보고서 작성 날짜: 05 / 06 / 2023

DNA 아미노산 서열을 활용한 단백질 안정성 예측 모델 개발

바이오데이터분석 2 조 김민정(20194730), 김소희(20203951), 김유진(20204353), 민정빈(20204535)

Abstract

생물학적 기능을 수행하는데 중요한 단백질은 DNA 의 염기서열로부터 결정된다. 과거에는 실험적인 데이터를 사용했기 때문에 시간과 노력이 많이 들었지만 최근에는 머신 러닝을 활용해 단백질 구조를 예측하는 방법이 대두되며, 그 정확도와 효율성이 향상되었다. 이는 단백질의 디자인부터 약물 스크리닝까지 생물학의 다양한 분야에 적용될 수 있다는 점에서 그 중요성이 증가하고 있다.

따라서 우리는 단백질 안정성을 예측할 수 있는 머신 러닝 기반의 모델을 제시하여, 그 성능과 실제 효율성을 측정하고자 한다. 모델의 성능을 높이기 위하여 아미노산의 서열을 통해 단백질의 안정성을 판단할 수 있는 여러가지 특성을 나열하고, 그 중요도를 고려하여 예측 모델을 만들고자 하였다. NNC 모델을 사용하여 단백질 안정성 예측 학습을 진행하였으며, 데이터의 L2 정규화 과정과 ROC curve 를 이용한 임계값을 통해 모델의 성능을 높이고자 하였다. 이를 통해 머신 러닝을 통한 단백질 예측이 가능하다는 것을 확인하였으며, 여러가지 지표를 통해 그 성능을 확인하였다. 이렇게 머신 러닝을 통해 단백질 안정성을 예측함으로서 생물 정보학 분야의 AI 활용 영역을 넓히고 또 다른 가능성을 제시할 수 있음을 시사한다.

1 Introduction

단백질은 우리 인체 내에서 생물학적 기능을 수행하는 중요한 분자로, 세포의 구조를 유지하고 효소 등을 통해 대사를 조절하는 역할을 담당한다. 그 기능은 3 차원 구조에 의해 결정되며, 구조가 체내에서 안정적으로 유지되어야 적절하게 기능할 수 있다. 하지만, 단백질은 생체 내 여러 조건에 의해 변형과 분해가 상대적으로 자주 일어나게 된다. 특히, 온도나 pH, 주위 조건 변화에 민감하게 반응하게 되면 안정성을 잃고, 그 기능을 제대로 수행하지 못할 수 있다. 이러한 단백질의 불안정성은 세포 기능의 저하, 질병의 발생, 약물 효과 저하 등의 문제를 초래할 수 있다.

단백질은 DNA 정보를 통해 만들어지는데, 그 특성은 저장된 염기서열마다 다르게 만들어진다. 일반적으로 단백질의 안정도를 측정하기 위해서 열분석법, 적외선 분광법, 자외선-가시분광광도법 등 다양한 방법을 사용해왔다. 열분석법은 단백질의 열적 성질을 연구하는 방법으로 열적 안정성, 전환온도, 열역학적 특성을 평가하는데 유용하며, 열중량분석과 열분석법 두가지가 대표적이다. 적외선 분광법과 자외선-가시분광광도법은 모두 전자기파를 사용하여 흡수 스펙트럼을 측정해 단백질이 흡수하는 특정 파장의 빛을 분석해 농도, 이온결합, 안정성평가 등 다양한 분야에서 사용하고 있다. 전통적인 생화학적 실험은 비용과 노력이 많이 소요된다는 점에서 머신 러닝 (Machine learning, ML) 기법을 사용해 단백질의 안정도를 예측하는 방법이 제안되었다. 머신 러닝은 많은 데이터를 기반으로 모델을 학습시키고, 학습된 모델을 사용해 새로운 입력 데이터에 대한 예측을 수행한다. 단백질의 안정도를 계산하기 위해 필요한 데이터셋으로는 서열정보, 구조정보 등이 있으며, 이를 종합적으로 활용해 모델을 학습시킨다. 이렇게 개발된 안정성 예측 모델은 실험에 비해 효율적이고 정확하게 단백질의 안정도를 평가할 수 있다는 장점이 있어 최근 많이 사용하고 있는 방법이다.

1.1 단백질 안정도 예측 머신러닝 모델

단백질 안정도를 계산하는데 사용하는 머신 러닝 모델은 대표적으로 CNN(Convolutional Neural Network), RNN(Recurrent Neural Network), GNN(Graph Neural Network) 이 있다. 가장 먼저 CNN 은 주로 단백질의 3 차원 구조 정보를 활용하여 안정도 예측하는 것에 사용된다. 주로 이미지 처리에 많이 사용되는 모델이지만, 단백질 서열을 1 차원으로 처리한 후에 적용한다. 단백질 서열 내에 존재하는 지역 패턴을 찾아내어 구조적인 정보를 추출하여 예측하는 방법으로 사용할 수 있어 단백질 안정도를 비교한다. RNN 은 단백질의 서열 특성을 고려한 안정도 예측에 주로 사용된다. 단백질의 아미노산 서열 정보를 통해 예측을 수행하게 된다. RNN 의 대표적인 종류인 LSTM(Long Short-Term Memory)은 서열 데이터셋을 처리하며 정보를 기억하고, 입력되는 정보를 함께 사용하여 예측을 수행하는데 특화되어 있다. 단기기억과 장기기억을 효과적으로 관리하기 때문에, 이를

바탕으로 순차적인 데이터를 처리하며 아미노산 간의 관계를 학습한다. 시간의존성 모델링이 가능하므로, 과거의 안정도 정보가 현재 안정도에 미치는 영향을 예측하는데 활용할 수 있다. GNN 은 단백질을 그래프로 표현하여 안정도 예측에 사용하는 모델이다. 아미노산이 연결된 구조를 그래프로 처리한 후, 각 노드의 특성과 상호작용을 고려하여서 그래프를 구성한다. 이는 구조적인 특성을 잘 인지하여 안정성 예측에 활용한다. GNN 을 활용하면, 아미노산의 거리, 결합강도 및 전하 상호작용 등의 특성을 파악해 단백질의 안정성 관련 특징을 가져올 수 있다. Transformer 는 이외에도 transformer, Ensemble Model, Latent Deep Learning Model 등의 모델을 활용해볼 수 있으며, 꾸준하게 단백질 안정도 예측과 관련된 모델은 개발 중이다.

본 연구에서는 DNA 에 저장된 아미노산의 서열을 바탕으로 단백질의 안정도 계산을 위한 머신 러닝 기법을 제안한다. 단백질의 아미노산 서열 자료를 활용해 PyTorch 를 사용하여 신경망 모델을 학습시켰다. 데이터 전 처리 과정을 통해 주어진 아미노산 서열을 가공하여 모델에 입력할 수 있는 형태로 전환하고, 아미노산 서열의 길이를 맞추어서 정규화한다. PyTorch 라이브러리에서 제공하는 클래스 'nn.Module' 모델을 학습시켜 손실 값(loss)과 정확도(accuracy)를 출력한다. PyTorch 는 딥러닝을 위한 오픈 소스 프레임워크로, 'nn.Module'은 그 중에서도 핵심적인 역할을 담당하는 클래스이다. 'nn.Module'은 신경망 모델의 기본 구성 요소로, 속받은 클래스에서는 모델의 구조를 변경하거나 파라미터를 업데이트하는 메서드들을 사용할 수 있다. 주어진 DNA 아미노산 서열의 데이터를 수집하고, 이를 기반으로 단백질의 안정성을 예측하는 모델을 개발하였다. 개발된 모델은 새로운 아미노산의 서열이 주어졌을 때 단백질의 안정성에 대한 평가를 진행하였으며, 이는 기존의 실험적인 방법보다 뛰어난 성능과 높은 정확도를 보여줄 것으로 기대한다.

1.2 단백질 안정도 특성 추출

본 연구에서는 ifeature 를 추출해서 사용하였다. Ifeature 란 "informative feature"로 계산 생물학 분야에서 사용되는 중요한 개념이다. 생물학적 시스템과 현상을 이해하는데 도움을 주는 특징을 정리한다. 여러 생물학적 데이터에서 추출하고 있으며, 단백질 서열 및 구조 분석을 위한 포괄적인 특징을 추출하는데 사용하는 도구이다. 서열을 기반 으로 하는 특징으로는 아미노산 구성, 분자량, PI, 지수 흡광 계수, 박막 상태, 등장 빈도, 아미노산 인덱스 및 화학 속성 등이 있으며, 구조 기반 특징으로는 단백질의 이중 헬릭스, 베타 시트, 각도, 거리 등이 포함된다. 단백질 분류 및 구조예측, 데이터베이스 구축 등에 사용될 수 있으므로 생물 정보학 분야에서 중요한 도구로 사용되고 있다.

이렇게 단백질의 안정성 예측이 가능해지면 다음과 같은 효과를 기대해볼 수 있다. 첫째, 비용과 시간이 절감될 것이다. 단백질의 안정성예측이 가능해 진다면, 기존에 실험을 통해 평가하는 것보다 시간과 비용을 절감할 수 있어 효율성을 높일 수 있다. 둘째, 대규모 단백질 스크리닝에 유용하게 사용되어 여러 후보 단백질 중에서 안정성이 높은 단백질만을 선정해 공학디자인이나 약물 개발의 대규모 스크리닝 단계에 유용하게 사용할 수 있다. 셋째, 새로운 단백질을 디자인 하는데 활용하여 기존의 단백질보다 안정성을 향상시킨 단백질을 개발할 수 있다. 이는 개발된 ML 을 사용하여 아미노산의 패턴이나 특정 구조적 특징을 조합하여 보다 생체 내에서 안정적인 단백질을 만들어낼 수 있을 것이다. 새로운 단백질 개발은 최종적으로 신약 개발로 연관된다. 특정 약물과 단백질의 상호작용을 예측하여 약물의 안정성을 예측할 수 있고, 안정성을 증가시켜 부작용은 줄이고 효과를 높일 수 있다. 나아가 개인 맞춤형 치료에 적용하여 환자의 유전적 다양성을 고려해 가장 안정적인 단백질 치료제를 제공할 수 있을 것으로 기대한다.

이처럼 기존에 알려진 DNA 의 아미노산 서열로부터 단백질 안정성 예측이 가능해지면, 단백질의 설계와 치료법 개발의 효율성을 향상시킬 수 있다는 점에서 그 중요성이 대두된다. 따라서 본 연구에서는 높은 정확도를 가진 모델을 개발하여 효과적으로 단백질의 안정성을 예측할 수 있는 방안을 제안하고자 한다.

2 Methods

2.1 데이터 수집 과정

주어진 아미노산 Sequence Text 파일 데이터를 코딩에 사용하기 위해서는 항목별로 list 에 저장하는 작업이 필요하다. Genenames/ ProteinID/ seq/ Label 4 가지 항목 별로 list 를 만들어 data list 로 저장함을 통해 코딩에 사용할 수 있는 데이터로 정리했다. 정확한 학습을 위해서는 데이터 전처리 과정이 필요한 경우 전처리를 실시한다.

2.2 단백질 안정성에 기여하는 feature

단백질 stability 를 높이는 feature 들을 계산하여 학습에 효율성을 높이는 작업이 필요하다. 대표적으로 단백질 stability 를 높이는 feature 에는 Amino acid composition, secondary structure, domains/motifs 등이 있다. 또한 단백질의 3D structure 와 단백질 에너지의 안정성 또한 stability 에 큰 역할을 한다.

하지만 주어진 amino acid sequence 를 가지고는 3D structure 나 에너지량 같은 feature 를 알 수 없다[1]. 따라서 amino acid sequence 를 가지고 파악할 수 있는 feature 들을 중심으로 단백질의 안정성을 높이는 특성들을 조사하였다.

1. Hydrogen bonding

수소 결합은 단백질의 분자 구조를 유지하고 단백질의 3 차 구조를 형성하는 데에 중요한 역할을 한다. 이러한 수소 결합은 단백질 내에서 아미노산 잔기 간, 아미노산 잔기와 용매 분자 간,

아미노산 잔기와 다른 생체 분자 간에 형성될 수 있고, 단백질의 알파 나선, 베타 시트, 턴 등의 이차 구조를 형성하는 데에 중요한 역할을 한다. 또한 수소결합은 변성, pH 변화, 환경 변화 등에 더 잘 견디도록 단백질을 안정화시키는 데 도움을 줄 수 있다. 수소결합이 일어나는 아미노산의 종류로는 **글리신 (Gly), 세린 (Ser), 트레오닌 (Thr), 티로신 (Tyr), 아스파라진 (Asn), 글루타민 (Gln) 아르진 (Arg), 리신 (Lys), 히스티딘 (His)** 등이 있다. 따라서 이러한 아미노산과 단백질 안정성 사이의 관련성이 존재한다.

2. Hydrophobic

Hydrophoicity 는 단백질 안정성에 긍정적인 영향을 미칠 수 있다. hydrophbic 아미노산은 주로 단백질 내부에 위치하면서 내부 구조의 안정성을 높이는데 역할을 할 수 있다. 또한 단백질 접힘 과정에서 중요한 역할을 할 수 있다. 단백질이 수용액과 상호작용할 때, hydrophbic 가 증가할수록 단백질은 수용액과의 상호작용을 최소화 시키기 위해 단백질 접힘 등을 통해 안정한 상태를 만들려고 한다. 따라서 hydrophbic 은 단백질 안정성에 도움을 준다고 할 수 있다. Hydorphbic 아미노산으로는 **glycine (Gly), alanine (Ala), valine (Val), leucine (Leu), isoleucine (Ile), proline (Pro), phenylalanine (Phe), methionine (Met), and tryptophan (Trp)** 등이 있다.

3. Disulfide bond

Disulfide bond 는 단백질 구조에 안정성을 높이는데 큰 역할을 한다. Cysteine 에 있는 –SH group 이 disulfide bond 를 형성하여 단백질 구조 및 접힘에 기여하고 적절한 기능을 수행하도록 한다. 따라서 disulfide bond 를 형성하는 **cysteine** 이 단백질 안정성에 긍정적인 영향을 미칠 수 있다.

4. Acidity/alkaline (pI_value)

일반적으로 단백질은 특정 pH 범위에서 최적의 안정성을 나타내는데, 이를 등전점(pI, isoelectric point)이라고 한다. 등전점은 단백질의 전체 아미노산 조성에 의해 결정되며, 단백질이 중성인 상태로 전기적으로 중립화되는 pH 값을 의미한다. pH 변화는 단백질의 구조 변화, 안정성 등에 영향을 미칠 수 있기 때문에 단백질의 등전점은 중요한 특성이라고 할 수 있다. 일반적으로 생체 내의 pH 와 유사할 때, 단백질이 안정적으로 존재할 수 있다. 따라서 단백질의 pI 가 7.0~7.4 정도 일 때 안정적으로 존재 할 가능성이 높다. 따라서 아미노산 서열을 이용하여 단백질 각각의 pI 를 계산하여 단백질 안정성 평가에 이용할 수 있다.

5. AAC/ CTDC

AAC 는 Amino Acid Composition 의 약자로, 단백질에서 각 아미노산의 상대적인 출현 빈도를 계산하여 단백질의 특징을 표현하는 방법이다. 아미노산은 단백질을 구성하는 기본 단위로 그 구조와 기능에 영향을 주기 때문에 중요한 특성으로 볼 수 있다. AAC 를 계산하기 위해서는 먼저 단백질의 아미노산 시퀀스를 분석하여 각 아미노산의 개수를 세는 작업이 필요하다. 아미노산이 전체 시퀀스에서 차지하는 상대적인 비율을 계산하게 된다. 예를 들어, 아미노산 시퀀스에서 Alanine 이 20 개, Leucine 이 10 개, Valine 이 5 개 나타나는 경우, Alanine 전체 시퀀스에서 약 57.1% (20/35), Leucine 약 28.6% (10/35), Valine 약 14.3% (5/35)의 상대적인 출현 빈도를 계산할 수 있다. 정 아미노산의 출현 빈도가 높을수록 해당 아미노산은 단백질 구조나 기능에 중요한 역할을 할 가능성이 높으므로 다음과 같은 계산을 통해 단백질의 안정성에 미치는 영향을 알아낸다.

CTDC(Composition, Transition, Distribution, Composition)는 아미노산 시퀀스의 구조적 특성을 고려하여 단백질의 특징을 추출하는 방법으로 다음 네가지 요소로 구성된다. 먼저 Composition 은 아미노산 시퀀스에서 특정 아미노산 조합의 출현 빈도를 나타낸다. 시퀀스 내의 특정 길이의 아미노산 조합을 바탕으로 그 빈도를 계산해 구조적 특징을 파악한다. Transition 은 아미노산 시퀀스에서 인접한 아미노산 쌍의 전이 빈도를 나타내며, 시퀀스 내에서 연이은 두 아미노산이 나타내는 쌍을 계산한다. Distribution 은 아미노산 시퀀스에서 특정 아미노산의 위치 분포를 나타내며, 시퀀스를 분석하여 특정 아미노산이 시퀀스 내에서 어디에 위치하는지를 확인한다. 마지막 Composition 은 아미노산 시퀀스에서 특정 아미노산 조합의 출현 빈도를 계산한다. CTDC 는 아미노산 서열의 구조적 특성을 고려하여 단백질의 특징을 추출하는 방법으로, 단백질의 구조와 기능을 이해하는 데 도움을 줄 수 있다는 점에서 그 장점이 있다.

6. 그 외 고려 가능한 특징 - Secondary structure

Secondary structure 는 단백질 구조와 안정성에 중요한 영향을 미친다. 아미노산 서열은 단백질의 부분적인 secondary structure 정보를 파악하는 데 도움을 줄 수 있다. Secondary structure 는 단백질 구조에서 중요한 요소이고 이는 단백질 안정성과 관련이 있을 수 있다.

α-헬릭스 (Alpha-helix) 구조는 α-헬릭스는 평면적이고 나선 모양의 구조를 가지며, 내부 수소결합이 안정성을 높일 수 있다.

베타 시트 (Beta-sheet)는 단백질 내에서 평면적인 시트 구조를 형성하는 secondary structure 중 하나이다. 턴은 주로 α-헬릭스나 베타 시트 간의 전환 부위로 나타나는 secondary structure 로서, 글라이신과 같은 아미노산이 단백질의 구조적 유연성을 제공하며, 단백질의 접힘과 안정성에 중요한 역할을 한다.

7. Feature normalization

Feature 마다 상대적인 값이 다르면 실제 학습에 있어서 데이터를 정확하게 해석하고 학습하지 못할 가능성이 있다. 따라서 모든 feature 들의 기여도를 일정하게 맞추기 위해 feature 들의 값을 정규화 시키는 작업이 필요하다.

각각의 feature 에 대한 Def 함수를 설정하였고, 정규화 과정을 위해 normalization 수식을 이용하여 함수의 결과 값을 토큰화

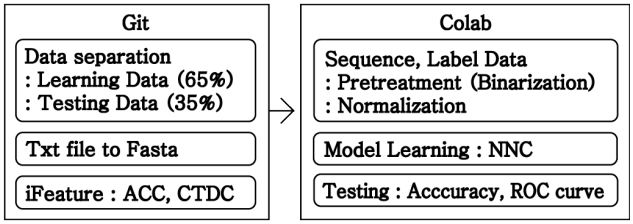
$$X = \frac{x - x_{min}}{x_{max} - x_{min}}$$

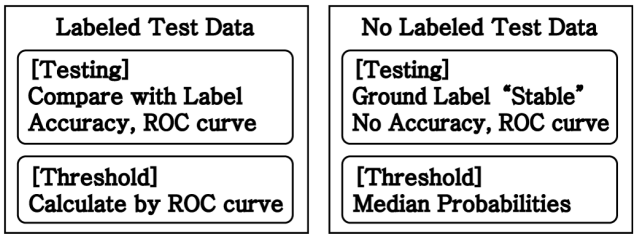
패딩으로 정규화 시켰다. Min-Max(Normalization) 수식을 이용하여 정규화를 진행하였다.

2.3 기계학습 과정을 수행한 과정

데이터 수집과 분류, 전처리, 모델 학습과 테스트 과정을 거쳤다. 머신러닝 모델은 빠른 학습 수행을 위해 GPU 를 제공하는 Colab 에서 제작하였다. Stability Label 이 있는 Test data 와 Label 이 없는 Test data 에 대해 두가지 코드를 작성하였다. 데이터 전처리와 특징 추출에 대한 코드와 결과 자료를 Git 에 올렸다..

Training & Test set





1. 데이터 수집 및 전처리

(1) 학습 데이터, 테스트 데이터 분류

> labeled Sequence Data :

주어진 약 700 개의 sequence 데이터를 txt 파일로 저장하고, 학습과 테스트 데이터로 랜덤하게 2:1 비율로 분류하였다.

> No labeled Sequence Data :

기존에 주어진 sequence 데이터를 모두 학습 데이터로 사용하였다.

테스트 데이터는, 실제 테스트와 같은 환경을 조성하기 위해 uniprot 에서 랜덤한 Sequence data 100 개를 받아 사용하였고, label 을 모두 Stable 로 지정하였다.

(2) 데이터 전처리 및 특징 추출

Sequence 의 특징을 추출하기 위한 전처리 과정으로 txt 파일을 fasta 파일 포맷으로 전환하였고, iFeature 에서 제공하는 코드를 활용하여 ACC, CTDT 특징을 추출하여 csv 파일을 얻었다.
학습 데이터와 테스트 데이터 모두 각각 같은 과정을 진행하였고, Google Drive 에 저장하여 colab 에서 사용하였다.

> labeled Sequence Data :

Sequence 의 Gene 이름과 Protein 의 ID 를 알 수 있는 기존 데이터는 fasta 파일에 이를 조합하여 넣었다

> No labeled Sequence Data :

Squence 만 주어져 정보를 알 수 없는 데이터는 "Protein1"과 같은 임의의 이름으로 지정하였다.

(3) 입력 데이터 구성

아미노산 서열과 ACC, CTDT 데이터를 0~1 사이 값으로 변환하고, L2 정규화를 통해 데이터를 전처리 하였다. 학습 데이터와 테스트 데이터 모두 같은 과정을 진행하였다.

2. 모델 학습

(1) 학습 모델 제작

PyTorch 의 'nn.Module'를 사용하여 모델을 제작하였다. 배치 사이즈는 400, 학습률은 0.0015, epoch 는 429 로 설정하여 모델을 학습시켰다. 모델 생성과 학습, 테스트를 50 회 반복하고, 테스트 결과의 최빈값을 사용하여 최종 결과와 정확도를 결정한다. 또한, 예측 확률의 평균을 계산하고 ROC curve 를 생성한다. 구글 코랩의 GPU 기능을 활용하여 보다 효율적이고 빠른 학습을 수행할 수 있었다.

(2) 테스트 분류 임계값 설정

모델 학습 후 테스트 데이터의 결과값을 도출하기 위하여 임계값을 지정해야 한다. 임계값은 테스트 결과의 정확도와 직접적으로 관련 있기 때문에, 적절한 임계값의 설정이 필요하다.

> labeled Sequence Data :

이미 Stability 를 알고 있는 테스트 데이터의 경우 Accuracy 와 ROC curve 를 계산할 수 있기 때문에, 가장 정확한 임계값을 ROC curve 를 통해 설정할 수 있다. 주어진 label 과 모델이 예측한 Probability 를 통해 ROC curve 를 계산하고, True Positive Rate('tpr')와 False Positive Rate('fpr')가 가장 큰 지점을 임계값으로 지정한다.이 지점은 모든 확률 값에 대한 최적의 분류 임계값으로, 가장 높은 정확도를 계산할 수 있다.

> No labeled Sequence Data :

Label 이 없어 Accuracy 와 ROC curve 를 계산할 수 없기 때문에, ROC curve 를 활용하여 정확성이 높은 임계값을 설정하지 못한다. 대안으로, 정확도가 다소 낮지만 Label 없이도 설정 할 수 있는 Probability 의 중앙값을 임계값으로 설정하여 Stability 를 분류하였다.

(3) 높은 정확도를 도출하기 위한 노력

ROC curve 를 활용하여 임계값을 설정하였다. 이를 통해 Probability 의 중앙값을 임계값으로 설정하는 것보다 더 높은 정확도를 가지는 모델을 제작할 수 있었다.

모델을 한번만 생성하고 학습하는 경우 테스트 시 40%대의 낮은 정확도를 보였다. 정확도를 높이기 위하여 모델을 여러번 생성, 학습 후 테스트를 진행하고, 최빈값을 통해 최종 결과와 정확도를 도출해 내었다. ROC curve 를 활용하여 임계값을 설정할 경우 60%대의 정확도를, 예측 확률의 중앙값을 임계값으로 설정할 경우 52~56% 사이의 정확도를 유지하는 모델을 제작할 수 있었다.

.

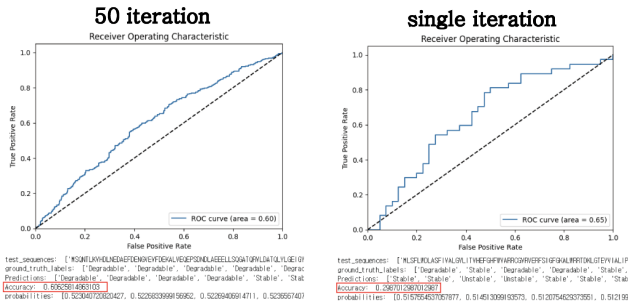


그림 1 ROC curve

3 Results and Discussion

3.1 Feature 내용 소개

올바른 feature 설정은 머신 러닝 모델을 만들 때 매우 중요한 요인으로 작용한다. 그러나 과도한 feature 를 사용하거나 중복되는 feature 를 사용하게 되면, 학습 데이터에 존재하는 노이즈나 이상치에 대해서까지 과도하게 학습하여 새로운 데이터에 대한 일반화가 제대로 이루어지지 않는 overfitting 이 발생할 수 있기 때문에 학습 데이터의 크기를 고려하여 크게 두가지의 feature 를 모델 학습에 사용하였다.

본 연구에서는 아미노산 시퀀스에서 여러가지 특징을 추출할 수 있는 i-feature 를 사용하였다. 단백질의 안정성을 아미노산 시퀀스로 판단하기 위해서는 아미노산 시퀀스와 단백질 안정성의 관계를 정의할 수 있어야한다. 아미노산 시퀀스를 바탕으로 단백질의 다양한 특징을 추출할 수 있는 함수들을 제공하며, 해당 tool 로 추출한 특징은 수치 데이터로 표현할 수 있어 머신러닝 알고리즘에 사용하기 적합했기 때문에 머신러닝 모델 개발에 사용하기 편리했다. 다음은 본 연구에서 우선 시하여 고른 특징들을 나타낸다.

- AAC (Amino Acid Composition)

AAC 는 각각의 아미노산이 전체 아미노산 시퀀스에서 차지하는 상대적인 비율을 나타낸다. 아미노산 시퀀스를 분석하여 아미노산의 종류별로 카운트를 수행하고, 해당 카운트를 전체 아미노산 개수로 나누어 상대적인 비율로 변환한다. 이를 통해 각 아미노산의 상대적인 출현 빈도를 계산하여 단백질의 특징을 표현할 수 있다. 즉, 단백질 안정성에 영향을 미치는 특정 아미노산의 출현 빈도를 파악할 수 있으므로 해당 특징을 추출하여 모델에 학습시켰다.

- CTDC (Composition, Transition, Distribution, Composition)

CTDC 는 아미노산 서열의 구조적 특성을 고려하여 단백질의 특징을 추출한다. 아미노산 시퀀스의 조합, 전이, 분포 및 조합 정보를 사용하여 단백질 안정성과의 상관관계를 계산한다. AAC 는 아미노산 시퀀스의 배열과 상관 없이 특정 아미노산의 빈도만 계산하기 때문에, CTDC 를 함께 사용함으로써 단백질 안정성 예측 모델의 성능을 향상시키고자 하였다

다음은 추출한 각 특징을 추출한 값과 FASTA 형식으로 바꾼 자료의 일부이다.

spoT_P0AG240.08689458689458690.011396011396011397O.048433048
prlF_P153730.072072072072072070.009009009009009009O.090090090
osmF_P333620.140983606557377040.00.049180327868852460.049180
fbaB_P0A9910.117142857142857150.011428571428571429O.062857142
cyoA_P0ABJ10.107936507936507940.003174603174603174600.0380952
yajQ_P0A8E70.067484662576687120.00.085889570552147240.0736196
btuE_P066100.065573770491803280.016393442622950820.054644808
rplM_P0AA100.098591549295774640.00.049295774647887320.056338

>spoT_P0AG24
MYLFESLNQLIQTYLPEDQIKRLRQAYLVARDAHEGQTRSSGEPYITHPVAVACILAEMKLDYETLMAALLHDVIEDTPATYQDMEQLFGKSVAE
>prlF_P15373
MPANARSHAVLTTESKVTIRGQTTIPAPVREALKLKPGQDSIHYEILPGGQVFMCRLGDEQEDHTMNAFLRFLDADIQNNPQKTRPFNIQQGKI
>osmF_P33362
MPLLKLWAGSLVMLAAVSLPLQAASPVKVGSKIDTEGALLGNIILQVLESHGVPTVNKVQLGTTPVVRGAITSGELDIYPEYTGNGAFFFKDENI
>fbaB_P0A991
MTDIAQLLGKDADNLLQHRCMTIPSDQLYLPGHDYVDRVMIDNNRPPAVLRNMQTLYNTGRLAGTGYLSILPVDQGVEHSAGASFAANPLY
>cyoA_P0ABJ1
MRLRKYNKSLGWLSLFAGTVLLSGCNSALLDPKGQIGLEQRSLILTAFGLMLIVVIPAILMAVGFAWKYRASNKDAKYSPNWSHSNKVEAVVV

그림 2 AAC feature 을 각각 csv 와 FASTA 형식으로 나타냄

3.2 모델 학습

3.2.1 모델 학습 및 테스트 절차

(1) 레이블이 부여된 아미노산 시퀀스의 TXT 파일을 FASTA 형식으로 변환한다.

(2) 학습 데이터와 테스트 데이터를 랜덤하게 2:1 비율로 분류한다.

(3) Ifeature 를 사용하여 FASTA 파일로부터 단백질의 특성을 추출한다.

(4) 아미노산 시퀀스와 추출한 특성을 노드로 설정하고 모델 학습을 진행한다.

(5) 학습된 결과와 레이블을 비교하여 손실 값을 결정한다.

(6) Tensor 의 중앙값과 ROC curve 를 사용하여 임계값을 설정한다.

(7) 배치 사이즈는 400, 학습률은 0.0015, epoch 는 429 으로 설정하여 모델을 학습시킨다.

(8) 모델 학습을 50 회 반복하고, 최빈값을 사용하여 결과와 정확도를 결정한다. 또한, 예측 확률의 평균을 계산하고 ROC curve 를 생성한다.

3.3 모델 평가

위와 같은 방식으로 모델을 만든 후, 모델의 정확도를 평가 하기 위해 여러가지 지표를 이용하여 단백질 예측 모델링의 성능을 평가하였다.

1. ROC & AUC (Area Under the Curve)

ROC 곡선은 이진 분류 모델의 임계값을 변화시키면서 계산되는 거짓 양성 비율(FPR)에 대한 진짜 양성 비율(TPR)의 그래프를 나타내며, 모델의 성능을 평가할 때 이용한다.

AUC 는 ROC 곡선의 아래 면적을 나타내고, 역시 모델의 분류 성능을 종합적으로 나타내는 지표로 사용된다. 일반적으로 AUC 의 값이 0.5 이상이면 성능이 우수하다고 평가한다.

우리는 ROC curve 를 사용하여 임계 값을 설정하는데 사용하였다. 이때, AUC 값은 0.6 이였으며, ACUURACY 또한 0.6 정도로 계산되었다.

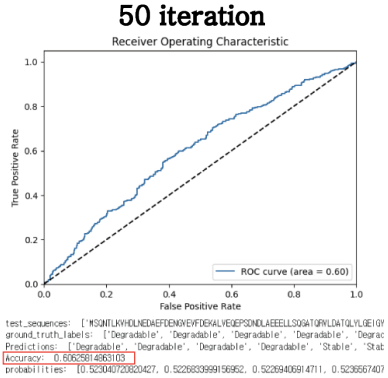


그림 3 모델 성능 평가 ROC

2. F1 SCORE

F1 SCORE 는 정밀도와 재현율의 조화 평균 값으로 모델의 성능을 평가하는 지표로 사용된다. 값이 1 에 가까울수록 모델의 성능이 우수하다고 할 수 있다. [F1 = 2 * (정밀도 * 재현율) / (정밀도 + 재현율)]

이때 정밀도는 정밀도는 양성으로 예측한 샘플 중 실제로 양성인 비율을 의미하며 [TP / (TP + FP)], 재현율은 실제 양성인 샘플 중 모델이 양성으로 정확하게 예측한 비율 [TP / (TP + FN)]을 의미한다. 즉, 모델이 실제 양성인 샘플을 얼마나 잘 찾아내는지를 나타낼 수 있다. 이를 통해 정확도 와 함께 모델의 성능을 평가하는 지표로 사용할 수 있다.

TRUE POSITIVE(TP)	27
TRUE NEGATIVE(TN)	0
FALSE POSITIVE(FP)	0
FALSE NEGATIVE(FN)	24
ACCURACY	0.5294
F1 SCORE	0.69

표 1 최종 test 결과 (total: 51 dataset)

최종 TEST 결과 ACCURACY 는 0.5294 이며, F1 SCORE 는 0.69 이다. ACCURACY 로 판단하기에는 다소 아쉬운 수치로, 예상했던 ACCURACY 0.6 에 미치지 못했다.

3. 로그 손실 (Log Loss)

로그 손실은 모델의 예측 확률과 실제 라벨 사이의 차이를 계산한 후 평균을 취한 값으로, 확률 기반 분류 모델의 성능을 평가하는 지표로 사용할 수 있다. 로그 손실은 크로스 엔트로피 손실(Cross Entropy Loss) 또는 로지스틱 손실(Logistic Loss)이라고도 불리며, 실제 확률값과 예측 확률값 사이의 차이를 측정한다.

$$\text{logloss} = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log p_i + (1-y_i)\log(1-p_i))$$

$$= -\frac{1}{N}\sum_{i=1}^{N}\log \hat{P}_i$$

로그 손실은 위와 같은 식으로 계산할 수 있으며, 값이 낮을수록 좋은 성능으로 해석할 수 있다. PyTorch 에서는 로그 손실을 계산하는 함수로 'nn.BCELoss()' 또는 'nn.BCEWithLogitsLoss()'를 사용할 수 있다. 'nn.BCELoss()'는 확률값에 시그모이드 함수를 적용한 후 로그 손실을 계산하고, 'nn.BCEWithLogitsLoss()'는 시그모이드 함수를 거치지 않은 확률값에 대해 로그 손실을 계산한다.

주어진 GROUND DATA 와 실제 예측 PROBABILITY 값을 통해 로그 손실 값을 계산한 결과 Log loss : 0.6486790776252747 정도로 계산되었다.

3.4 모델 결과 정리 및 오차 원인 분석

본 연구에서는 결과의 예측을 Predictions: ['Stable', 'Degradable', 'Stable', 'Stable', 'Stable'] 의 형태로 나오도록 모델을 학습시켰으며, 전체적인 loss 는 평균 0.6 accuracy 는 평균 0.이 나올 수 있도록 다양한 생화학적 특징과 머신러닝 모델을 사용하여 예측을 시도하였다. 추후 여러 지표를 활용하여, 예측 모델의 성능을 평가하였으며, 좋은 결과가 나온 값도 있었지만 종합적으로 성능이 크게 뛰어나다고 판단하기에 부족하다고 생각했으며, 이는 여러가지 이유에서 모델 학습이 부족했음을 의미한다.

그 중 가장 큰 부분은 feature 의 부족이라고 판단하였다. 우리는 i-feature 에서 특성을 추출하여, AAC 와 CTDC 를 선정하였다. 물론 두가지 feature 모두 아미노산의 배열과 관련되어 있으며, 이를 통해 단백질의 안정성을 예측하기에 도움이 되었다고 생각한다. 하지만 단백질의 안정성에는 아미노산의 배열 뿐만 아니라, hydrophobicity, hydrogen bonding, PH 등 많은 요소를 함께 고려해야 한다. 특히 단백질의 안정성과 구조에 많은 영향을 미치는 secondary structure 에 대한 feature 를 넣지 못한 것이 정확도를 높이지 못한 하나의 원인이 될 수 있다고 생각한다. I-feature 에서 secondary structure 의 특성을 얻을 수 있는 수식이 존재하였으나, 이를 코딩을 통해 데이터를 변환하는 과정에서 오류가 계속 발생하였고, 결국 feature 로 사용할 수 없었다. 물론 secondary structure 를 아미노산의 서열만으로 예측하기에 한계가 있으나, 만약 feature 로 사용했으면 조금이나마 정확성을 높이는데 기여했을 것이라 생각한다.

또한 모델을 학습시키는 과정에서, 학습 및 테스트 데이터 설정, epoch 횟수 및 배치 사이즈 등의 최적의 학습을 수행 할 수 있는 조건들의 설정 또한 최종 accuracy 에 영향을 미쳤을 것이라고 생각된다. 머신러닝을 실행하는 과정에서 epoh 과 drop out 수를 조절하며 과적합되는 것을 막고 최대한 높은 정확도를 가져올 수 있는 값을 설정하였지만, 최적화가 되기에는 부족한 점이 있을 수 있었다.

마지막으로, 모델 선정에서의 오류이다. 아미노산 서열을 바탕으로 단백질 구조를 예측할 때에는 충분하지 않은 데이터를 바탕으로 결과 예측을 진행해야하고 순서와 인접한 아미노산의 종류 등을 고려해서 분석을 진행하여야하기 때문에 SVM 이나 LSTM 과 같은 기타 모델을 고려해보았다. 하지만, 본 연구에서는 최종적으로 다른 모델을 선택하였고, 다른 모델이 가질 수 있는 장점을 고려하지 못하였다는 점에서 아쉬움이 있다.

4 Conclusion

우리는 머신 러닝 기반의 단백질 안정성 예측 모델링을 제작하였고, 이를 다양한 지표를 활용하여 성능을 평가 하였다. 단백질 안정성을 예측하기 위해서 다양한 단백질 안정성에 영향을 미치는 특성들을 조사했으며, 이를 학습에 이용 가능한 형태로 변환시키는 작업을 진행하였다. 결과적으로 i-feature 를 통해 아미노산 배열을 읽고 단백질 안정성을 예측하는 모델을 제작할 수 있었으며, 이를 다양한 지표들을 통해 평가하는데 성공하였다.

이는 단백질의 안정성을 머신 러닝을 통해 예측할 수 있고, 기존의 실험을 통한 평가보다 빠르고 효율적인 예측이 가능함을 의미한다. 따라서 이러한 머신 러닝 기반의 단백질 안정성 예측 모델의 신약개발 분야에서 잠재적 이용 가능성이 기대되며, 그 중요성이 더욱 강조될 것이다.

References

1. Kamerzell, T.J. and C.R. Middaugh, *Prediction machines: applied machine learning for therapeutic protein design and development.* Journal of Pharmaceutical Sciences, 2021. **110**(2): p. 665-681.
2. Kulshreshtha, S., et al., *Computational approaches for predicting mutant protein stability.* J Comput Aided Mol Des, 2016. **30**(5): p. 401-12.
3. Liu, S., C. Liu, and L. Deng, *Machine Learning Approaches for Protein(-)Protein Interaction Hot Spot Prediction: Progress and Comparative Assessment.* Molecules, 2018. **23**(10).
4. Marabotti, A., B. Scafuri, and A. Facchiano, *Predicting the stability of mutant proteins by computational approaches: an overview.* Brief Bioinform, 2021. **22**(3).
5. Ozen, A., et al., *Machine learning integration for predicting the effect of single amino acid substitutions on protein stability.* BMC Struct Biol, 2009. **9**: p. 66.
6. Pan, Q., et al., *Systematic evaluation of computational tools to predict the effects of mutations on protein stability in the absence of experimental structures.* Briefings in Bioinformatics, 2022. **23**(2).
7. Pancotti, C., et al., *A Deep-Learning Sequence-Based Method to Predict Protein Stability Changes Upon Genetic Variations.* Genes (Basel), 2021. **12**(6).
8. Santos, J. and H. Rivas, *Evolution of Amino Acid Properties in the Context of Protein Secondary Structure Prediction*, in *2021 IEEE Congress on Evolutionary Computation (CEC)*. 2021. p. 426-433.