Uniwersytet Warszawski
Wydział Fizyki

Aleksander Wiński
Nr albumu: 370493

Zastosowanie metod uczenia głębokiego do przewidywania białkowych struktur drugorzędowych typu helisa π na podstawie sekwencji i informacji ewolucyjnej

Praca licencjacka
na kierunku Zastosowania fizyki w biologii i medycynie
specjalność Projektowanie molekularne i bioinformatyka

Praca wykonana pod kierunkiem
dr Stanisława Dunina-Horkawicza
Centrum Nowych Technologii UW

Warszawa, lipiec 2018

Oświadczenie kierującego pracą

Oświadczam, że niniejsza praca została przygotowana pod moim kierunkiem i stwierdzam, że spełnia ona warunki do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data Podpis kierującego pracą

Oświadczenie autora (autorów) pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przez mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data Podpis autora (autorów) pracy

Streszczenie

W pracy został przedstawiony problem, zaproponowane rozwiązanie i omówione wyniki przewidywania regionów π-helikalnych na podstawie sekwencji i informacji ewolucyjnej. Π-helisy stanowią rzadki element struktury drugorzędowej białek. Dostępne metody przewidywania struktury drugorzędowej na podstawie sekwencji osiągają słabe rezultaty w wykrywaniu π-helis. Przygotowano odpowiedni zbiór sekwencji (0.85% reszt π-helikalnych) i wykorzystano uczenie głębokie do określenia, które reszty w sekwencji tworzą π-helisy. Wytrenowany model osiągnął na zbiorze testowym 50% precyzji i 42% skuteczności.

Słowa kluczowe

π-helisy, uczenie maszynowe, uczenie głębokie, struktura drugorzędowa, bioinformatyka

Dziedzina pracy (kody wg programu Socrates-Erasmus)

13.2 Fizyka

Tytuł pracy w języku angielskim

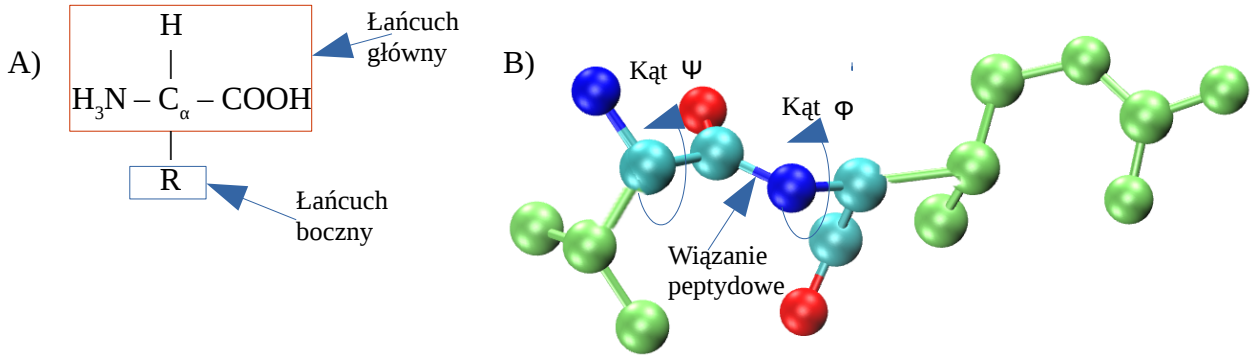
The use of deep learning methods to predict protein π helix secondary structure based on sequence and evolutionary information

Spis treści:

1. Wstęp

Podstawowe informacje o białkach

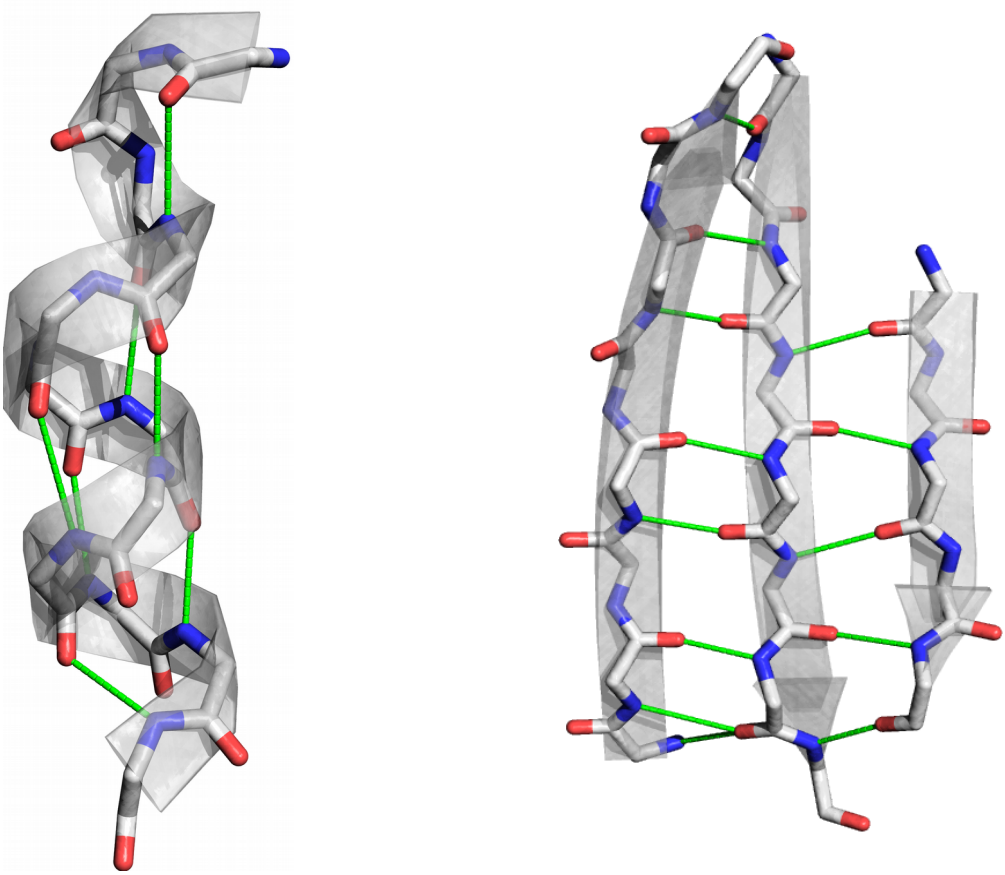
Białka są makromolekułami pełniącymi różnorodne i kluczowe funkcje w organizmach komórkowych i wirusach. Między innymi, katalizują reakcje chemiczne, transportują inne molekuły, odpowiadają za odpowiedzi immunologiczne organizmu, a także biorą udział w przekazywaniu sygnałów. Białka zbudowane są z aminokwasów połączonych wiązaniem peptydowym. Aminokwasy są związkami organicznymi posiadającymi grupę karboksylową (-COOH) oraz aminową (-NH_2). Aminokwasy biogenne, których w przyrodzie występuje 20, składają się z łańcucha głównego jednakowego dla wszystkich oraz łańcucha bocznego, który decyduje o ich specyficznych właściwościach fizykochemicznych. Dwa aminokwasy tworzące wiązanie peptydowe przedstawione są na Rysunku 1.1.



Rysunek 1.1. A) Wzór chemiczny aminokwasu. B) 2 aminokwasy (walina i arginina) połączone wiązaniem peptydowym. Łańcuch główny (azot – ciemnoniebieski, tlen – czerwony, węgiel – turkusowy), łańcuchy boczne – kolor zielony. Strzałką wskazane wiązanie peptydowe oraz kąty φ i Ψ.

Kolejne połączone ze sobą aminokwasy tworzą sekwencję białkową, w której poszczególne aminokwasy określane są jako reszty. W określonym środowisku białko zwija się do konformacji natywnej. Struktura ta zależy od składu aminokwasowego sekwencji, gdyż aminokwasy różnią się od siebie wieloma właściwościami takimi jak: wielkość łańcucha bocznego, ładunek elektrostatyczny czy hydrofobowość. Za stabilność struktury białka w dużej mierze odpowiadają wiązania wodorowe między tlenem grupy karbonylowej (CO), a azotem grupy amidowej (NH). Wiązania te odpowiadają za utworzenie

elementów struktury drugorzędowej białka, z których najczęściej występującymi są α-helisy i β-kartki, przedstawione na Rysunku 1.2. Alternatywnie, struktury drugorzędowe mogą być zdefiniowane przez charakterystyczne wartości kątów torsyjnych w łańcuchu głównym: φ (wokół wiązania N-C_α) oraz Ψ (wokół wiązania C_α-C) - Rysunek 1.2.



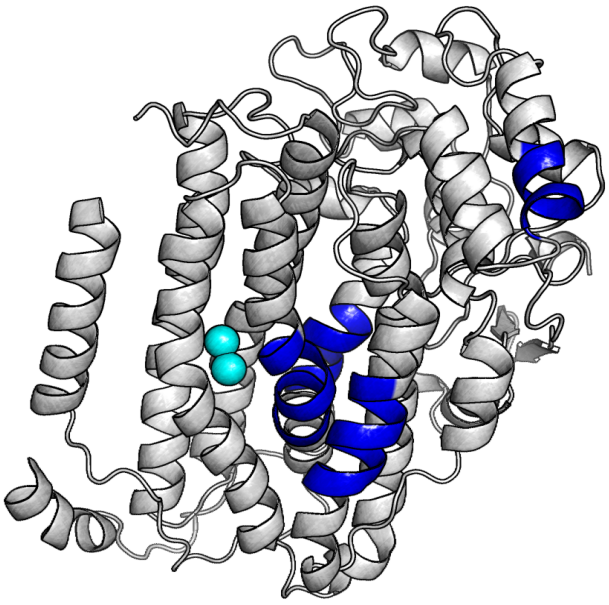
Rysunek 1.2. α-helisa i β-kartka z zaznaczonymi na zielono wiązaniami wodorowymi. Tleny grupy karbonylowej oznaczone na czerwono, a azoty grupy amidowej na niebiesko.

Różnice w miarach kątów czy układzie wiązań wodorowych pozwalają na zdefiniowanie poszczególnych elementów struktury drugorzędowej. Stosując algorytmy takie jak DSSP[1], można zaklasyfikować reszty do jednej z ośmiu rodzajów struktury drugorzędowej:

- α-helisa (H)
- 3_{10} helisa (G)

- π-helisa (I)
- skręt z wiązaniami hydrofobowymi (T)
- β-kartka (E)
- β-mostek (B)
- skręt bez wiązań hydrofobowych (S)
- fragment nieustrukturyzowany (C)

Oddziaływania między aminokwasami odległymi w sekwencji wpływają na wzajemne ułożenie fragmentów struktury drugorzędowej i przyjęcie określonej konformacji nazywanej strukturą trzeciorzędową. Rysunek 1.3 przedstawia przykładową strukturę trzeciorzędową białka.



Rysunek 1.3. Struktura trzeciorzędowa białka (identyfikator PDB: 1mty łańcuch D). Na ciemnoniebiesko zaznaczone zostały π-helisy, a na turkusowo jony żelaza.

Aby określić strukturę przestrzenną białka najczęściej stosuje się krystalografię rentgenowską, która polega na badaniu obrazu dyfrakcyjnego powstającego po naświetleniu kryształu białka promieniami rentgenowskimi. Inną metodą jest NMR, czyli spektroskopia magnetycznego rezonansu jądrowego, która dzięki rejestracji widm korelacyjnych od kilku rodzajów jąder, pozwala na określenie

odległości między nimi. Dzięki takim doświadczeniom możliwe było stworzenie bazy struktur, z których największą jest PDB (Protein Data Bank) z 131672 rozwiązanymi strukturami.
Doświadczalne określenie struktury jest złożonym procesem, krystalografia rentgenowska wymaga na przykład uzyskania kryształu białka pozwalającego na wykonanie obrazu o wystarczającej rozdzielczości, co w przypadku na przykład białek błonowych, czyli zakotwiczonych w błonie komórkowej, jest dużym problemem. Dlatego rozwijane są metody obliczeniowe prowadzące do określania struktury białek, a narzędzia do przewidywania struktury drugorzędowej są ich przykładem.

Celem mojej pracy było uzupełnienie dostępnych programów do przewidywania struktury drugorzędowej, o narzędzie pozwalające przewidywać regiony π-helikalne. Stworzono program PiPred, który jest dostępny publicznie na serwerze Instytutu Maxa Plancka w Tybindze – https://toolkit.tuebingen.mpg.de. W dalszej części pracy opisane zostaną π-helisy oraz sposób wykonania i implementacji programu PiPred.

Π-helisy jako rzadka struktura drugorzędowa białek

Helisy, najczęstszy element struktury drugorzędowej białek, są określone przez powtarzający się wzór wiązań wodorowych między wodorem grupy amidowej (NH) i tlenem grupy karbonylowej (CO). W przypadku najczęściej występujących α-helis oddziaływanie to występuje między resztą *i* oraz *i+4* w sekwencji aminokwasowej. Natomiast w π-helisach wiązania wodorowe powstają między resztą *i* oraz *i+5*, co powoduje, że ich struktura jest mniej upakowana. Względna niestabilność π-helisy wynika z występowania nieoptymalnych zakresów kątów torsyjnych w resztach ją tworzących, słabszymi oddziaływaniami van der Waalsa oraz wysokim kosztem entropowym związanym z ułożeniem 5 reszt, tak aby powstało wiązanie wodorowe[2]. W konsekwencji π-helisy są najczęściej krótkie, zazwyczaj o długości 7 reszt aminokwasowych i stosunkowo rzadkie, obecne tylko w 15 % struktur białek[3]. Pomimo rzadkiego występowania, π-helisy są często obecne w funkcjonalnych regionach białek, tworząc miejsca wiążące dla ligandów i jonów metali[2]. W porównaniu do α-helis, π-helisy różnią się preferowanymi aminokwasami[4]. Aminokwasy aromatyczne występują zazwyczaj na końcach, podczas gdy

aminokwasy polarne preferowane są w centrum helisy[3]. Sugeruje się, że π-helisy wyewoluowały w wyniku insercji jednego aminokwasu w rejonach α-helikalnych[5]. π-helisy występują więc najczęściej w środku α-helis bądź na ich N- lub C-końcu.

Warto zauważyć, że istnieją narzędzia, które dla znanej struktury białka, przyporządkowują elementy struktury drugorzędowej. Popularne programy, takie jak DSSP czy STRIDE[6] preferują w przyporządkowaniach α-helisy, co powoduje, że π-helisy są często pomijane. W związku z powyższym w pracy do konstrukcji zbioru danych zawierających poprawne przypisania reszt tworzących π-helisy został wykorzystany algorytm zaproponowany w pracy Cooley-a i wsp.[5], który zostanie opisany w kolejnym rozdziale pracy.

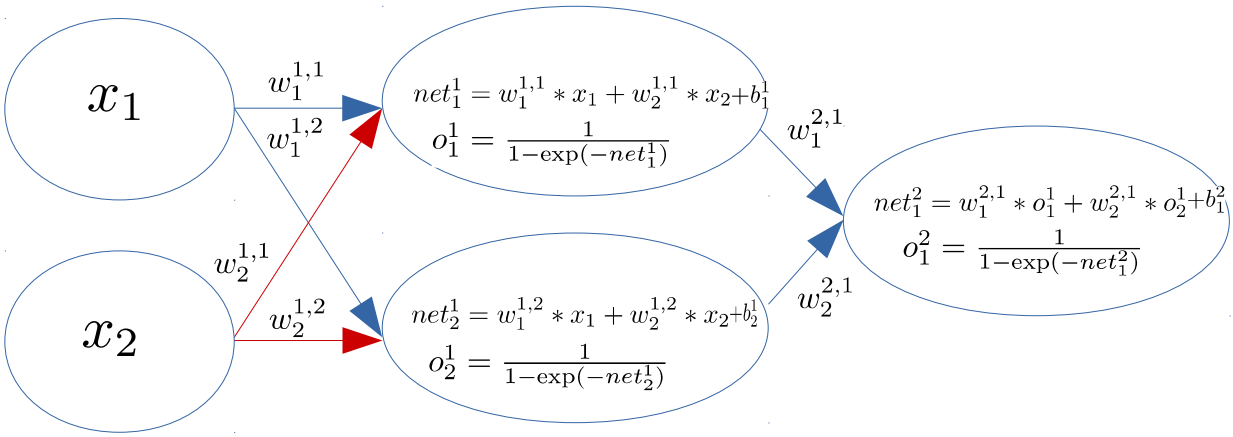
Przewidywanie struktury drugorzędowej na podstawie sekwencji jest aktywnym obszarem badań i istnieje wiele metod zaproponowanych do rozwiązania tego problemu. Nowoczesne metody często wykorzystują sieci neuronowe i uczenie głębokie, osiągają skuteczność w granicach od 75 do 85% dla 3-stanowego przewidywania oraz do 70% przy przewidywaniu 8 elementów struktury drugorzędowej. Najlepsze metody takie jak DeepCNF[7] czy DCRNN[8] wykazują jednak skuteczność przewidywania π-helis na poziomie 0%. Jest to spowodowane bardzo małą liczbą sekwencji zawierających π-helisy, w zbiorach danych takich jak CB6133 czy CB513 używanych do trenowania tego typu programów.

Ze względu na role pełnione przez π-helisy w strukturach białek, ich przewidywanie na podstawie sekwencji byłoby korzystne i umożliwiło określanie potencjalnych miejsc funkcjonalnych takich jak miejsca wiązania ligandów. W pracy zostanie omówiona metoda, PiPred, oparta o głębokie sieci neuronowe, trenowana na zbiorze z prawidłowo przypisanymi resztami tworzącymi π-helisy.

Głębokie sieci neuronowe

W tym podrozdziale zostaną opisane sieci neuronowe w kontekście wykorzystania w uczeniu z nadzorem. Sieci neuronowe złożone są z warstw, a te z kolei z neuronów. Warstwę wejściową stanowią zebrane egzemplarze cech, a ostatnią przewidziane wartości odpowiadające tym egzemplarzom. Sieci głębokie pomiędzy warstwą wejściową i wyjściową zawierają warstwy ukryte, nazwane tak ze względu na tworzenie przez nie nowych cech poprzez

przekształcenia algebraiczne już istniejących. Przykładowo egzemplarzami mogą być osoby, cechami wiek oraz waga a wartością, która ma być przewidziana 1, gdy osoba uprawia sport i 0 w przeciwnym wypadku. Sieć odpowiadająca temu przykładowi została przedstawiona na Rysunku 1.4. Jest to sieć gęsta, czyli taka w której każdy neuron danej warstwy jest połączony z każdym neuronem z warstwy poprzedniej. Złożona jest z warstwy wejściowej zawierającej dwie cechy, jednej warstwy ukrytej złożonej z dwóch neuronów oraz warstwy wyjściowej złożonej z jednego neuronu. Warstwa ukryta tworzy nowe cechy o_1^1 i o_2^1. Wartością wyjściową tej sieci jest prawdopodobieństwo, że dana osoba uprawia sport.



Rysunek 1.4. Przykład sieci neuronowej. Indeks górny dla o, net i b oznacza numer warstwy (warstwa wejściowa ma numer 0), zaś indeks dolny oznacza numer neuronu w warstwie. Indeks górny w to odpowiednio numer warstwy i numer neuronu w warstwie, zaś dolny odpowiada numerowi cechy mnożonej przez w . Definicje oznaczeń podane poniżej.

W najprostszym modelu zadaniem neuronu jest obliczenie iloczynu skalarnego (net) wektora cech wejściowych $[\mathrm{x}_1, ..., x_n]$ (są to wyjścia z neuronów z poprzedniej warstwy) i wektora wag $[\mathrm{w}_1, ..., w_n]$ przesuniętego o próg b (Równ. 1):

$$net = \sum_{i=1}^{n} w_i * x_i + b \qquad (1)$$

Istotnym elementem sieci głębokich jest funkcja aktywacji, czyli funkcja według, której obliczana jest wartość wyjścia neuronu o dla danego wyniku net.

e

Możliwą funkcją aktywacji jest na przykład funkcja sigmoidalna, wyrażona jak w (Równ. 2):

$$f(net) = \frac{1}{1-\exp(-net)} \tag{2}$$

Rolą funkcji aktywacji jest wprowadzenie do obliczeń nieliniowości, bez której sieć można byłoby zastąpić regresją liniową. Uczenie sieci polega na optymalizacji wag w i progów b, dla każdego neuronu w sieci tak, żeby zminimalizować funkcję straty U. Do tego celu wykorzystywany jest algorytm propagacji wstecznej, polegający na aktualizacji wag i progów zaczynając od neuronów warstwy wyjściowej w kierunku warstwy wejściowej, podążając zgodnie z gradientem przyjętej funkcji straty (Równ 3. i 4.).

$$\vec{w} = \vec{w} - \eta * \nabla U(f(net), y) \tag{3}$$

$$b = b - \eta * \frac{\partial U(f(net), y)}{\partial b} \tag{4}$$

Gdzie $\vec{w}$ jest wektorem wag dla danego neuronu, U funkcją straty, y wartością, której neuron wyjściowy ma się nauczyć (na przykład w przypadku klasyfikacji binarnej $y \in 0, 1$) , a η hiperparametrem określającym długość kroku w kierunku przeciwnym do gradientu. Zbyt duży krok może wpłynąć na pominięcie minimum funkcji straty, zaś zbyt mały wymaga bardzo dużej liczby iteracji algorytmu. Funkcja straty określa jak bardzo przewidziane wartości różnią się od wartości prawdziwych dla danego zestawu cech. Przykładową funkcją straty jest binarna entropia krzyżowa, opisana Równ 5:

$$-1/N * \sum_{n=1}^{N} [y_n * log(f(net_n)) + (1 - y_n) * log(1 - f(net_n))] \tag{5}$$

N oznacza liczbę wszystkich egzemplarzy cech, y_n wartość prawdziwą dla danego egzemplarza cech, zaś $f(net_n)$ wartość przewidzianą przez sieć. W pojedynczej iteracji na podstawie cech wejściowych obliczane jest wyjście sieci, następnie korzystając z algorytmu propagacji wstecznej aktualizowane są wagi dla każdego neuronu, według gradientu funkcji straty. Taką iterację nazywamy epoką (*ang. epoch*).

W sytuacji gdy wartość wyjściowa dla danego egzemplarza cech zależy również od egzemplarzy go otaczających (na przykład struktura drugorzędowa zależy od ciągu występujących po sobie aminokwasów w sekwencji), wykorzystuje się sieci konwolucyjne i rekurencyjne na przykład LSTM[9]. Pierwsze pozwalają na znajdowanie zależności między elementami sekwencji mieszczącymi się w oknie o stałej długości, drugie zaś pozwalają określić, które z poprzednich elementów sekwencji mają wpływ na wartość wyjściową dla danego elementu.

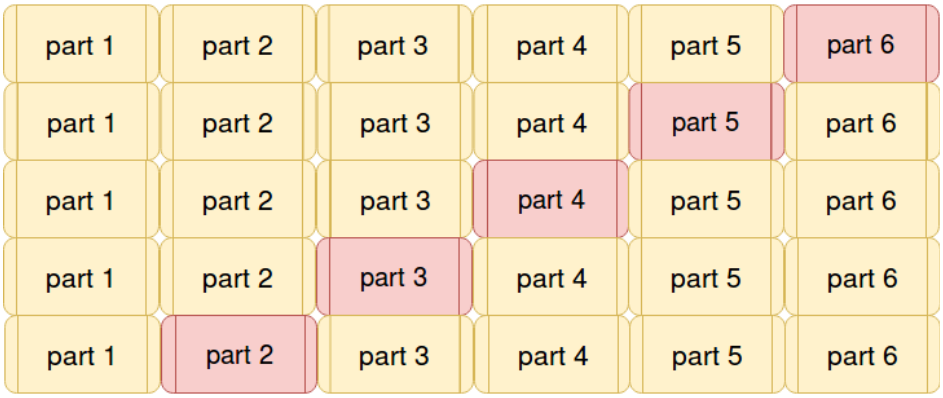
2. Przygotowanie danych i budowa modelu

Zbiór danych stanowiły struktury i odpowiadające im sekwencje pobrane z serwera CullPDB[10]. Są to sekwencje przefiltrowane do 50 % podobieństwa dla danego łańcucha, odpowiadające strukturom białek rozwiązanym z użyciem krystalografii rentgenowskiej (rozdzielczość wyższa niż 3.0 Å, czynnik R mniejszy niż 1.0). Sekwencje zostały dodatkowo przefiltrowane programem cd-hit[11] do 50% podobieństwa. Struktura drugorzędowa dla każdej reszty aminokwasowej została określona z użyciem programu DSSP. Dla każdej sekwencji wygenerowano macierz preferencji aminokwasów dla danej pozycji (PSSM – *ang. position specific scoring matrix*) używając trzech iteracji programu psiblast[12] oraz wartości oczekiwanej E równej 0.001. Wyniki DSSP dotyczące przypisania regionów π-helikalnych zostały poprawione w oparciu o metodę opisaną przez Cooley-a i wsp.[5]. Wszystkie łańcuchy z CullPDB zostały przeskanowane i regiony π-helikalne zostały przypisane, gdy występowały w nich wiązania wodorowe typu π o energii mniejszej niż 0.5 kcal/mol (według DSSP) oraz kąt torsyjny dla wszystkich reszt w tym rejonie mieścił się w zakresie definiującym π-helisy (-180°<φ<0°, -120°< ψ<45°). Ze względu na duże różnice w geometrii w znanych π-helisach, jedna nie π-helikalna reszta otoczona przez π-helikalne regiony była uznawana za część π-helisy. Na końcu, 8-stanowe wyniki DSSP zostały sprowadzone do 4 kategorii (‘H’ – helisa, zawiera etykiety ‘H’ oraz ‘G’;

‘E’, zawiera ‘E’ oraz ‘B’; ‘C’, zawiera ‘S’, ‘T’, ‘C’; oraz ‘I’ – π-helisa, zawiera tylko etykiety ‘I’) - Rysunek 2.1.

Rysunek 2.1 Fragment sekwencji (PDB id: 4yli łańcuch F) i odpowiadająca jej struktura drugorzędowa.

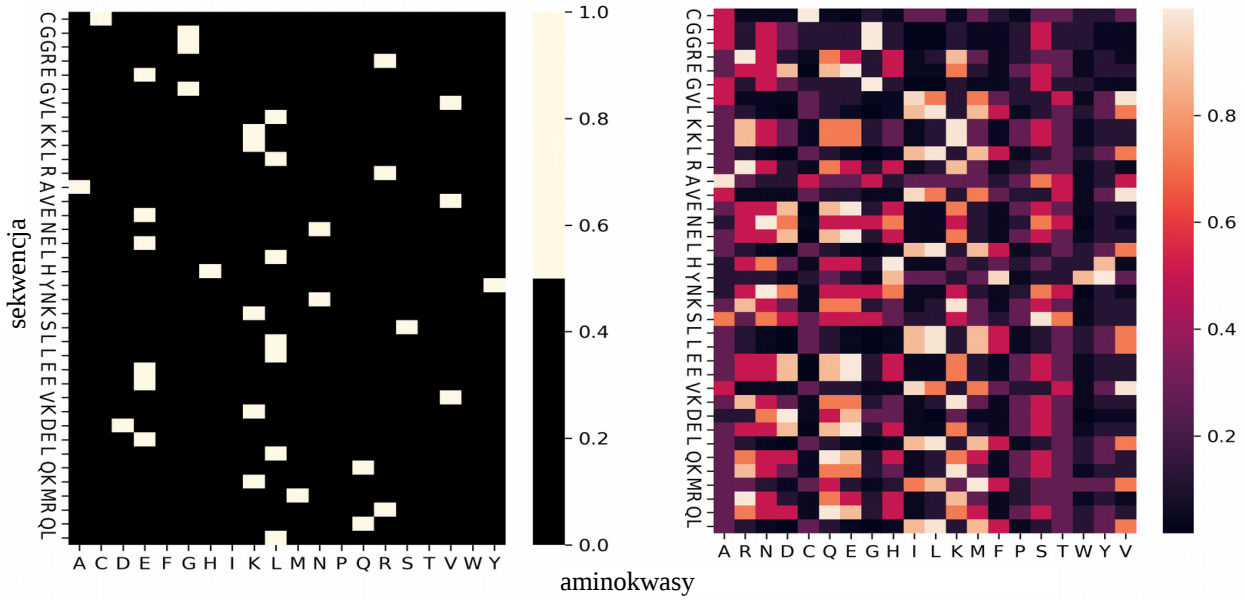
Cały zbiór danych zawierał 22649 sekwencji z odpowiadającymi etykietami struktury drugorzędowej. Wybrano losowo 1215 sekwencji do zbioru testowego zaś pozostałe stanowiły zbiór treningowy. W każdym ze zbiorów reszty π-helikalne stanowiły około 0.85% wszystkich reszt. Zbiór treningowy podzielono na 6 części. Pierwsza złożona z 15584 została wykorzystana wyłącznie do trenowania modelu. Pozostałe to zbiory walidacyjne, wykorzystane w procesie 5 krotnej walidacji krzyżowej, do optymalizacji liczby epok treningu, po których model najlepiej przewiduje regiony π-helikalne. Przebieg walidacji krzyżowej, przedstawia Rysunek 2.2.



Rysunek 2.2. Przebieg walidacji krzyżowej. Część 1 (*ang. part*) składała się z 15584 sekwencji, pozostałe części zawierały po 1170 sekwencji każda. W każdym z 5 przebiegów walidacji (kolejne wiersze na rysunku) model trenowano na częściach oznaczonych na żółto, zaś sprawdzono na częściach oznaczonych na czerwono.

Ponadto zagwarantowano, że każda sekwencja zawierająca π-helisę zarówno w zbiorze testowym jak i zbiorach walidacyjnych była w co najwyżej 30% identyczna z dowolną sekwencją z całego zbioru danych. Zbiór danych składał się z sekwencji o długości od 30 do 700 aminokwasów. Każda z nich została zakodowana jako macierz o wymiarach 700 na 44, gdzie 20 cech stanowiły zakodowane przy użyciu kodu 1 z n (*ang. ‘one-hot encoding’*) aminokwasy, kolejne 20, dane z PSSM przeskalowane do zakresu 0-1 przy użyciu funkcji

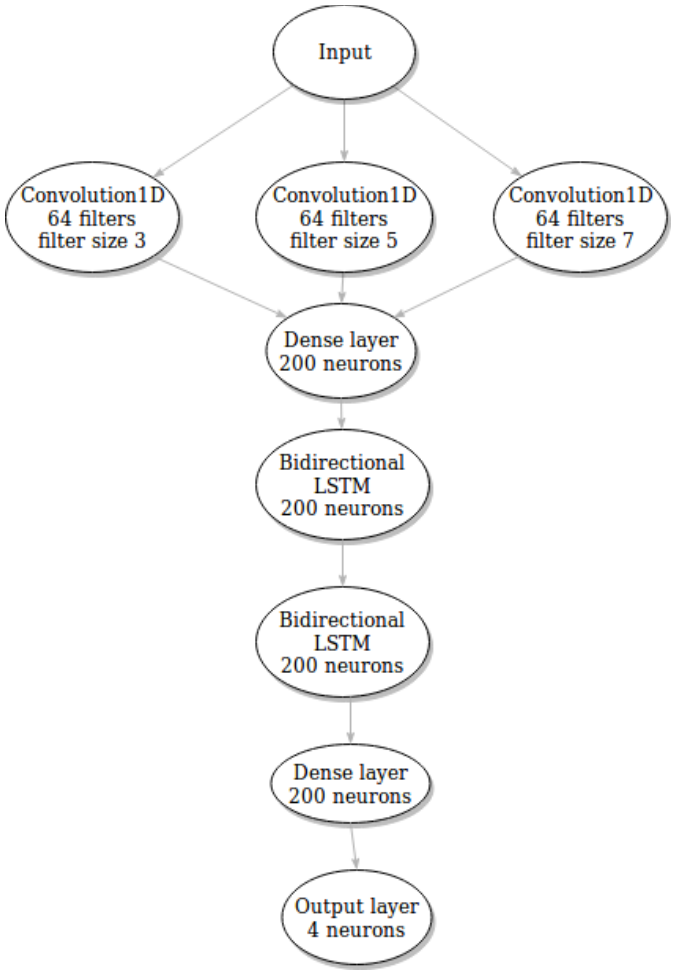
sigmoidalnej, a ostatnie 4 cechy to zakodowane również przy użyciu kodu 1 z n, etykiety struktury drugorzędowej. Jako, że wejście do sieci musi mieć określony rozmiar, wiersze macierzy dla sekwencji krótszych niż 700 uzupełniono zerami. Graficzna prezentacja omówionego sposobu kodowania (bez wierszy z samymi zerami), została przedstawiona dla przykładowej sekwencji długości 39 aminokwasów, na Rysunku 2.1.



Rysunek2.1. Przykład zakodowanej sekwencji i odpowiadająca jej macierz PSSM

Zakodowane sekwencje stanowiły wejście do głębokiej sieci neuronowej. Architekturę sieci przedstawiono na Rysunku 2.2. Użyto trzech jednowymiarowych warstw konwolucyjnych zawierających 64 neurony oraz wykorzystujących okna o rozmiarach odpowiednio 3, 5 oraz 7. Jako funkcję aktywacji wejścia wykorzystano tangens hiperboliczny. Wyjście z każdej z tych warstw stanowiła macierz rozmiarów 700 x 64. Tak więc ostatecznie wyjściem z warstw konwolucyjnych były 3 macierze, które połączono według wierszy otrzymując macierz rozmiarów 700 x 192. Wykorzystano następnie warstwę gęstą złożoną z 200 neuronów, z których każdy korzystał z wyjścia warstwy poprzedniej (macierz 700 x 192). Kolejną częścią sieci były dwie dwukierunkowe (przetwarzające sekwencję od początku do końca i odwrotnie) warstwy LSTM złożone z 200 neuronów każda. Ostatnie dwie warstwy to warstwy gęste złożone odpowiednio z 200 i 4 neuronów. Hiperparametry, takie jak liczby neuronów czy funkcje aktywacji dobrano na podstawie trenowania i walidacji modelu, ze względu na koszt czasowy obliczeń, na niewielkich podzbiorach (około 1000) sekwencji, dla różnych zestawów parametrów i wybraniu tych, dla których model osiągnął najlepszy wynik F1 w walidacji.

Warstwa wyjściowa aktywowana funkcją softmax pozwoliła zwrócić dla każdej reszty prawdopodobieństwo utworzenia przez nią każdej z 4 struktur drugorzędowych.



Rysunek 2.2 Architektura sieci. Warstwa wejściowa, następnie 3 warstwy konwolucyjne z 64 filtrami o rozmiarach odpowiednio 3, 5 i 7, następnie warstwa gęsta, 2 dwukierunkowe warstwy LSTM oraz warstwa gęsta i warstwa wyjściowa.

Taka architektura umożliwiła rozpoznanie zależności między kilkoma sąsiednimi aminokwasami (warstwy konwolucyjne), a także między oddalonymi od siebie resztami (warstwy LSTM), wpływającymi na utworzenie danej struktury drugorzędowej. Warstwy gęste pozwoliły natomiast na klasyfikację aminokwasów na podstawie cech wyprodukowanych przez warstwy konwolucyjne i LSTM. Do trenowania sieci wykorzystano algorytm optymalizacji Adam[13] ze stałą uczenia 0.003. Funkcją straty byłą ważona entropia krzyżowa, z wagami dla etykiet obliczonymi jako $max(log(n_i/n), 1)$ gdzie n_i to liczba reszt z danej kategorii (H,E,I,C) zaś n to liczba wszystkich reszt. Sieć trenowano przez 50 epok dla każdej walidacji, zapisując wagi gdy metryka F1 (średnia harmoniczna ze skuteczności i precyzji) dla π-helisy mierzona na zbiorze walidacyjnym, uległa poprawie. Skuteczność to stosunek liczby reszt

π-helikalnych prawidłowo przewidzianych do liczby wszystkich reszt π-helikalnych w zbiorze, zaś precyzja to stosunek liczby reszt π-helikalnych prawidłowo przewidzianych do liczby wszystkich przewidzianych reszt π-helikalnych. Architektura sieci została zaimplementowana w module do języka Python o nazwie Keras[14].

3. Wyniki

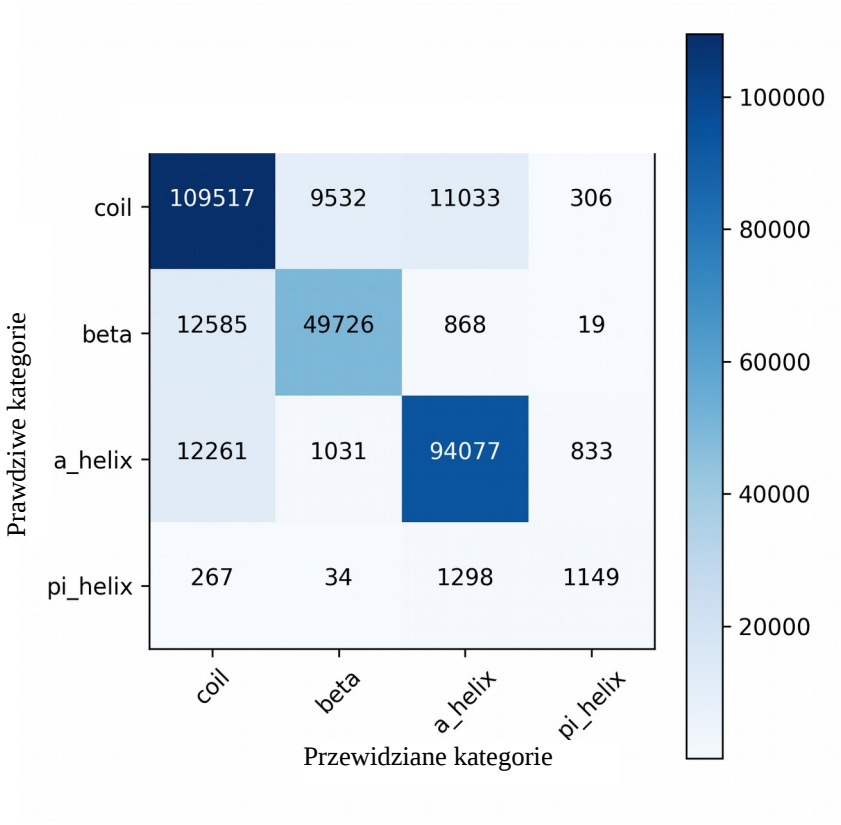
Wyniki 5-krotnej walidacji krzyżowej przedstawione są w Tabeli 3.1

Tabela 3.1. Skuteczność i precyzja dla najlepszej epoki każdej z 5 części walidacji.

Nr walidacji	Nr epoki	Skuteczność	Precyzja	F1
1	46	0.50	0.46	0.48
2	24	0.47	0.48	0.47
3	35	0.49	0.49	0.49
4	30	0.41	0.43	0.42
5	33	0.44	0.46	0.45

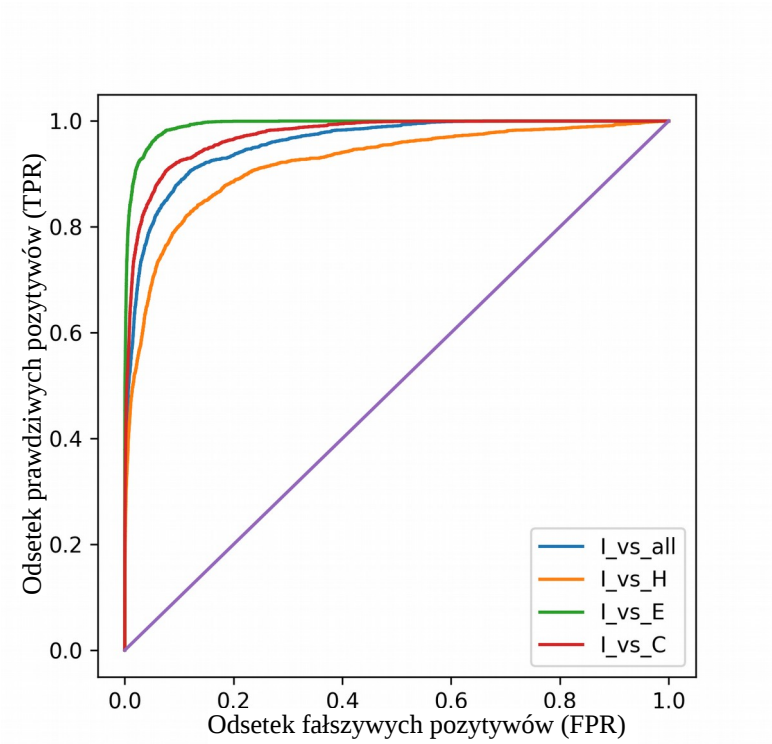
Wyniki dla każdej części są podobne. Zarówno precyzja jak i skuteczność wyniosły średnio 46%. Ostateczny predyktor został zbudowany z zestawów wag uzyskanych dla 3 przebiegów walidacji z najlepszymi wartościami metryki F1. Prawdopodobieństwo dla każdej z 4 kategorii jest średnią z 3 prawdopodobieństw, przewidzianych przez model, używając kolejno każdego z 3 zestawów wag. Pozwoliło to na uzyskanie stabilnego predyktora dającego zbliżone rezultaty dla niepodobnych do siebie sekwencji zawierających π-helisy. Tak przygotowany model został sprawdzony na zbiorze testowym, zawierającym 1215 sekwencji nieużywanych w trenowaniu modelu. 306 z nich zawiera co najmniej jedną π-helisę (w sumie 406 π-helis). PiPred wykazał 50% precyzji oraz 42% skuteczności, mierzonych dla każdej reszty aminokwasowej. Macierz pomyłek dla wszystkich kategorii przedstawia Rysunek 3.1.

PiPred najczęściej myli ze sobą α- i π-helisy, na co wpływa powiązanie ewolucyjne (π-helisy często powstają z α-helis przez insercje, delecje lub substytucje aminokwasu) tych struktur, a także bardzo duża częstość występowania α-helis.



Rysunek 3.1. Macierz pomyłek dla wszystkich kategorii struktury drugorzędowej.

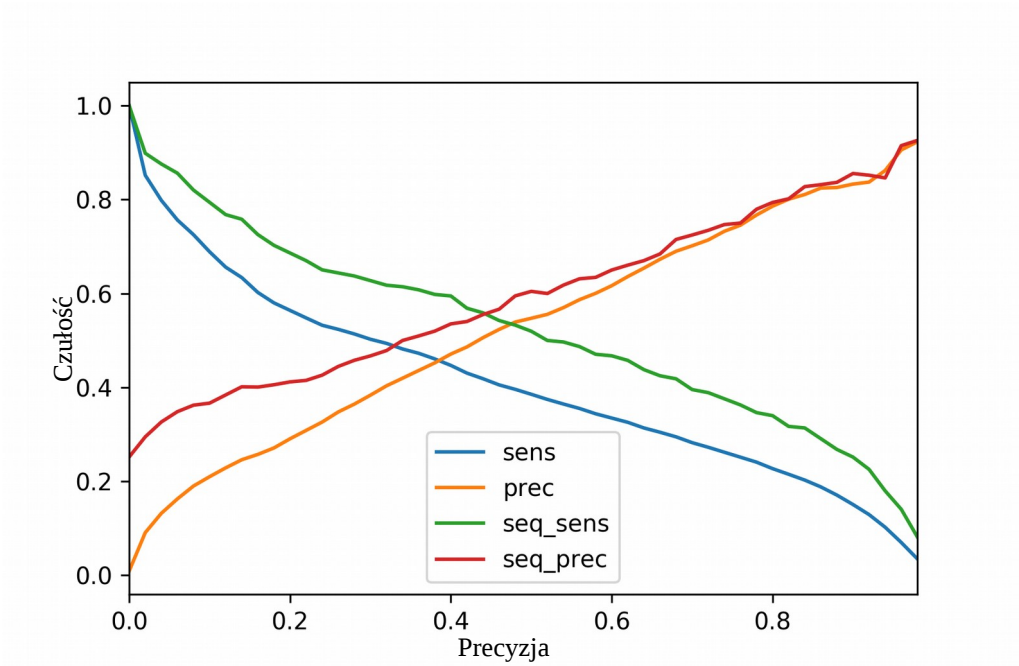
Biorąc pod uwagę częstość występowania π-helis (0.85% wszystkich reszt), PiPred osiąga dobre rezultaty, co potwierdza krzywa ROC ang. Receiver Operating Characteristic (wykres stosunków fałszywie pozytywnych i prawdziwie pozytywnych przewidywań) przedstawiona na Rysunku 3.2. Losowemu przewidywaniu odpowiada prosta i pole (AUC ang. Area Under Curve) pod nią równe 0.5, zaś idealnemu klasyfikatorowi AUC równy 1. Wyniki predyktora przy odróżnianiu π-helis od pozostałych kategorii (H, E, C) określone jako pola pod krzywymi to w kolejności tych kategorii 0.91, 0.96, 0.95. Dla pozostałych kategorii łącznie, pole to wynosi 0.92.



Rysunek 3.2. Krzywa ROC dla odróżniania π-helis od pozostałych kategorii (I_vs_all - stosunek FPR do TPR przy odróżnianiu reszt π-helikalnych od pozostałych, I_vs_H – od α-helis, I_vs_E – od β-kartek, I_vs_C – od fragmentów niesutrukturyzowanych)

Wszystkie przedstawione do tej pory wyniki odnoszą się do reszt aminokwasowych. Jeśli natomiast rozpatrzymy regiony π-helikalne i uznamy za prawidłowo przewidziany region z co najmniej jedną dobrze przewidzianą resztą to skuteczność i precyzja wynoszą odpowiednio 38% i 59%.

W celu określenia zachowania skuteczności i precyzji zarówno dla reszt jak i segmentów wykonano wykres zależności tych metryk od prawdopodobieństwa, przy którym dana reszta jest uznawana jako część π-helisy.
Wykres jest punktem odniesienia, w jakim stopniu można być pewnym wyników predyktora. Precyzja przy akceptowaniu reszt z prawdopodobieństwem ponad 90% wynosi około 90% zaś skuteczność około 15%. Predyktor jest w stanie przewidywać π-helisy w różnych kontekstach strukturalnych (w środku, na N- i C- końcu α-helis).



Rysunek 3.3. Wykres wartości metryk od prawdopodobieństwa akceptacji reszty jako π-helikalna (sens – skuteczność dla reszty, prec – precyzja dla reszty, seq_sens – skuteczność dla segmentów, seq_prec – precyzja dla fragmentów).

4. Wnioski

Predyktor osiągnął dobre wyniki (42% skuteczności, 50% precyzji), biorąc pod uwagę liczność π-helis (0.85% wszystkich reszt). Związane jest to między innymi z charakterystycznym składem aminokwasowym i kolejnością występowania danych reszt w tych regionach, czego sieć neuronowa (warstwy konwolucyjne) nauczyła się w wyniku trenowania na dużym zbiorze sekwencji. Biorąc pod uwagę wyniki predyktora i rolę π-helis w strukturze białek, może on być narzędziem służącym do przypisywania miejsc aktywnych i pomagającym w modelowaniu białek błonowych. Może być wykorzystany również przy projektowaniu sekwencji, czyli budowaniu sekwencji, które zwijają się do zadanej struktury przestrzennej.

5. Bibliografia

1. Kabsch, W. & Sander, C. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* **22,** 2577–2637 (1983).
2. Weaver, T. M. The pi-helix translates structure into function. *Protein Sci.* **9,** 201–6 (2000).
3. Fodje, M. N. & Al-Karadaghi, S. Occurrence, conformational features and amino acid propensities for the π-helix. *Protein Eng. Des. Sel.* **15,** 353–358 (2002).
4. Tikhonov, D. B. & Zhorov, B. S. Conservation and variability of the pore-lining helices in P-loop channels. *Channels* **11,** 660–672 (2017).
5. Cooley, R. B., Arp, D. J. & Karplus, P. A. Evolutionary Origin of a Secondary Structure: α-Helices as Cryptic but Widespread Insertional Variations of α-Helices That Enhance Protein Functionality. *J. Mol. Biol.* **404,** 232–246 (2010).
6. Frishman, D. & Argos, P. Knowledge-based protein secondary structure assignment. *Proteins Struct. Funct. Bioinforma.* (1995). doi:10.1002/prot.340230412
7. Wang, S., Peng, J., Ma, J. & Xu, J. Protein Secondary Structure Prediction Using Deep Convolutional Neural Fields. *Sci. Rep.* **6,** 18962 (2016).
8. Li, Z. & Yu, Y. Protein Secondary Structure Prediction Using Cascaded Convolutional and Recurrent Neural Networks. in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence* 2560–2567 (2016).
9. Hochreiter, S. & Schmidhuber, J. J. LSTM. *Neural Comput.* **9,** 1–32 (1997).
10. Wang, G. & Dunbrack, R. L. PISCES: A protein sequence culling server. *Bioinformatics* **19,** 1589–1591 (2003).
11. Li, W. & Godzik, A. Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* **22,** 1658–1659 (2006).
12. Altschul, S. F. *et al.* Gapped BLAST and PSI-BLAST:a new generation of protein database search programs. *Nucleic Acids Res.* (1997). doi:10.1093/nar/25.17.3389

13. Kingma, D. P. & Ba, J. L. Adam optimizer. *arXiv Prepr. arXiv1412.6980* 1–15 (2014). doi:http://doi.acm.org.ezproxy.lib.ucf.edu/10.1145/1830483.1830503

14. Chollet, F. Keras Documentation. *Keras.Io* (2015).