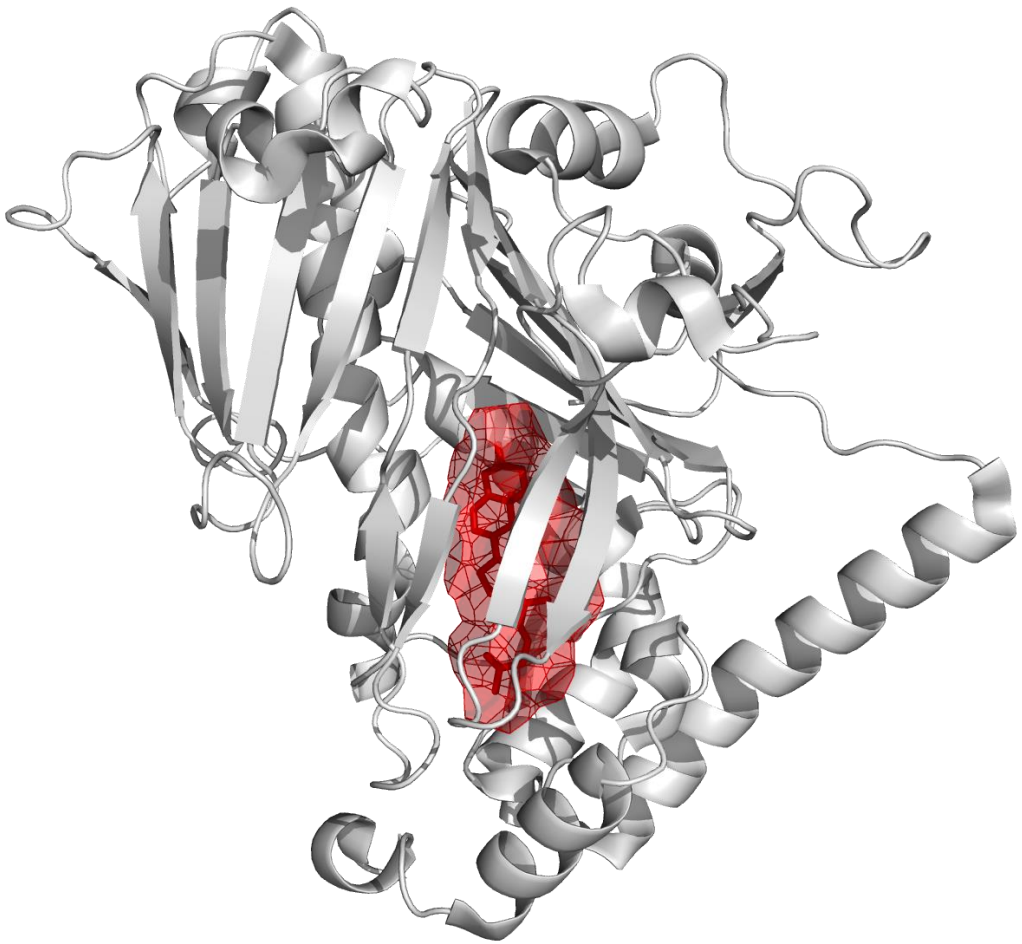
Nauczanie maszynowe – projekt zaliczeniowy

Autor: Michał Szczygieł

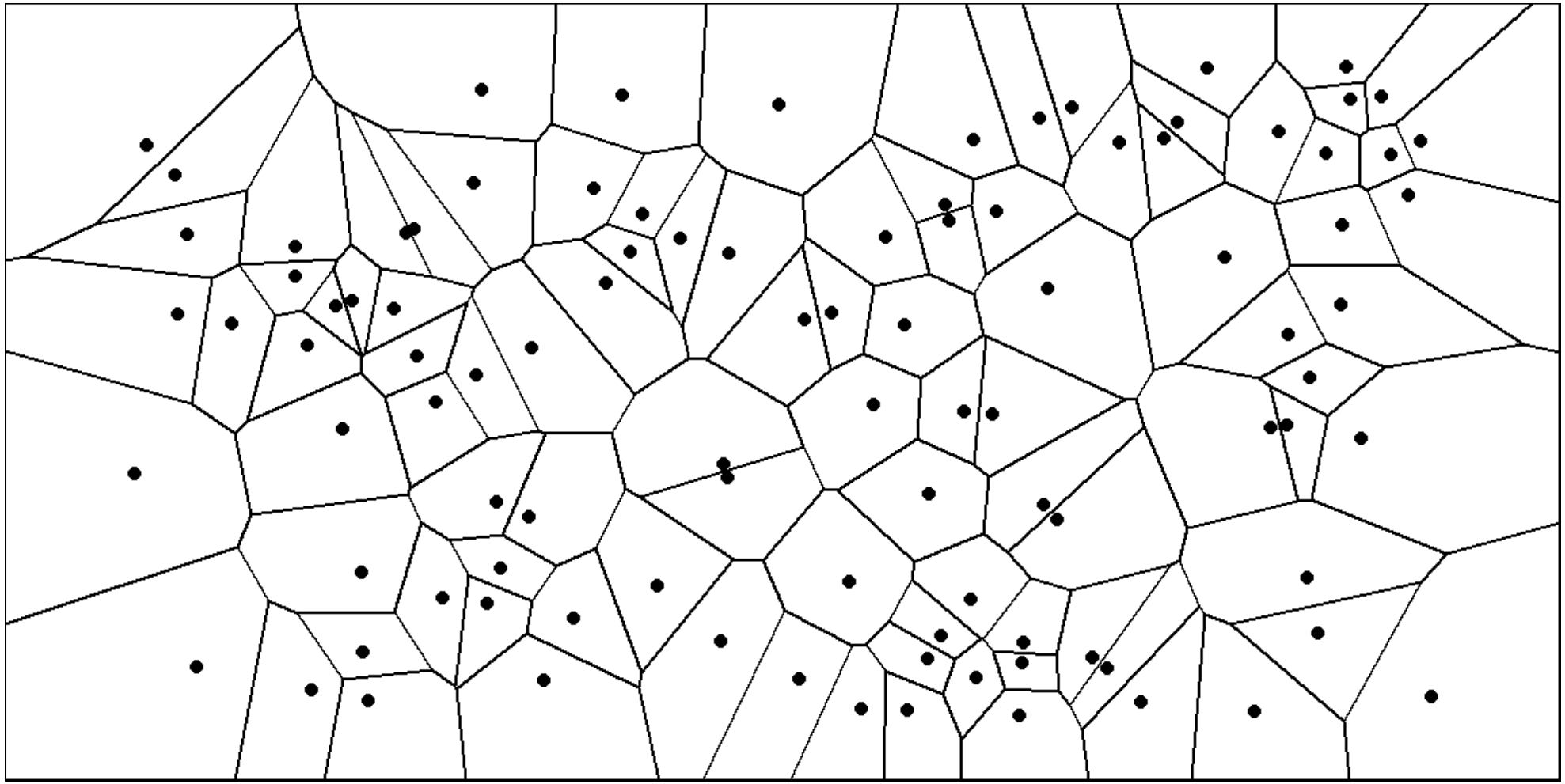
Grupowanie miejsc wiążących ligandy:

- Reprezentacja struktur cząsteczek białek ze związanymi ligandami (z PDB):
  - Struktury zostały pobrane z bazy danych RCSB PDB,
  - Brano pod uwagę jedynie warianty ludzkie (narzuca to rozsądne ograniczenie na rozmiar danych)
- Reprezentacja miejsc wiązania ligandów,
- Redukcja wymiarowości - **PCA**,
- Grupowanie - **KMeans**

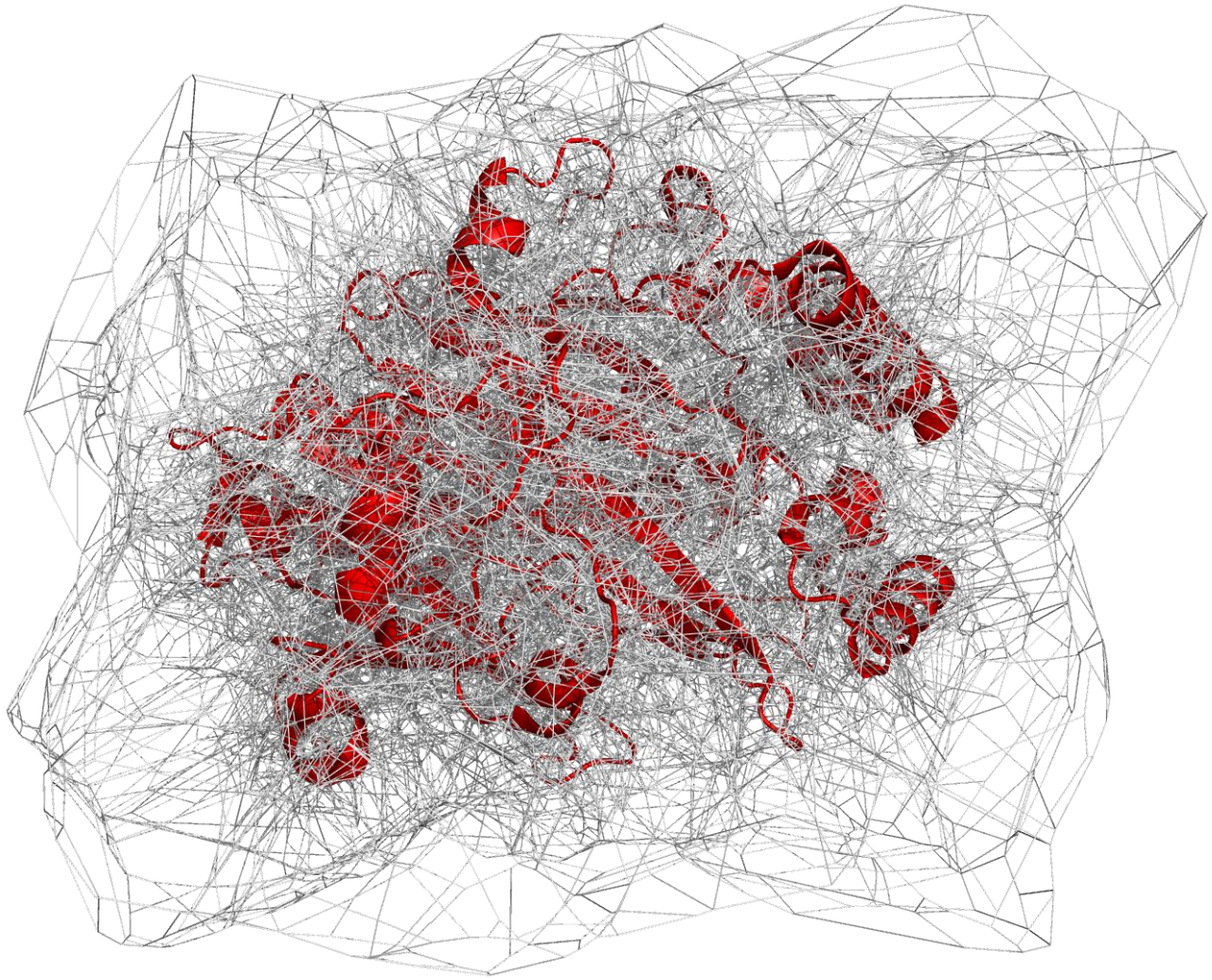
Jak reprezentować struktury białek?



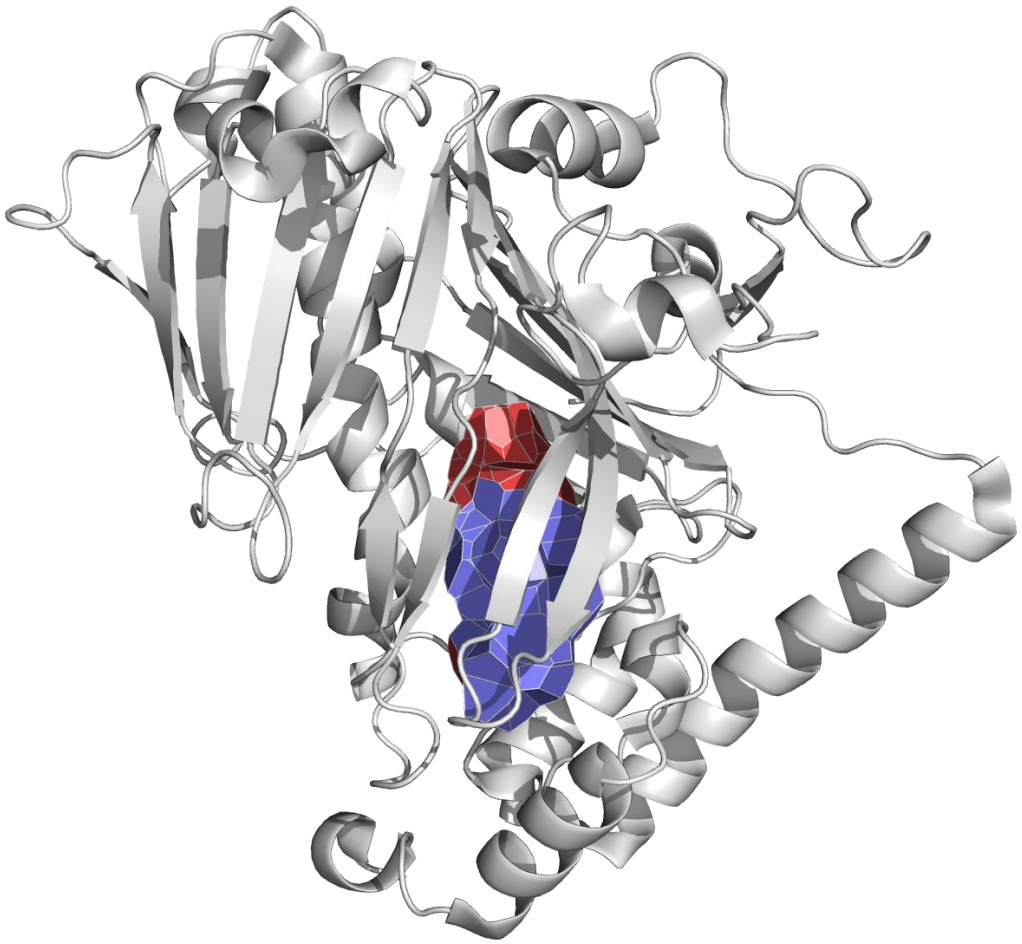
Diagramy Woronoja:

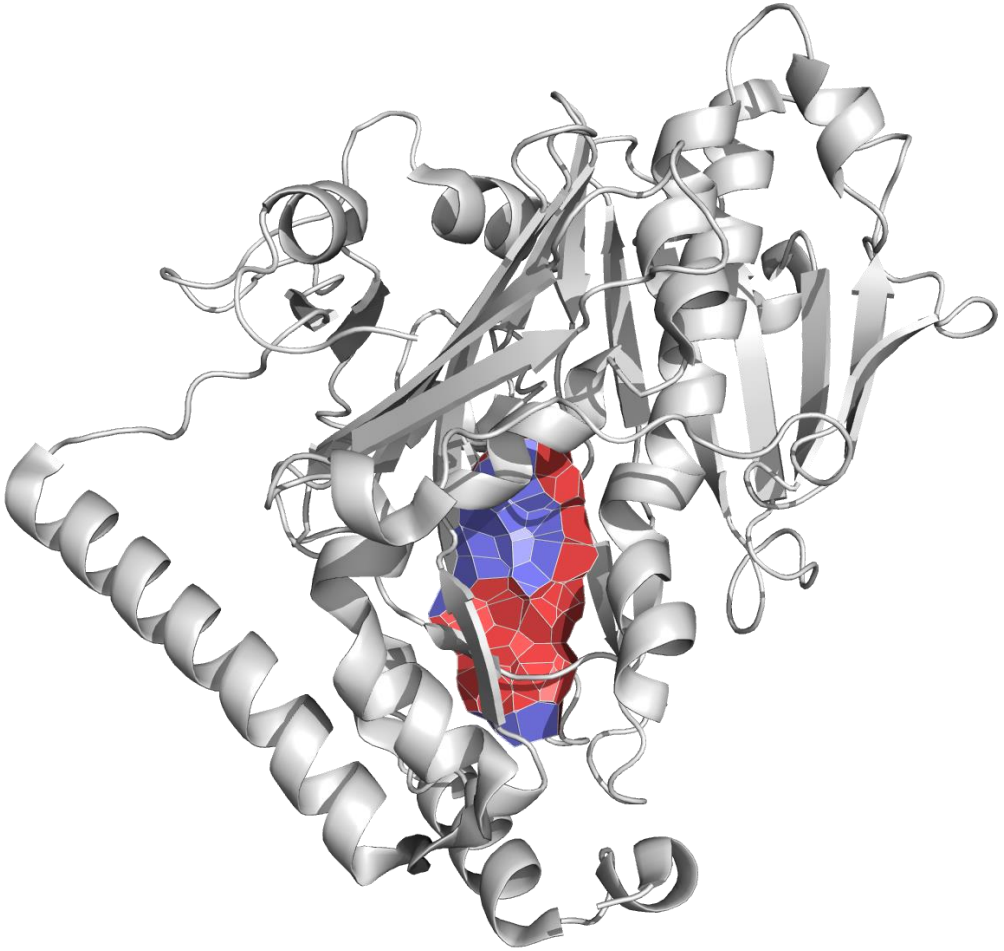


Diagramy Woronoja w 3D:



Reprezentacja przy pomocy DW jest prosta i praktyczna:





Matematyczna reprezentacja miejsc wiązania ligandów:

- Cząsteczka liganda stanowi pewien zbiór atomów,
- Każdemu atomowi liganda można przypisać wektor reprezentujący **>w jakiś sposób<** otoczenie tego atomu,
- Następnie można złożyć wektory od poszczególnych atomów w jeden duży wektor reprezentujący całe miejsce wiązania danego liganda.

W ten sposób przeprowadzana jest wektoryzacja miejsc wiązania ligandów.

Jakie wektory przypisywać atomom ligandów?

Można przypisać np. taki wektor 4D:

V = [masa, objętość, hydrofobowość, punkt izoelektryczny]

Wszystkie powyższe wartości dotyczą sąsiadującego aminokwasu. Jeśli sąsiadujących aminokwasów jest więcej to można wziąć średnią z wektorów od każdego aminokwasu.

Reprezentacja – ciąg dalszy:

Do tej pory opis sąsiedztwa ligandów zawierał informacje o fizykochemii (reszt aminokwasowych). Brak jednak informacji związanych z „przestrzennością”. Aby osiągnąć namiastkę tej „przestrzenności” można uzupełnić wektory embeddingowe o dodatkowy wymiar reprezentujący odległość. Idąc dalej, w przypadku sąsiedztwa wielu aminokwasów, zamiast liczyć średnią arytmetyczną wektorów można policzyć ich średnią ważoną przez odległości.

Redukcja wymiarowości:

W pokazanej reprezentacji każde miejsce wiązania liganda jest dane jako długi wektor liczb zmiennoprzecinkowych. Przykładowo, dla cholesterolu miejsce wiązania reprezentowane jest jako wektor 112 wymiarowy.

Teraz, można wykonać redukcję wymiarowości aby wyciągnąć z tych danych informacje w formie skondensowanej.

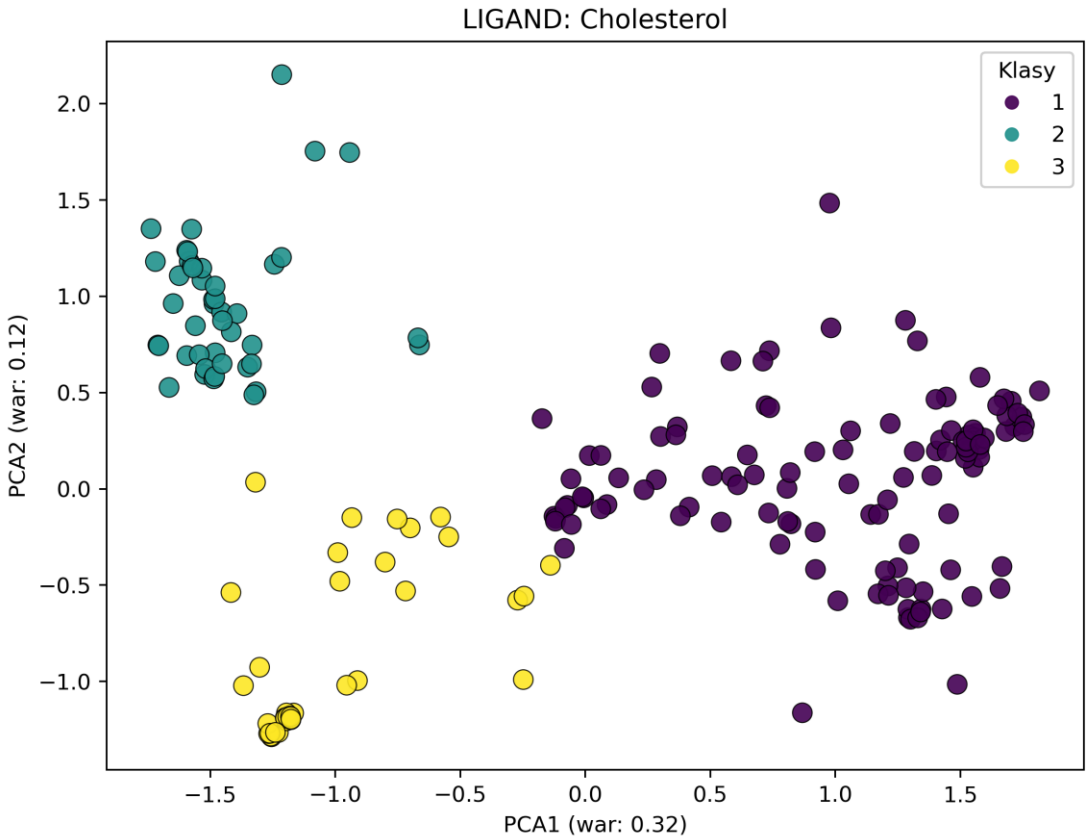
Do tego celu dobrze nadaje się algorytm PCA (wymiarowość redukowano przy zachowaniu >80% wariancji).

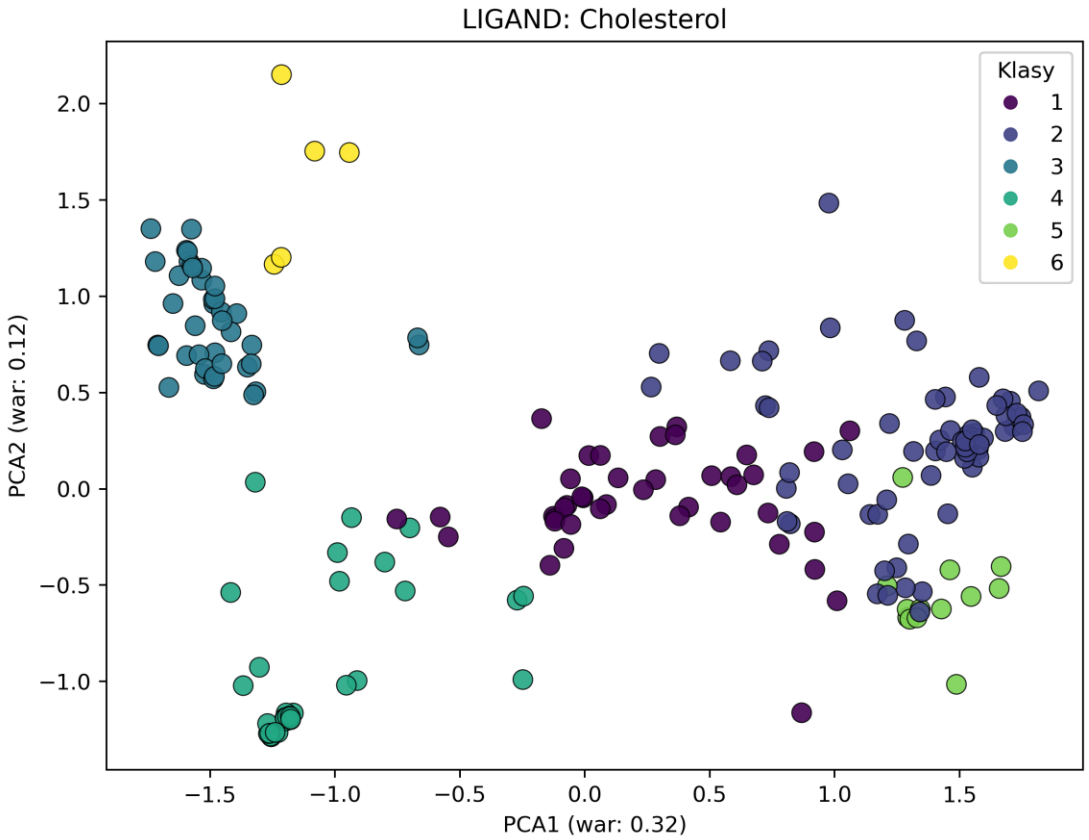
Grupowanie miejsc wiążących:

Po wykonaniu redukcji wymiarowości okazuje się, że poszczególne miejsca wiązania można z powodzeniem opisać ~10 wymiarowymi wektorami. Co więcej, początkowe 2 wymiary w takiej reprezentacji dość dobrze separują grupy – możliwa jest wizualizacja skupisk miejsc wiązania na płaszczyźnie.

Można również wykonać grupowanie w przestrzeni zredukowanej przy pomocy algorytmu KMeans – tutaj nastąpi automatyczne przypisanie klas, przydatne do innych analiz.

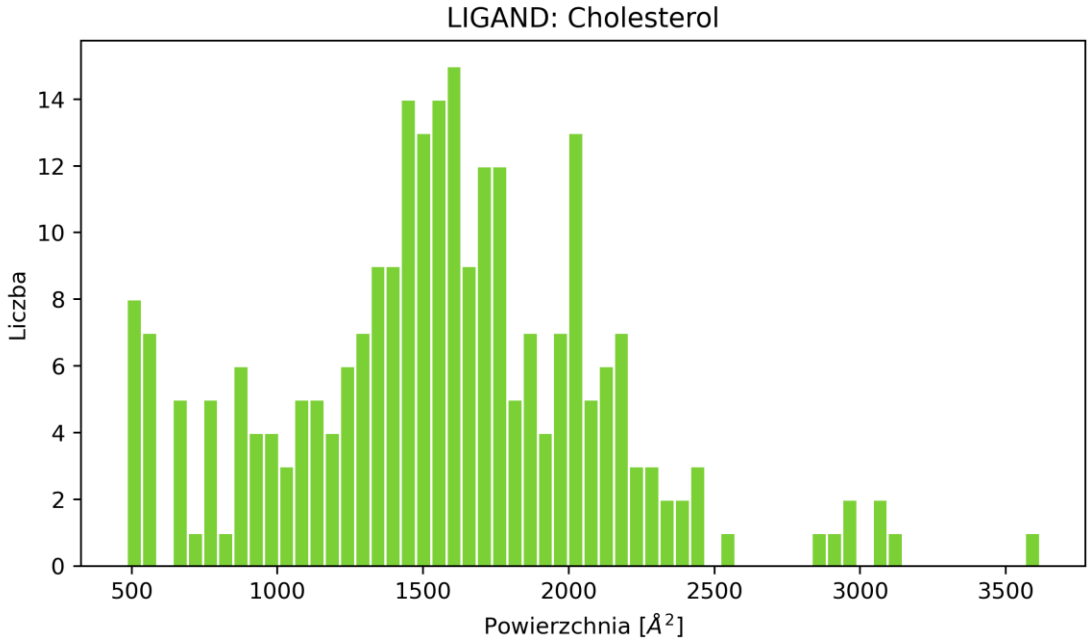
Wyniki (CLR – redukcja do 12D, var: 81.7%):

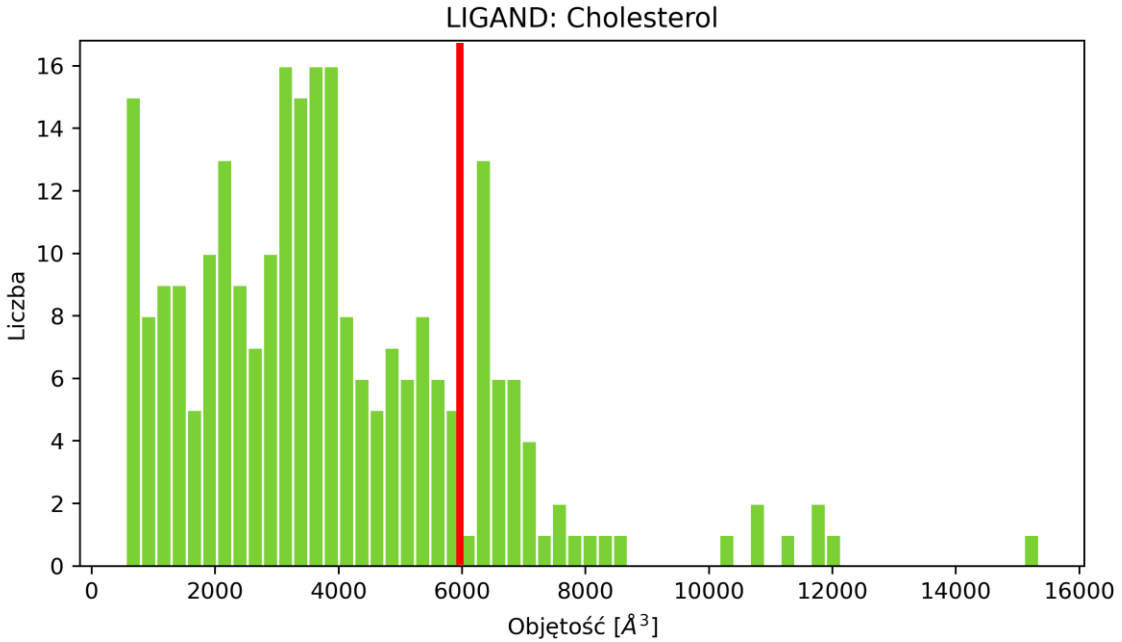




Grupowanie 209 miejsc wiązania CLR (ze 170 struktur)

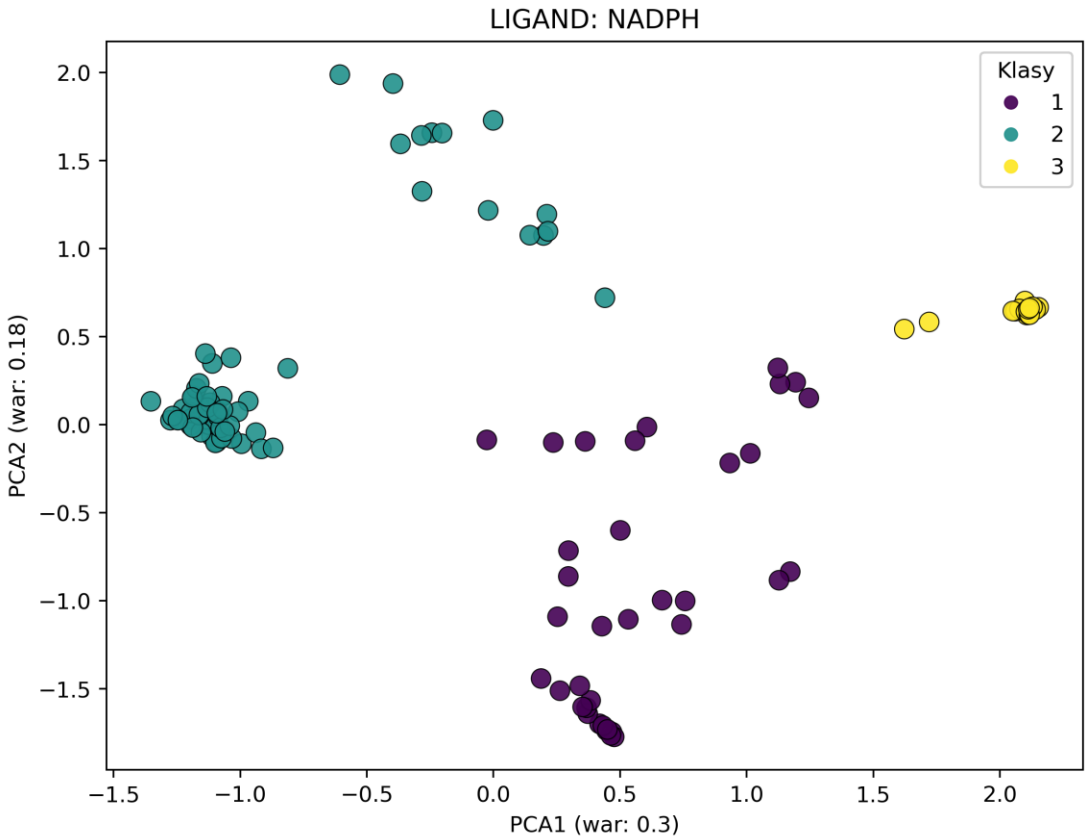
Wyniki (CLR, powierzchnie i objętości domen):

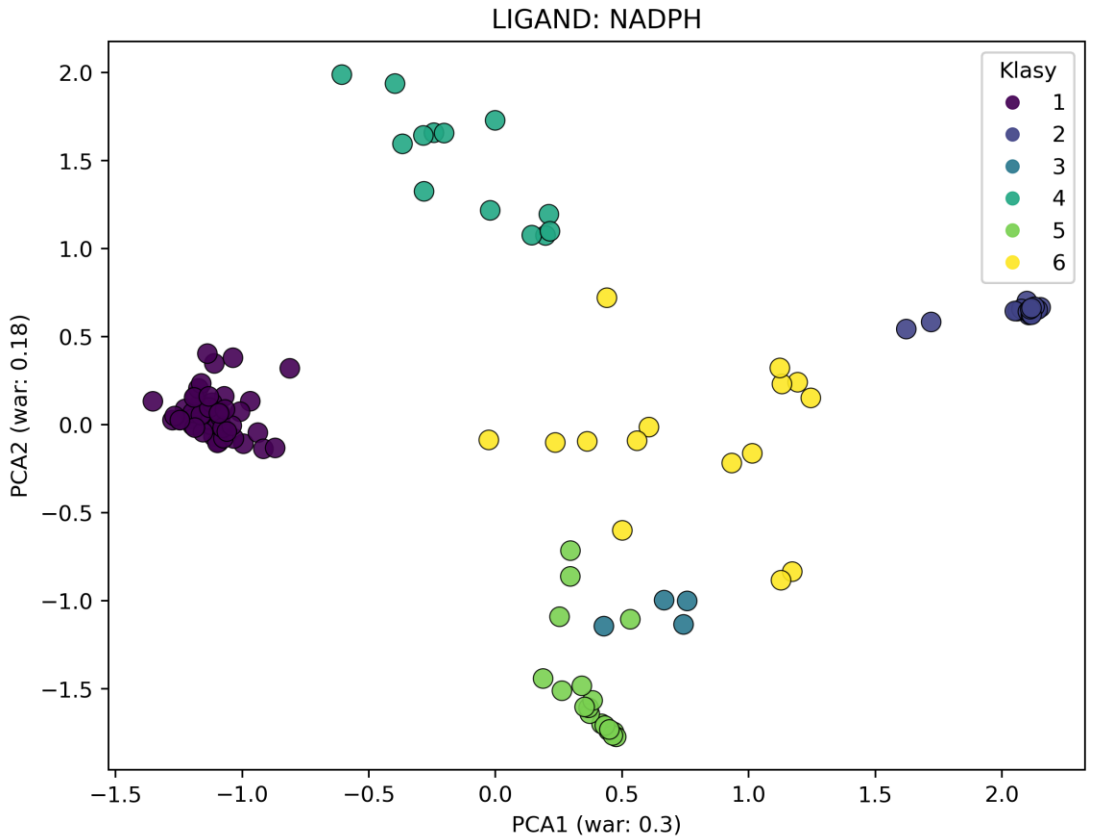




Całkowita liczba znalezionych miejsc wiązania: 254

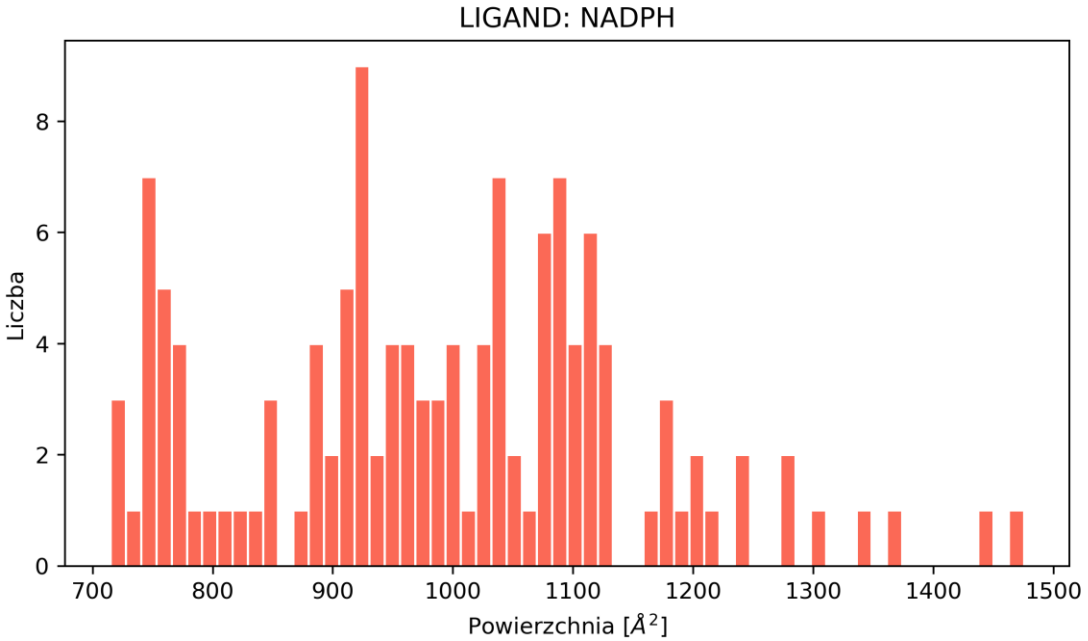
Wyniki (NADPH – redukcja do 9D, var: 80.9%):

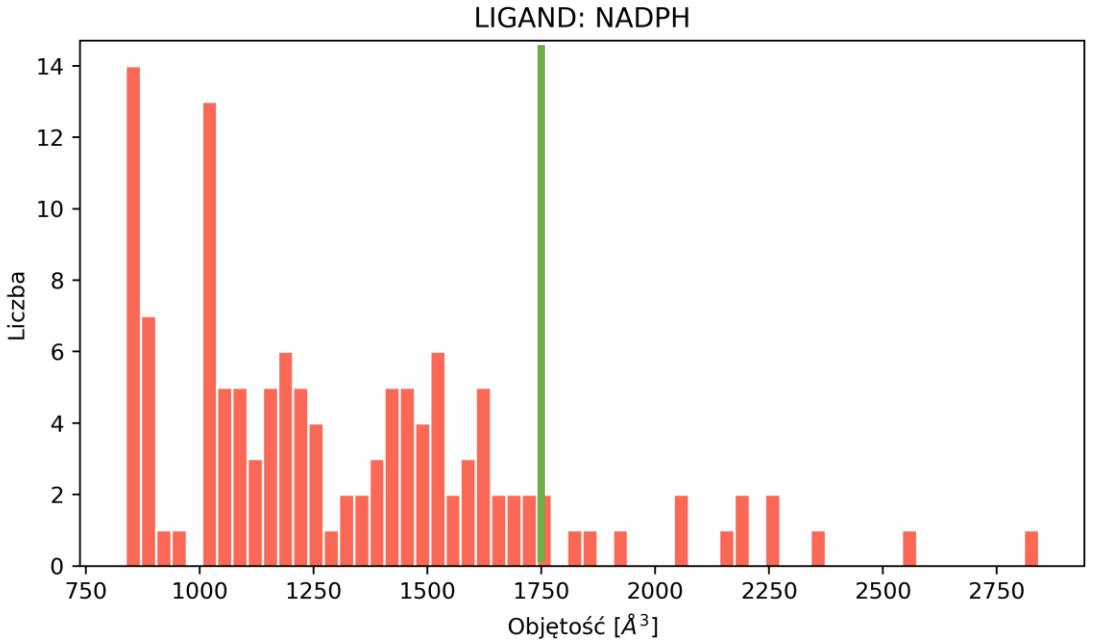




Grupowanie 113 miejsc wiązania NADPH (ze 140 struktur)

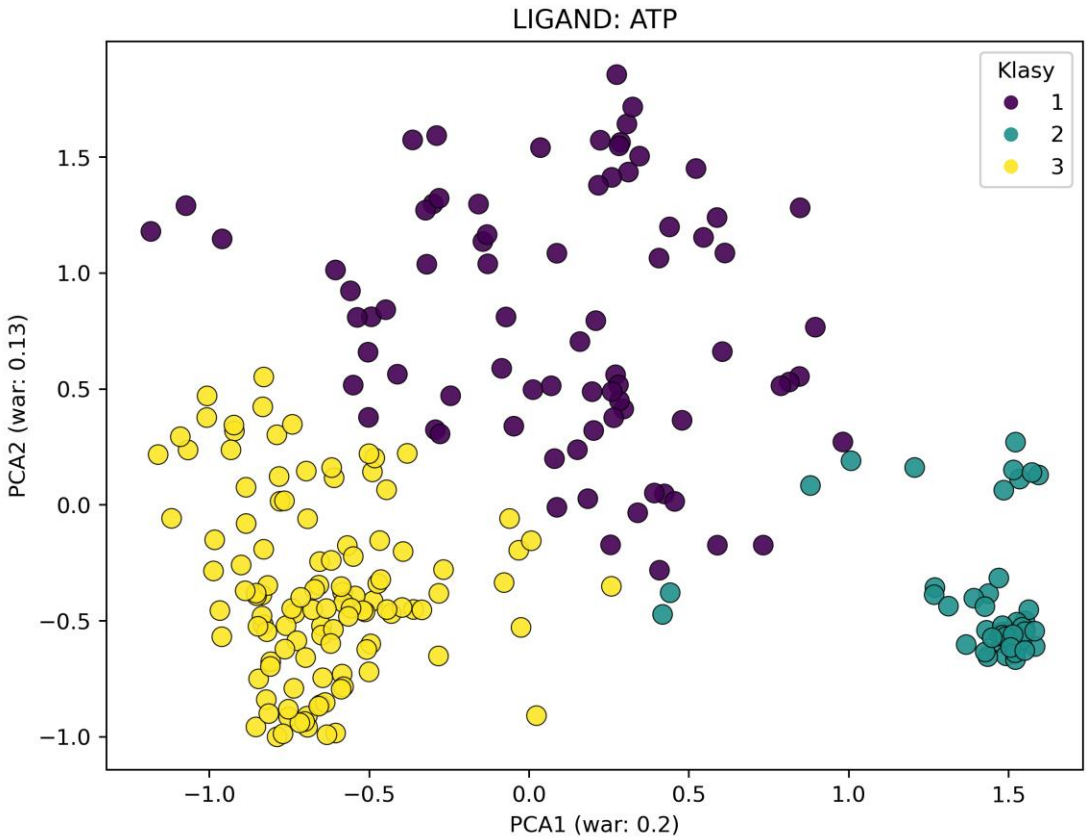
Wyniki (CLR, powierzchnie i objętości domen):

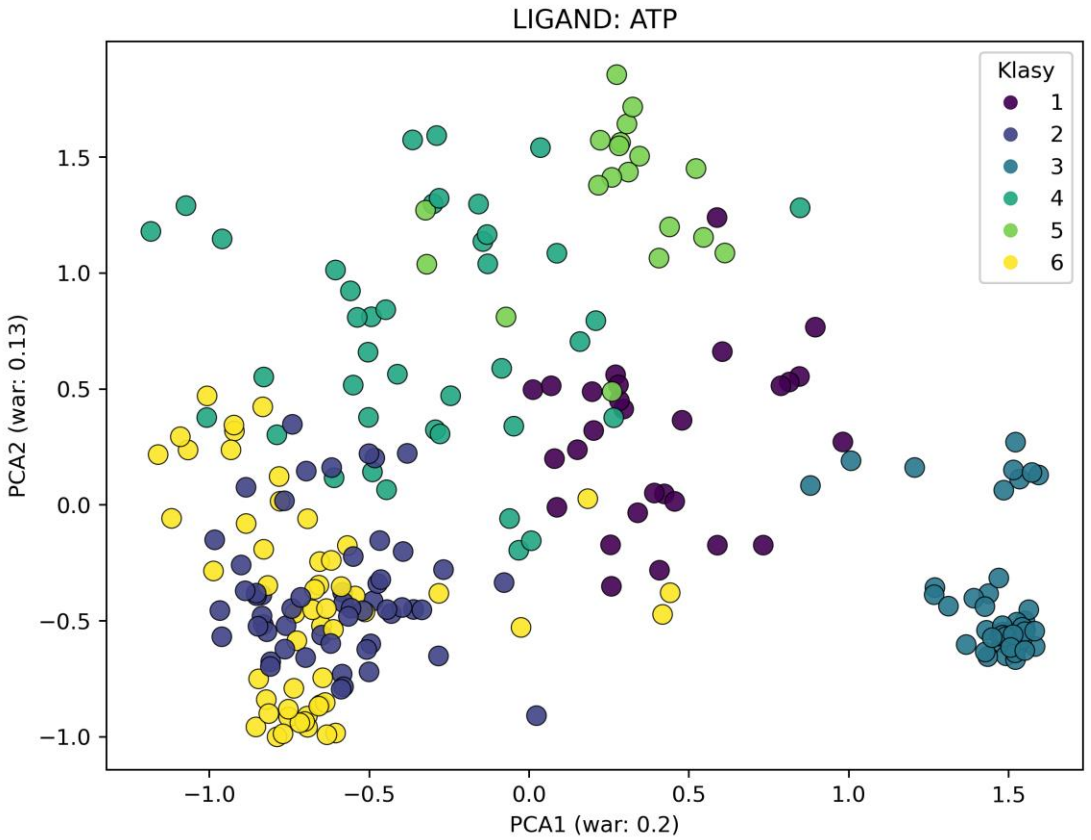




Całkowita liczba znalezionych miejsc wiązania: 128

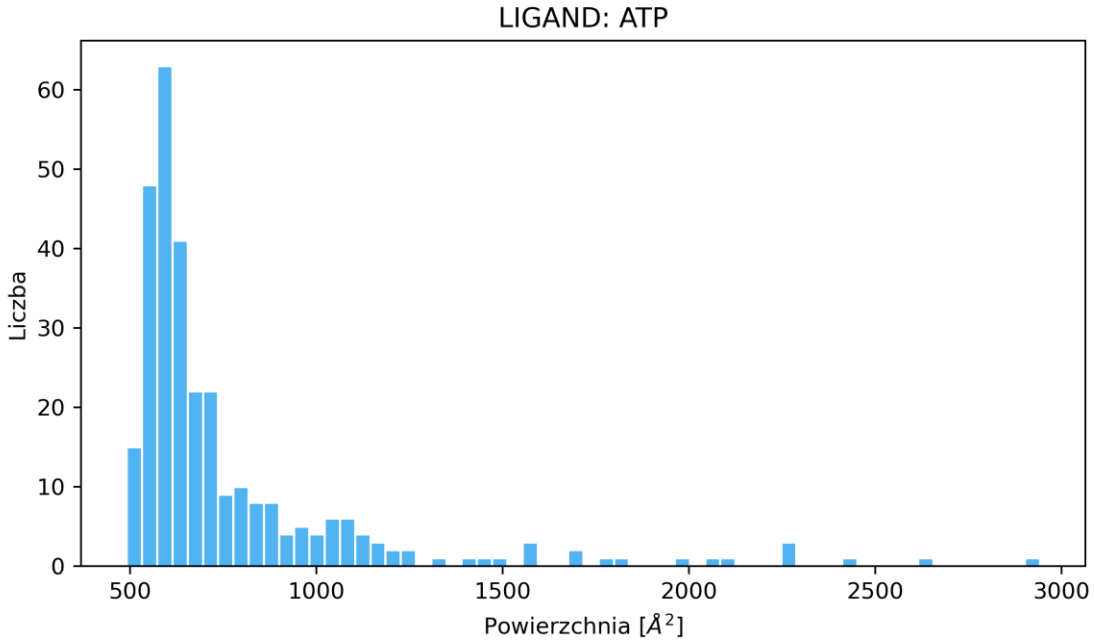
Wyniki (ATP – redukcja do 17D, var: 80.8%):

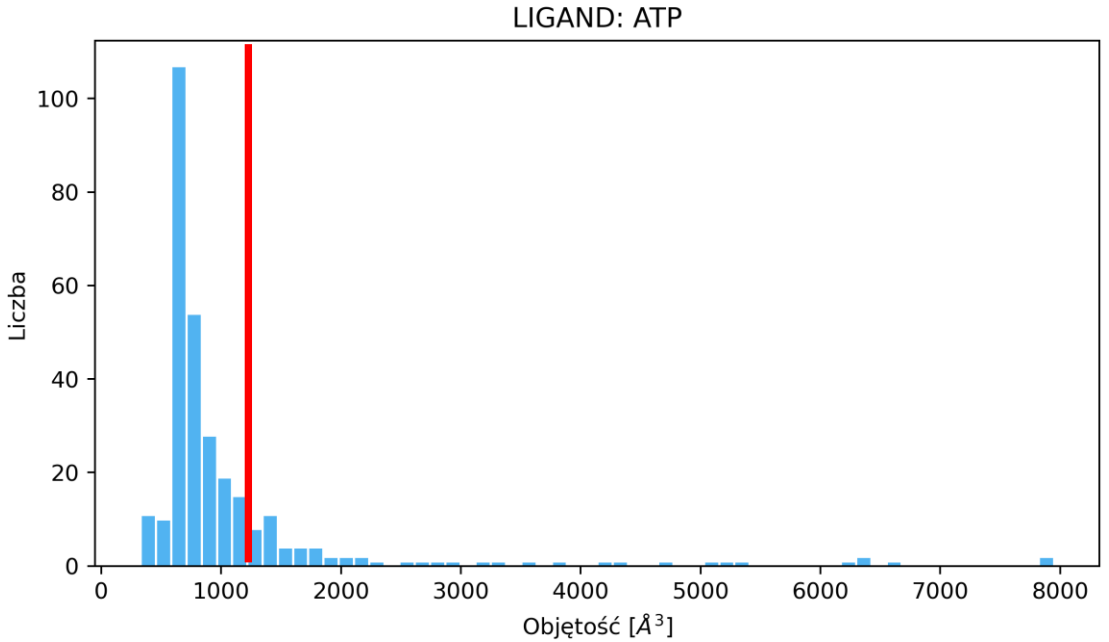




Grupowanie 242 miejsc wiązania ATP (z 307 struktur)

Wyniki (ATP, powierzchnie i objętości domen):





Całkowita liczba znalezionych miejsc wiązania: 302

Wnioski:

- Wektoryzacja miejsc wiązania jest skutecznym sposobem ich reprezentacji,
- Możliwe było przeprowadzenie grupowania miejsc wiążących w obrębie pojedynczych ligandów.
- Ograniczeniem pokazanej tutaj reprezentacji jest to, że nie dopuszcza ona jednoczesnego udziału różnych typów ligandów.

Predykcja ligandu na podstawie kieszeni wiążącej:

- Reprezentacja kieszeni wiążących – jeszcze raz. Poprzednia reprezentacja się nie nadaje ze względu na to, że teraz trzeba zakodować miejsca wiązania dla różnych ligandów,
- Metoda: wielowarstwowe modele neuronowe – klasyfikatory

Reprezentacja One-Hot:

W tej reprezentacji miejsce wiążące kodowane jest jako 20 wymiarowy wektor zer i jedynek na pozycjach korespondujących z określonymi aminokwasami. Innymi słowy jest to forma zapisu obecności lub braku konkretnego aminokwasu w otoczeniu liganda. Jest to słaba reprezentacja – nie rozróżnia ligandów dla których kieszenie wiążące mają podobny skład oraz nie reprezentuje przestrzenności.

Model:

```
model = Sequential([
    Dense(20, activation='relu’),
    Dense(128, kernel_initializer="normal", activation='relu’),
    Dense(64, kernel_initializer="normal", activation='relu’),
    Dense(32, kernel_initializer="normal", activation='relu’),
    Dense(16, kernel_initializer="normal", activation='relu’),
    Dense(3, kernel_initializer="normal", activation='sigmoid’),
])

model.compile(optimizer='adam’,
              loss="categorical_crossentropy",
              metrics=['accuracy’])

hist = model.fit(samples_train, labels_train,
              batch_size=8, epochs=50,
              validation_data=(samples_test, labels_test))
```

Dane uczące:

- Wykorzystano tu dane z poprzedniego zagadnienia – 3 typy ligandów, 3 kategorie. Opis kieszeni wiążących w reprezentacji one-hot.
- Podział danych na dwie kategorie: treningowe (80%) i testowe (20 %) (walidacyjne zbędne bo model uczy się bardzo szybko),
- W sumie 300 miejsc wiążących (po 100 na kategorię)

Wyniki:

Po 50 epokach model osiąga dokładność na poziomie ~90% (zbiór testowy).

Po 150 epokach model całkowicie dopasowuje się do danych – dokładność na zbiorze treningowym wynosi wtedy 1.0. Na zbiorze testowym jest to dokładność na poziomie ~95%

Jeszcze raz, tylko dla 6 kategorii ligandów:

- 6 kategorii (dodatkowe: BGC, GLC, UPG), 600 miejsc wiążących, po 100 na kategorię,
- Podział na dane treningowe i testowe taki sam jak poprzednio,

```
model = Sequential([
    Dense(20, activation='relu’),
    Dense(256, kernel_initializer="normal", activation='relu’),
    Dense(128, kernel_initializer="normal", activation='relu’),
    Dense(64, kernel_initializer="normal", activation='relu’),
    Dense(32, kernel_initializer="normal", activation='relu’),
    Dense(16, kernel_initializer="normal", activation='relu’),
    Dense(6, kernel_initializer="normal", activation='sigmoid’),
])
```

Wyniki:

- Po 50 epokach model osiąga dokładność na poziomie ~74% (zbiór testowy),
- Po 150 epokach model osiąga dokładność ~78% (kontynuowanie uczenia nie przynosi poprawy)
- Dołożenie kategorii wymagało powiększenia sieci, mimo to jej dokładność spadła.

Inne kodowania:

- Reprezentacja wektorowa (taka sama jak w pierwszym problemie) ale dla zachowania stałej wymiarowości między różnymi ligandami wymagane jest dodanie paddingu w postaci zer.
- Reprezentacja wektorowa ale konieczność stałej wymiarowości można obejść stosując sieci rekurencyjne.

Dziękuję za uwagę