巻頭言

されどパスワード

伊藤 和人
情報メディア基盤センター長

あなたはどんなパスワードをお使いだろうか。「password」「123456」「asdfghjk」「1qaz2wsx」などだったら直ちに変更して頂きたい。これらはよく使われるパスワードランキング上位の常連であり、簡単にばれてしまう。他にも数字のみ、数字と辞書に載っている単語の組み合わせ、人や物の名前もパスワードとして不適切である。

パスワードスプレーというサイバー攻撃手法がある。パスワードとしてよく用いられる文字列のリストを用意し、PC やサーバなどネットワークに接続されている情報システムに対してリスト中の文字列を順にパスワード候補としてログイン（サインイン）を試みる手法である。ログイン ID の利用者の情報からパスワードを類推するといった高度な手法ではない。とにかく、どこかにログインできる情報システムがないか、世界中を試して回る単純な攻撃である。そのパスワード候補リストには冒頭に示したような文字列も含まれているだろう。攻撃は PC などのコンピュータを使って自動化されていて、攻撃者はどこかのシステムにログインできるのを待っている。一定時間内に何回かログインに失敗するとアクセスを遮断するセキュリティツールも存在するが、敵もさるもの、ツールに設定された遮断条件を免れるような時間間隔でログイン試行を繰り返す。1 分間に 1 回だけの試行であっても、1 か月間継続すれば 43,200 回になる。複数の ID に同時にログイン試行すれば、ID の数だけ攻撃回数は増える。単純な攻撃ではあるが防御は難しく、大量に行われると脅威となる。

「root」「admin」「administrator」「user」といった ID は真っ先にパスワードスプレー攻撃の対象となる。自分が利用している情報システムにこれらの ID があれば十分な強度のパスワードを設定して頂きたい。使用しない ID は削除すること、外部からログインする必要がなければ外部からのアクセスを一切遮断するといった対策も検討して頂きたい。

ログインされても自分の PC やサーバには大した情報は入っていないから大丈夫などと考えてはいけない。ログインを許した PC から自分になりすまして悪意あるメールの送信や、その PC がさらにパスワードスプレーなどの攻撃元として悪用される可能性もある。パスワードを使いまわしている別の重要なシステムにログインされる恐れもある。

ログインにおける多要素認証という仕組みがある。この仕組みでは、パスワード（1 つ目の要素）が破られても、スマホなどに送付される認証コード入力やスマホアプリで承認操作（2 つ目の要素）をしないとログインを許可しないため、不正ログインを防ぐ方式として導入されている。しかし現時点では多要素認証の利用は限定的であり、依然としてパスワードは重要である。強度の高いパスワードについては、情報処理推進機構（IPA）の「チョコっとプラスパスワード」のページなどが参考になる。十分な強度のパスワードを利用することと、パスワードの使いまわしをしないことをお願いしたい。

日本語における基本色カテゴリーの情報科学的分析

栗木一郎

１．はじめに

我々は視覚から外界の情報の８割を取得していると言われている。これは末梢から脳に向かう神経繊維の数をもとにした推測だと言われている。視覚で得られた情報を共有する際の１つの方法として言葉を用いる。特に私が研究対象としている色の感覚は、手で触ったり物体の形で共有する事ができない概念のため、言葉が使われる事が多い。色を伝えるための言葉を「色名（しきめい）」と言う。通常、１つの色名は複数の少しずつ異なる色を集めたグループに対して適用される。この色のグループを、専門的には「色カテゴリー」と言う [1]。

図１には異なる 330 の色が示されており、それぞれの色の違いには容易に気づけると思われる。一方で、例えば「緑」と呼べる色の領域を囲むことを考えると、相当な数の色がその領域に含まれるだろう。「黄色」であったらその色数はぐっと減るかもしれないが、やはり複数色をまとめた領域になる。一方で、果物が収穫に適した成熟度にあるかを色で見極める場合、その色の違いを言葉で表現することは非常に難しくなる。摘果の是非というカテゴリー化を脳内で行っているにも関わらず、言葉とは対応が取りづらい。このように日常的には色名と色カテゴリーは非常に密接に関連しているが、言葉と対応しない色カテゴリーも少なからず存在する。

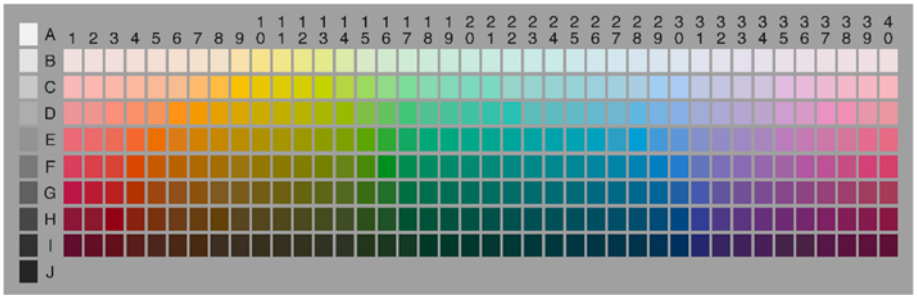


図１．WCS カラーサンプル[2]（330 色）

日本語には色名が多数あることが報告されているが、日常的に誰もが共通で使える言葉でなければ公共の意思疎通の役には立たない。誰もが共通して使える色名を「基本色名」と言うが、日本語にはいくつの基本色名があるのか、という問いに答えるのは意外に難しい。

大きな問題の１つは同義語の存在である。例えば、外来語の「ピンク」と「桃色」とが指す色は同じなのか違うのか、誰が正解を知っているだろうか。一つの評価方法として、全く同じ構成要素からなる色の集合（＝色カテゴリー）に対し、異なる人が違う色名を当てていた場合、それらの色名は同義語とみなす事ができるだろう。この「同義」の定義は言語的

な研究におけるものとは異なるが、機能的には同じことを意味していると考えられる。

もう１つの問題は、色名に関する語彙数の違いである。ある人は緑を１つの大きなグループとして扱うのに対し、別の人は複数の色名（例えば、ターコイズ、エメラルド、等々）で細分化していた場合、その２人は全く意思疎通ができないのか。例えば後者の人が細分化された緑系の色名で命名した色カテゴリーを総合した色票群と、前者の人が単一色名（緑）で命名した色票群と完全に一致すれば、前者は後者の概念をより細かく分割しているだけ、と評価できるだろう。

２．日本語の基本色名数

このような評価指標を数値的に求める方法について考えてみる。例えば、「ある人が色名 A として集めた色票(図１のような標準の色サンプルのこと)のグループ(＝色カテゴリー)と、別の人が色名 B として集めた色票のグループ（＝色カテゴリー）の一致度」を評価すれば、用語の一致／不一致に関する判断を回避できる。すなわち、同じ色票セットの各色に対して複数の人が命名したデータを分析し、類似性の高い色カテゴリーがいくつ存在するかを明らかにすれば良い、という数値的問題に落とし込むことができる[3-5]。語彙数の問題も、ある人の複数の色カテゴリーを集約した境界線が、他の人の異なる数の色カテゴリーを集約した境界線と一致した時は同一色カテゴリーだと評価すれば問題は解決する。ここで注意していただくと、この評価手法では、色に関する言葉（色名）は主役ではなく、色カテゴリーと呼ばれる色票セットの類似度が主な評価対象になる。つまり、パターンの類似度で数値化すれば良いことになる（図２）[3-5]。

英語に関しては過去に同様の研究が行われていた[5]ため、我々も同じ手法を日本語に適用してみることにした[6]。具体的には、57 人の日本語を母語とする実験参加者（主に大学の教員、学部生、大学院生）に対し、先行研究[1-5]と同じ 330 枚の標準色票（図１）の色票を１枚ずつ呈示し、それぞれを自由な色名で呼んでもらう。ただし、色名の用法には制限があり、修飾語（「薄い」赤など）や複合語（「黄緑」、「赤紫」など）は使用できず、また特定の物にしか使われない色名（髪に対する「ブロンド」など）は使用できないが、物体名を用いることは可能とした。これは基本色名を導出する研究で用いられる基本的なルールの一つで、単一語彙に基づく制限 (monolexemic restriction) [1]と呼ばれる。

１人の参加者の中で、ある色名で呼ばれた色票に１、それ以外の色票に０を割り当てると、要素数 330 の２値ベクトルが作れる。１人の参加者に対し、その人が使った色名数と同じ数の２値ベクトルが得られる。１人あたりの平均の使用色名数は 17.8 ± 6.7 語（平均 ± 分散）だったため、被験者全体で約 1,000 個の２値ベクトルが得られた。この２値ベクトルの例を以下に示す。

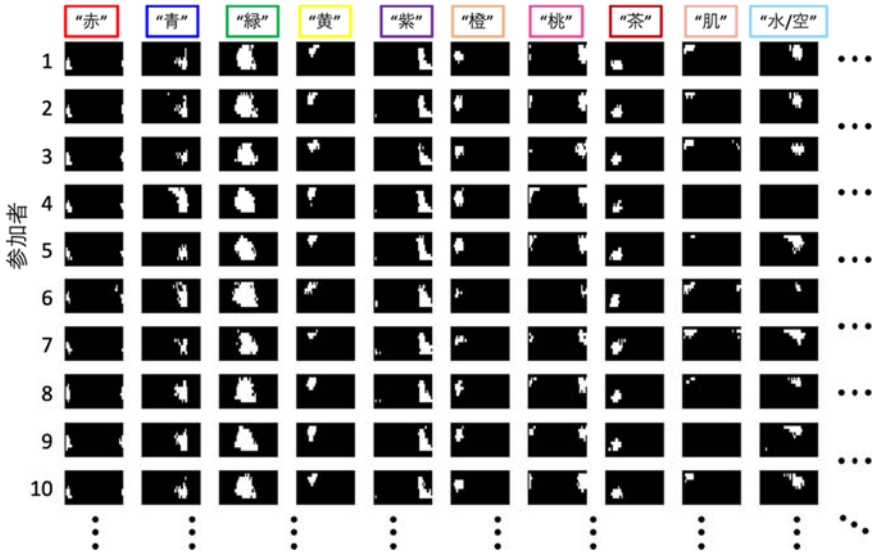


図２．２値ベクトルの例。

図２の個々の黒い四角は図１の色座標（10 行 41 列）と対応しており、画素の各点が図１の個々の色票に相当している。白い領域は２値ベクトルの１、黒は０を示している。行方向は実験参加者の違い（個人差)、列方向は色カテゴリーの違いを示している。各列の上に示した色名は各色カテゴリーの代表的なものである。縦方向に見比べて頂くと、多少の個人差はあるものの、白い領域の位置や大きさが概ね似通っていることが見て取れる。

この２値ベクトルを、パターンの類似性に従って最少数のクラスターに分割することにより、実験参加者間の基本色名の数を推定できる。クラスター分割には k-平均法を用い、最適なクラスター数の導出には gap 統計量[7]を基準とする方法を用いた。k-平均法における距離の導出には、ピアソンの相関係数を用いた。分割数 k の時の Gap 統計量 G(k)はクラスター分割後の残差総和をもとにした値であり、最適なクラスター分割数に近づくに従い、k の値を１つ増やすごとに値が小さくなる。k+1 と k の Gap 統計量の差分 G(k+1) – G(k) を計算し、その値が正から負に初めて転じる k の１つ手前を最適値とする定義の方法を用いた[4-6]。

その結果、日本人話者 57 名のデータの最適クラスター数は 19 と判定され、それらは以下の通りである：赤、黄、緑、青、ピンク、オレンジ、茶、紫、白、黒、灰、水、肌、黄土、紺、クリーム、抹茶、エンジ、山吹 [6]。ただしこれらの色名は、各クラスターで使用者が最も多かったものをラベルとして用いているため、例えば、多数が水色と呼んだのと同じ色票群を別の参加者が空色と呼んだ場合もあった。上記の色名のうち、赤～灰までの 11 色名は多くの国の言語で共通して現れる色名で「ユニバーサル基本色名[1]」と呼ばれる。

図３．k-平均法によるクラスタリングの結果。有彩色の最適クラスター(k=16)。グレースケールは、各クラスターに分類された参加者全員（N=57）の２値ベクトルの平均値である。

一方、日本人話者の全員が同じ色カテゴリーを全て使用したわけではなく、ある者は水色を使わなかったり、紺、クリーム、エンジに相当する色カテゴリーを使わなかったりした。このような色カテゴリー（色名ではない）の使用者数を対数軸に表したグラフ（図４）を見ると、上記の色名のうち肌色、黄土色あたりまでは 80%以上の参加者が使用しているため、フラットな領域として現れる。一方で、参加者によって使われなかった色名は急峻なスロープの上に乗っている。従って、水、肌、黄土の３カテゴリーは、ユニバーサル基本色名には含まれない日本語固有の基本色カテゴリーと呼んで差し支えないと思われる[6]。

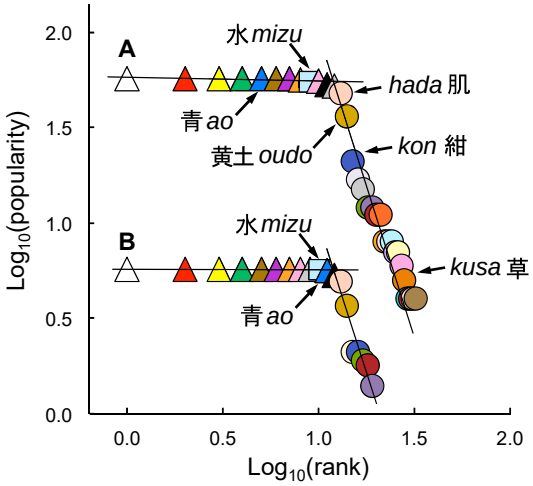


図４．色名の使用者数と順位。いずれも対数軸で表示。

３．水色、青、緑

水色のカテゴリーは興味深く、北米英語における研究と比較すると、青（blue）の領域から分割しているように見える。日本語における基本色名の先行研究[8]を見ると、その研究の参加者にも水色という色名の使用者は見られた。しかし、被験者間の一致度が低いために、ある実験参加者が水色と呼んだ色票に対し、他の参加者が青と呼んだ比率（青と水色の重複率）が平均約 80%と高かった、と報告している。つまり、参加者間において色カテゴリーの境界に個人差が大きかったことを意味している。今回の研究データにおける分離度を比較するため、改めて２つの色カテゴリーの重複率を計算してみた。先行研究（N=10）と我々の研究（N=57）では参加者数が異なるため、我々のデータから 10 人分をランダムにサンプリングして重複率を計算するプロセスを 10,000 回くりかえし、その度数分布を作成した（図５）。その結果、水色と青の重複度の中央値は約 60 %で、赤とピンクの色カテゴリーの重複度とほぼ一致していたことから、我々の実験への参加者の間では、赤とピンクの色カテゴリー境界と同程度の確かさで水色と青のカテゴリーが分離していたと言える。

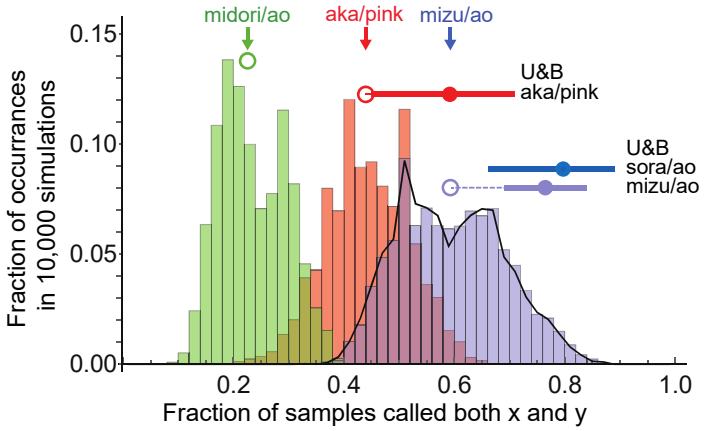


図５．重複率（横軸）の頻度分布。

日本語には、緑色のものを青と呼ぶ習慣がある。例えば、青虫、青菜、青々とした新緑、青信号、などである。日本語では青と緑の色カテゴリー境界が曖昧なのだろうか？世界には色名の語彙が少ない言語・文化があり、その中には青と緑を１つのカテゴリー”grue（グルー：green と blue の混合による造語）”として扱う人々もいる事が知られている。その grue カテゴリーは複数の言語間で共通の色カテゴリーとして現れる事が示されている[4]が、果たして日本語の「青」は、この grue カテゴリーなのだろうか。

先ほどの、水色と青の重複率を計算したのと同じ方法で、今回のデータの青と緑の色カテゴリー境界の重複率を計算してみたところ[7]、中央値が約 20%と非常に低く、新たな分離が発見された水色と青の重複率（約 60%）よりも小さい値である。そもそも、全ての被験者が青と緑の両方を異なる色票群（色カテゴリー）に対して用いており、k-平均法によるクラスター分割でも k の値が小さい頃から既に青と緑の色カテゴリーは分割していた。一方

で、水と青の色カテゴリーは k の数が 9 辺りになるまで現れなかったことから、青と緑の境界は青と水色の境界よりも重複率が低い事を別の形で裏付けている。

青と緑の色カテゴリーの境界は乳幼児の時点で分かれている事が、神経生理学的応答によって調べられている。Yang ら[9]は近赤外分光法を用いて 5–6 ヶ月児 16 名の後頭側頭部（OT 領域）の脳活動を計測したところ、乳幼児が観察しているモニターに 2 色の緑の交代を示した時より、同じ色差を持つ緑と青の交代を示した時に有意な脳活動の上昇を認めたと報告している。この結果は、色のカテゴリー分割は言語の獲得とは別に神経レベルでは進行しており、言語との対応や色カテゴリーの成形（他者とのコンセンサス）は成長の途上で得られていくことを示唆している[10]。

こうした色カテゴリーの使用に関する個人の傾向を、どの色カテゴリーの語を何%の色票に用いたか、というベクトルで表す事ができる。例えば日本語の場合、19 色カテゴリーの各々に何%の色票を割り当てたかを算出すれば、参加者数（N=57）と同じ 19 次元のベクトルを作る事ができる。例えば水色や肌色というカテゴリーを全く使わない人に対しては、それらの色カテゴリーに対応した要素はゼロとする。このように定義したベクトル群を k-平均法によってクラスター解析し Gap 統計量を用いて最適化すると、２つの類型に分類することができた。1 つは水色を多く使う類型、もう 1 つは水色をあまり使わない類型に分かれた。このような解析方法を Lindsey & Brown [6]はモチーフ（motif）解析と呼んでいる。

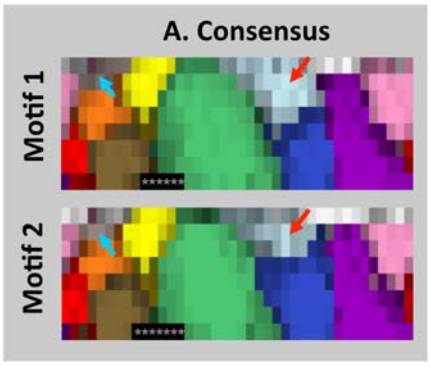


図６．日本語の色名使用の２つの類型（モチーフ）

５．日本語の色カテゴリーの相互情報量

最後に、日本語の色カテゴリーはどの程度の情報伝達の効率を持っているか、グループ内の相互情報量（group mutual information; GMI）という指標によって評価してみた[6, 11]。その定義は以下の通りである。

$$GMI(C_S; C_R) = \sum_{s,\ r} p_N(s,\ r) log_2 \left(\frac{p_N(s,\ r)}{p_N(s) p_N(r)} \right) \quad (1)$$

これは、発話者の色名が受話者に伝える色の曖昧さを削減しうる情報量を表す指標である。ここで C_S、 C_R は発話者（sender: S）と受話者（receiver: R）の選択した色票に対す

る色名、$p_N(s, r)$は発話者・受話者が色サンプル s, r を選択する同時確率を要素とするマトリクス、$p_N(s)$, $p_N(r)$はその周辺確率を示す。

この相互情報量は色名が増え、かつ参加者間の一致度が高いほど大きな値を示す。例えば、ユニバーサル基本色名の 11 色名が最も理想的な状態で効率よく用いられた場合の理想値は 3.46 ビット（$\log_2(11) = 3.46$）であり、32 色名の場合には 5.0 ビットが理想値となる。北米の英語話者[5]（N=51）の使用色名数は 21.9 ± 7.6 語（平均±分散)、基本色カテゴリー数は 20 で、そのデータによる計算結果は 1.83 ± 0.04 ビット（平均 ± 95%信頼区間；信頼区間は 2,000 回のブートストラップにより求めた）となった。使用色名数が 17.8 ± 6.7 語（平均±分散)、基本色カテゴリー数が 19 の日本語[6]（N=57）の結果は 2.04 ± 0.03 ビット（平均 ± 95%信頼区間）となった。それぞれの言語において、基本（最適）色カテゴリー数に分割した際に同じカテゴリーに属するものをまとめた「名寄せ」処理を行なった場合の GMI は、生データよりわずかに上昇し英語では 2.15 ± 0.03 ビット（平均 ± 95% 信頼区間）だったのに対し、日本語では 2.28 ± 0.02 ビット（平均 ± 95%信頼区間）とわずかに高い値を示した。これらの結果は、色名数の多少よりも日本人話者の間の個人差の少なさが原因であると考えられる[6]。

６．おわりに

ここで紹介した k-平均法を用いた色名呼称データの分析手法は、言語としての性質（語源や変遷の歴史）に影響されることなく、抽出された色票群の類似性から同義語を特定し、基本色カテゴリー数を客観的指標により導出する方法として、どの言語でも用いる事ができる。例えば中国語の標準語である北京語については、研究によって基本色名数が大きく異なっておりユニバーサル基本色名に満たない 6 とする研究も、ユニバーサル基本色名とほぼ同じ 11 とする研究もある。これらの研究で齟齬が生じている理由の１つが同義語の扱いである。中国では長い歴史の間に異なる民族の支配が入れ替わり、それぞれの民族が主として用いた語などが現在の共通語の中に複数の同義語として残っている。例えば、単一語彙（１文字）で表される日本語の赤の同義語だけでも以下の６つがある：赤 chì、 丹 dān、紅 hóng、 血 xuè、 赭 zhě、朱 zhū [12]。ユニバーサル基本色名を提唱した Berlin & Kay [1]の定義を厳密に適用すると、この同義語の存在は基本色を構成する要件のうち「顕著性（その色カテゴリーに関して、話者間に共通する第一選択肢であること)」を満たさない。最も厳しくこの規則を適用すると基本色名の数が 6 となる[1]というのが中国語の色名研究では特に問題となった点である。この中国語のデータに k-平均法を用いて解析を行なった研究については、また別の機会に紹介したいと考えている。

また、２値ベクトルの形に直しているため、多言語の実験データを混ぜて解析することもできる。実際に、英語と日本語のデータをモチーフ解析する試みも実施している[6]。その結果、モチーフは３種類現れ、１つは基本 11 色で主に分類するユニバーサル型、２つは水色カテゴリーを用いる日本型、３つめは青と緑の中間色にティール（teal）という色名を当

てる北米型であった（図７）。これらのモチーフに分類される内訳を見ると、日本型（motif 2）に分類されたのは日本語話者 N=51 と英語話者 N=2、北米型（motif 3）に分類されたのは英語話者 N=13 と日本語話者 N=2、ユニバーサル型（motif 1）は英語話者 N=36 と日本語話者 N=4 だった。この結果は、色カテゴリーの境界は言語の影響を強く受けるものの、個人の用法は必ずしも言語グループに束縛されないことも明らかになる。

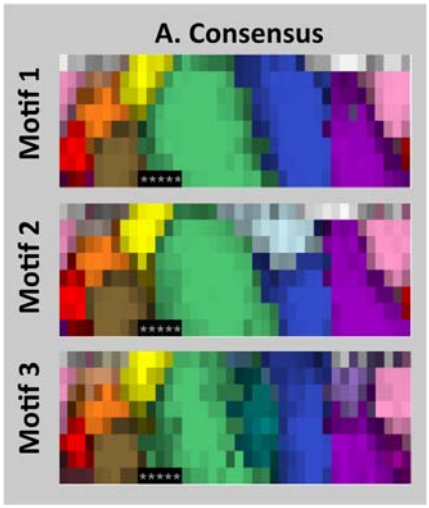


図７．英語・日本語話者のデータ（N=108）から得られた３つの類型（モチーフ）。Motif 2 は水色の領域が、motif 3 は青と緑の間の"teal"と紫と別に"lavender"の領域があることが特徴。

こうした情報科学に基づく分析手法を用いることによって新しい発見が得られ、特に今回報告した日本語の研究[6]では、基本色カテゴリーの数（= 19）と水色の青からの分離が挙げられる。またこの方法を他の言語データにも適用し、色情報表現の文化的要因と個人差について考察を深めていきたいと考えている。

【参考文献】

[1] Berlin, B., & Kay, P. (1991). *Basic color terms: Their universality and evolution*. Univ of California Press.

[2] Kay, P., Berlin, B., Maffi, L., Merrifield, W. R., & Cook, R. (2009). *The world color survey.* Stanford, CA: CSLI Publications.

[3] Lindsey, D. T., & Brown, A. M. (2006). Universality of color names. *Proceedings of the National Academy of Sciences*, *103*(44), 16608-16613.

[4] Lindsey, D. T., & Brown, A. M. (2009). World Color Survey color naming reveals universal motifs and their within-language diversity. *Proceedings of the National Academy of Sciences, 106*(47), 19785-19790.

[5] Lindsey, D. T., & Brown, A. M. (2014). The color lexicon of American English. *Journal of vision, 14*(2), 17-17.

[6] Kuriki, I., Lange, R., Muto, Y., Brown, A. M., Fukuda, K., Tokunaga, R., Uchikawa, K., Lindsey, D.T., & Shioiri, S. (2017). The modern Japanese color lexicon. *Journal of Vision, 17*(3), 1-1.

[7] Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63*(2), 411-423.

[8] Uchikawa, K., & Boynton, R. M. (1987). Categorical color perception of Japanese observers: Comparison with that of Americans. *Vision research, 27*(10), 1825-1833.

[9] Yang, J., Kanazawa, S., Yamaguchi, M. K., & Kuriki, I. (2016). Cortical response to categorical color perception in infants investigated by near-infrared spectroscopy. *Proceedings of the National Academy of Sciences, 113*(9), 2370-2375.

[10] Kuriki, I. (2019). Emergence and separation of color categories: An NIRS study in prelingual infants and a k-means analysis on Japanese color-naming data. *Current Opinion in Behavioral Sciences, 30*, 21-27.

[11] Lindsey, D. T., Brown, A. M., Brainard, D. H., & Apicella, C. L. (2015). Hunter-gatherer color naming provides new insight into the evolution of color terms. *Current Biology, 25*(18), 2441-2446.

[12] Sun, V. C., & Chen, C. C. (2018). Basic color categories and Mandarin Chinese color terms. *Plos one, 13*(11), e0206699.

連続ウェーブレット変換を用いた植物生体電位の交流成分分析

長谷川　有貴

１．はじめに

近年、植物の生育にかかわる環境因子を人工的に制御し、無農薬で周年供給が可能な農業システムである植物工場の普及が進められているが、運用コストが高く、栽培の効率化や栽培作物の高品質化が求められる。このため、生長促進や生産物の付加価値を向上させる環境制御の指標としての栽培作物の生理活性評価が重要となる。そこで本研究では、細胞内外の電位差により発生し、植物の生理活性と密接に関係して応答することが知られる植物生体電位に着目した。これまでの研究で、我々は植物生体電位における直流成分の解析結果を基に光源制御システムを開発し、光合成や呼吸等の生理活性を非侵襲で評価できることなどを報告してきた[1]が、直流成分のデータを用いた評価には時間がかかることや、一定時間ごとの光源の消灯が必要であることが課題であった。

そこで本研究では、より短時間での評価を可能とするために交流成分に着目し、時間周波数解析手法である連続ウェーブレット変換（Continuous Wavelet Transform、以下 CWT）を用い、周囲環境の変化に対する植物生体電位応答の交流成分特性について解析、評価した。

２．実験方法

2.1　植物生体電位の測定方法

植物生体電位を測定する実験環境の概略図を図 1 に示す。実験では、研究室内にある植物工場（温度 25 ℃、湿度 50%）で栽培したリーフレタスを測定対象として用い、植物体の下方の茎に挿入した脳波用針電極を基準電極とし、光がよく当たる葉の葉脈に挿入した電極との電位差を高入力インピーダンス対応のディジタルマルチメータで測定した。また、植物の光合成活性と生体電位応答の関係を比較するため、生体電位測定と同時に、NDIR(Non Dispersive InfraRed)型 CO_2 ガスアナライザを用いて測定容器内の CO_2 濃度を測定した。

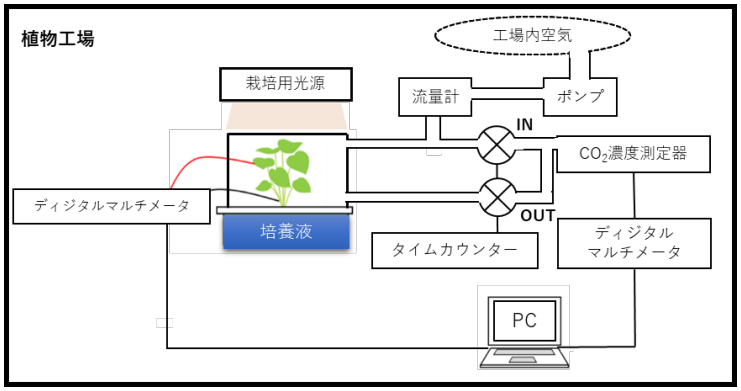


図 1　植物生体電位測定の実験環境の概略図

2.2　連続ウェーブレット変換（CWT）

CWT は時間-周波数変換手法であり、心電図や地震データなどの非定常信号の解析に適した手法の一つである。CWT によって得られた結果は、時間、周波数、応答値の 3 つの情報を持ち、これをカラーマップで表示することで時間-周波数特性を視覚的に把握することが可能となる。実数の集合を$\mathbf{R}$とし、$\mathbf{R}$ 上の二乗可積分関数の空間を$L^2(\mathbf{R})$ で表す。平均値がゼロで原点を離れると急激に振幅が小さくなるような時間の関数$\varphi(t)$（$L^2(\mathbf{R})$に内包される）が、許容条件式(1)を満たすとき$\varphi(t)$はマザーウェーブレット(以下 MW)と呼ばれる。

$$C_\varphi = \int_{-\infty}^{\infty} \frac{|\hat{\varphi}(\omega)|^2}{|\omega|} d\omega < \infty \tag{1}$$

$\hat{\varphi}(\omega)$は$\varphi(t)$のフーリエ変換であり、ωは角周波数で$\omega = 2\pi f$である。MW が与えられたとき、任意関数$x(t)$の連続ウェーブレット変換は式(2)で定義される。

$$W(a,b) = \int_{-\infty}^{\infty} x(t)\hat{\varphi}_{a,b}(t)dt \tag{2}$$

ただし$\hat{\varphi}(t)$は$\varphi(t)$の複素共役とし、$\varphi_{a,b}(t)$は MW を用いてスケールの拡大縮小および時間 b のシフトによりつくられたウェーブレット関数系であり式(3)で表される。本研究では Morse ウェーブレットを MW として用いており、式(4)の逆フーリエ変換である。また、βとγは両端の減衰やウェーブレットの幅のパラメータであり、任意の値を取ることで MW として用いられる。

$$\varphi_{a,b}(t) = a^{-\frac{1}{2}}\varphi\left(\frac{t-b}{a}\right) \tag{3}$$

$$\Phi_{\beta,\gamma}(\omega) = U(\omega)a_{\beta.\gamma}\omega^{\beta}e^{-\omega^{\gamma}} \tag{4}$$

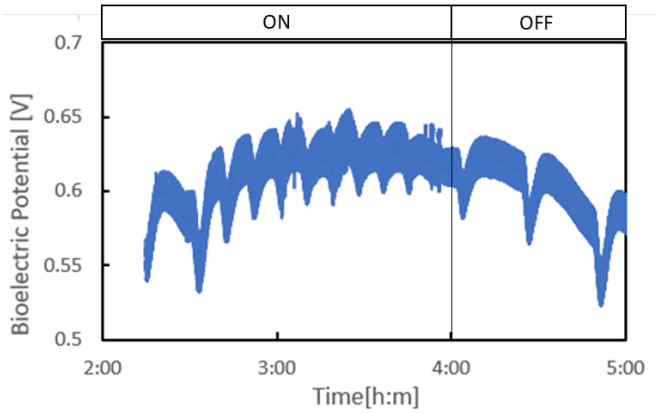
この定義からわかるように、フーリエ変換が時間tの関数から周波数fの関数への分解であるのに対して CWT は周波数(1/a)と時間 b の 2 変数関数$W(a,b)$への分解である[2]。CWT による時間周波数解析は、周波数に応じて時間分解能および周波数分解能を適切に変化させることができ、経時的に変化していく信号の周波数解析に有効である。本研究では、Matlab2016b により、Morse ウェーブレットを用いて連続ウェーブレット変換を行った[3]。

３．実験結果

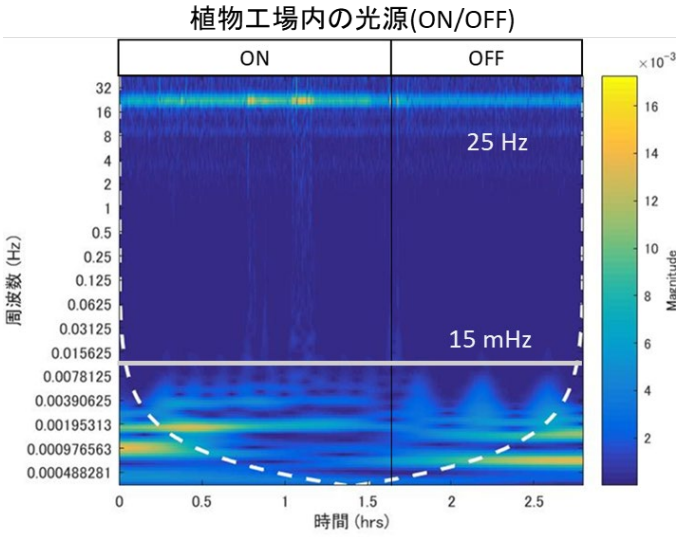
サンプリング周波数を 100 Hz として測定した植物生体電位応答とその CWT 結果を図 2 に示す。光照射時には、2 mHz 付近と 1 mHz 付近に応答が見られ、光遮断時には、1 mHz と 0.5 mHz 付近に応答が見られた。また、カラーマップの暖色が低周波側に存在していることから植物生体電位応答は 15 mHz 以下に集中する傾向があると考えられる。さらに、光の照射、遮断に関わらず、25 Hz 付近に応答が見られており、光合成ではなく、定常的な生理活性に関連する応答の可能性が考えられる。よって、光に対する植物生体電位応答の交流成分は 15 mHz 以下あるいは 50 Hz 以上の周波数領域に存在する可能性が考えられる。

次に、光条件を、光照射 8:00 ~ 22:00（14h）、光遮断 OFF 22:00 ~ 8:00（10h）とした長

期的な光周期と、2 時間毎に光照射、遮断を繰り返す短期的な光周期下で測定した、リーフレタスの植物生体電位応答の CWT 結果を図 3 に示す。長期的な光周期下での CWT 結果（図 3(a)）から、光照射時には 0.5 mHz 付近と 0.25 mHz 付近に応答が見られ、光遮断時には、0.5 mHz 付近から 0.25 mHz 付近へとその応答が徐々に変化していく様子が見られた。また、短期的な光周期下での CWT 結果（図 3(b)）から、光照射時には 1 mHz 付近に応答が見られ、光遮断時に 0.5 mHz 付近に応答が見られた。以上の結果より、植物生体電位応答の交流成分の光に対する特性は、光照射時のピーク値での周波数と光遮断時のピーク値での周波数が 2 ： 1 になる傾向があると考えられる。

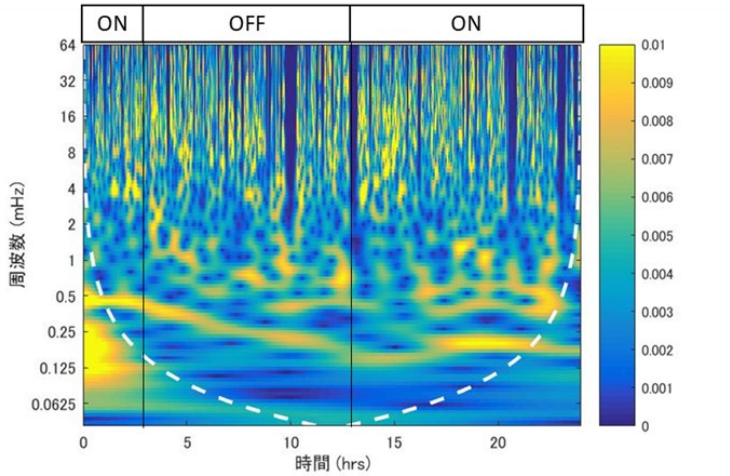


(a) リーフレタスの植物生体電位応答

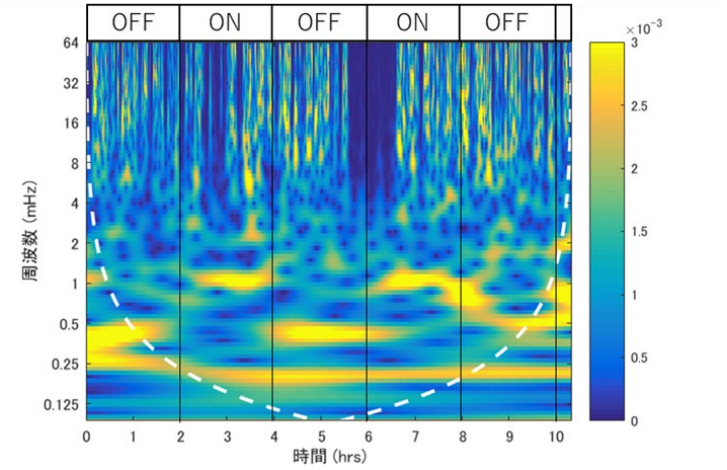


(b) (a)の CWT 結果

図 2 リーフレタスの植物生体電位応答とその CWT 結果（fs = 100Hz）



(a)長期的な光周期下



(b)短期的な光周期下

図 3 各光周期下におけるリーフレタスの植物生体電位応答の CWT 結果

４．まとめ

本研究では、植物生体電位の時間的要素を含む交流成分特性の評価を目的として、時間周波数解析手法の一つである CWT を用いた植物生体電位の交流成分分析を行った。測定対象として、リーフレタスを用い、サンプリング周波数 100 Hz での植物生体電位測定と得られた生体電位データに対して CWT を行った。その結果、光に対する植物生体電位応答の交流成分は 15 mHz 以下あるいは 50 Hz 以上の周波数領域に存在する可能性が示唆されるとともに、25Hz 付近に、光照射の有無に関わらず応答が見られたことから、光合成活性以外の定常的な生理活性に関係した応答が 25 Hz 付近に見られる可能性が示唆された。

以上の結果から、植物生体電位応答の時系列データに対して CWT を行うことで、植物の光照射、遮断時の生理活性の変化を観測が可能であり、新たな評価指標を定義することで植物の生理活性を定量的に評価できる可能性が示された。

謝辞

本レポートは、2021 年 3 月に理工学研究科博士前期課程を修了した盛本裕貴さんの修士論文を基に作成しました。

参考文献

[1] Fumiya Murohashi, Hidekazu Uchida, Yuki Hasegawa, “Evaluation of photosynthetic activity by bioelectric potential for optimizing wavelength ratio of plant,” International Journal of Biosensors&Bioelectronics, Volume 4, Issue 6, pp. 281-287, 2018.

[2] 章忠, 川畑洋昭, ”連続ウェーブレット変換による時間-周波数解析,”計算機統計学, Vol. 10, No. 2, pp. 93-101, 1997.

[3] 小野順貴, “短時間フーリエ変換の基礎と応用, ”日本音響学会誌 72 巻 12 号, pp. 764-769, 2016.

軽量プログラミング言語による分子ドッキングソフトウェアの開発

松永 康佑、山口 皓平

1. はじめに

タンパク質などの生体分子は細胞内でさまざまな分子と相互作用することで、生命活動の維持に必要な仕事を行っている。例えば、酵素は基質と結合しその反応を触媒することで、通常の環境では決して起こり得ない化学反応を実現させている。また、多様なタンパク質間の相互作用はシグナル伝達にとって重要であり、どのタンパク質とタンパク質が相互作用するのかを知ることは細胞の分子生物学にとって重要であるだけでなく、病気の原因となるタンパク質を特定する上でも有益である。タンパク質間相互作用は、抗体による抗原の中和という免疫系の観点からも重要である。また、人工物の話題へ移ると、低分子の薬剤が標的となるタンパク質のどこに？どの程度の親和性で結合するのか？という問題は、創薬におけるスクリーニングプロセスとして基本的な問題である。

このように、生体分子や薬剤分子がどのように相互作用して結合するのか(もしくはしないのか)という問題は基礎的な分子生物学だけでなく、免疫系の理解や、医療・創薬応用を含めた重要な課題である。この問題に対して、計算科学からのアプローチとして主に 2 つの手法が存在する。一つは、分子動力学(Molecular Dynamics; MD［1］)シミュレーションを用いるものである。中でもいわゆる古典的な MD シミュレーションでは、計算機の中で質点として表現した生体分子の原子座標をニュートンの第二法則に基づいて時間発展させていく手法であり、非常にシンプルである。シンプルではあるが、その応用範囲は広く、例えばタンパク質が他のタンパク質と結合するポーズ、低分子薬剤のポケットへの結合ポーズ、結合ポーズの結合自由エネルギー計算、結合・乖離のレート、など多くの物理量を計算することができる。ただし、MD シミュレーションの欠点として計算量が大きいという問題がある。典型的な系では、例えばタンパク質を数千原子で表現し、周りの水分子やイオンなどの溶媒を数万原子で表現する。MD では、こうした数万原子に働く力を全て求めて座標を更新することで 1 ステップの計算となる。普通、1 ステップで更新できる時間幅は 1 フェムト秒(10^{-15} 秒)であり、これを 100 ナノ秒(10^{-9} 秒)からマイクロ秒(10^{-6} 秒)まで計算するのが普通である。これには最新のソフトウェアとハードウェアを用いたとしてもおよそ 1 日～10 日かかってしまう。普通、例えばバーチャルスクリーニングによって薬剤の候補を絞る計算では、数百～数万の低分子が候補となるので、1 薬剤に 1 日かかってしまうだけでも実用に耐えることはできない。

計算科学のもう一つのアプローチとして歴史的によく用いられているのがドッキング計算である。ドッキング計算では、MD シミュレーションのように物理法則に従って分子を動かすことはせずに、経験的なスコア関数を用いてそれが大きくなるような結合ポーズを探

すという計算を行う。典型的には、例えばタンパク質と低分子薬剤の結合では、まず大きい方のタンパク質分子(こちらを Receptor と呼ぶ)を原点に固定する。そして小さい方の低分子薬剤(こちらを Ligand と呼ぶ)の構造を剛体として扱い、6 自由度(並進と回転)の空間を網羅的に探索し、最もスコア関数が大きくなる低分子薬剤のポーズ(並進座標と回転角)を求める、という計算を行う。このアプローチは MD に比べて計算量が小さく、比較的短時間(例えば数十秒〜数分)で計算が終了するため、バーチャルスクリーニングなどの大量の処理に用いられている。また、最初にドッキング計算を行った後で、薬剤の候補を絞り、その後で MD シミュレーションを適用するという組み合わせのアプローチもよく使われる。

ドッキング計算を行うソフトウェアは商用のものからフリーのものまで様々なものが開発されている。フリーのもの(但し商用利用は制限がある)として、例えば AutoDock［2]、ZDOCK［3]、HADDOCK［4]などのソフトウェアが有名である。これらは比較的歴史が古く、例えば 2000 年前後から開発が行われているので、ソースコードが古かったり、また一部ではそもそもソースコードが公開されていないものもある。一方で近年のハードウェア・ソフトウェアの進歩は目覚ましく、Python や Julia といった軽量な言語を用いて、メンテナンスしやすい比較的短いソースコードの分量で、更に GPU を用いた科学計算向きの高速な処理が実現可能になってきている。また近年のディープラーニングの進歩の恩恵を享受するためには、ニューラルネットのフレームワークとそのまま接続することのできるドッキングソフトウェアが必要である。ニューラルネットワークのフレームワークと接続することができれば、例えば、スコア関数の一部をグラフニューラルネットや畳み込みニューラルネットに置き換えたり、スコア関数のパラメータを微分可能にすることで誤差逆伝播法で実験構造データを訓練データとしてパラメータを訓練させることができる。そこで、メンテナンスしやすい軽量言語を用いて高速な GPU 計算を実現し、ニューラルネットフレームワークとも接続できるドッキングソフトウェアの実現を目指して、ここでは Julia 言語を用いたドッキングソフトウェアの実装について紹介する。以下では、まず実装の元となった ZDOCK の計算アルゴリズムについて紹介した後で、Julia を用いたそれらの CUDA 実装について紹介する。最後に今後の展望について述べる。

2. 背景知識：ZDOCK の計算アルゴリズム

ZDOCK は(当時)ボストン大学(現在は UMass Chan Medical School)の Zhiping Weng らによって開発されたドッキングソフトウェアで、スコア関数と計算アルゴリズムの両方を定義している[3]。まずスコア関数から説明する。スコア関数は半経験的なものであり、以下の 3 つの項を足し合わせたもので定義される。

$$S = \alpha S_{SC} + S_{DS} + \beta S_{\mathrm{ELEC}}$$

ここで、Sがトータルスコアで、S_{SC}は形状相補性(Shape Complementarity)、S_{DS}は脱水和の自由エネルギー、S_{ELEC}は静電相互作用である。αとβはそれぞれの項の寄与を調整するパラメータであり、$\alpha = 0.01$と$\beta = 0.06$が用いられる。

ZDOCK の特徴は、3 つのスコアを計算する際に、原子座標をグリッド化して分子構造をグリッドで表現することである。具体的には、1.2 Å の幅を持つグリッドを作成して、最寄りもしくは近隣に存在する原子の化学的特徴を元にグリッド点に値(複素数値)を割り当てる。グリッド化した後で、それぞれのスコア(S_{SC}, S_{DS}およびS_{ELEC})を Receptor のグリッドと Ligand のグリッドとの積で定義する。関数系が対象ならば、Ligand を併進させることは畳み込み積の形になるので、スコア計算に 3 次元 FFT(Fast Fourier Transform)を用いることができる。これによって並進空間の探索の計算量を$O(N^3)$から$O(N^2 \log N)$へと削減することができる(ここで Nはグリッド点の数)。例えば、S_{SC}の計算のためには、以下のようなルールでグリッドに値を割り合てる: まず原子を surface atom と core atom に分類する。Surface atom というのは分子表面にあり溶媒と相互作用できる原子のことで、これは溶媒露出表面積(Solvent Accessible Surface Area; SASA)を 1 Å 以上持つ原子と定義される。それ以外は core atom で、これは分子内部に埋もれた原子のことである。そしてそれぞれの原子の座標から van der Waals 距離(にファクターを掛けたもの)以内に存在するグリッド点に対し、surface atom か core atom かで異なる値を割り当てていく。例えば Receptor のグリッドでは、以下のように割り当てられる(図 1)。

$$R_{SC}(l,m,n) = \begin{cases} 1 & \text{surface of Receptor} \\ \rho i & \text{core} \\ 0 & \text{otherwise.} \end{cases}$$

ここで、ρは 9 であり、iは虚数である。Ligand のグリッドL_{SC}でも同じような割り当てを行うことで、S_{SC}は以下のように定義される。

$$S_{SC}(o,p,q) = \mathrm{Re}\left[\sum_{l=1}^{N}\sum_{m=1}^{N}\sum_{n=1}^{N} R_{SC}(l,m,n)\cdot L_{SC}(l+o,m+p,n+q)\right] - \mathrm{Im}\left[\sum_{l=1}^{N}\sum_{m=1}^{N}\sum_{n=1}^{N} R_{SC}(l,m,n)\cdot L_{SC}(l+o,m+p,n+q)\right].$$

このように複素数間の積として定義することで、core atom 同士が重なった時のペナルティーなども表現できる。既に上に述べたように、これは直接計算されるのではなく逆空間で FFT によって効率良く計算することができる。S_{DS}およびS_{ELEC}も同じ考え方で定義される。

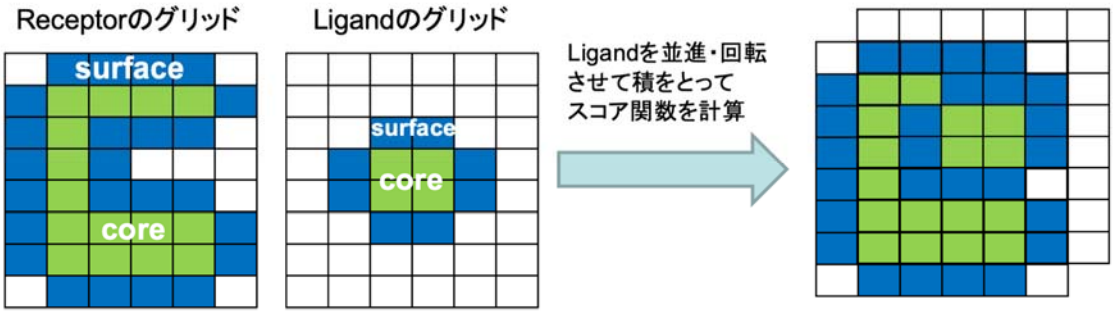


図 1：分子のグリッド化と ZDOCK のスコア関数

3. Julia 言語による実装

前節で説明した ZDOCK のスコア関数を軽量言語である Julia [5]を用いて実装し、更に GPU を用いて高速に処理するために Julia の CUDA.jl パッケージ[6]を用いて CUDA で GPU 用カーネル関数を実装した。スコア関数の計算で計算に時間がかかる場所(ボトルネック)は主に 2 つあり、(i) 分子のグリッド化と (ii) FFT である。このうち、FFT は既に用意されている GPU 用の高速 FFT ルーチンを呼び出すだけであり、CUDA.jl パッケージを用いると CPU 向けのコードがそのまま GPU で実行される。一方で分子のグリッド化は既存のルーチンが存在しないので自前で GPU 用のカーネル関数を実装する必要がある。

NVIDIA 製の GPU 向けのカーネル関数を実装するには、同社が提供している CUDA 拡張を用いる必要があるが、C 言語で書かなければいけないため、ソースコード量が大きくなりがちで開発やメンテナンスのコストが高い。Julia 言語用の CUDA.jl パッケージは、Julia 言語を用いて CUDA を書くことを可能としており、最低限の長さのコードで気軽に高速な計算を実現することができる[6]。したがって今回は CUDA.jl を用いてグリッド化のコードを書いた。グリッド化は、原子座標と全てのグリッド点座標との距離を計算し、それが van der Waals 半径以内だったら値を割り当てる、という処理を全ての原子に対して行う。これは原子毎に並列に行える計算であり、各原子の計算を CUDA コアへ割り当てて並列計算することができる(図 2)。

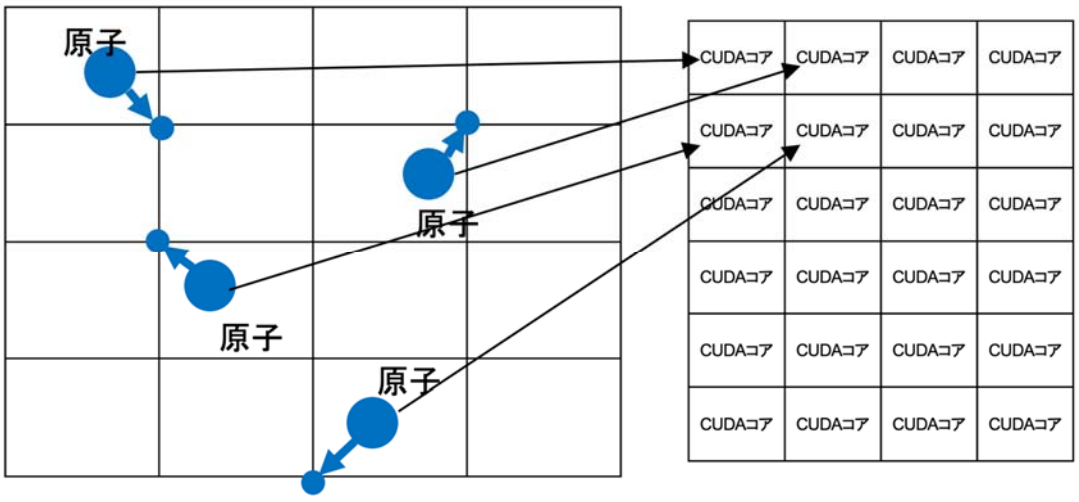


図 2：グリッド化の概念図と CUDA コアへの割り当て

実際のコードの例を図 3 に示す。これは原子の近隣(van der Waals 半径以内)のグリッド点へ指定された値を足していくカーネル関数である。厳密に言えば、グリッド点における値を足す演算だけはスレッドセーフではないため、単一のスレッドが排他的に実行することを保証するために atomic 演算を用いている。図が示すように、Julia 言語の文法を使うこ

とができるので、C 言語で書くのに比べて比較的コンパクトに書くことができ、メンテナンスの負担も軽減できる。実行速度は後に示すように C 言語で実装するのと遜色ない性能が出る。

```
function spread_neighbors_gpu_add!(grid::CuDeviceArray{T},
        x::CuDeviceVector{T}, y::CuDeviceVector{T}, z::CuDeviceVector{T},
        x_grid::CuDeviceVector{T}, y_grid::CuDeviceVector{T}, z_grid::CuDeviceVector{T},
        rcut::CuDeviceVector{U}, charge::CuDeviceVector{T}) where {T, U}
    natom = length(x)
    tid = threadIdx().x
    gtid = (blockIdx().x - 1) * blockDim().x + tid  # global thread id
    nx, ny, nz = size(grid)

    iatom = gtid
    if iatom <= natom
        for ix = 1:nx
            dx = x[iatom] - x_grid[ix]
            if abs(dx) > rcut[iatom]
                continue
            end
            for iy = 1:ny
                dy = y[iatom] - y_grid[iy]
                if abs(dy) > rcut[iatom]
                    continue
                end
                for iz = 1:nz
                    dz = z[iatom] - z_grid[iz]
                    if abs(dz) > rcut[iatom]
                        continue
                    end
                    d = dx*dx + dy*dy + dz*dz
                    if d < rcut[iatom]*rcut[iatom]
                        CUDA.@atomic grid[ix, iy, iz] += charge[iatom]
                    end
                end
            end
        end
    end


    return nothing
end
```

図 3：近隣のグリッド点へ指定された値を割り当てる CUDA カーネル関数

4. 性能評価

実装したコードを用いて、速度評価と結合ポーズの精度の評価を行った。テスト系として PDB ID: 1CGI を用いて、ZDOCK と同じグリッド幅 1.2 Å を用いて計算を行った。Surface atom と core atom を定義するための SASA などの計算は一度だけなので、計算時間から除外する。CPU1 コアのみで行った場合は 389.97 秒の実行時間であった一方で、GPU1 枚を用いた場合は 8.24 秒の実行時間であった。したがっておよそ 50 倍の速度改善が確認できた。ただし、CPU1 コアのみとの比較なのでこれを複数コア使えるように並列化を行えばその差が縮まると予想される。GPU 計算の内訳を見ると、グリッド化と FFT 共に同程度の実行時間であった。したがって我々が実装したグリッド化のカーネル関数がボトルネックでないことを確認できた。一方で、GPU コアの使用率は 50%前後だったので、今後は使用率

を高めるために原子だけでなくグリッド点における並列化も検討していく。

図 4 に計算された結合ポーズと実験構造との比較を示す。Ligand の結合ポーズは、スコア関数の値の高いものから 10 個までを重ねて描画している。正解構造をマゼンタで描画しているが、10 個の構造のうちいくつかは正解結合ポーズ付近に重なっていることが確認できる。同じような結果は ZDOCK を用いたドッキング計算でも確認できており、ZDOCK のスコア関数を精度良く再現できていることがわかる。

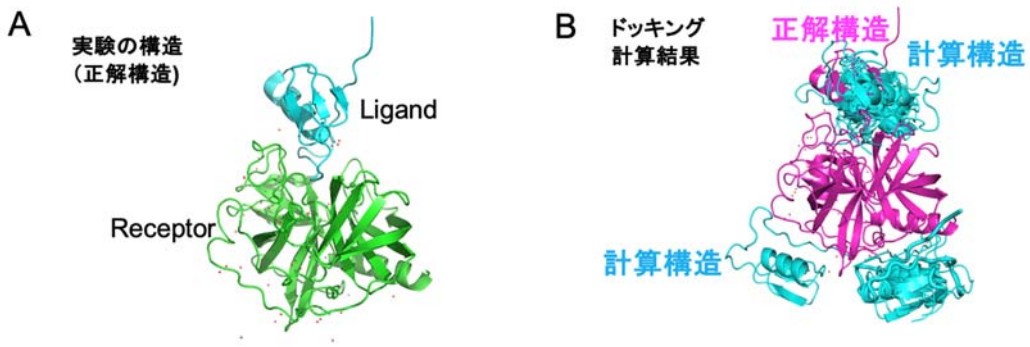


図 4：テスト系複合体（PDB ID：1CGI）を用いた結合ポーズの精度評価

5. まとめと今後の展望

本稿では、軽量プログミング言語である Julia を用いた高速なドッキング計算ソフトウェアの開発について簡単に紹介した。元々Julia 言語は CPU での高速性を目的として設計されているが、ドッキング計算の一部を CUDA.jl パッケージを用いてカーネル化することで CPU1 コアに比べて更に 50 倍の高速化を確認することができた。

はじめに触れたように、今後実装したドッキングコードをニューラルネットや機械学習のフレームワークと接続することで、スコア関数の一部をニューラルネットに置き換えたり、スコア関数のパラメータを誤差逆伝播で最適化することを検討している。特に、ZDOCK のスコア関数のパラメータとして最適化できそうなものは、溶媒和自由エネルギーの計算に必要な ACE スコアや、各項の寄与率を決定するαとβである。通常これらのパラメータを網羅的に動かして最適化することは計算量的に困難であるが、ニューラルネットと同じように損失関数を定義して勾配降下法を適用できるならば現実的な計算量で最適化することができると期待される。

6. 謝辞

本研究は以下の共同研究の成果の一部である。経済産業省戦略的基盤技術高度化支援事業 JPJ005698。

参考文献

［1］埼玉大学 情報メディア基盤センター 2020 年度年報

［2］ Goodsell, D. S. and Olson A. J. Automated docking of substrates to proteins by simulated annealing, *Proteins* **8**, 195–202 (1990)

［3］ Chen, R. and Weng, Z., Docking unbound proteins using shape complementarity, desolvation, and electrostatics, *Proteins* **47**, 281–294 (2002)

［4］ Dominguez, C., Boelens, R. and Bonvin, A. M. J. J. HADDOCK: A Protein−Protein Docking Approach Based on Biochemical or Biophysical Information, J. Am. Chem. Soc. **125**, 1731–1737 (2003)

［5］ Bezanson, J., Edelman, A., Karpinski, S. and Shah, V. B. Julia: A Fresh Approach to Numerical Computing. *SIAM Rev.* **59**, 65–98 (2017)

［6］ Besard, T., Foket, C. and De Sutter, B. Effective Extensible Programming: Unleashing Julia on GPUs. IEEE Transactions on Parallel and Distributed Systems **30**, 827-841 (2019)