散係数，セカンドメッセンジャーの特性等を記述した神経細胞や神経細胞ネットワークの動作をシミュレーションすることができ

る。これらのシミュレータは、モデルを組み立て、動かす、効率よく取り扱うためのツールを提供し、実験家に広く受け入れられている。NEURON、GENESIS はこれまでにそれぞれ 200 以上の科学論文の中で用いられてきた。

シミュレーションプラットフォームの主要な部分とシミュレーション環境を使うための多様なモードは以下のようにまとめられる。シミュレーション言語のインタープリタ、GUI、偏微分方程式を計算し、入出力するための計算エンジン、データファイル、スクリプトファイル、あらかじめコンパイルされたオブジェクトライブラリである。これらは、神経細胞や神経細胞ネットワークモデルを作るために高度なシミュレーション言語を用いる。コマンドは、コマンドプロンプトまたは GUI により対話的に実行されるか、スクリプトファイルによって逐次的に実行される。特定の実験に対応するシミュレーションは、モデル自体を作成し、GUI を立ち上げるコマンド列をスクリプトファイルに書くことでセットアップされる。スクリプト言語とモジュールは極めて強力な、スクリプトを数行書くだけで非常に洗練されたシミュレーションを実行することができる。

2.3 モデルデータベースに登録されたモデルのユーザビリティの問題

モデルデータベースに登録されているモデルを利用し、統合するためには、2 つの問題がある。

1. モデルの入力、出力および内部状態の変数そして変数が明確に定義されていない。ユーザはモデルを実行し、シミュレーションの結果が詳しく記述されている論文を読むことによってそれらを判別しなければならない。
2. モデルはスタンドアロンで実行されるという仮定で設計されている。各モデル毎にパラメータを調節するための GUI がついている。

モデルデータベースに登録されているモデルは神経科学の研究者によって作成されたものである。モデルのシミュレーションの結果はピアレビュー論文に掲載されている。モデル自体もデータベースのキュレータによってレビューされ、最小限の修正を加えられている。従って、モデルの妥当性は保証されている。しかしながら、モデルには変数とパラメータの説明が含まれていない。これが第一の問題であ

る。どれが神経細胞に対する入力、出力、内部状態変数が分からない。これはモデルの元々の目的を考えると不思議ではない。モデルは特定の科学的問題を解くために作られる。他の研究者が組み立てて統合することを意識して作られるわけではない。従って、モデルのユーザである別の研究者は、変数やパラメータの意味を明らかにするために、それぞれのモデルについて一つずつ、モデルを実行し、シミュレーション結果を解析し、論文をすべて読まなければならない。

第二の問題は、シミュレータの使いやすさ、利便さを目指した以下に述べる二つの特長の副作用によるものである。一つ目の特長は、GUI が充実していることである。ユーザは GUI を通じてモデルを作り、シミュレーションを行い、探索的にパラメータを設定し、電圧や電流や刺激を制御し、変数を時間や位置の関数としてグラフプロットすることができる。二つ目の特長は、GUI をカスタマイズしたり、データ解析をしたり、最適化したりする上で便利な、完全なプログラミング言語を有するスクリプト環境を有することである。この二つの特長により、データとロジックとインタフェースはスクリプトの中に混在している。従って、使いやすさを向上させるためには、モデルを再構成する必要がある。

3 モデル再構成とインタフェース手法

本研究では、モデルデータベースに登録されている神経細胞モデルの再構成及びインタフェースする方法を提案する。これは神経系の統合された機能の理解のために、非常に重要なステップである。基本的な考え方は以下の通りである。

1. 入出力変数およびパラメータを少ない労力でシステムチェックに同定するべきである。
2. モデルの修正は最小限に抑えるべきである。
3. シミュレーションの環境そのものには手を加えない。

第一の考え方は、モデルデータベースにモデルが 100 個以上登録されているため、モデルを構成するファイルをすべて解読するのに多大な労力を要するというに基づいている。そこで、入出力変数やパラメータを効率的に抽出するためのルールを提案する。第二の考え方は、モデルがすでにレビューさ

れ，スタンドアロンでの動作が保証されていることを考慮する．この利点は保存されなければならない．そこで，モデルに最小限の，入出力のためのスクリプトを追加することとする．第三の考え方は，シミュレーション環境が常にアップデートされることを考慮するものである．シミュレータ自体を書き換えることにすると，バージョンアップするたびにシミュレータを書き換えなければならなくなる．そこでシミュレータに変更を加えずにインタフェースプログラムを用意することにした．提案する方法は，三つの考え方に対応する以下の三つのフェーズにより構成される．

3.1 モデルの入出力の同定

モデルファイルからのプロットコマンドの抽出

モデルはスクリプト言語で書かれたコマンド列で構成される．モデル自身を構成し，さらに，特定のシミュレーションのための GUI を生成する．モデルは大量の変数とパラメータを含んでいる．あるものはモデル内部のみで用いる．あるものはグラフをプロットするために用いられる．それぞれの言語には値に存在範囲がある．モデルが想定された条件で用いられないと得られる結果も矛盾したものになってしまう．GUI は統合化の点で基本的に不要である．最初はこれらをコメントアウトすることを考えた．しかしその性質をよく考えてみると，グラフにプロットされる変数やパラメータはモデルの入出力の候補であることが分かった．値の存在範囲に関する手がかりもつかめるだろう．従って，入出力を同定する第一のステップは，グラフプロットコマンドをモデルファイルから抽出することである．NEURON シミュレータの場合，`addvar` 及び `plot` である．グラフにプロットされる変数は，出力の有力な候補である．プロットを繰り返し実行する中で，値を変化させるパラメータは入力候補と考えられる．

論文のグラフからの単位や意味の抽出

登録されたモデルのシミュレーションの結果は，論文に説明されている．結果のグラフは対応する論文の図の中に見つけることができる．変数やパラメータの単位や意味は，図の中に表示されている．これらを変数やパラメータの注釈に用いることができる．このステップは自動ではなく，手動により実施される．

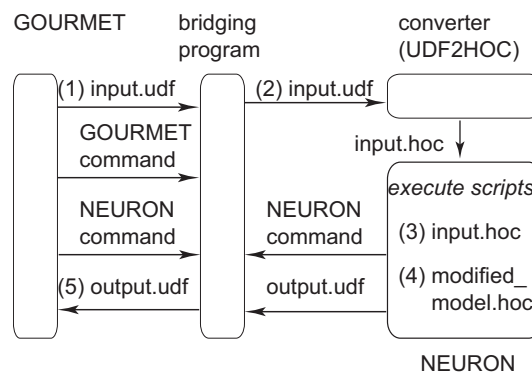


図 1 System structure of the integrated simulation environment. It comprises integrative simulation platform (GOURMET), neural simulator (NEURON), conversion script(UDF2HOC), and the bridging program.

3.2 モデルの再構成

入出力データの UDF による定義

入出力変数とパラメータが明らかになったところで，これらを分かりやすく表現する方法を考える．ここでは，データフォーマットとして，User Definable Format (UDF)[13] と呼ばれるフォーマットを採用する．UDF は，XML などと比較してオーバーヘッドが少なく，実験データに含まれる変数の型，単位，意味を記述する際に最もシンプルな表記法を提供するからである．

この UDF はもともと，土井らによる OCTA プロジェクト [14] と呼ばれる，ソフトマテリアルのマルチスケールシミュレーションプロジェクトの中で開発されたものである．具体的には，変数の定義部 `global_def` とデータ部 `data` で構成され，ユーザが自由に構造を決めることができる．詳細は，実験の章で具体的に説明する．

入力と出力の変換

入出力を表現する記法を定義したが，このままではシミュレータが読み書きすることができない．そこで，入力については，UDF ファイルをシミュレータが解読可能な形式に変換するプログラムを用意し，シミュレータを起動する前に変換する．NEURON シミュレータの場合，UDF2HOC というスクリプトにより HOC フォーマットに変換する．出力については，モデルファイルにスクリプトを追加し，シミュレータが UDF フォーマットで結果を出力する

ようにする。NEURON シミュレータの場合、HOC フォーマットのモデルファイルの中に追加する。

3.3 インタフェースプログラムを介したモデルの起動

異なるシミュレーション環境のモデル同士を接続することを考えて、統合化シミュレーションプラットフォームからモデルを起動することとする。統合化シミュレーションプラットフォームとして GOURMET を採用する。この GOURMET は UDF と一緒に開発されたもので、UDF の Editor、Viewer であり、異なるシミュレータを起動する機能も備えている。このような特徴により、ユーザはシミュレーション結果を容易に確認することができる。

GOURMET から起動されるプログラムは、UDF が送受信でき、GOURMET コマンドを受信することができ、NEURON コマンドを送信できる必要がある。そこで、そのような機能を有するインタフェースプログラムを作成し、そのプログラムが NEURON と GOURMET とをつなぐようにした。このようにすることで、NEURON そのものを書き換えずに、外部プログラムとインタフェースすることができる。図 1 に描いたインタフェースプログラムの動作を以下に説明する。

1. UDF 形式の入力データ (input.udf) が GOURMET からインタフェースプログラムへ送信されると、
2. インタフェースプログラムはこれを HOC ファイル (input.hoc) に変換するプログラム (UDF2HOC) に渡して起動し、
3. NEURON においてこの HOC ファイル (input.hoc) を実行した後、
4. オリジナルのモデルに出力を UDF 形式にするスクリプトを追加したもの (modified_model.hoc) を実行する。
5. 実行結果 (output.hoc) は UDF 形式で、インタフェースプログラムを介し NEURON から GOURMET へ送られる。

4 インタフェース同定と再構成実験

4.1 海馬錐体細胞モデル

我々は ModelDB に登録されていたモデルに提案する手法を適用した。本論文で取り上げる海馬の神

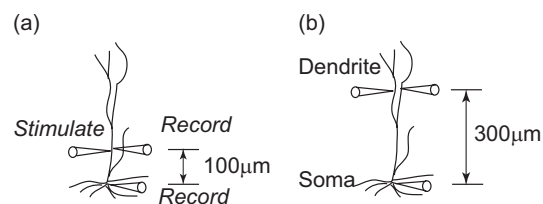


図 2 Experimental apparatus of the hippocampal pyramidal neural cell model. EPSPs were recorded for (a) a proximal (100 μm from the soma) and (b) more distal (300 μm from the soma) synaptic inputs.

経細胞モデルは ModelDB に登録されていたモデルの一つである。このモデルを例に取り、以下に適用結果を説明する。このモデルは、海馬錐体細胞において、細胞体の興奮性シナプス後電位 (excitatory postsynaptic potential: EPSP) の強さがシナプス入力的位置によらないことを示すものである [15]。モデルの開発者は、海馬 CA1 の錐体細胞の先端樹状突起全般に渡って、dual whole-cell recordings の手法により、興奮性シナプス入力に対する応答の強さと反応速度を同定した。開発者らは、樹状突起における EPSP の強さが、細胞体から遠ざかるにつれて増し、樹状突起のフィルタリング効果を相殺するかのようであることを発見した。そのことが、細胞体における EPSP の強さに対するシナプス入力的位置依存性を減少させるよう働いていることを明らかにした。具体的には、樹状突起における入力と細胞体との距離を、実験とシミュレーションの両方において変化させた。細胞体から近い距離 (細胞体から 100 μm) と、遠い距離 (細胞体から 300 μm) の樹状突起に対してシナプス入力に加えられ、EPSP が測定された (図 2)。実験結果は、シナプス伝導率の連続的な増加が各々の樹状突起への入力の強さを正規化する役割を果たしていることを示唆するものであった。

4.2 手法の適用結果

モデルの入出力同定

次の 3 行は、海馬 CA1 の錐体細胞のモデルを表すスクリプトファイルの 1 つ "main_example_epsps.hoc" から抽出されたものである。

```
g2[synapse_pos_cntr].plot(t)
```

```
g2[synapse_pos_cntr].addvar("s.v(.5)",1,1)
g2[synapse_pos_cntr].addvar("d[i].v(.5)",2,1)
```

この3行から、出力をプロットするグラフの横軸が”t”，グラフの縦軸は”s.v(.5)”及び”d[i].v(.5)”であると推定できる．対応する論文 [15] のグラフには、前者が細胞体における EPSP を表し、後者が、樹状突起において記録された EPSP を表すと定義されている．従って、モデルの出力変数の組は、時間”t”，細胞体における EPSP”s.v(.5)”，樹状突起における EPSP”d[i].v(.5)”で構成される．横軸の単位は [ms] であり、縦軸の単位は [mV] である．そして、”i”に関するコメントがファイル中に見つかった．

```
// i is dendritic compartment for synapse
```

パラメータ”i”へ、値”38, 153”が代入された．これらは、シナプスコンパートメントの参照番号である．この番号は、電流が注入された場所から細胞体までの距離を示している．シミュレーション結果と論文のグラフを比較し、前者のコンパートメントは、細胞体から近い部分（細胞体から 100 μm ）であり、後者のコンパートメントは、細胞体から遠い部分（細胞体から 300 μm ）であると分かった．従って、このモデルの入力パラメータは、シナプスコンパートメントの参照番号”i”である．グラフプロットの関数を含む行が自動抽出された後は、論文とモデルスクリプトを比較して、変数とパラメータの意味と単位を明らかにするのに、30 分程度かかった．

モデルの再構成

総括すると、モデルの出力変数は、時間”t”，細胞体における EPSP”s.v(.5)”，樹状突起における EPSP”d[i].v(.5)”である．横軸の単位は [ms] であり、縦軸の単位は [mV] である．このモデルの入力パラメータは、シナプスコンパートメントの参照番号”i”である．入力には単位がない．入力は一つの値しか含まないため、再構成したモデルの出力データ形式について述べる．モデルを構成するスクリプトファイルに、出力データを以下のようにフォーマットするようなコマンドを追加した．出力データの定義部は以下のように定義される．”:”の前の第一項は変数、”:”の後の第二項はデータの方、[] で囲まれた部分は単位で、二重クォーテーションで囲まれた最後の部分は変数の注釈である．

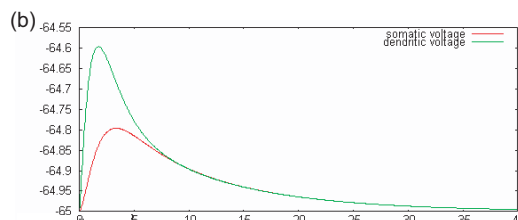
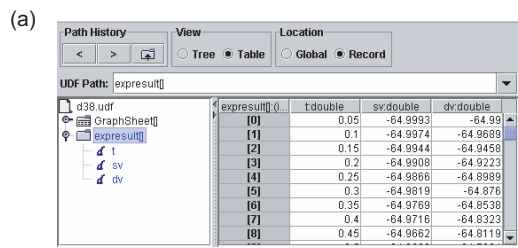


図 3 Simulation results which are displayed in integrated simulation platform, GOURMET. (a) The output data in UDF format is represented in the tree structure and table. (b) The output data is plotted on the graph.

```
\begin{global_def}
expresult[]:{
t: double [ms] "time"
sv: double [mV] "somatic voltage"
dv: double [mV] "dendritic voltage"
}
\end{global_def} .
```

出力データのデータ部は、以下のものであった．即ち、時刻、細胞体電位、樹状突起の電位のセットで構成される．

```
\begin{data}
expresult[]:[
{0.05 -64.9993 -64.99}
{0.1 -64.9974 -64.9689}
{0.15 -64.9944 -64.9458}
(snip)
{39.9 -64.9951 -64.9951}
{39.95 -64.9951 -64.9951}
{40 -64.9952 -64.9952}
]
\end{data} .
```

モデルのインタフェース

インタフェースプログラムを介し、GOURMET から NEURON シミュレータにコマンドを送り、モデルを実行した。その結果得られるデータを GOURMET において表示したものを図 3 に示す。図 3 (a) は、出力データをツリーおよびテーブル表示したもの、図 3 (b) はこれをグラフにプロットしたものである。NEURON を外部プログラムから制御し、結果を分かりやすく整理した形で得ることができた。以上を通じ、本手法の有効性が示された。

5 考察

5.1 データフォーマットの選定理由とその他のフォーマットの可能性

我々は簡便性および柔軟性の観点から、ユーザ定義可能なフォーマット (UDF) を選んだ。データおよびモデルの変数やパラメータの注釈が、モデルに埋め込まれた知識を共有し、モデルを統合するために、非常に重要である。UDF は注釈に必要な、シンプルなインタフェースを提供する。生命科学のための XML に基づくデータ形式は、SBML[16] をはじめ、さまざまなものが提案されている。NeuroML[17] は神経生理学データを表示するための候補の 1 つである。UDF データはタグの構造が単純であるので、NeuroML のような他のデータ形式に容易に変換することが可能である。UDF 形式を有するモデルの入出力データは、標準形式が異なる、またはそのような標準形式が存在しない他のスケールのモデルに接続することが可能である。

5.2 モデルのインタフェースをどのように拡張するか

我々が説明に用いた海馬錐体細胞モデルにおいて、注入する電流の大きさは固定されており、電流を注入する位置が可変であった。モデルの出力は EPSP であり、スパイク信号時系列など、他の変数は含んでいない。このようなモデルを、別の種類の神経細胞モデル、例えば SpikeNET[18] のようにスパイク入力が必要とするモデルと結合しようとする時、問題が起こるのである。注目している以外のパラメータや変数を、入出力インタフェースとして追加する必要があるだろう。パラメータや変数のある程度の標準化が必要になると考えられる。神経細胞のインタフェースは他の細胞と比べて比較的分かりやすいの

で、注目すべき入出力の数はある程度減らすことができる。他の種類の細胞については、どのようなインタフェースにするべきかの議論がより困難になると考えられる。こういった問題についても、今後の課題として検討していきたい。

5.3 本手法を他のシミュレーション環境に適用するには

本稿では、典型的な神経細胞シミュレーションの環境の 1 つ、NEURON を用いて提案手法について説明した。異なるシミュレーション環境に本手法を適用するためには、入出力に関わる変数を抽出するために用いる関数の名前を対応する別の名前で置き換えて抽出すればよい。もう一つの典型的な神経細胞シミュレーションの環境である GENESIS の対応する関数に含まれる変数を抽出すれば、この手法は GENESIS 環境において動くモデルにも同様に適用可能である。さらに、シミュレーション環境毎にインタフェースプログラムを用意する必要がある。

6 結論

生命科学、神経科学知識の構造化を目的として、データベースに登録されたそのままでは使いづらい神経細胞モデルを対象とし、再利用可能な形に再構成する手法を提案した。手法は以下の三段階で構成される。

- まず、プロット関数を手がかりに入出力変数とパラメータを明らかにする。単位や意味は、対応する論文から見つけることができる。
- 次に、明らかにした入出力データを UDF (User Definable Format) の形に整理する。
- 最後に、シミュレータを統合するプラットフォームにつないで、データを受け渡しできるようにする。

神経細胞モデルデータベース ModelDB に登録されている、NEURON シミュレータ上で動く海馬の苔状線維モデルに適用して本手法の有効性を示した。他のモデルと結合可能なモデルとアノテーションされたシミュレーションデータが得られた。生命科学や神経科学の分野においてデータや知識を統合するための様々な試みの中で一つの独自の方法を提案するものである。今後は、ユーザビリティが向上したモデルとデータをネットワーク上で公開していく。さ

らに，これらのモデルを組み合わせる異種のモデルから構成される大規模な神経細胞ネットワークモデルを構築する．NEURON シミュレータ以外の，異なる種類のシミュレーション環境において実行可能なモデルを，同様に共通のインタフェースでつなぐ．そして，スケールの異なる現象を記述するデータやモデルを扱うことが可能な神経系マルチスケールシミュレーションプラットフォームへと展開する．

謝辞

本研究は，「科学技術振興機構戦略的創造研究推進事業 さきがけ」神経系の双方向マルチスケールシミュレータの開発，「文部科学省科学研究費補助金特定領域研究生命システム情報（領域番号#014）の支援を受けた．UDF 技術について御教示頂いた東京大学土井正男教授に謝意を表す．

参考文献

- [1] Joshi-Tope, G., Gillespie, M., Vastrik, I., D'Eustachio, P., Schmidt, E., de Bono, B. et al.: Reactome: a knowledgebase of biological pathways, *Nucleic Acids Res.*, Vol. 33, No. Database Issue, pp. D428–32 (2005).
- [2] Kanehisa, M. and Goto, S.: KEGG: Kyoto Encyclopedia of Genes and Genomes, *Nucleic Acids Res.*, Vol. 28, pp. 27–30 (2000).
- [3] Shepherd, G. M., Mirsky, J. S., Healy, M. D. and Singer, M. S.: The Human Brain Project: neuroinformatics tools for integrating, searching and modeling multidisciplinary neuroscience data, *Trends Neurosci.*, Vol. 21, No. 11, pp. 460–8 (1998).
- [4] Koslow, S.: Should the neuroscience community make a paradigm shift to sharing primary data?, *Nat. Neurosci.*, Vol. 3, No. 9, pp. 863–5 (2000).
- [5] Chicurel, M.: Databasing the brain, *Nature*, Vol. 406, No. 6798, pp. 822–5 (2000).
- [6] Otake, M. and Takagi, T.: Reassembly and interfacing neural models registered on biological model databases, *Genome Informatics*, Vol. 16, p. In Press (2005).
- [7] Tomita, M., Hashimoto, K., Takahashi, K. and Shimizu, T. S.: E-CELL: software environment for whole-cell simulation, *Bioinformatics*, Vol. 15, No. 1, pp. 72–84 (1999).
- [8] Bower, J., Beeman, D. and Hucka, M.: The GENESIS Simulation System, *The Handbook of Brain Theory and Neural Networks, Second edition* (Arbib, M., ed.), MIT Press, Cambridge, MA, pp. 475–478 (2003).
- [9] Beeman, D. and Bower, J. M.: Simulator-independent representation of ionic conductance models with ChannelDB, *Neurocomputing*, Vol. 58–60, pp. 1085–1090 (2004).
- [10] Usui, S.: Visiome: neuroinformatics research in vision project, *Neural Netw.*, Vol. 16, No. 9, pp. 1293–300 (2003).
- [11] Migliore, M., Morse, T. M., Davison, A. P., Marengo, L., Shepherd, G. M. and Hines, M. L.: ModelDB: making models publicly accessible to support computational neuroscience, *Neuroinformatics*, Vol. 1, No. 1, pp. 135–9 (2003).
- [12] Hines, M. L. and Carnevale, N. T.: The NEURON simulation environment, *Neural Computation*, Vol. 9, pp. 1179–1209 (1997).
- [13] Doi, M.: Challenge in polymer physics, *Pure Appl. Chem.*, Vol. 75, No. 10, pp. 1359–1615 (2003).
- [14] Doi, M.: OCTA, <http://octa.jp/>.
- [15] Magee, J. C. and Cook, E. P.: Somatic EPSP amplitude is independent of synapse location in hippocampal pyramidal neurons, *Nat. Neurosci.*, Vol. 3, No. 9, pp. 895–903 (2000).
- [16] Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H. et al.: The Systems Biology Markup Language (SBML): A Medium for Representation and Exchange of Biochemical Network Models, *Bioinformatics*, Vol. 19, No. 4, pp. 524–531 (2003).
- [17] Goddard, N. H., Hucka, M., Howell, F., Cornelis, H., Shankar, K. and Beeman, D.: Towards NeuroML: model description methods for collaborative modelling in neuroscience, *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, Vol. 356, No. 1412, pp. 1209–28 (2001).
- [18] Delorme, A. and Thorpe, S.: SpikeNET: An Event-driven Simulation Package for Modeling Large Networks of Spiking Neurons, *Comput. Neural Syst.*, Vol. 14, pp. 613–627 (2000).