修 士 学 位 論 文

題 目（和文の題目には英文を、英文の題目には和文を併記すること。）

（和文）

ニューラルネットワークを用いたドメインリンカーの予測

（英文）

Prediction of domain linker using neural network

東京農工大学大学院工学府 博士前期課程

生命工学 専攻　　平成 28 年度入学

学 籍 番 号　16641106

氏 名　河村 直樹

指 導 教 員 氏 名　黒田 裕

受 領 印

目次

1．序論

1-1. 背景・目的

タンパク質は生体内において化学反応の触媒、生体構造の形成、遺伝子発現の調整など様々な役割を持っている。そのため、それらの機能の解明は生命現象の理解のために非常に重要である。

タンパク質の構造・機能の解析を行う場合は、実験的手法を用いることが多い。例えば、分光測定、熱測定による物性の測定、X 線結晶構造解析、NMR による構造の解析が広く行われている。しかし、ミスフォールディングなどが原因で、解析のためのタンパク質試料の作成が難しいケースも多い。特に、複数のドメインを持つ多ドメインタンパク質は、巨大であるために発現精製、結晶化が非常に困難である。そこでドメインという構造的・機能的な単位に分割して構造を解析するという手法が有効とされる。しかし、実験的手法でドメイン境界領域(以下リンカーとする）を決定するためには多大な人員的・物的リソースが必要とされる。

多ドメインタンパク質の迅速な解析を行うために、計算的手法を用いてアミノ酸配列からドメインを予測する技術がこれまでに多く開発されてきた。しかし、アミノ酸組成や長さはタンパク質ごとに大きく異なるため、ドメイン領域自体を予測することは難しい。一方、リンカー領域は配列の特徴を比較的検出しやすいため、リンカーをターゲットとする予測器が多く開発されている。Domcut[1]はリンカー領域とドメイン領域におけるアミノ酸出現頻度の差に基いてリンカー部位を予測する。

一般的になっているのは機械学習を用いた予測である。ニューラルネットワークは機械学習の中で一般的な手法であり、リンカー予測にも応用されている[2]。PPRODO[3]もニューラルネットワークを用いた予測器であり、特徴量として二次構造情報や PSSM を用いることで正答率を向上させている。DomPro[4]は Recursive Neural Network を用いている点が特徴である。 DoBo[5]はアルゴリズムにサポートベクターマシン(SVM)を用い、進化的情報を利用することによってさらに精度を向上させている。本研究室の先行研究においても、SVM を用いたリンカー予測機 DROP[6]が開発されている。しかし、リンカー部位を正確に予測することは依然難しい問題であり、DROP の正答率は 3 割程度にとどまっている。

2006 年ごろから、再びニューラルネットワークが大きな注目を集め始め、画像処理・音声処理等のベンチマークにおいて SVM を大きく上回る精度を持つことが示されている。精度向上の一因に、データに潜在する構造をニューラルネットワーク自身が特徴として抽出するため、人間に設計できなかった複雑な特徴を認識できるようになったことが挙げられる。その他の関連技術も日進月歩で開発されている。

ドメインリンカー予測のさらなる精度の向上を目指すため、最新の手法を利用したニューラルネットワークを用いてリンカーを対象とした予測器の開発を目的とした。

1-2. ニューラルネットワークによる予測

本研究では、ニューラルネットワークを用いてタンパク質一次構造からドメインリンカー部位の予測を試みた。ニューラルネットワーク、畳込みニューラルネットワークの原理について説明する。

1-2-1.　ニューラルネットワークの原理

ニューラルネットワークは神経細胞をモデルとしている。神経細胞(ニューロン)は周りの複数のニューロンと接合しており、電気信号を用いて情報伝達を行っている。また、他ニューロンからの入力がある閾値を超えると電気信号を出力する仕組みになっている。この複雑な相互作用により生物は情報処理を行っていると考えられている[7]。ニューラルネットワークは、これをシミュレートすることで様々な問題の解決を目指したモデルである。

ニューラルネットワークはユニットと呼ばれる素子からなる。このユニットは以下の図のように複数の実数値を入力値として受け取り、1 つの実数値を出力する。

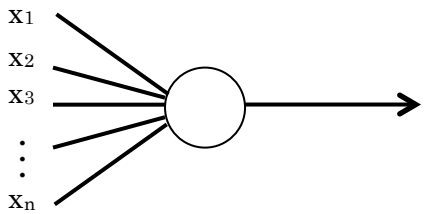


図 1.　ニューロンの模式図

ユニットを表す式は以下のようになる。

$$z = f\left(\sum_i w_i x_i + b\right) = f(\mathbf{W}\boldsymbol{x})$$

w_iはニューロン同士の結合強度をモデル化した値であり、重みと言われる。b はバイアスと呼ばれる。関数 $f(\ )$ は活性化関数と呼ばれる。閾値 $-b$ を超えると出力信号を発するようにモデル化していると考えられる。このユニットを以下の図のように複数用いて層を作り、層を重ねることでニューラルネットワークを構成できる。

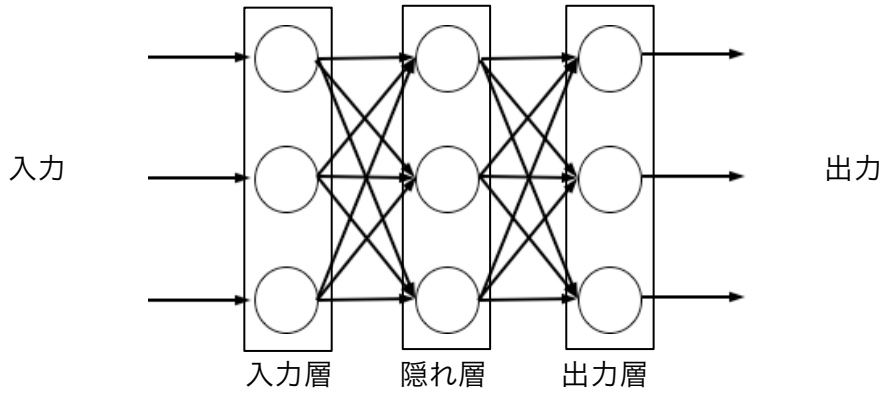


図 2.　ニューラルネットの模式図

図 2 のニューラルネットワークは入力値が入力層、隠れ層、出力層の方向にのみ伝達され、戻ることがないため、feedforward neural network と呼ばれる。また、隠れ層のニューロン全てが次の出力層のニューロンと結合しているため、隠れ層は全結合層である。

・ミニバッチ学習によるニューラルネットの学習

feedforward neural network は入力を受け取り、ある値を出力する。この出力はニューロンの重み、バイアスの値に依存する。これらのパラメータを変更することによってデータにフィッティングさせる事ができる。パラメータwは訓練データ集合を用いて、訓練データを入れた際の出力値と目標値の差が小さくなるように調整する。実際は、出力値と目標値のズレを数値化する誤差関数$E(w)$を最小化するようにパラメータを変化させる。しかし、変化させるべき対象のパラメータは膨大なため、解析的に解くことは不可能である。そこで、勾配降下法を用いて近似解を求める。パラメータの更新式は以下のようになる。

$$w^{(t+1)} = w^{(t)} + \Delta w^{(t)}, \quad \Delta w^{(t)} = -\eta \nabla E\left(w^{(t)}\right)$$

ここで、$w^{(t)}$は時刻tでのパラメータの値である。ηは学習率と呼ばれ、1 ステップでの移動距離を決定する。ニューラルネットワークの学習では通常、誤差逆伝播法を用いて高速に勾配を計算する。

勾配降下法はその性質より、誤差関数の局所解に陥る。最適解と大きく差がある場合、適切な予測を行うことは出来ない。そのため、局所解に陥ることを出来るだけ避けるため、訓練データの部分集合のみを用いた誤差関数に基いてパラメータを更新する。これをミニバッチ学習とよび、局所解に陥る可能性を下げることができる。

1-2-2. 畳み込みニューラルネットワークの原理

畳み込みニューラルネットワーク(convolutional neural network, CNN)は動物の視覚神経系をモチーフにしている。動物がモノを見たとき、視覚情報は網膜で受け取られ、視覚野に入力される。Hubel と Wiesel は特定の方位、または位置に対して選択的に応じるニューロンが存在していることを発見した[8]。そのような細胞には単純型細胞、複雑型細胞という 2 種類の細胞がある。単純型細胞は位置選択性が厳密であり、複雑型細胞はそうではない。この振る舞いは、複雑型細胞が単純型細胞を束ねることで実現していると考えられている。つまり、束ねられた単純型細胞のどれかの発火シグナルを受け取るだけで複雑型細胞が発火することで複雑型細胞のロバスト性が確保されていると考えられている。

この仮説を元に CNN は考え出された。CNN は畳み込み層とプーリング層という 2 種類の層と交互に積み重ねた構造を持つ。畳み込み層では通常のニューラルネットワークと異なり、結合が疎になっており、重みを共有する。これは局所的な受容野を持つ単純型細胞を模倣している。プーリング層は畳み込み層の出力を取りまとめる役割を持ち、複雑型細胞を模倣する。これらの手法により、通常のニューラルネットワークよりパラメータ数を削減することに成功している。

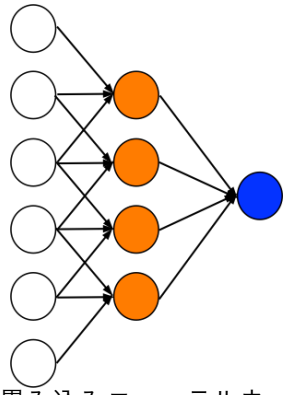


図 3.　畳み込みニューラルネットの模式図

CNN は特に画像処理で大きな成果を挙げている。2012 年一般物体認識のコンテスト ILSVRC で CNN を用いた Krizhevsky らのグループが物体認識と物体検出の 2 部門で圧勝したことが広く知られている[9]。それ以降、CNN の研究は大きく広がり、画像認識の標準的な方法として知られるようになった。また、ディープラーニング技術を一般に広める要因になった。

バイオインフォマティクスにおいても広く用いられており、二次構造予測[10-11]、DNA 結合タンパク予測[12-13]、バイオ画像データの分析[14-15]、など様々な問題で既存の手法を超える成果を挙げている。

2．手法

2-1. データセットの作成

ニューラルネットワークで学習させるためのマルチドメインタンパク質のデータセットを作成した。

2-1-1. ドメインデータの取得

本研究では先行研究に従い、構造ドメインを「マルチドメインタンパク質中に存在するドメインのうち、他ドメインとの相互作用が少なく、切り離しても単独でフォールドが可能なもの」と定義した。

ドメインデータセットを作成するため、既に公開されている以下 2 つのデータベースからタンパク質を取得した。ドメイン境界はそれぞれのデータベースで用いられている定義を使用した。

SCOPe[16](http://scop.berkeley.edu/)はタンパク構造分類のデータベースで、プログラムと人間の目でクラスわけを行っている。タンパク質は高位から Class、Fold, Superfamily, Family の 4 階層で分類されている。今回用いたバージョンは SCOPe 2.06 である。今回、12 の Class 中 a: All alpha proteins, b: All beta proteins, c: Alpha and beta proteins (a/b), d: Alpha and beta proteins (a+b), e: Multi-domain proteins (alpha and beta), f: Membrane and cell surface proteins and peptides, g: Small proteins, h: Coiled coil proteins のタンパク質を抽出した。使用しなかった Class は i: Low resolution protein structures, j: Peptides, k: Designed proteins, i: Artifacts の 4 クラスである。この 4 クラスは構造ドメインとなり得ないか、予測のためのデータセットに適さないと考えたため、取り除いた。結果、32029 個のタンパク質を取得した。

CATH[17]も同じくタンパク質構造分類のデータベースであり、プログラムと人の目によってクラス分けされている。今回、バージョン 4.1.0 を用いた。実際に利用したファイルは”CATH-domain-boundaries-v4_1_0.txt”である。
このファイルでは”segment”・”domain”・”fragment”という 3 つの定義を用いてタンパク質であるアミノ酸シーケンスの領域を分割している。”segment”はアミノ酸シーケンスが連続しているタンパク質ドメイン、”domain”は連続していない discontinuous domain を含んだドメイン、”fragment”はドメインではない領域を指している。我々が今回ドメインと定義しているものに discontinuous domain は含まないため、segment を 1 つのみ持つ domain のみを抽出した。結果、63054 個のタンパク質を取得した。

その後、それぞれのデータセットからマルチドメインタンパク質(2 ドメイン以上持つタンパク)を抽出し、２つのデータセットをマージした。SCOPe データセットと CATH データセットでタンパク質が重複した場合、SCOPe データセットの定義を優先した。マージした結果、タンパク質の数は 71085 個となった。FASTA ファイルがないタンパク質を取り除き、最終的に 69680 個となった。これを統合データセットと呼ぶ。

2-1-2. 代表配列選出

統合データセットは重複した配列が多く含まれている。そのため、これをそのまま機械学習のデータセットとするとデータの情報に偏りが生じるおそれがある。そのため、クラスタリングアルゴリズムである blastclust[18]を用いて identity 30%、coverage 80%にてクラスタリングを行い、クラスタリングされたタンパク質から 1 つを抽出し、代表配列とした。クラスタからの代表選出のアルゴリズムを以下に示す。まず、クラスタ内タンパク一つ一つをクラスタ内タンパクに対して blast をかけた。次にクラスタ内のタンパク質間で両者から見て identity が 30%以上のものを類似しているとした。最後に類似しているタンパクが最も多いものを代表とした。類似数が同じタンパク質がある場合、アミノ酸配列が最も長いものを選択した。クラスタ内タンパク質が 2 つのものも同じくアミノ酸配列が長いものを選択した。

2-1-3. 目視によるデータセットの作成

SCOPe・CATH によるドメイン定義は本研究での定義と異なっている。そのため、統合データセット中に今回の学習データとして適さないタンパク質を含んでいる。そのため、目視によってタンパク質を分類した。まず、統合データセットに対して代表配列選出を行った結果 4465 タンパク質となった。その後、目視の対象とするタンパク質は 3 ドメイン以下とし、4 ドメイン以上を省いた。結果、代表配列数は 4132 となった。このデータセットを Class 1、Class 2、Class 3 にクラス分けした。本研究の定義に完全に当てはまると考えられるタンパク質は Class 1、ある程度当てはまると考えられるタンパク質は Class 2、当てはまらないと考えられるタンパク質は Class 3 とした。Class 1 及び Class 2 のタンパク質セットをマージし、Dataset 2 とした。Dataset 2 のタンパク質数は 2430 となった。

2-1-4. ドメイン間相互作用を元にした構造ドメインの決定

構造ドメインを定義しているデータベースである IS-Dom[19]では、マルチドメインタンパクのドメイン間相互作用によって構造ドメインを判定している。この手法によって同定された構造ドメインは Independent Structual Domain(ISD)と呼ばれる。ISD の判定は水素結合数、疎水性クラスタ数によって行われる。

2-1-5. 水素結合

水素結合の検出は HBPLUS[20]を用いて行った。これは、タンパク質分子内の主鎖間(MC-MC)、主鎖-側鎖間(MC-SC)、側鎖間(SC-SC)の水素結合の有無を距離に基いて検出する。これを用いてタンパク質残基間の全水素結合を検出した。

2-1-6. 疎水性クラスタ

疎水性クラスタ(HPC)は独自に定義した概念であり、疎水性アミノ酸 Phe, Tyr, Trp, Met, Ala, Val, Ile, Leu の 8 種類の残基の側鎖の C-β以降の炭素原子 3 つによって形成される。3 つの原子中、任意の 2 つの原子の距離が 5Å 以内にある場合、1 つの疎水性クラスタを形成しているとする。

2-1-7. 二次構造

二次構造情報は DSSP[21]によって決定した。DSSP は水素結合から 8 種類の二次構造を残基に割り当てる。DSSP の出力ファイルの STRUCTURE 列を参照し、8 種類の内、G、H、I はヘリックス、E、B はストランド、その他はコイルとした。

2-1-8. ISD による判定条件

IS-Dom のデフォルトである(MC-MC 11、MC-SC 9、 SC-SC 7、Distance 5.0Å、HPC 7)としたマルチドメインタンパク中のドメイン全てが ISD の場合のみ ISD マルチドメインとした。判定の例を次に挙げる。例えば 2 つのドメインを持つマルチドメインタンパク質を考え、SCOPe、または CATH で 2 つのドメインが定義されているとする。この 2 つのドメインの「開始残基の直前」・「終了残基の直後」で切ったとき、失われる水素結合、疎水性クラスタの数が上記の条件を全て下回った場合に ISD マルチドメインとする。

2-1-9. ISD によるデータセットの作成

目視によるクラス分けは極めて主観的であるため、先行研究で用いられていた、相互作用による構造ドメインの選出とデータセットの作成も行う。統合データセットから ISD タンパク質のみ抽出し、29755 タンパク質となった。その後、代表配列を選出して、2516 タンパク質となった。これを Dataset 1 と呼ぶ。

2-1-10. ISD に基づいたリンカーの拡張

SCOPe、CATH ではドメインのみが定義されており、マルチドメインタンパク質の場合、N 末端側のドメインの終了残基、C 末端側のドメインの開始残基が得られる。この 2 つの残基をリンカー残基とした。

実際のリンカーは 2 残基以上ある場合が考えられるので、拡張する必要がある。そのため、ISD に基いてリンカー残基を拡張する。以下の図は拡張のイメージである。四角が 1 残基を表し、D,L はそれぞれドメイン残基、リンカー残基を表している。

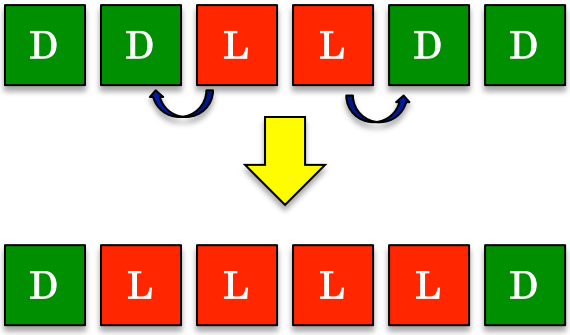


図 4. リンカー拡張の模式図

拡張の際の手順及び条件を示す。

・拡張の際は N 末方向、C 末方向にそれぞれ独立に拡張する。一方向のみに拡張されることもあり得る。

・拡張の際は 1 残基ずつ伸ばしていく。伸ばす回数の最大値は 10 とする。

・拡張候補の残基の二次構造を確認し、コイル以外なら拡張を止める。

・候補の残基と、さらに一つ前の残基との間で切断した場合、ISD の判定の条件を全て下回った場合、候補の残基をリンカー残基とする。判定をクリアできない場合、拡張を止める。

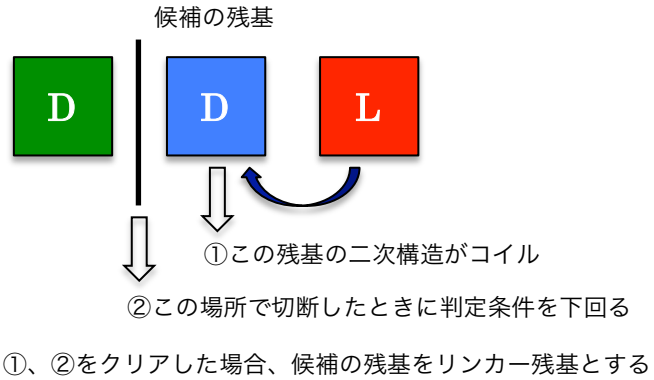
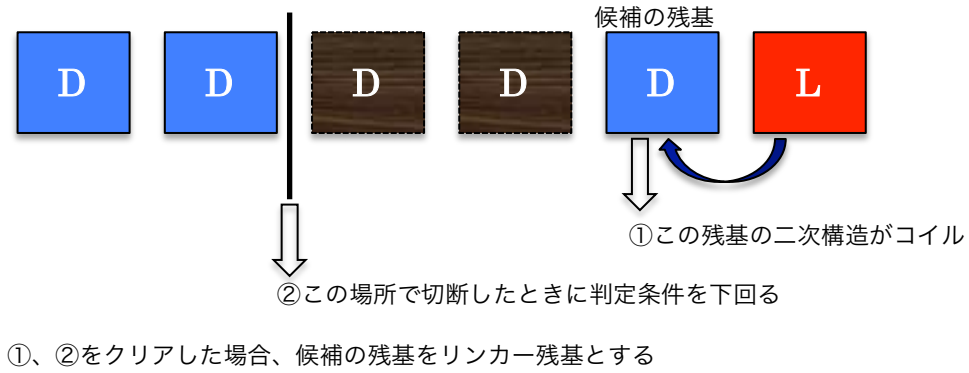


図 5. リンカー拡張の判定条件

・候補残基、および候補残基の一つ先の残基が disorder の場合、disorder でない残基が現れるまで先の残基を探索し、始めに現れた disorder でない残基を判定残基とする。判定残基の二次構造、および判定残基のひとつ先の残基と判定残基の間の ISD 判定を行い、クリアしたら候補残基をリンカー残基とする。

・候補残基のひとつ先の残基が disorder の場合、disorder でない残基が現れるまで先の残基を探索し、始めに現れた disorder でない残基の一つ手前で切断し、判定する。



・判定残基のひとつ先の残基が disorder の場合、disorder でない残基が現れるまで先の残基を探索し、始めに現れた disorder でない残基の一つ手前で切断し、判定する。

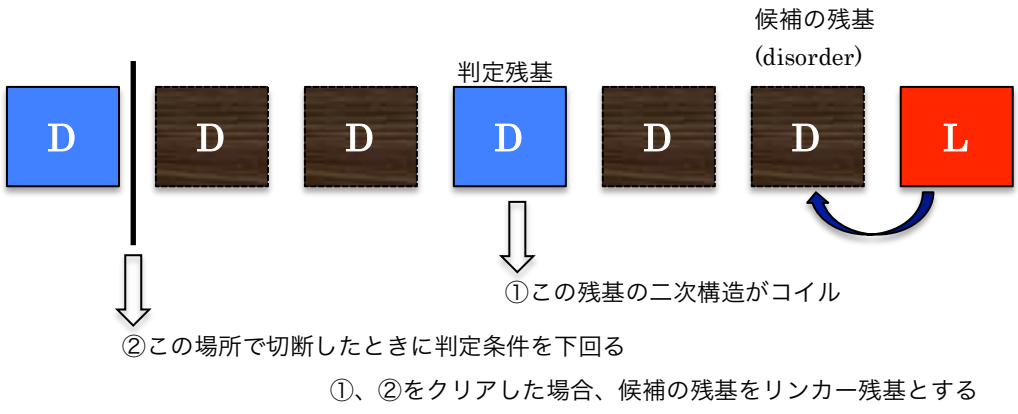


図 6. disorder の際のリンカー拡張の判定条件

2-1-11. 残基番号の変換

SCOPe および CATH のドメイン境界は PDB に準拠している。今回、予測に使う入力データは FASTA 形式のアミノ酸配列であるため、アミノ酸配列に基づく残基番号に変換する。変換のために Biopython[22]を用いてタンパク質を PDB ファイルから一次構造に変換した。その後、clustalw2[23]を用いて FASTA ファイルとアラインメントを行い、PDB ファイルの残基番号をアミノ酸配列に基づいた番号に変換した。また、ドメイン境界データの残基番号も変換した。

2-1-12. データセットの分割

機械学習を行い、その性能をテストするため、Dataset2 の 2430 個中、1830 個をトレーニングデータセット、300 個をバリデーションセット、300 個をテストセットとした。

2-1-13. トレーニングデータセットの拡張

今回作成したデータセットは小さく、機械学習においてデータが不足する可能性がある。そのためトレーニングデータセットの拡張を行った。1830 個のトレーニングデータセットのタンパク質を含む 30%クラスタ内で 70%クラスタリングを行う。その 70%クラスタから 1 つ代表を選出した。しかし、70%クラスタに 30%クラスタの代表が含まれている場合、それを代表とした。これによって新たに得られた 1108 個のタンパク質はトレーニングデータセットに加え、計 2938 個となった。

2-2. 入力データの作成

2-2-1. 特徴量ベクトルの作成

入力はアミノ酸配列である。タンパク質は様々な長さのアミノ酸配列からなる一方、ニューラルネットワークの入力は固定長である必要があるため、アミノ酸配列の一部分を抽出して入力とする。切り取る部位を window と呼ぶ。window の中心に位置する残基について、リンカーの確率を予測する。

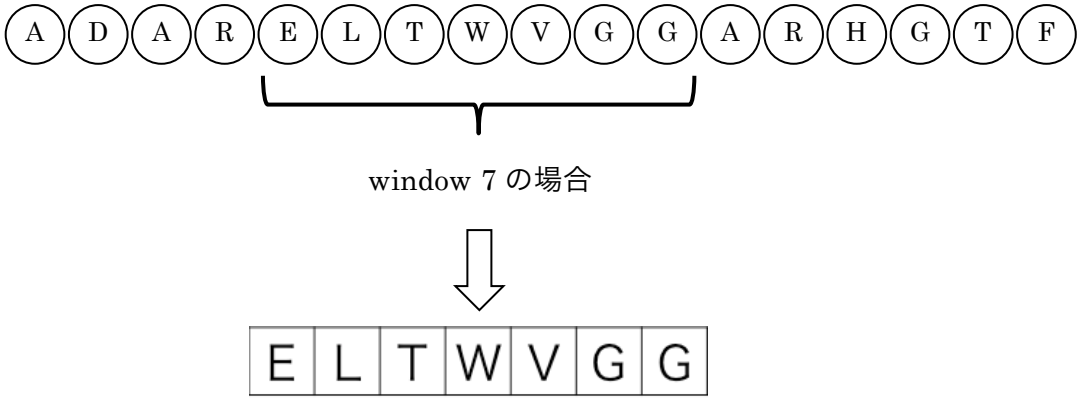


図 7. window によるアミノ酸配列の切り取り

次に、残基レベルでの特徴量を作成する。以下の種類の特徴を使用した。

① アミノ酸の種類

one-hot encoding の長さ 21 のベクトルを用いる。20 種類のアミノ酸に対応する位置が 1 となっており、他は全て 0 である。21 番目は X を表す。

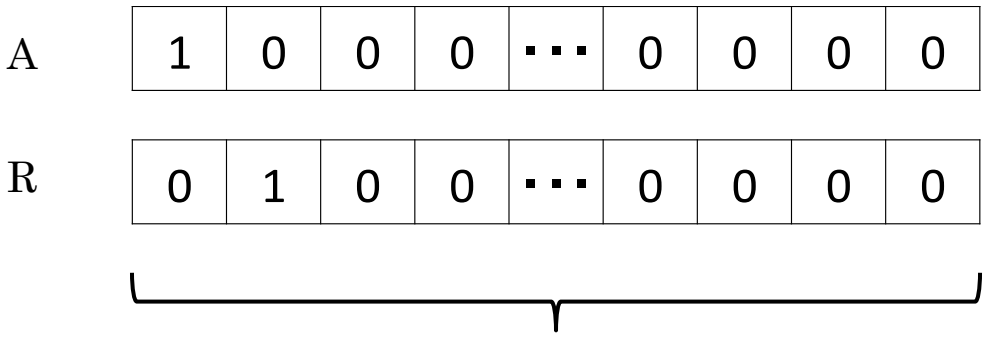


図 8. アミノ酸配列の one-hot encoding

② アミノ酸 PSSM

ベクトル長は 20 となる。PSSM は PSI-BLAST[18]によって計算した。

得た PSSM はおよそ-15 ～15 までの整数値をとる。この値をニューラルネットへ直接入力することは望ましくないため、次のように標準化を行った。

$$PSSM_{std} = \frac{PSSM - PSSM_{mean}}{PSSM_{\mathrm{sd}}}$$

$PSSM_{mean}$, $PSSM_{\mathrm{sd}}$はそれぞれ 1 タンパク質内の全ての PSSM 集合の平均、標準偏差である。

③ 二次構造

予測するリンカー部位はほとんどコイルであるため、二次構造は非常に重要な情報となる。

DSSP、spider2[24]を用いて三状態(ヘリックス・シート・コイル)二次構造情報を得た。

DSSP はタンパク質立体構造を入力として用いるので、二次構造を決定することが出来る。そのため one-hot encoding の長さ 3 のベクトルとして二次構造を表した。DSSP を用いることが出来るのは構造情報が既に得ている場合のみであるため、この情報はトレーニングの際のみ用いた。

予測の際には spider2 によって得られた三状態二次構造予測情報を用いた。ベクトル長は同じく 3 で、三状態の確率を表す。

④ 露出溶媒表面積(ASA)

spider2 により予測した ASA の値を用いた。以下の式で標準化した。

$$ASA_{std} = \frac{ASA - ASA_{\min}}{ASA_{\max} - ASA_{min}}$$

$SA_{\max}$ 、ASA_{min} は、1 タンパク質内の ASA の最大値、最小値である。

⑤ マルチプルアラインメント(MSA)による情報

・共進化

アミノ酸残基はタンパク質内で相互作用しており、一方の残基が変異すると、相互作用している他方の残基に影響を及ぼし、変異することがある。これを共進化とよび、タンパク質間の相互作用予測に広く用いられている。タンパク質間相互作用はリンカー予測に重要な情報をもたらすと考えられるため、特徴量として採用した。

共進化の検出に、残基間の相互情報量がよく用いられる[25]。相互情報量は、情報理論において、2 つの確率変数の相互依存の尺度を表す量であり、以下の式で表される。

$$MI_{x,y} = H_x + H_y - H_{x,y}$$

H_xはアラインメントされた配列のある列 x のエントロピーであり、以下の式で表される。

$$H_x = -\sum_{i=1}^{20} p(x_i) \log_{20} p(x_i)$$

$p(x_i)$はあるアミノ酸 i が、その列において現れる確率である。

$H_{x,y}$はアラインメントされた配列の任意の 2 列 x, y の結合エントロピーであり、以下の式で表される。

$$H_{x,y} = -\sum_{i=1}^{20}\sum_{j=1}^{20} p(x_i, y_j) \log_{20} p(x_i, y_j)$$

$p(x_i, y_j)$は任意の 2 列 x, y に i, j のアミノ酸が現れる確率である。

HHblits[26]を用いて MSA を行い、相互情報量マトリックスを作成した。ただし、相互情報量がある閾値以下の場合は 0 とした。その後、以下の式で特徴量 MI loss, Interaction を算出した。

$$MI\ loss_x = \frac{1}{x(L-x)} \sum_{i=0}^{x} \sum_{j=x+1}^{L} MI_{i,j}$$

$MI_{i,j}$は i 番目と j 番目の残基の相互情報量、タンパク質の L はアミノ酸長である。これは x 番目のアミノ酸配列残基 で切断した時に失われる相互情報量の平均を表している。

$$Interaction_x = \sum_{i=1}^{x-1} MI_{x,i} + \sum_{i=x+1}^{L} MI_{x,i}$$

これは x 番目のアミノ酸残基と他の全てのアミノ酸残基との相互情報量の合計を表している。

MI loss、 Interaction, エントロピーH_x の 3 つを特徴量として用いた。

以上の特徴量を連結し、残基の特徴量とした。その後、window 中の残基全ての特徴量を連結し、1 つの入力値とした。

2-2-2. ラベルの作成

ドメインは 0、リンカーは 1 とした。出力値がリンカーの確率となるように意図した。

2-2-3. ミニバッチの作成

ドメインリンカー部位はアミノ酸配列の一部分であるため、ドメイン部位のデータ量とリンカー部位のデータ量は不均衡である。これらを自然の比率で入力すると、ニューラルネットのトレーニングが成功しない可能性が高い。具体的には、全てドメインと判定するように学習してしまうと考えられる。これを避けるため、データセットをドメイン部位とリンカー部位に分け、ドメインデータセット、リンカーデータセットを作製した後、それぞれから等量ずつ取り出し、結合してミニバッチとした。ミニバッチのサイズは 64 とした。

2-3. ニューラルネットワークのモデル

始めに以下のモデルを使用した。単純な CNN である。実装は全て Tensorflow[27]で行った。

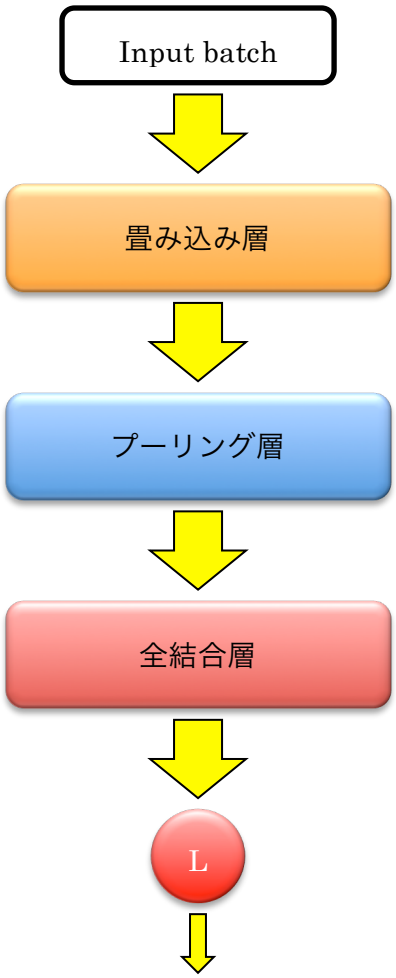


図 9. CNN のアーキテクチャの模式図

それぞれの層について説明する。

・input batch

入力するミニバッチである。3 次元配列となっていて、高次元から、バッチ, window, 特徴量ベクトルである。

・畳み込み層

入力を以下の式で畳み込む。

$$u_{im} = \sum_{k=0}^{K-1} \sum_{p=0}^{L-1} x_{i+p,k} h_{pkm} + b_{km}$$

ここで、x は入力値、h はフィルタ、b はバイアスである。K はチャネルであり、特徴量ベクトルの長さに一致する。L はフィルタ長、m はフィルタ数である。　最終的な出力は、この畳み込みの結果に活性化関数を作用させたものとなる

$$z_{im} = f(u_{im})$$

活性化関数は、以下に示す ReLU を使用した。

$$f(\mathrm{x}) = \max(x, 0)$$

プーリング層では max pooling を用いた。これによってリンカー部位の位置普遍性を確保することを試みた。

$$u_{ik} = \max_{\mathrm{p} \in \mathrm{P_i}} (z_{pk})$$

次に、以下に示す疎結合したニューラルネットワークを使用した。特徴量に加え、window の位置ごとに結合するニューロンを変えた。

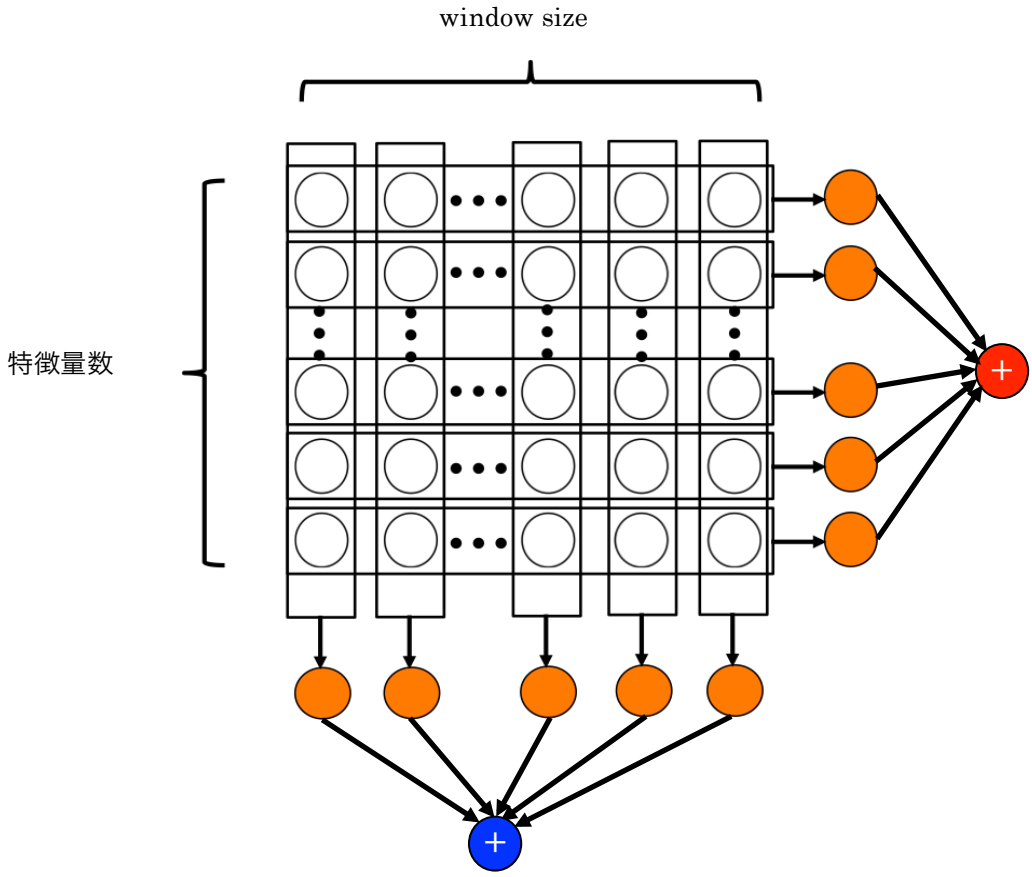


図 10. 疎結合ニューラルネットワークのアーキテクチャの模式図

オレンジ色のニューロンは四角で囲った部分とのみ結合する。赤・青のニューロンはオレンジ色のニューロンの数値の和である。赤・青のニューロンの数は同じであり、便宜上フィルタ数と呼ぶ。赤・青のニューロンは全結合層に入力される。式は畳み込みの場合と同じである。

$$u_m = \sum_{k=0}^{K-1} \sum_{p=0}^{L-1} x_{p,k} h_{pkm} + b_{km}$$

ここで、x は入力値、h はフィルタ、b はバイアスである。K はチャネルであり、赤の場合、特徴量ベクトルの長さに、青の場合 window size に一致する。L はフィルタ長であり、赤の場合、window size に、青の場合特徴量ベクトルの長さに一致する。m はフィルタ数である。

次にトレーニングの手法について説明する。これらは 2 つのアーキテクチャで共通である。

誤差関数は以下の式を使用した。

$$E(\boldsymbol{w}) = -\sum_{n=1}^{N} \mathrm{w} * \mathrm{y_n} \log P(L|\boldsymbol{x}_n) + (1 - y_n) \log(1 - \log\big(P(L|\boldsymbol{x}_n)\big)$$

$P(L|\boldsymbol{x}_n)$はリンカーの確率を表している。2 値分類であるため、$P(D|\boldsymbol{x}_n)$をドメインの確率とすると、$P(D|\boldsymbol{x}_n) + (L|\boldsymbol{x}_n) = 1$ である。$\boldsymbol{w}$ は重みのベクトルである。
w は正例に対する重みである。実際のタンパク質はリンカーに対してドメインの残基がはるかに多いため、重みが 1 であるとドメインをリンカーであると予測してしまうことが多くなってしまう。そのため、リンカーへの重みを軽くすることでこれを防ぐ。重みを軽くすることによって Sensitivity を下げ、Precision を上げることができる。

全結合層では dropout[28] を使用した。dropout は 1 回のバッチによる学習ごとにランダムに選択した一部の全結合層の出力をマスクし、部分ネットワークで学習する手法である。アンサンブル学習と似た効果が得られる。

正則化に L2 loss を用いた。L2 loss はパラメータの自由度を制限する手法で、誤差関数に正則化項を設ける。

$$E_{wd}(\boldsymbol{w}) = E(w) + \lambda \boldsymbol{w}^{\mathrm{T}} \boldsymbol{w}$$

λ は正則化の大きさを決めるパラメータである。

損失関数を最小化するための勾配降下法のアルゴリズムとして Adam [29]を用いた。

2-4. 予測結果の評価

2-4-1. 残基レベルでのリンカー部位の予測と評価

モデルの残基レベルでの評価を行うため、10- fold cross validation によって判定を行った。これはトレーニングデータを 10 分割し、9 つをトレーニングに用いて、残りの 1 つをテストに用いる手法である。予測結果は TP(True Positive)、FP(False Positive)、TN(True Negative)、FN(False Negative)に分類される。

表 1. 予測結果のクラス分け

		予測された部位の答え	
		リンカー部位	ドメイン部位
予測結果	リンカーと予測	TP	FP
	ドメインと予測	FN	TN

これを元に Precision、Sensitivity、F score を算出する。式は以下のようになる。

$$Precision = \frac{TP}{TP + FP}$$

$$Sensitivity = \frac{TP}{TP + FN}$$

$$F\ score = \frac{2 \cdot Precision \cdot Sensitivity}{Precision + Sensitivity}$$

Precision はリンカーと予測した残基中の正答の割合である。Sensitivity は全リンカー残基中、予測できた割合である。F score は 2 つの調和平均をとっている。Precision と Sensitivity はトレードオフの関係にあるため、F score をみてモデルを比較する。

2-4-2. タンパク質レベルでのリンカー部位の予測

今回のニューラルネットは、ある残基についてリンカーの確率を出力する予測器であるが、実際に必要な情報はあるタンパク質についてリンカーの有無および位置である。タンパク質レベルの予測は、次のように行った。はじめにタンパク質の全ての残基をニューラルネットで予測する。これによって全残基のリンカー確率を得ることが出来る。次に、そのデータに対し畳み込みを行った。これはリンカーの位置情報を予測に用いるためである。リンカーの前後の残基はリンカーである可能性が高いため、畳み込みを行うことでより精度を上げることが出来ると考えられる。最後に、リンカー部位を決定する。決定方法は 2 種類用いた。最大値の残基をリンカーであると判定する方法と、あらかじめ決めておいた閾値を超えた残基はリンカー部位とする方法である。前者は全てのタンパク質が必ず 1 つリンカーを持っていることを前提とする簡易な方法である。後者は、シングルドメインや 3 つ以上のドメインをもつマルチドメインタンパク質を正しく予測するために一般化した方法である。

2-4-3. タンパク質レベルでの正解の判定

モデル選択のために正答率を比較する必要があるが、この正答率は全てタンパク質レベルで算出した。トレーニングデータセットを用いて判定し、10- fold cross validation を行った。トレーニングデータは全てポジティブデータ(リンカーを 1 つ以上持つタンパク質)であるため、最大値をリンカーとする方法を用いた。タンパク質 1 つにつきリンカー残基を 1 つ予測し、実際のリンカー部位中の残基であった場合、正解とする。予測残基がリンカー部位中の残基でなかった場合、予測残基とリンカー部位の残基との最短距離を計算する。この値を margin と呼ぶ。リンカー部位中の残基を当てた場合、margin は 0 である。判定に用いたタンパク質のうち、margin が X 以下であったタンパク質の割合を margin X 正答率と定義する。X は 0 以上の整数である。

2-4-4. バリデーションセットを用いた評価と閾値の調整

性能を見積もるバリデーションセットでのテストは閾値を用いる判定方法を用いた。ネガティブデータにも対応する必要があるためである。予測結果は 4 つに分類される。

表 2. タンパク質レベルでの予測結果のクラス分け

		予測された部位の答え	
		リンカー部位	ドメイン部位
予測結果	リンカーと予測	TP	FP
	シングルドメインと予測	FN	TN

マルチドメインタンパク質に対して予測を行った時、閾値を超え、マルチドメインであると判定した上で、リンカー部位を正答出来た場合は TP にカウントする。

マルチドメインタンパク質に対して予測を行った時、閾値を超え、マルチドメインであると判定した上で、リンカー部位を誤答した場合は FP にカウントする。

マルチドメインタンパク質に対して予測を行った時、閾値を超えず、シングルドメインであると判定した場合は FN にカウントする。

ネガティブデータに対しては、以下のように評価する。

シングルドメインタンパク質に対して予測を行った時、閾値を超えず、シングルドメインと判定した場合は TN にカウントする。

シングルドメインタンパク質に対して予測を行った時、閾値を超えて、マルチドメインと判定した場合、FP にカウントする。

残基レベルでの評価と同様に、性能の比較のために Precision、Sensitivity、F score を用いた。

ニューラルネットワークの出力値は残基レベルであるため、タンパク質全ての残基を予測した後、フィルターで出力値を畳み込み、最大値が threshold 以上であればマルチドメインとして、リンカー部位を予測する。この threshold とフィルターの値を最適化する。

バリデーションデータである 300 個のマルチドメインと、ネガティブデータとしてバリデーション用の 300 個のシングルドメインタンパク質のデータを用いて、threshold とフィルターの最適化を行った。

2-4-5. テストデータによるテストと他の予測器との比較

同じドメイン境界部位を予測するソフトウェアを使い、テストデータ(マルチドメイン 300 個、シングルドメイン 300 個)を予測した。それら結果と今回開発した基本モデルを比較した。

比較に用いたソフトウェアは Domcut、DomPro、Dobo、DROP である。

予測する際の統一したルールを以下のように決定した。閾値を調整した際のルールと異なっている。

1. シングルドメインかマルチドメインかを始めに判定する。マルチドメインをシングルドメインと判定した場合、FN にカウントする。
2. マルチドメインと判定した場合、予測するリンカー部位は 1 箇所のみとする。複数予測するソフトが多く、ソフトウェア・タンパクごとに適切な予測数を決定することが難しいためである。2 リンカー以上ある場合、予測できなかったリンカー数だけ FN にカウントする。
3. 予測部位は 1 残基とする。この予測残基がリンカー領域から前後 5 残基以内に入っていた場合、TP に、入っていなかった場合、FP および FN にカウントする。

Precision、Sensitivity 等の算出は先行研究に従い、ポジティブデータのみで行った。定義は 1-1 であるが、以下の式と同義である。

$$Precision = \frac{正答数}{マルチドメインと予測した数}$$

$$Sensitivity = \frac{正答数}{ポジティブデータ内のリンカーの総数}$$

この算出方法の場合、ランダムに当てた場合の評価も行うことが出来る。始めに、タンパク残基の 1 つをランダムに指定し、上述の 2,3 によって正誤を判定する。この場合、全てのタンパク質をマルチドメインであると考えることになる。乱数は一様分布と二項分布を用いた。

3. 結果と考察

3-1. ドメイン・リンカー残基の分析

データセットをドメイン残基とリンカー残基に分け、ドメインセット・リンカーセットの PSSM の平均をアミノ酸ごとに算出し、差をとることで比較した。
計算式は、あるアミノ酸における、リンカーの PSSM の平均 — ドメインの PSSM の平均 となる。

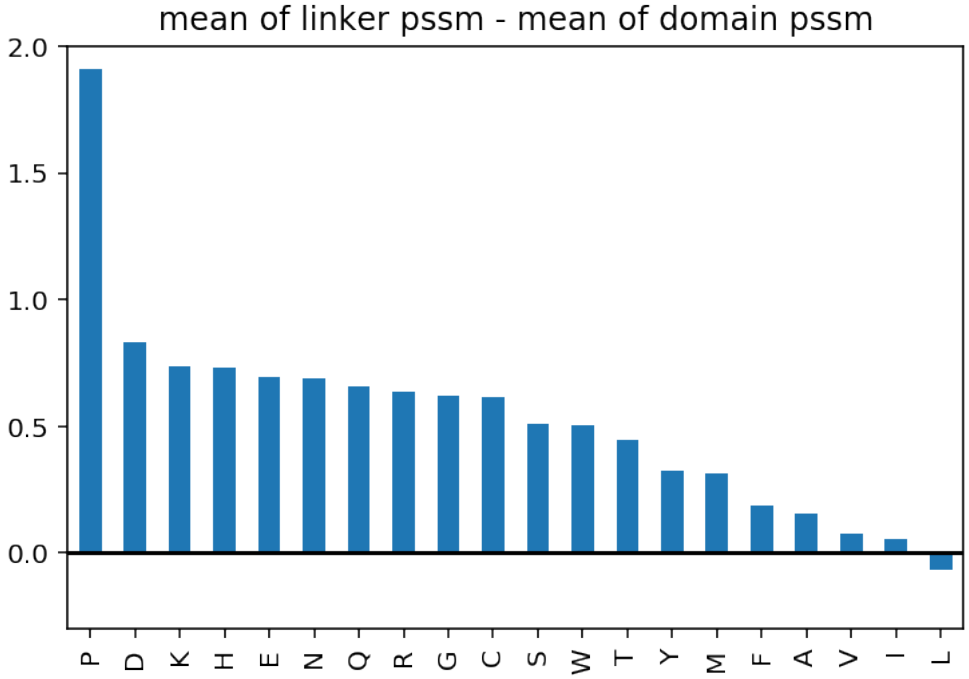


図 11. ドメイン/リンカー部位の PSSM 平均の差

プロリンが非常に高い値になっている。これはリンカーのプロリンの PSSM は平均的にドメインの PSSM より高く、プロリンになりやすいことを意味する。リンカー部位は機能や構造を持っていないことが多く、プロリンへの変異が受け入れやすいと考えられる。一方、ドメインの残基は構造を破壊する可能性が高いプロリンへは変異しにくいと考えられる。

次にコイル領域のみを取り出し、ドメイン・リンカーの差を比較した。

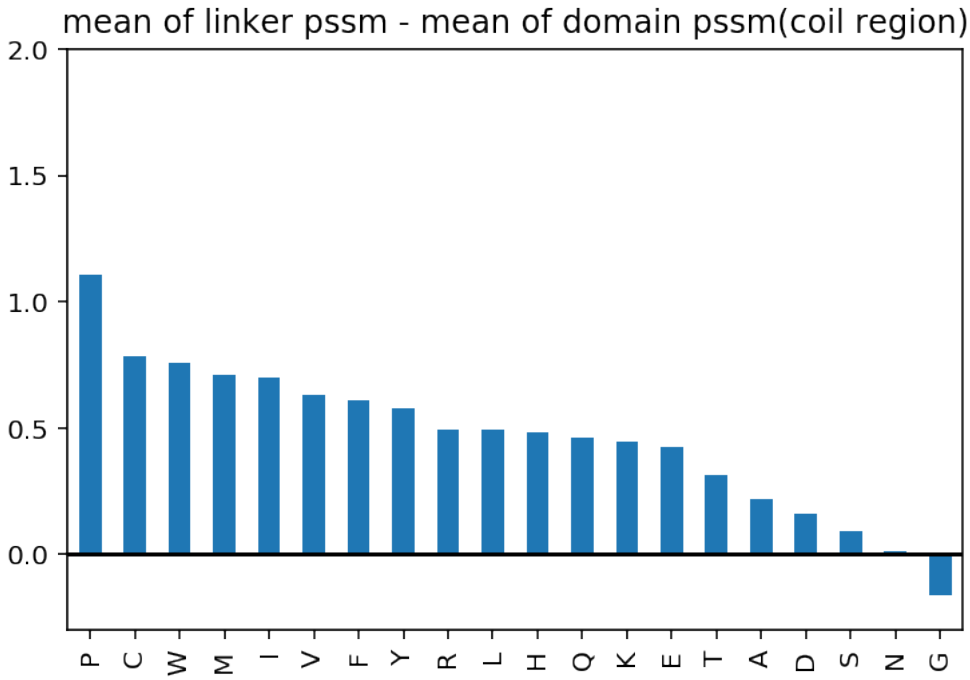


図 12. コイル領域における、ドメイン/リンカー部位の PSSM 平均の差

プロリンは図 11 と変わらず高い値を示している。一方、グリシンはコイル領域だけで考える場合は低い値を示している。また、アスパラギンはドメイン・リンカーの差はないことが分かる。その他のアミノ酸の PSSM は正の値をとっているため、ドメインと比較してリンカー部位は変異を受け入れやすい。これは、リンカー部位は機能を持っていないことが多いため、変異による進化圧を受けないためであると考えられる。

以上の分析より、PSSM はリンカーを予測する上で重要な情報であると示唆される。

3-2. 畳み込みニューラルネットワークでの予測

トレーニングデータセットのタンパク質を用いて畳み込みニューラルネットワークを学習させ、cross validation によって性能を見積もる。また、トレーニングデータ全体に対しての予測性能と比較することで、過学習の有無を判断する。

特徴量をアミノ酸・PSSM・二次構造・ASA のみを用いた場合(model 1)と、それに MSA・位置情報を加えた場合(model 2)の 2 つの条件で予測する。

3-2-1. model 1 による予測

特徴量としてアミノ酸・PSSM・二次構造・ASA のみを用いて CNN で予測をおこなった。

ハイパーパラメータは以下のものを用いた。最適なパラメータを見つけるために、全結合層のニューロン数、畳み込みフィルタ数、エポック数、L2 loss weight は複数試した。

表 3. model1 のハイパーパラメータ

バッチサイズ	64
学習率	0.001
全結合層のニューロン数	32, 64, 128
window size	5
特徴量ベクトルの長さ	45
畳み込みフィルタ数	10, 20, 30
畳み込みフィルタのストライド	1
プーリングウインドウ	2
プーリングのストライド	2
エポック数	10,20,30
dropout rate	0.5
L2 loss weight	0.005,0.01
誤差関数中の w(正値への重み)	0.4,0.5

以上の全ての組み合わせで予測を試し、クロスバリデーションテスト(CV)とトレーニングデータ(Train)を用いて正答率を算出した。その際クロスバリデーションテストよりトレーニングデータセットでの予測率が大きく上回っていた場合、過学習となっていると思われる。そのため、マージン 5 での正答率に 2%以上差がある場合、過学習と考え除外した。表　は全てのハイパーパラメータの組み合わせと正答率を示している。

表 4. ハイパーパラメータの組み合わせとマージン 3, 5 正答率

全結合相ニューロン数	畳み込みフィルター数	エポック数	L2 loss weight	w	margin 3 CV 正答率	margin 5 CV 正答率	margin 3 Train 正答率	margin 5 Train 正答率
32	30	30	0.005	0.4	0.28169	0.32102	0.29094	0.33128
32	30	30	0.01	0.4	0.27998	0.32033	0.28444	0.32410
128	30	30	0.005	0.4	0.27827	0.31691	0.28923	0.32718
32	20	30	0.005	0.4	0.27997	0.31690	0.28103	0.31863
32	30	30	0.01	0.5	0.27827	0.31656	0.28752	0.32821
32	30	30	0.005	0.5	0.27759	0.31589	0.28718	0.32923
32	30	20	0.005	0.5	0.27519	0.31520	0.27932	0.31829
32	20	30	0.01	0.4	0.27758	0.31417	0.28137	0.31761
32	20	30	0.005	0.5	0.27449	0.31383	0.28513	0.32308
32	20	30	0.01	0.5	0.27621	0.31315	0.27863	0.31692
128	30	30	0.005	0.5	0.27690	0.31280	0.28752	0.32718
64	20	30	0.005	0.5	0.27449	0.31211	0.28068	0.31795
128	30	30	0.01	0.5	0.27384	0.31179	0.28479	0.32342
32	20	20	0.01	0.4	0.27758	0.31108	0.28205	0.31692
32	10	30	0.005	0.4	0.27757	0.31106	0.28068	0.31316
32	20	20	0.005	0.5	0.27759	0.31075	0.28068	0.31487
64	30	30	0.01	0.4	0.27040	0.31075	0.28239	0.32034
32	20	20	0.005	0.4	0.27347	0.31074	0.28205	0.31658
32	20	20	0.01	0.5	0.27417	0.31041	0.28171	0.31692
128	30	30	0.01	0.4	0.27382	0.31040	0.28103	0.32000
32	10	30	0.005	0.5	0.27859	0.31038	0.27966	0.31350
32	30	20	0.005	0.4	0.27178	0.31008	0.27932	0.31692
32	10	30	0.01	0.4	0.27826	0.31004	0.28274	0.31521
32	30	20	0.01	0.5	0.26905	0.30973	0.28103	0.31897
64	10	30	0.005	0.5	0.27246	0.30971	0.27966	0.31282
64	10	30	0.005	0.4	0.27587	0.30939	0.28205	0.31214
128	20	30	0.005	0.4	0.27141	0.30937	0.27658	0.31282
128	20	30	0.005	0.5	0.27176	0.30904	0.27453	0.31179
64	30	30	0.005	0.4	0.27278	0.30903	0.28376	0.32000
64	20	20	0.005	0.5	0.27451	0.30835	0.28342	0.32000

全結合層ニューロン数 32, 畳み込みフィルタ数 30, エポック数 30 , L2 loss weight 0.005, w 0.4 の組み合わせが margin 5 CV 正答率 32.1%と最も高くなっている。そのため、このパラメータを採用した。

図 13 はマージン 0, 3, 5, 7, 10 での正答率のグラフである。図 14 は x 軸をマージン、y 軸を正答率としたグラフである。

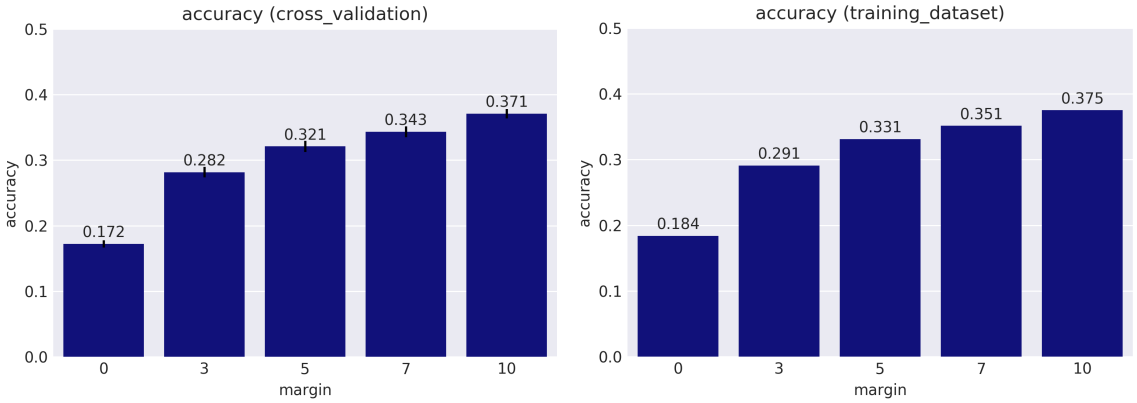


図 13. CV・Train に対するマージン正答率(model 1)

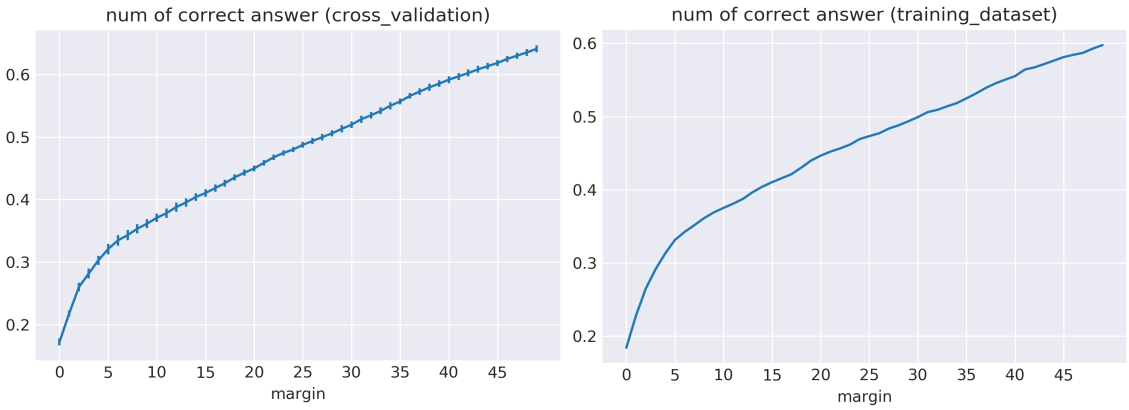


図 14. CV・Train に対するマージン・正答率のグラフ(model 1)

リンカーを当てることが出来ているのは 17.2%と低い。また図 14 より、0 — 5 付近で急激に上昇し、あとは線形に同じ傾きで上昇していることがわかる。この予測器が当てているのは margin 5 までであり、あとはランダムに当たっていると考えられる。よって、意味のある正答率は margin 5 の値までであると考えられる。CV での margin 5 正答率は 32.1%である。

図 15-19 は、クロスバリデーションテスト及び、トレーニングデータセットに対して予測を行ったとき、用いたパラメータごとの予測率を示したものである。予測率が高くなっているパラメータが良いパラメータであると考えることが出来る。

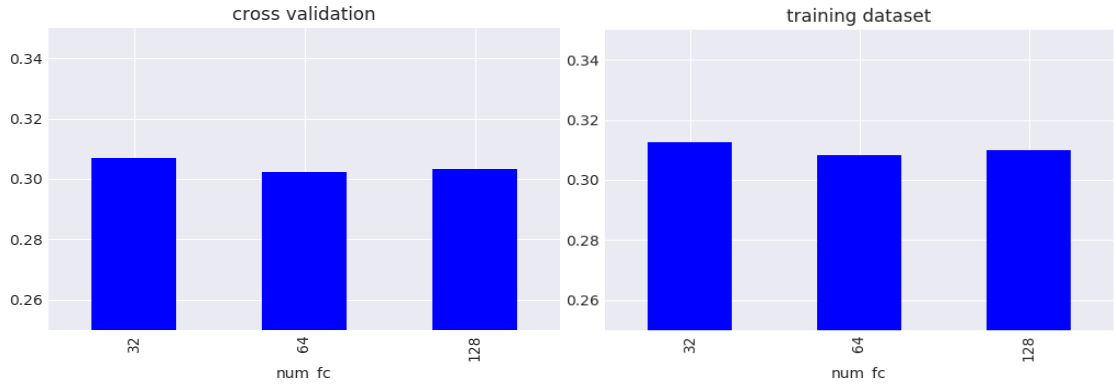


図 15. 全結合層ニューロン数による正答率の変化(model 1)

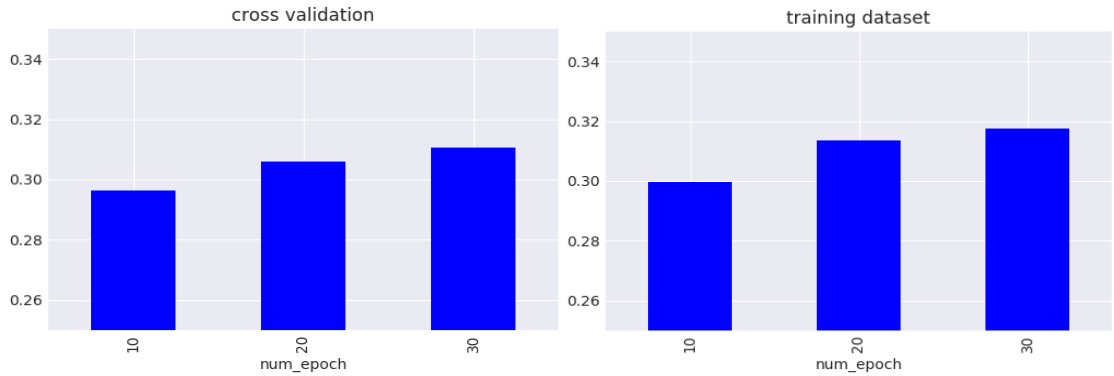


図 16. エポック数による正答率の変化 （ model 1)

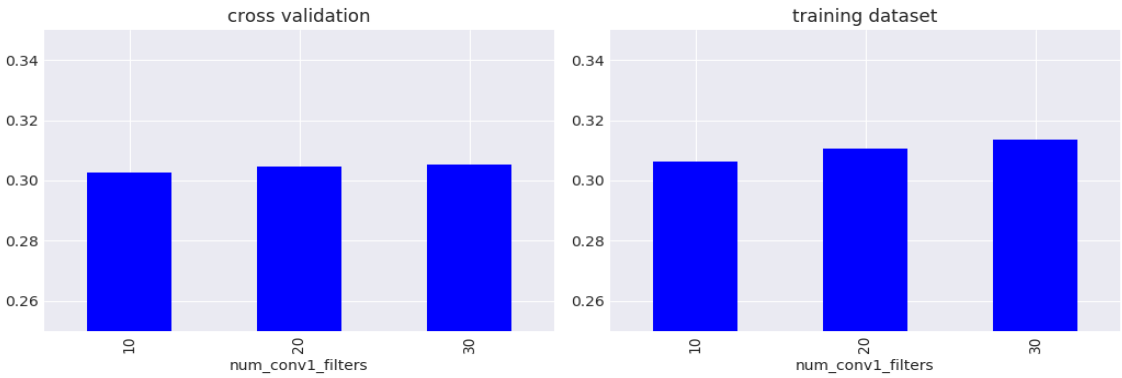


図 17. 畳み込みフィルタ数による正答率の変化 （ model 1)

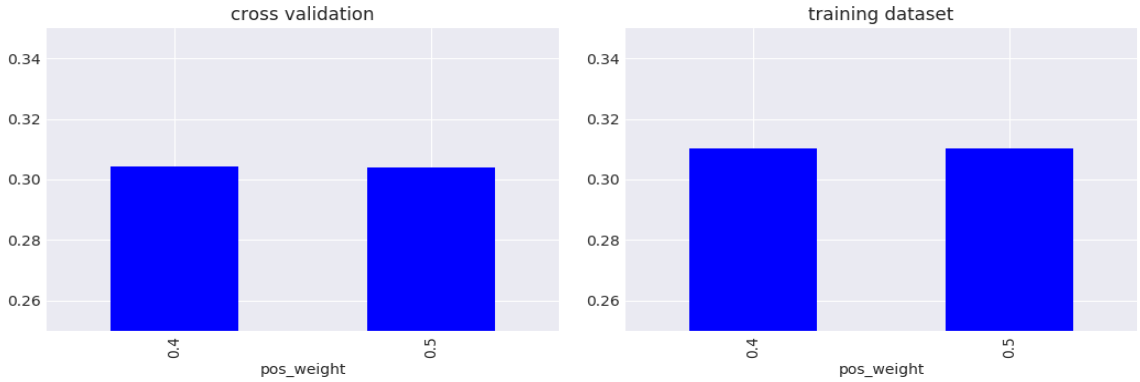


図 18. w による正答率の変化 （ model 1）

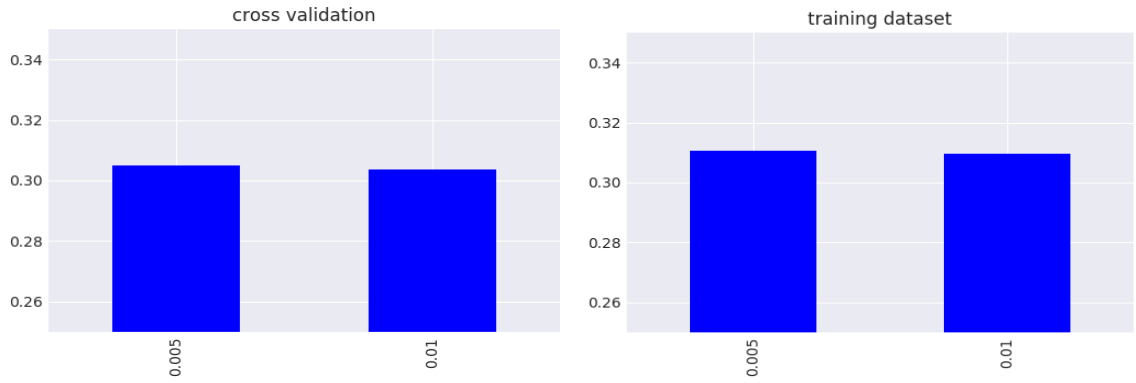


図 19. L2 loss の係数による正答率の変化 （ model 1）

全結合層ニューロン数は 32 が高くなっている。エポック数は 10 より 20,30 が高くなっているが、training dataset では更に高い値を示しており、多くしすぎることによる過学習の危険性もある。畳み込みフィルタ数も同様に多くすることによって正答率が上昇している。L2 loss weight と w は大きな違いが見られない。そのため、これら 2 種類のパラメータは次からの実験では 1 つに固定した。

3-2-2. model 2 による予測

MSA 情報と位置情報を新たに加え、学習させる。パラメータは以下の通りとした。model1 と同様に、マージン 5 での正答率に 2%以上差がある場合、過学習の可能性があるため除外した。表 5 にハイパーパラメータの組み合わせ、表 6 はマージン正答率である。

表 5. model2 のハイパーパラメータ

バッチサイズ	64
学習率	0.001
全結合層のニューロン数	32, 64
window size	11
特徴量ベクトルの長さ	49
畳み込みフィルタ数	10, 20, 30
畳み込みフィルタのサイズ	4, 7, 11
畳み込みフィルタのストライド	1
エポック数	10, 20, 30
dropout rate	0.5
L2 loss weight	0.01
誤差関数中の w(正値への重み)	0.4

表 6. ハイパーパラメータの組み合わせとマージン 3, 5 正答率

全結合相ニューロン数	畳み込みフィルター数	畳み込みフィルターサイズ	エポック数	margin 3 CV 正答率	margin 5 CV 正答率	margin 3 Train 正答率	margin 5 Train 正答率
64	30	11	20	0.3862	0.4228	0.4054	0.4372
64	20	11	20	0.3794	0.4183	0.3989	0.4355
32	30	11	20	0.3788	0.4129	0.3876	0.4215
32	20	11	20	0.3784	0.4099	0.3821	0.4150
64	30	4	20	0.3722	0.4043	0.3862	0.4181
32	10	7	20	0.3650	0.4019	0.3760	0.4071
64	10	7	20	0.3680	0.4019	0.3732	0.4075
64	10	11	20	0.3637	0.4006	0.3602	0.3972
32	10	11	20	0.3538	0.3941	0.3551	0.3962
32	30	4	20	0.3606	0.3934	0.3845	0.4129
64	20	11	10	0.3575	0.3917	0.3609	0.3999
64	30	7	10	0.3599	0.3879	0.3739	0.4027
64	20	4	20	0.3465	0.3841	0.3599	0.3900
32	10	4	30	0.3499	0.3831	0.3664	0.3969
32	20	4	20	0.3407	0.3766	0.3585	0.3914
32	20	11	10	0.3356	0.3739	0.3472	0.3825
64	10	11	10	0.3367	0.3736	0.3507	0.3845
32	10	11	10	0.3390	0.3722	0.3332	0.3640
32	30	11	10	0.3336	0.3688	0.3531	0.3832
64	10	4	20	0.3414	0.3688	0.3541	0.3767
32	20	7	10	0.3336	0.3660	0.3510	0.3849
32	10	7	10	0.3305	0.3653	0.3394	0.3671
64	10	7	10	0.3284	0.3640	0.3397	0.3729
64	10	4	10	0.3175	0.3448	0.3171	0.3445
64	20	4	10	0.3141	0.3431	0.3298	0.3623
32	20	4	10	0.3103	0.3394	0.3277	0.3572
32	10	4	10	0.3034	0.3294	0.3100	0.3401

全結合層ニューロン数 64, 畳み込みフィルタ数 30, フィルタサイズ 11, エポック数 20 の組み合わせが margin 5 CV 正答率 42.28%と最も高くなっている。そのため、このパラメータを採用した。

図 20 はマージン 0, 3, 5, 7, 10 での正答率のグラフである。図 21 は x 軸をマージン、y 軸を正答率としたグラフである。

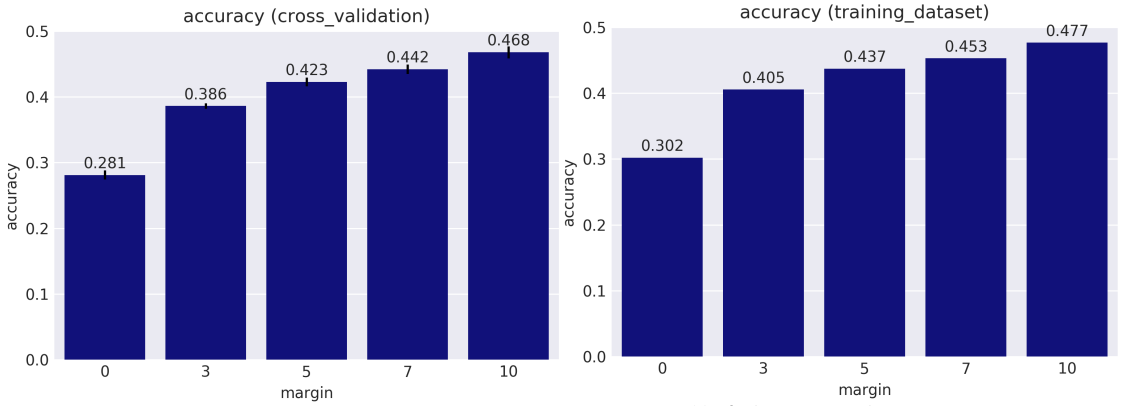


図 20. CV・Train に対するマージン正答率(model 2)

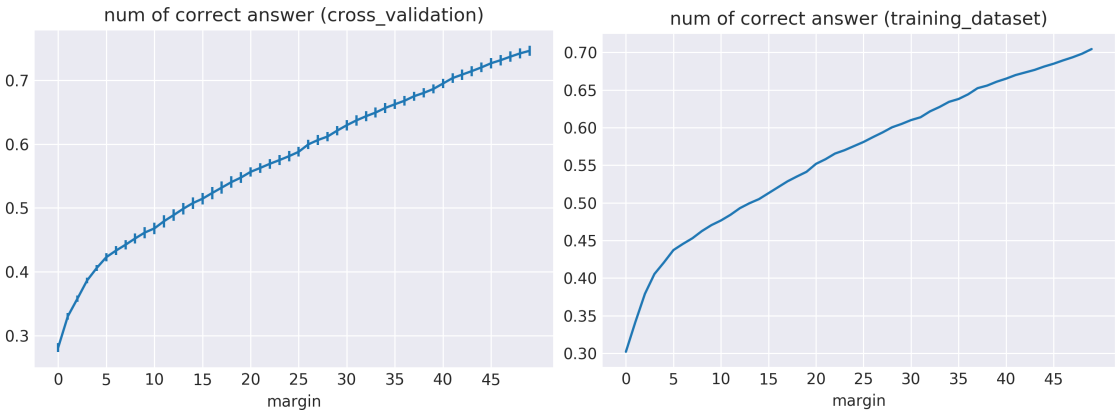


図 21. CV・Train に対するマージン・正答率のグラフ(model 2)

共進化情報と位置情報を入れることで、CV において Margin 5　正答率が 8%以上増加している。これらの特徴量がドメインリンカーを予測する上で重要であることがわかる。Training dataset との正答率の差も僅かであり、過学習が起こっていないであろうと推測される。

また図 21 より、0 — 5 付近で急激に上昇し、あとは線形に同じ傾きで上昇している。これは model 1 と同じ傾向が見られている。

図 22-25 は、クロスバリデーションテスト及び、トレーニングデータセットに対して予測を行ったとき、用いたパラメータごとの予測率を示したものである。

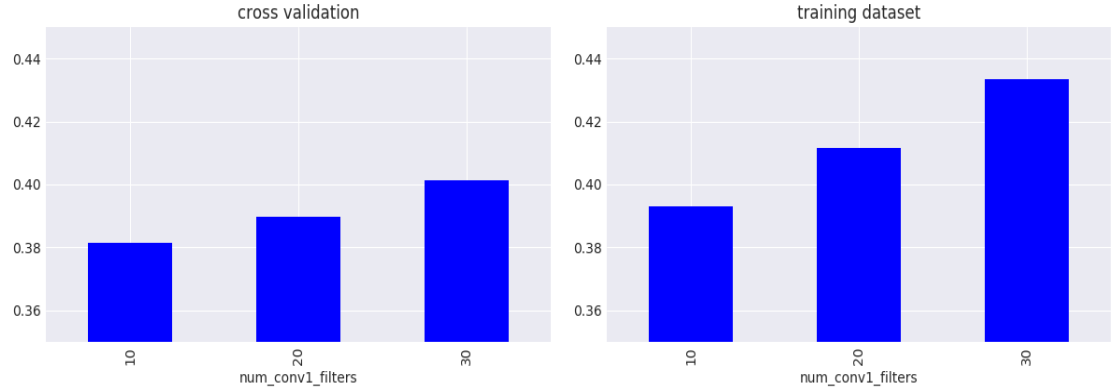


図 22. 畳み込みフィルタ数による正答率の変化(model 2)

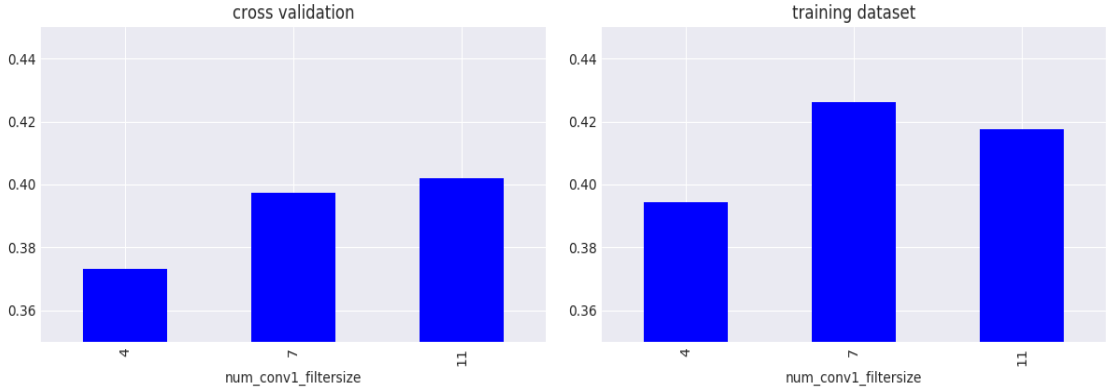


図 23. 畳み込みフィルタサイズによる正答率の変化 （ model 2)

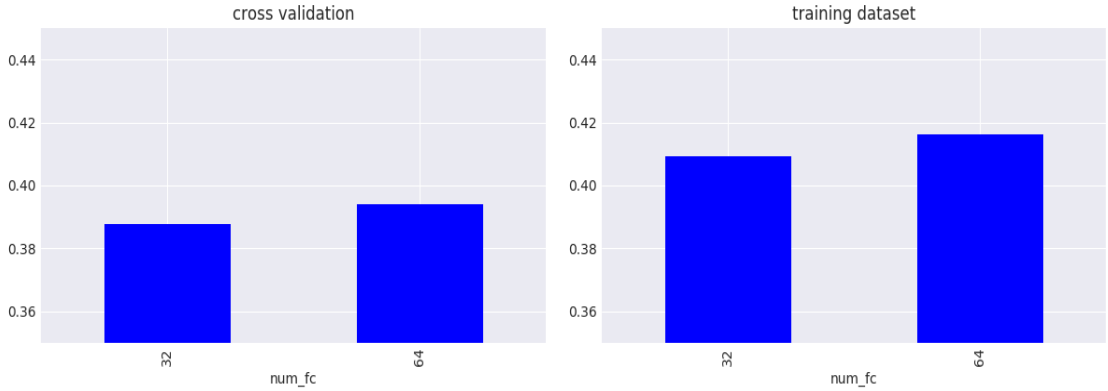


図 24. 全結合層ニューロン数による正答率の変化(model 2)

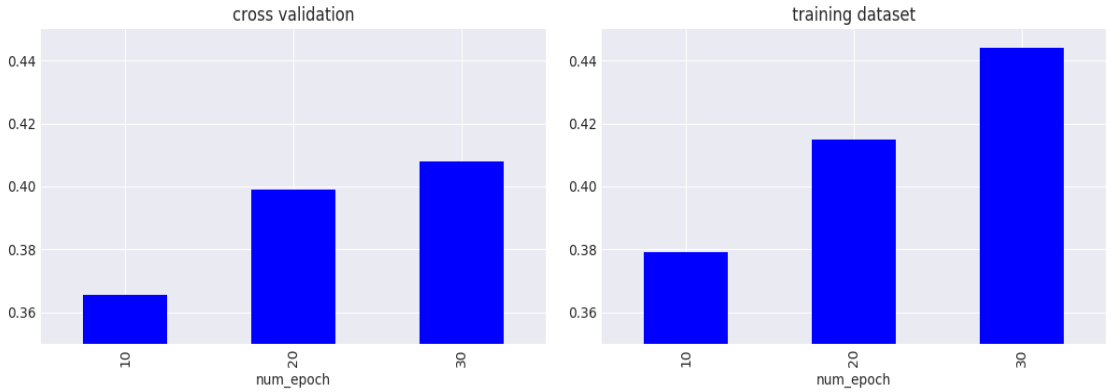


図 25. エポック数による正答率の変化(model 2)

図 22 の畳み込みフィルタ数は、30 が最も高い値を示している。

図 23 の畳み込みフィルタサイズはトレーニングデータセットでは 7 が最大であるが、CV においては 11 が上回った。これは 7 が過学習をしていることを示している。フィルタサイズ 11 は、畳み込み後の値が 1 つであるため、プーリングを行っていない。

エポック数は、増やすことで正答率が上昇した。表 00 には 10 もしくは 20 のみ現れているので、30 にすると過学習を起こすと思われる。そのため、今回は 20 を最適なパラメータとした。全結合層ニューロン数は大きな差がみられなかったが、僅かに 64 が上回っている。

以上の結果より、畳み込みは良い結果をもたらさないことがわかった。

3-3. 疎結合したニューラルネットワーク(model 3)による予測

0-0 で示したアーキテクチャでトレーニングする。特徴量は 0-0 のものを用いる。パラメータは 0-0 の結果を考慮して、以下のように設定した。

表 7. model3 のハイパーパラメータ

バッチサイズ	64
学習率	0.001
全結合層のニューロン数	64,128
window size	11
特徴量ベクトルの長さ	49
畳み込みフィルタ数	10, 15, 20, 25
エポック数	10,20, 30
dropout rate	0.5
L2 loss weight	0.01
誤差関数中の w(正値への重み)	0.4

表 8 に結果を示した。前回と同じように margin5 正答率が CV と Train で 2%以上異なっている場合は除外した。

表 8. ハイパーパラメータの組み合わせとマージン 3, 5 正答率

全結合層ニューロン数	畳み込みフィルター数	エポック数	margin 3 CV 正答率	margin 5 CV 正答率	margin 3 Train 正答率	margin 5 Train 正答率
64	15	20	0.3978	0.4331	0.4136	0.4458
128	10	20	0.3962	0.4324	0.3914	0.4215
64	10	20	0.3880	0.4242	0.4095	0.4413
64	20	10	0.3886	0.4218	0.4054	0.4376
128	15	10	0.3807	0.4136	0.3948	0.4270
64	15	10	0.3773	0.4095	0.3924	0.4256
64	10	10	0.3688	0.4006	0.3709	0.4078
128	10	10	0.3657	0.3979	0.3626	0.4003

全結合層 64, 畳み込みフィルタ数 15, エポック数 20 のモデルを採用した。エポック数 30 のものは全て除外されてしまっており、増加により過学習がおきたことがわかる。また、フィルタ数も同様に増加が過学習につながっている。

最良モデルの正答率と、マージンによる変化を図 26, 27 に示した。

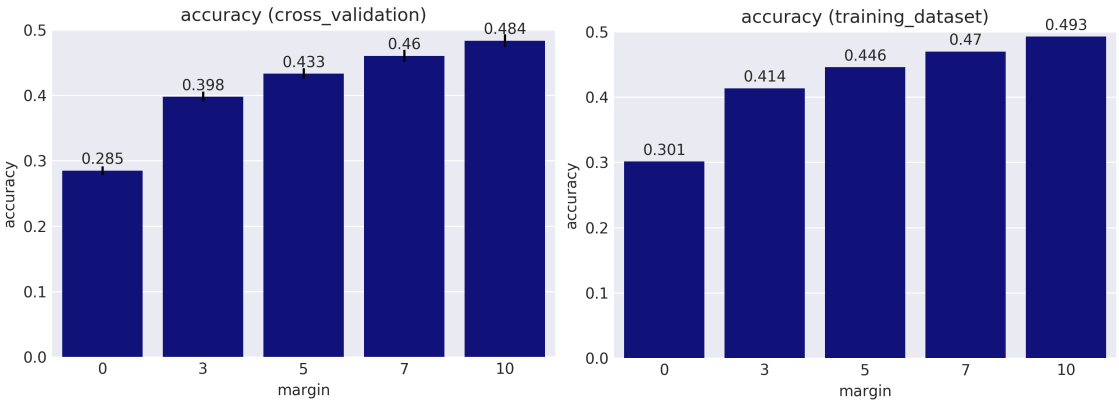


図 26. CV・Train に対するマージン正答率(model 3)

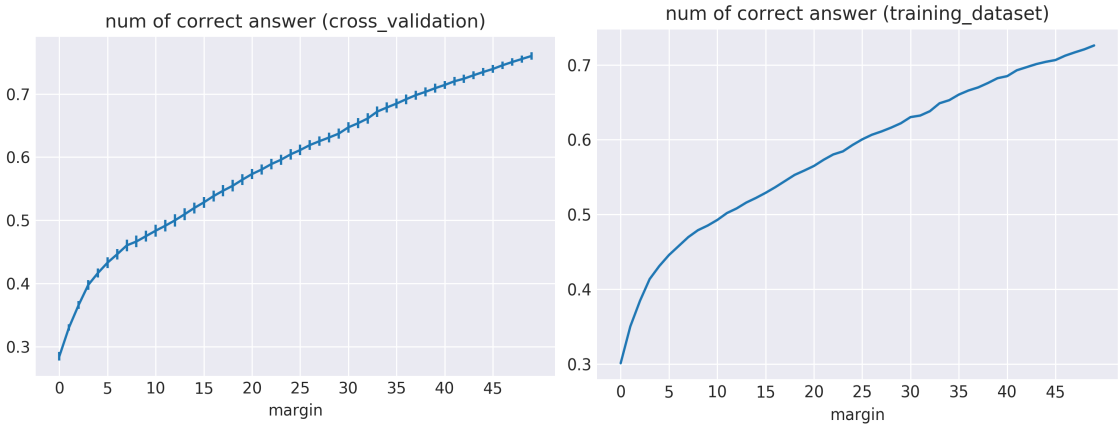


図 27. CV・Train に対するマージン・正答率のグラフ(model 3)

model2 と比較して margin 5 CV 正答率が 1%ほど上昇している。model2 と異なり、window のある位置の全ての特徴量が 1 つのニューロンにつながっている。このニューロンが特徴量を捉え、良い結果をもたらしたと考えられる。

このアーキテクチャ・特徴量・パラメータが最も良い結果となったので、このモデルを基本モデルとした。

3-4. ニューラルネットワークの学習状況

基本モデルのトレーニングの学習を追跡する。図 28 は 誤差関数の値の変化である。

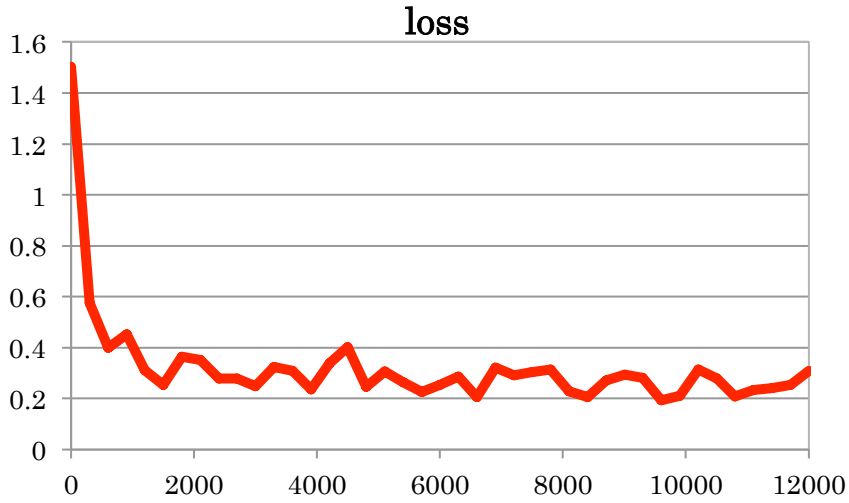


図 28. 基本モデルの誤差関数の値の変化

トレーニングの開始から 500 ステップほどで急激に誤差が減少しており、学習がうまく行われていることがわかる。2000 ステップを超えたところから減少がゆるやかになり、学習がこれ以上進まないと考えられる。

図 29 は L2 loss の値の変化である。

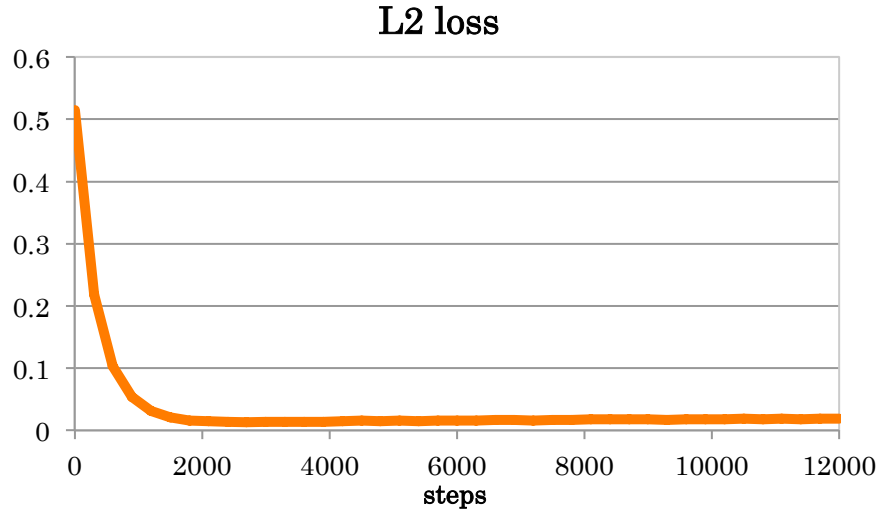


図 29. 基本モデルの L2 loss の値の変化

2000 ステップを超えた当たりからほぼ 0 となっており、正則化が行われていることがわかる。

3-5. 残基レベルでのリンカー予測の結果

基本モデルを用いて残基レベルでのドメイン・リンカーの 2 値予測を行い、リンカーの Precision, Sensitivity を算出した。テストのために cross validation を行った。

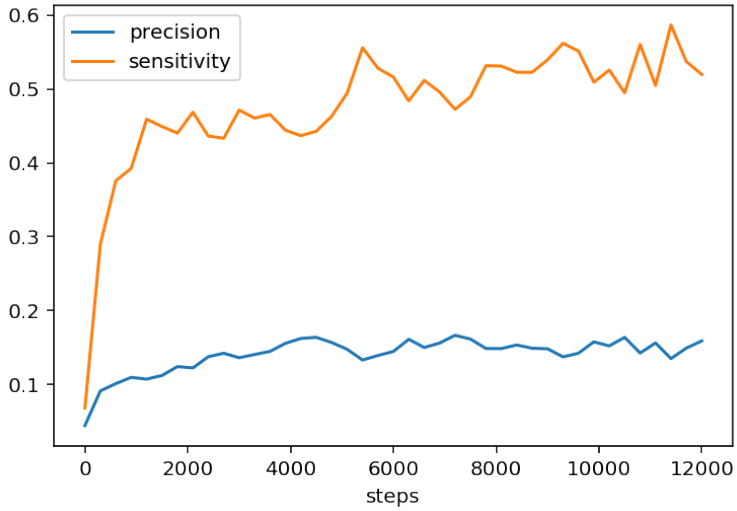


図 30. 基本モデルの残基レベルでの Precision、Sensitivity

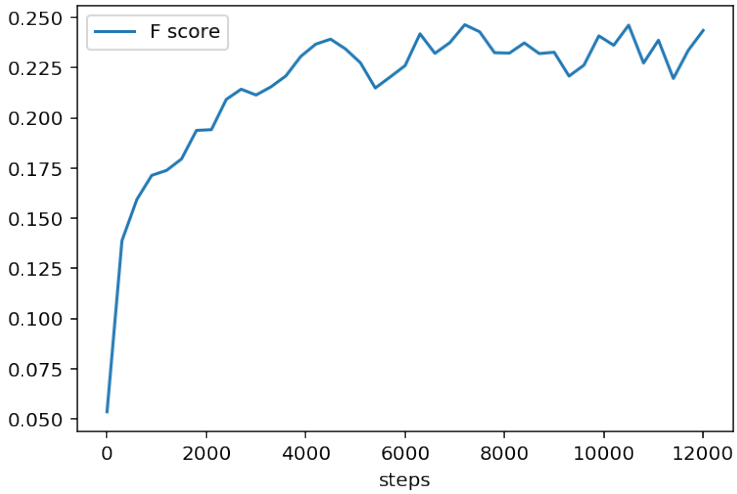


図 31. 基本モデルの残基レベルでの F 値

Precision が 10%台であり、極めて低い。Sensitivity も 50%程度である。残基レベルで正確に予測することは難しい課題である。ドメインリンカー部位が持っている情報のみではドメインリンカーを正確に予測することが出来ないと考えられる。特にタンパク質 3 次元構造に関する特徴が必要であると思われる。

F Score は 8000 ステップ以降安定した値を取っており、学習がこれ以上進まないことを示している。

3-6. 閾値の調整

ニューラルネットワークの出力値を以下のフィルターで畳み込み、閾値を上回ったらマルチドメインとする。フィルターと閾値を以下のように変え、バリデーションセット(マルチドメイン 300 個、シングルドメイン 300 個)を用いて評価した。

表 9. フィルターの種類

フィルター番号	フィルター
0	[1, 1, 1],
1	[0.5, 1, 0.5]
2	[1, 1, 1, 1, 1]
3	[0.5, 1, 1, 1, 0.5]
4	[0.5, 0.8, 1, 0.8, 0.5]
5	[1, 1, 1, 1, 1, 1, 1]

表 10. パラメータの組み合わせと F 値の比較

threshold	フィルター番号	Precision	Sensitivity	F score
0.6	0	0.20805	0.62000	0.31156
0.6	1	0.21099	0.65306	0.31894
0.6	2	0.23989	0.52976	0.33024
0.6	3	0.23018	0.54878	0.32432
0.6	4	0.23117	0.53614	0.32305
0.6	5	0.25161	0.42623	0.31643
0.65	0	0.22481	0.52096	0.31408
0.65	1	0.22084	0.54601	0.31449
0.65	2	0.26367	0.44086	0.32998
0.65	3	0.25857	0.44385	0.32677
0.65	4	0.25938	0.44865	0.32871
0.65	5	0.28800	0.33488	0.30968
0.7	0	0.25234	0.43316	0.31890
0.7	1	0.25460	0.44865	0.32485
0.7	2	0.30924	0.35648	0.33118
0.7	3	0.30315	0.36150	0.32976
0.7	4	0.30435	0.35981	0.32976
0.7	5	0.33862	0.26230	0.29561
0.75	0	0.29661	0.31818	0.30702
0.75	1	0.29675	0.33796	0.31602
0.75	2	0.35393	0.25200	0.29439
0.75	3	0.34896	0.27347	0.30664
0.75	4	0.35263	0.27347	0.30805
0.75	5	0.38028	0.20074	0.26277
0.8	0	0.32738	0.21912	0.26253
0.8	1	0.33526	0.23293	0.27488
0.8	2	0.38211	0.17091	0.23618
0.8	3	0.38760	0.18248	0.24814
0.8	4	0.38462	0.18315	0.24814
0.8	5	0.39130	0.12544	0.18997

threshold 0.7、フィルター [1, 1, 1, 1, 1] (window 5 でのスムージングに等しい)が最も F score が良い値となった。基本モデルとこれらのパラメータを用いて予測を行う。

3-7. テストデータによるテストと他の予測器との比較

マルチドメイン 300 個、シングルドメイン 300 個に対して、シングルドメイン/マルチドメインの二択を my model 及び他のモデルで予測した。図 32, 33 はその比較である。

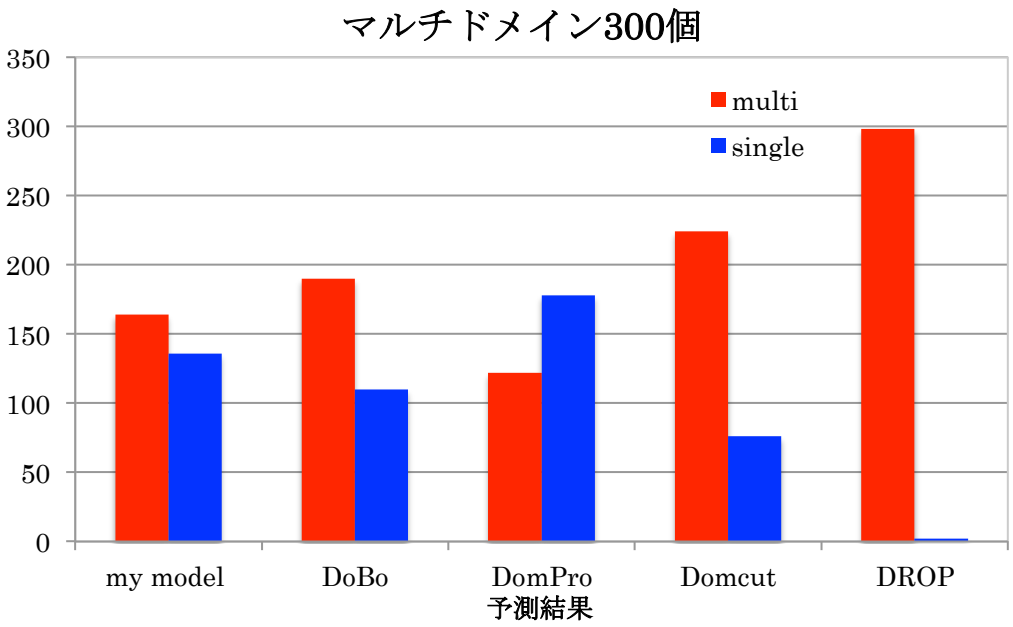


図 32. マルチドメインに対するマルチ/シングルの予測結果の比較

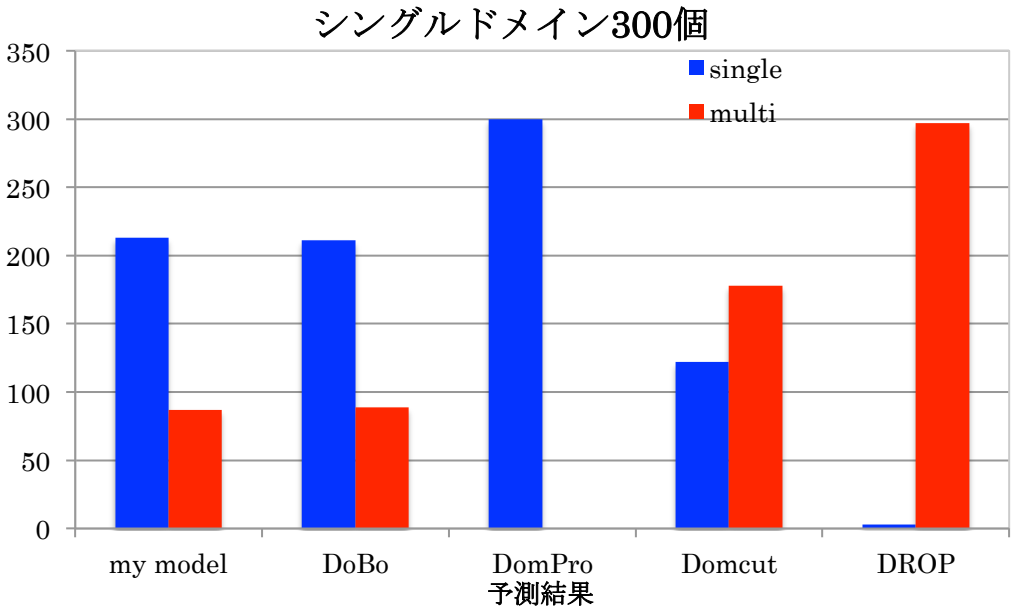


図 33. シングルドメインに対するマルチ/シングルの予測結果の比較

my model はマルチドメインデータセットに対し、マルチドメインと予測した割合が、55%であり、半分を上回った。シングルドメインに対しても 71%の割合で正答しており、予測対象がシングルドメインかマルチドメインか不明な場合であっても機能することが分かった。

DROP はほぼ全てマルチドメインと判定している。これは threshold が低すぎることが原因である。

DomPro はシングルドメインに対して正答率が 100%である。しかし、マルチドメインに対してもシングルドメインであると判定していることが多く、正答率は 41%である。

DoBo はマルチ・シングル両方で正答率が 5 割を上回っている。

図 34 はシングル・マルチドメインを合わせた正答率である。

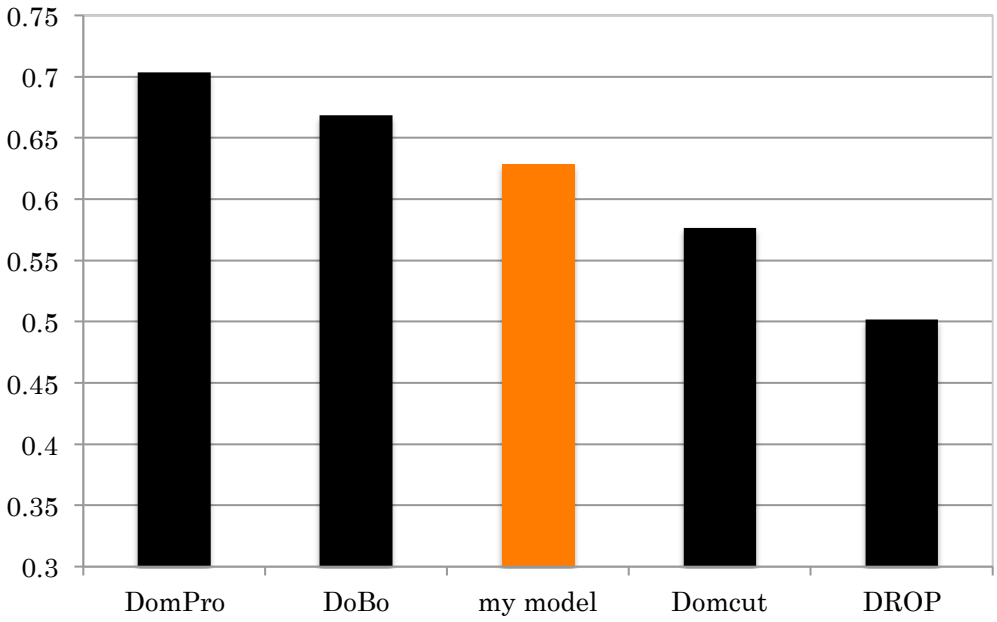


図 34. マルチ/シングル予測の正答率の比較

DomPro はシングルドメインに対して 100%正答しているため、合計の正答率が高くなっている。マルチドメイン判定の閾値を高くすることでこの分類を可能にしていると考えられる。これは DomPro の正答率向上のための戦略であると思われる。Dobo は my model と同じシングル・マルチともに正答を試みる戦略であると思われるが、my model を正答率で上回っている。今回のモデルは、シングル・マルチドメインの分類の改善が必要であると考えられる。

3-8. マルチドメインデータに対する予測結果の比較

マルチドメインデータに対して様々なソフトウェア、乱数でリンカー部位を予測した結果である。

	TP	FP	FN
my model	82	82	248
DROP	82	216	248
DoBo	51	139	279
DomPro	20	102	310
Domcut	20	204	310
ランダム(一様分布)	17.5	282.5	312.5
ランダム(二項分布)	51.7	248.3	278.3

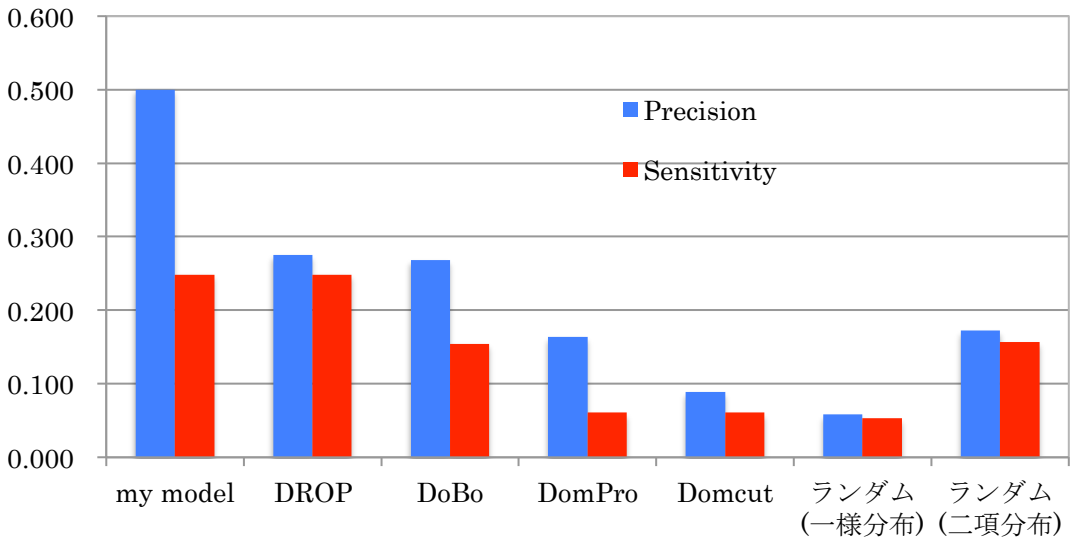


図 35. リンカー予測結果の Precision、Sensitivity による比較

my model は Precision が高い値を示している。マルチドメインであると判定した場合、50%で正答している。一方、Sensitivity は低く、リンカー全てを当てることが出来ていない。これはマルチドメインをシングルドメインであると判定しているケースが多いのが一因である。

二項分布によるランダムが比較的高い値を示しているが、これはリンカーがアミノ酸配列の中心に位置している場合が多いためである。

3-9.　F score による比較

・最後に、F score を用いて予測器の比較を行った。

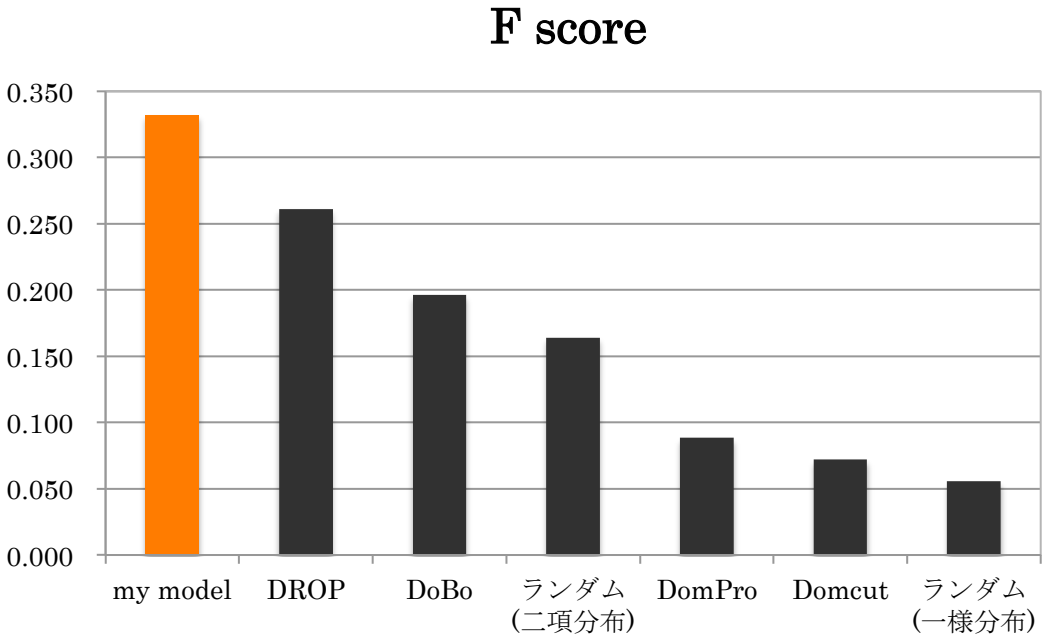


図 36. リンカー予測結果の F 値による比較

my model が最も高くなり、マルチドメインデータに対しては最良の予測器である。

今回、my model が最も高くなった理由はいくつか考えられる。最も影響を与えたのは ISD という独自の概念を用いて正誤を判定したことであると考えた。DROP 以外の予測器は ISD に関する情報を持っていないため、my model に有利に働いたと思われる。

HHblits を用いて高精度な MSA を行うことが出来たことも予測向上に貢献したと思われる。psi-blast よりも遠縁の配列を取得することが出来、精度の高い共進化情報を特徴量として用いることが可能になった。DROP は共進化情報を用いていないため、この点で上回ることが出来たと考えられる。

Dropout や Adam など近年開発された技術を用いたニューラルネットワークを用いたことも貢献した可能性もある。

3-10. 予測の例

実際に予測した結果を紹介する。

・成功例　PDBID: 2hcz, chain: X

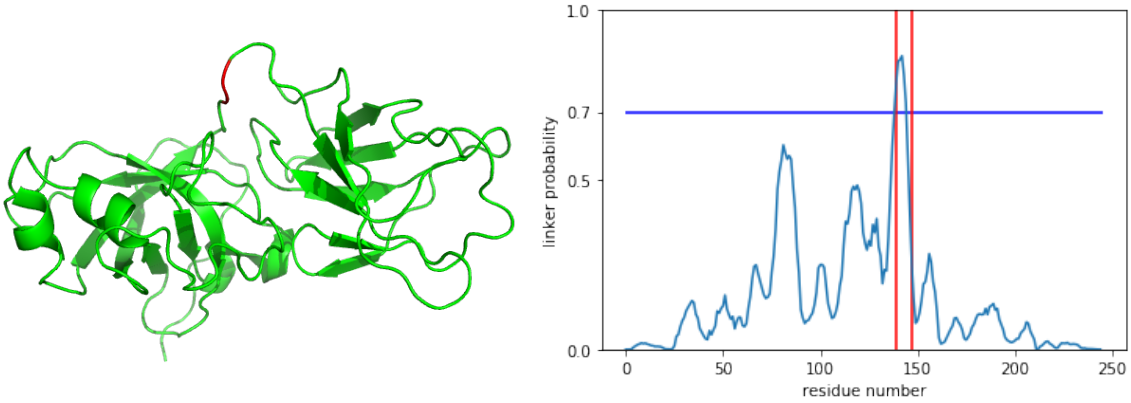


図 37. 成功例のタンパク質と予測結果

図 37 左の赤色で示した残基がリンカー残基である。また、図 37 右は x 軸が残基番号、y 軸がリンカー確率を表し、赤色の線で囲まれた部分がリンカー部分を示している。水色の折れ線グラフがリンカーの確率を表しており、青の線で表された閾値を超えるとリンカーと判定している。この例ではリンカー部分を予測できていることがわかる。

・失敗例 1　PDBID: 1a5t, chain: A

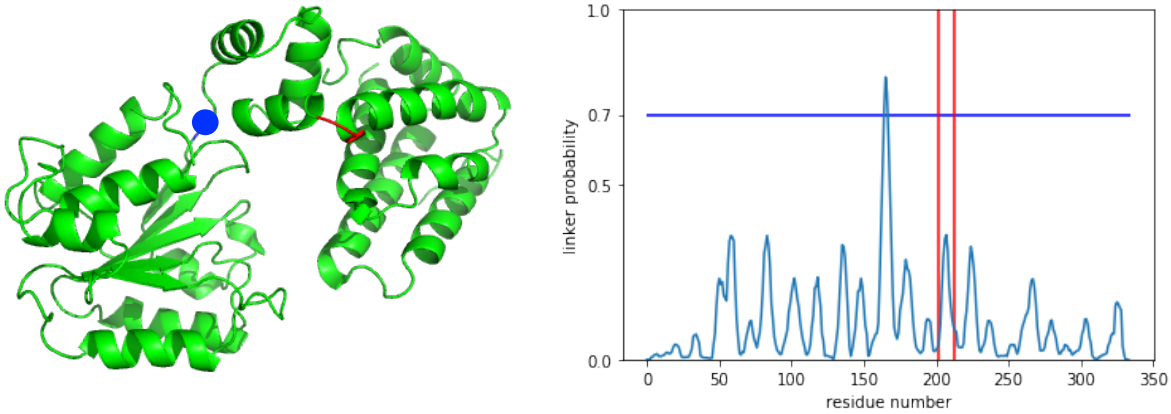


図 38. 失敗例のタンパク質と予測結果 1

青丸の部分が予測した残基である。リンカー定義部分から外れているが、目視での判断では、予測残基で切断してもドメインを分割可能であると思われる。

・失敗例 2　PDBID: 1bg6, chain: A

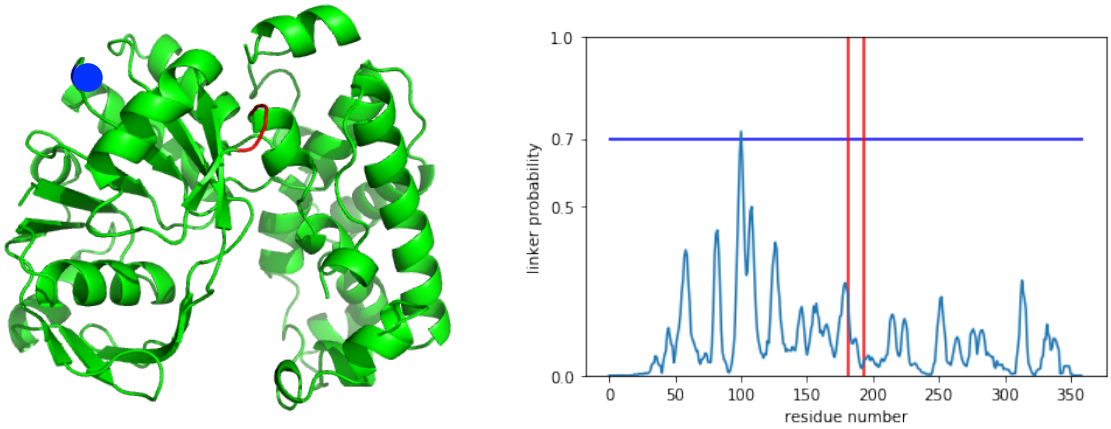


図 39. 失敗例のタンパク質と予測結果 2

正解のリンカー部分はほぼ反応がなく、青丸で示した 100 残基目がリンカーと予測している。
完全に失敗した例である。

・失敗例 3　PDBID: 1k9y, chain: A

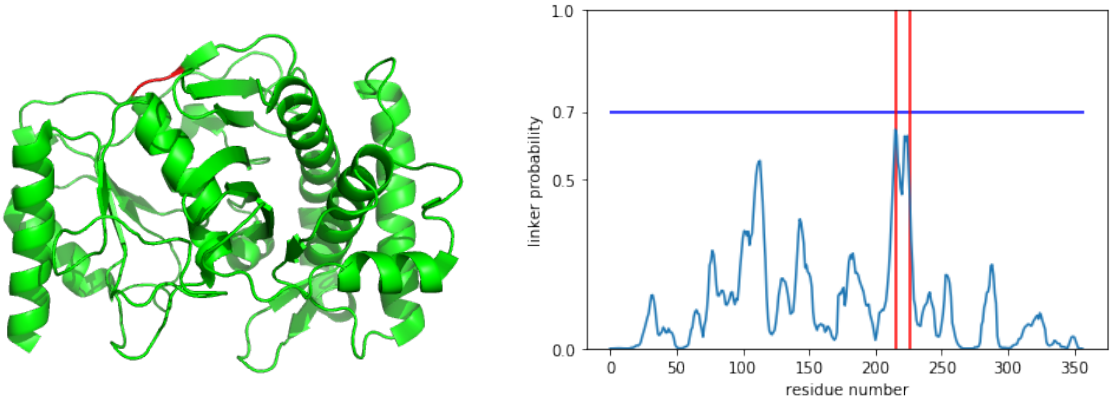


図 40. 失敗例のタンパク質と予測結果 3

マルチドメインであるにも関わらず、シングルと判定している。ドメイン間の相互作用が大きいためではないかと思われる。しかし、実際のリンカー部分において予測したリンカー確率が最大となっており、閾値を下げることでマルチドメインと判定し、リンカー部位を当てることが出来ただろうと考えられる。

4．結論

本研究では、近年開発されたニューラルネットワークを用いてドメインリンカー予測器を開発した。ニューラルネットワークの種類は畳み込みこみニューラルネットワークと疎結合したニューラルネットワークを用いた。また、先行研究で取り入れていなかった露出溶媒表面積予測と共進化情報を新たに取り入れることによって予測精度の向上を試みた。これによって過去の予測器を超える精度を達成することが出来た。

今後の課題として、Recurrent Neural Network を用いた予測がある。共進化情報は遠く離れた残基同士の相互作用を検出するが、今回の研究では十分に利用できていないため、可変長の配列から予測可能な RNN が適している可能性がある。また、今回、ドメイン・リンカーの 2 値によって予測しているが、リンカー残基内であっても、その特徴は大きく異なる可能性が高い。より一般化して、ある残基で切断した際のエネルギーロスの値を予測することによってより高精度になる可能性がある。その場合、今回よりもより大きなデータセットを用いる事ができるので、転移学習を適用できるかもしれない。また、オートエンコーダによって良い特徴量を抽出出来る可能性もある。このように課題は多く、近年のニューラルネットワークを用いることでより高精度に出来る可能性を秘めており、この予測結果はタンパク構造予測、ドメイン同定の有用な情報になると思われる。

5．参考文献

[1]Suyama, M., & Ohara, O. (2003). DomCut: prediction of inter-domain linker regions in amino acid sequences. BIOINFORMATICS APPLICATIONS NOTE, 19(5), 673–674.

[2] Satoshi Miyazaki, Yutaka Kuroda, Shigeyuki Yokoyama. (2002). Characterization and prediction of linker sequences of multi-domain proteins by a neural network. J Struct Funct Genomics. 15, 37-51

[3]Sim, J., Kim, S.-Y., & Lee, J. (2005). PPRODO: prediction of protein domain boundaries using neural networks. Proteins, 59(3), 627–632.

[4]Cheng, J., Sweredoski, M.J. & Baldi. (2006) DOMpro: Protein Domain Prediction Using Profiles, Secondary Structure, Relative Solvent Accessibility, and Recursive Neural Networks. P. Data Min Knowl Disc 13: 1

[5]Eickholt, J., Deng, X., & Cheng, J. DoBo: Protein domain boundary prediction by integrating evolutionary signals and machine learning.

[6]Ebina, T., Toh, H., Kuroda, Y. (2011) DROP: an SVM domain linker predictor trained with optimal features selected by random forest, Bioinformatics, 15;27(4):487-94

[7] 金澤一郎 (監修), 宮下保司 (監修), Eric R. Kandel (編集), James H. Schwartz (編集), Steven A. Siegelbaum (編集), Thomas M.Jessell (編集), A. J. Hudspeth (編集). カンデル神経科学 PRINCIPLES OF NEURAL SCIENCE Fifth edition. メディカル・サイエンス・インターナショナル

[8] Hubel DH, Wiesel TN. (1968) Receptive fields and functional architecture of monkey striate cortex. J Physiol. Mar 195(1), 215-243

[9]A. Krizhevsky, I. Sutskever, and G. E. Hinton. (2012). Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems 25, 1106–1114.

[10]Li, Z., & Yu, Y. (2016). Protein Secondary Structure Prediction Using Cascaded Convolutional and Recurrent Neural Networks.

[11]Busia, A., & Jaitly, N. Next-Step Conditioned Deep Convolutional Neural Networks Improve Protein Secondary Structure Prediction. (2017). http://arxiv.org/abs/1702.03865

[12]Zeng, H., Edwards, M. D., Liu, G., & Gifford, D. K. (2016). Convolutional neural network architectures for predicting DNA-protein binding. Bioinformatics, 32(12), i121–i127.

[13]Alipanahi, B., Delong, A., Weirauch, M. T., & Frey, B. J. (2015). Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nat Biotechnol, 33(8), 831–838.

[14]Ronneberger O, Fischer P, Brox T. (2015). U-Net: convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention–MICCAI, 234 - 241.

[15]Zhang, W., Li, R., Zeng, T., Sun, Q., Kumar, S., Ye, J., & Ji, S. (2015). Deep Model Based Transfer and Multi-Task Learning for Biological Image Analysis. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15, 7790(c), 1475–1484.

[16]Fox NK, Brenner SE, Chandonia JM. (2014). SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. Nucleic Acids Research 42:D304-309.

[17] Ian Sillitoe, Tony E. Lewis, Alison Cuff, Sayoni Das, Paul Ashford, Natalie L. Dawson, Nicholas Furnham, Roman A. Laskowski, David Lee, Jonathan G. Lees, Sonja Lehtinen, Romain A. Studer, Janet Thornton, Christine A. Orengo (2015). CATH: comprehensive structural and functional annotations for genome sequences, Nucleic Acids Research, vol.43, D1, 376–381

[18]Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucleic Acid Res., vol. 25 3389-3402

[19]Ebina Teppei, Umezawa Yuki, Kuroda Yutaka. (2013). IS-Dom: a dataset of independent structural domains automatically delineated from protein structures, Journal of Computer-Aided Molecular Design, 27(5),419-426

[20]Kabsch W, Sander C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers. 22 2577-2637.

[21]I.K. McDonald and J.M. Thornton. (1994). Satisfying Hydrogen Bonding Potential in Proteins, JMB 238:777-793.

[22]Cock PA, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, Friedberg I, Hamelryck T, Kauff F, Wilczynski B and de Hoon MJL .(2009). Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics, 25, 1422-1423
[23]Larkin MA, Blackshields G, Brown NP, Chenna R, McGettigan PA, McWilliam H, Valentin F, Wallace IM, Wilm A, Lopez R, Thompson JD, Gibson TJ, Higgins DG. (2007). Clustal W and Clustal X version 2.0. Bioinformatics. 23(21):2947-2948.
[24]Yang Y, Heffernan R, Paliwal K, Lyons J, Dehzangi A, Sharma A, Wang J, Sattar A, Zhou Y. (2017) SPIDER2: A Package to Predict Secondary Structure, Accessible Surface Area, and Main-Chain Torsional Angles by Deep Neural Networks. Methods Mol Biol. 1484 55-63.
[25]Martin, L. C., Gloor, G. B., Dunn, S. D., & Wahl, L. M. (2005). Using information theory to search for co-evolving residues in proteins, 21(22), 4116–4124.
[26]Remmert, M., Biegert, A., Hauser, A., & Söding, J. (2011). HHblits: Lightning-fast iterative protein sequence searching by HMM-HMM alignment. Nat Methods. 9(2) 173-175.
[27]Mart´ın Abadi *et al.* Large-Scale Machine Learning on Heterogeneous Distributed Systems http://arxiv.org/abs/1603.04467 (2016)
[28]Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, (2014). R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, 15, 1929–1958.
[29]Diederik P. Kingma, Jimmy Ba. Adam: A Method for Stochastic Optimization https://arxiv.org/abs/1412.6980

TensorFlow で学ぶディープラーニング入門 初版　中井　悦司 著　　マイナビ出版
機械学習プロフェッショナルシリーズ 深層学習 岡谷　貴之 著　　講談社
深層学習 -Deep Learning- 初版 監修 人工知能学会 神嶌敏弘 編集 麻生 英樹、安田宗樹、前田 新一、岡野原 大輔、岡谷 貴之、久保 陽太郎、ボレガラ ダヌシカ著 近代科学社
これならわかる深層学習 入門　瀧　雅人著　講談社

6. 謝辞

本研究を進めるにあたり、丁寧なご指導と助言を頂きました黒田裕教授に深く感謝致します。また、研究を手伝って頂いた松沢佑紀氏に感謝致します。また、議論を通じて多くの助言と示唆を頂いた黒田研究室の皆様に感謝致します。