PASS : Prediction of α-Helix Transmembrane Region by Separating ER Signal Sequence

PASS : ER 시그날 시퀀스 분리를 통한 단백질의 알파헬릭스 막횡단 부위 예측

Min-ho Jung[1], Young Joo Seol[1], Min Kyung Kim[2*], Hyun Seok Park[3], Seong-Joon Yoo[1]

[1]School of Computer Engineering, Sejong University, Seoul, Korea
[2]Center of Engineering Research, Ewha Womans University, Seoul, Korea
[3]Department of Computer Science and Engineering, Ewha Womans University, Seoul, Korea

*To whom correspondence should be addressed. E-mail: minkykim@ewha.ac.kr

Abstract

이 논문에서는 ER 시그날 시퀀스 서열의 존재 여부와 단백질에의 알파헬릭스 형태의 막횡단 부위를 예측하는 통합시스템을 개발하였다. 기존의 시스템과 달리 이 두 가지 예측을 하나의 통합된 시스템에서 수행하여 예측의 정확성을 높였다. 또한 인터넷에서 이용이 가능하도록 웹 서버(http://dblab.sejong.ac.kr/pass/index.html)를 구현하였다.

Introduction

알파헬릭스 막횡단 부위 예측기만 이용하여 예측을 수행하면 정확성이 떨어지는 사실이 밝혀져 있다[1]. 예를 들어 알파헬릭스 막횡단 부위 예측기를 먼저 수행한 경우라면, 이 알파헬릭스 막횡단 부위 예측기에 의해서 ER Signal Peptide[1]에 해당하는 부위도 막횡단 구간(Membrane Spanning Region)으로 인지될 수 있으며 Sideness에 있어서 여러 개의 막횡단이 나오는 경우에 Reading Frame방식으로 Inward, Outward가 결정되는바 전체의 Sideness에 있어 프래임 이동이 일어날 가능성이 있다. 따라서 예측기의 성능을 향상시키기 위해서는 시그날 시퀀스와 알파헬릭스 막횡단 부위가 하나의 서열에 함께 존재한다는 가정을 하고, 이 두개를 구분하여 예측할 수 있도록 하는 것이 보다 정확한 결과를 얻을 수 있다.

시그날 시퀀스, 막횡단 부위 예측을 위해 현재 우리가 이용할 수 있는 예측용 소프트웨어들로는 SignalP[2], TargetP, TMHMM[3]등의 시스템을 개발한 덴마크의 CBS(Center for Biological Sequence Analysis)가 가장 발전된 기술을 보유하고 있다. 그러나 이들을 이용하여 위와 같은 통합 예측을 하기 위해서는 여러 개의 서로 다른 프로그램을 순서대로 사용해야 한다. 또한 전체 예측기의 성격을 정확히 이해하고 사용해야 할 필요가 있다. 따라서 이 논문에서는 이러한 작업의 복잡성을 제거하기 위하여 통합 시스템을 구현하고, 이 시스템이 기존의 막횡단 부위 예측 소프트웨어만을

1) Signal Peptide은 단백질 서열에 단백질의 방향의 신호 역할을 하는 서열이다.

이용하여 예측할 때보다 성능이 향상됨을 보인다.

Prediction of Cleavage Sites

먼저 CSP(Cleavage Site Prediction) 즉 ER Signal Peptide 부위를 인지하는 모듈을 개발하였다. 이는 소포체 서열의 Signal Peptide가 끝나는 Cleavage Site[2]를 예측하기 위해서 신경망을 사용하였다. 시그날 시퀀스는 크게 두 부분으로 나눌 수 있다. 즉 N-terminal Site 에 있고 단백질의 방향을 결정하는 Signal Peptide와 실제로 소포체로 가는 단백질 서열에서 Signal Peptide가 잘려나간 나머지 단백질 즉, Mature Protein이 그것이다.

그림 1은 신경망을 훈련시키는 모델이다. 훈련 예는 SignalP에서 제공하는 것을 가공하여 사용하였다.[2]

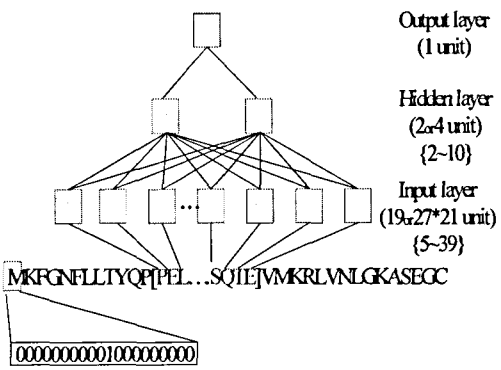

그림 1 Signal Peptide 예측을 위한 신경망 모델

위 그림 1에서 입력 서열은 Window Size를 5에서 39까지 선택 할 수 있는 Sliding Window[3] 기법을 응용한다. 이 논문에서는 Window Size를 C-Score를 위한 알고리즘은 27로 하였고, S-Score를 위한 알고리즘에서는 19로 하여 학습하였다.

본문에서 사용된 신경망은 2가지 종류가 있다. 그것은 C-Score(Raw Cleavage site Score) 신경망과 S-Score(Signal Peptide Score) 신경망 이다.

여기에서 C-Score 신경망이란 훈련 예를 Cleavage Site +1 아미노산만을 높은 점수(1)로, 나머지 아미노산은 낮은 점수(0)로 초기화하여 훈련시킨 신경망이고 S-Score신경망이란 훈련 예를 Cleavage Site 전 아미노산의 전체를 높은 점수(1)로, 나머지 Cleavage Site 이후 아미노산, 즉 Anchor 부분에는 낮은 점수(0)로 초기화하고 Signal Peptide를 포함한 시그날 시퀀스의 길이가 30이 되도록 훈련시킨 신경망이다.

훈련 예를 들어 Cleavage Site 가 아미노산 서열 13번째 아미노산과 14번째 아미노산 사이이면 C-Score 신경망의 훈련 예는 Cleavage Site +1 인 위치인 14번째 아미노산의 값만 높은 점수 즉 1로 하고, 나머지 아미노산의 값은 낮은 점수 즉 0으로 하여 훈련을 시키고 S-Score 신경망의 훈련 예는 Cleavage Site 이전인 13번째 전 아미노산 값 모두를 높은 점수 즉 1로 하고, 14번째 이후 아미노산의 값은 낮은 점수 즉 0으로 하여 훈련 하였다.

또한 예측 방법에는 2가지 방법이 있다. C-Score를 이용하는 방법과 C-Score와 S-Score를 혼합한 Y-Score를 이용하는 방법이다. C-Score를 이용하여 예측하는 방법은 단순히 C-Score 신경망에서 결과 값으로 예측하는 것이다. Y-Score를 이용하여 예측하는 방법은 C-Score 신경망에서 나온 결과 값 즉, C-Score와 S-Score 신경망에서 나온 결과 값 즉, S-Score를 혼합하여 Y-Score를 구해서 예측하는 방법이다.[4]

$$Y_i = \sqrt{C_i \Delta_a S_i}$$

수식 1

---

2) Signal Peptide와 Mature Protein을 구분하여 주는 Site를 Cleavage Site 라고 한다.

3) Sliding Window는 하나의 training 방법으로서 일정한 window size를 정해 하나의 긴 서열을 window size 만큼 잘라서 훈련시키고 그 다음 단계로 가서 훈련시키는 기법이다.

여기에서 $\Delta_d S_i$는 i 위치에서 앞뒤로 d 개의 S-Score 의 평균을 나타내는 것이다. 아래에는 $\Delta_d S_i$을 계산 하는 방법이다.

$$\Delta_d S_i = \frac{1}{d}\left(\sum_{j=1}^{d} S_{i-j} - \sum_{j=0}^{d-1} S_{i+j}\right)$$

수식 2

이 논문에서는 d를 8로 하여 Y-Score를 계산하였다.

여기서 신경망의 결과 값은 학습 할 때 결과층에 나온 실수형 데이터를 Cutoff[4]를 사용해서 Cutoff 보다 크면 1.0, Cutoff 보다 작으면 0.0으로 최종 결과 데이터를 구한다. 여기에서 Cutoff 값을 0.5를 사용했다.

Y-Score는 S-Score 가 Cutoff 이상인 위치에서 Cutoff 이하로 변하는 위치와 Y-Score 가 갑자기 변하는 위치가 일치 하는 위치가 Cleavage Site 이다.

이 훈련 예는 SignalP(http://www.cbs.dtu.dk/ftp/signalp/)에서 Flat File을 사용하였다.

단백질 서열은 아미노산 20개로 구성되어있는데 1개의 아미노산을 2진수 21비트로 변환시켜서 입력 값을 만든다. 예를 들어서 아미노산 중 'A'는 '100000000000000000000' 으로 'R'을 '010000000000000000000', 이런 식으로 20개의 아미노산을 표현한다.

표 1 아미노산 코드를 21진수의 2진 코드로 변환해주는 대조표

A	100000000000000000000
C	010000000000000000000
D	001000000000000000000
E	000100000000000000000
F	000010000000000000000
G	000001000000000000000
H	000000100000000000000
I	000000010000000000000
K	000000001000000000000
L	000000000100000000000
M	000000000010000000000
N	000000000001000000000
P	000000000000100000000
Q	000000000000010000000
R	000000000000001000000
S	000000000000000100000
T	000000000000000010000
V	000000000000000001000
W	000000000000000000100
Y	000000000000000000010
?	000000000000000000001
0	000000000000000000000
1	111111111111111111111

표 1은 아미노산을 21비트의 2진 코드로 변환해주는 대조표이다.

실험 성능은 Cross-Validation 방식으로 계산한다. 즉 훈련 예를 같은 크기의 5개의 집단으로 나누고, 그중 4개의 집단은 훈련 예로 나머지 1개는 실험 예로 사용하여 성능을 계산하여 가장 좋은 것으로 선택하여 사용하는 방식을 적용한다.

Pediction of α-Helix Region

알파헬릭스 막횡단 부위를 예측하는데 TMHMM과 유사한 은닉 마코브 모델(HMM)을 사용한 모듈을 개발하였다[3,5]. HMM은 Computational Biology에서 게놈의 확률적 구조, 프로틴 패밀리와 유전자 구조 등 성공적으로 많이 사용되어왔다. 기본적인 원리는 각 지역과 단백질의 특별한 부분에 State Set을 정하는 것이다. 생물학에 부합하는 모델을 세우는 것은 모델을 분석하거나 재구성하는데 도움을 준다[5]. 그리고 그런 의미를 지닌 다양한 모델을 실험함으로서 생물학의 의미가 있는 법칙들을 배울 수 있다. 알파헬릭스 형태의 막횡단 단백질 구조는 세포막을 중심으로 Inner Loop와 Outer Loop 그리고 막횡단 지역을 모델링 할 수 있다. 그 각각의 State들에서는 그 지역에 특정한 20개의 아미노산이 나올 확률이 일치된다. 또 각 State들은 생물학적으로나 구조적으로 연결 되어있다.

4) Cutoff 는 threshold 즉, 임계치를 나타낸다.

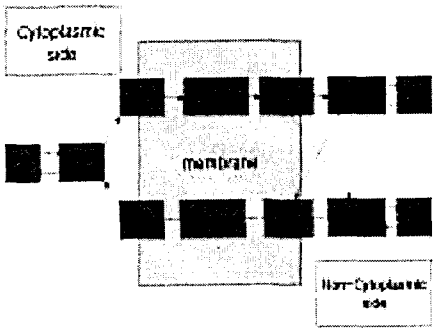

그림 2 HMM의 전체 모델 구조

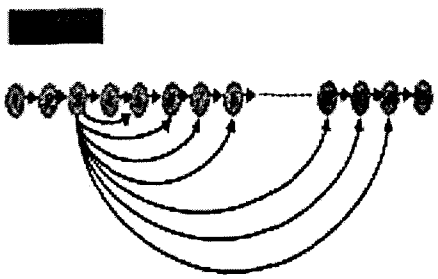

그림 3 막횡단 헬릭스 코어의 states

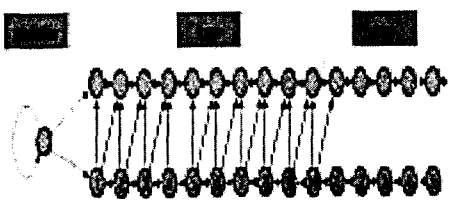

그림 4 Cytoplasmic side와
Non-cytoplasmic side의 states

그림 2,3,4에서 TMHMM의 구조를 보여주고 있다. 그림 2를 보면 막횡단 헬릭스 코어 부분에서 앞부분에 5개의 State, 뒷부분에 5개의 State, 그리고 중간부분에는 25개의 State를 할당했다. 중간부분은 세 번째 State에서 다음 State로 24번째 State까지만 다양하게 전이 할 수 있게 만듦으로서 5에서 25개까지의 State전이가 가능하도록 만들었다. 따라서 막횡단 헬릭스 코어 부분에서 15에서 35개까지의 전이가 가능하다. 그림 4에서 각각의 Loop 에서는 20개의 State를 사다리꼴의 모델을 세움으로서 1개 이상의 전이가 가능하다. 그리고 Loop에서 Globular State로도 전이가 가능하다. Globular State는 자기 자신으로의 전이와 Loop의 전이가 가능한 State이다. 그림 2에서 이름이 같은 부분의 아미노산 분산 확률은 모두 같게 했다. 결국 모델에 적용되는 Parameter는 7개의 부분 * 20 개의 아미노산 분산 확률 + 막횡단 헬릭스 코어 중간부분에서 21개 Loop(자기 자신에게로 가는 State 포함)에서 35*3개로서 총 266개가 된다. 신경망에서는 보통 수천 개의 Parameter가 필요한 점과 현저히 비교된다.

HMM을 기계학습(Machine Learning) 하는데 Maximum Likelihood Estimation 방법을 쓰는 HMM 학습의 대표적인 방법인 Baum-Welch Reestimation을 사용한다. TMHMM은 학습을 단계별로 진행한다.[3] 여기서는 TMHMM에서 했던 'Soft Boundaries'의 방법을 쓰지 않고 원래 데이터 그대로의 레이블을 사용했다. 훈련이 끝난 다음에는 가장 큰 확률을 지닌 패스를 찾기 위해 Viterbi 알고리즘을 사용한다.

훈련 예은 TMHMM에서 사용한 160개의 단백질 서열을 사용했다. 그것은 108개의 *Multi Spanning*과 52개의 *Single Spanning*을 가지고 있다. Cross-Validation을 하기 위해서 TMHMM에서 만든 각 서열이 25%(Needleman-Wunsch alignment에서) 이하의 유사도를 지닌 서열로 나누어진 10개의 Set을 사용했다. TMHMM 에서와 같이 9개의 Set으로 학습을 한 다음 나머지 한 개 Set으로 테스트를 했다. 이와 같은 방법으로 나머지 다른 Set에 대해서 10번 수행하였다. 이 모델은 Cross-Validation을 통해서 160개의 단백질 훈련 예에 대하여 단백질의 모든 위상을 맞게 예측하는데 55.6%의 성능을 나타냈다. TMHMM은 76.9%로 다른 예측기 보다 높은 성능을 보인 이유는 바로 각 토폴로지 경계 부분을 희석을 시키는 단계를 거쳐서 그 서열로 훈련을 하기 때문으로 생각된다.

Prediction Model

본 연구에서는 PASS(Prediction of a-helix Transmembrane Region by Separating ER-Signal Sequence)라는 예측 프로그램을 개발하였다. 이는 위의 두 가지 알고리즘을 통합한 시스템으로 다음과 같은 기능을 갖는다. 첫 번째는 Signal Peptide만을 예측하는 기능이다. 두 번째는 알파헬릭스 막횡단에 해당하는 위상만을 예측하는 기능이다. 세 번째는 Signal Peptide 예측 결과를 알파헬릭스 막횡단 부위 예측기에 전달하여 위상을 예측하는 기능이다. 즉, PASS는 하나의 웹에서 사용자가 한번에 Signal Peptide를 통해 알파헬릭스 막횡단 부위 예측기를 거쳐서 사용자가 원하는 것을 예측하여 주는 시스템이다.

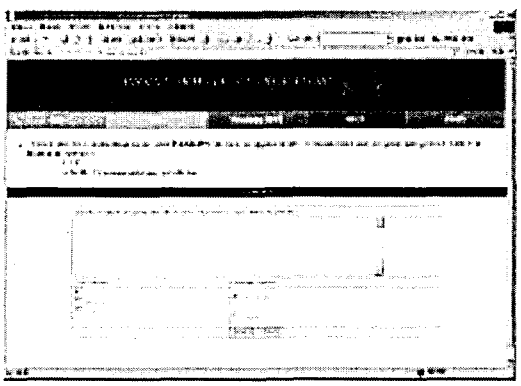

그림 5 PASS 의 초기 화면

그림 5는 PASS를 웹상에서 서비스하는 초기 화면이다.

PASS를 선택하면 Signal Peptide와 막횡단 위상을 함께 예측한 결과를 볼 수 있다. 입력한 서열의 Signal Peptide로 예측된 서열을 alpha-TM으로 전달하게 된다. 때문에 alpha-TM이 Signal peptide를 막횡단으로 잘못 예측하는 경우가 발생하지 않는다.

그림 6은 입력 서열에서 Signal Peptide로 예측된 서열을 제외한 나머지 단백질 서열의 알파헬릭스 막횡단 위상 예측 결과를 텍스트 형식으로 보여준다. 여기서 "i"는 세포막 안쪽을, "M"은 막 횡단을 나타내고 "o"는 세포막 밖을 나타낸다.

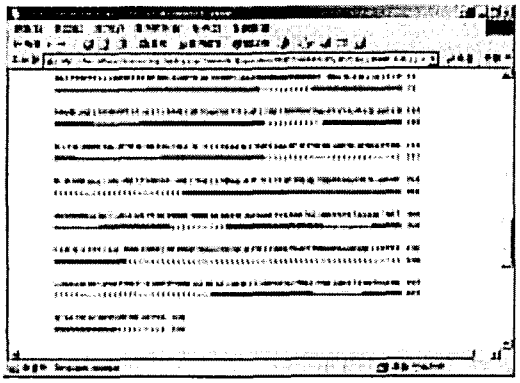

그림 6 막횡단 위상 예측 결과

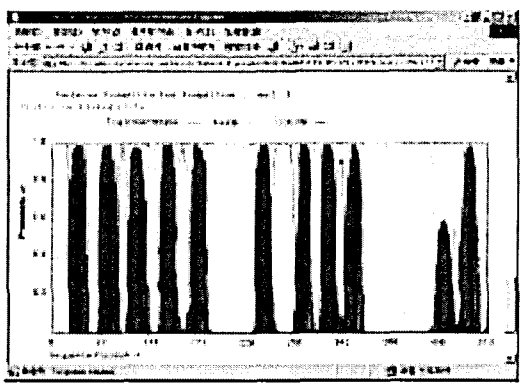

그림 7 막횡단 위상 예측 그래프

그림 7은 Graph Type의 Option을 선택했을 때의 결과이다. 막대그래프와 선 그래프는 단백질 서열의 Posterior 확률을 나타낸다. 빨강은 막횡단 단백질일 확률을, 초록색은 세포막 안쪽일 확률을 나타내고 파랑은 세포막 밖일 확률을 나타낸다. 각각 수치가 제일 높은 곳이 해당하는 지역으로 예측이 된다. 그림 7을 보면 단백질 서열의 알파헬릭스 막횡단 Topology를 한눈에 볼 수 가 있다.

Experiment

PASS에서 Signal Peptide를 고려할 때와 고려하지 않을 때의 막횡단 단백질 예측 성능 비교

아래 표 2 에서 방법 1은 기존의 TMHMM에서 사용되었던 Data Set으로 학습하고 예측 했을 때의 성능이고, 방법 2는 기존의 TMHMM에서 사용되었던 데이터 셋에서 시그날 펩타이드를 갖는 단백질을 제거한 나머지 150개로 훈련을 한 다음 160개의 데이터 셋으로 예측했을 때의 성능이다. 위상 예측의 정확도는 막횡단의 위상과 위치가 모두 맞게 예측된 백분율이고, 위칙 예측 정확도는 막횡단의 위상에 상관없이 막횡단의 부위만 예측된 백분율, Single TM Sensitivity는 찾아낼 것중 제대로 찾아낸 것의 백분율을 의미한다. 마지막으로 Single TM Specificity는 예측된 세그먼트 중 제대로 예측된 세그먼트의 백분율이다. 표 3 결과에서 볼 수 있듯이 Signal Peptide를 갖는 단백질이 훈련 셋으로 쓰인 경우 전체 예측성능이 저하됨을 알 수 있다.

표 2 Signal Peptide를 분리 예측하는 경우와 고려하지 않는 경우의 예측 성능

방법	훈련셋 크기	위상 예측 정확도	위치 예측 정확도	Single TM Sensitivity	Single TM Specificity
1	160	65.0	74.4	96.4	96.2
2	150	68.1	76.3	96.4	96.7

기존 막횡단 단백질 예측 프로그램과 PASS의 성능비교

실험을 통해 입증된 Signal peptide를 갖는 단백질을 SWISS-PROT DB, Möller's TM Data set 과 TMPDB에서 가져와서 CLUSTALW를 이용해 30% 이하의 유사도를 갖는 서열들을 필터링했다. Eukaryote에 속하는 A4_HUMAN, AMD2_XENLA, CD7_HUMAN, GHR_HUMAN, GLPA_HUMAN, GP21_RAT, HA12_MOUSE, MPRD_BOVIN, OSTB_YEAST, PGDR_MOUSE, RIB1_HUMAN, RIB2_HUMAN, RMP1_HUMAN, RMP2_HUMAN, RMP3_HUMAN, STS_HUMAN, GLK2_RAT, FRIZ_DROME, LSHR_RAT과 Prokaryote에 속하는 COAB_BPFD, COAB_BPPF1, COX2_PARDE와 CYOA_ECOLI로 훈련 셋을 만들었다.[1] 표 3은 훈련 예에 각각 Method의 막횡단 Topology 예측 결과이다. No. of TM Segments and Position은 TM Segment의 개수와 위치를, TM Topology는 전체 위상을 의미한다. TM Segment의 위치는 최소 11개의 Residues가 오버랩 되면 맞게 예측하는 것으로 했다. 결과를 보면 PASS가 다른 예측 프로그램에 비해 성능이 월등히 높다는 사

표 3 Signal Peptide가 있는 단백질에 대한 막횡단 단백질 예측 프로그램의 성능 비교

	Prediction Accuracy(%)	
TM Topology Prediction Methods	No. of TM Segments and Position	TM Topology
KKD	26.1	-
TMpred	4.3	4.3
TopPred II	8.7	8.7
DAS	0.0	-
TMAP	21.7	13.0
MEMSAT 2	43.5	39.1
SOSUI	17.4	-
PRED-TMR 2	56.5	-
TMHMM 2.0	69.6	69.6
HMMTOP 2.0	47.8	47.8
PASS	82.6	82.6

실을 알 수 있다. Signal Peptide를 고려하지 않는 다른 예측 프로그램에서는 Signal Peptide를 막횡단 부위로 잘못 예측하는 경우가 생기기 때문에 성능이 떨어진다. Signal peptide를 처리하면 TMHMM의 경우는 PASS와 비슷한 결과가 나온다. PASS에서는 Signal Peptide를 먼저 예측해서 Signal Peptide를 제거하고 막횡단 Topology를 예측하기 때문에 Signal Peptide를 막횡단 부위로 잘못 예측하는 경우가 생기지 않는다.

Discussion

이 논문에서 Signal Peptide가 있는 아미노산 서열이 훈련 예로 사용될 경우 막횡단 Topology를 예측이 저하됨을 알 수 있었다. 또한 예전에는 Signal Peptide와 막횡단 Topology 예측을 따로 해야 하는 번거로움이 발생하였으나, PASS를 이용하면 Signal Peptide를 먼저 예측하고 나머지 아미노산 서열로 막횡단 Topology를 예측하여 보다 작업의 복잡성을 줄일 수 있을 뿐만 아니라 예측의 성능을 높일 수 있다.

References

[1] Demelo M. Lao, Masafumi Arai, Masami Ikeda and Tochio Shimizu (2002), The presence of signal peptide significantly affects transmembrane topology prediction. BIOINFORMATICS Vol 18, pp.1562-1566

[2] Henrik Nielsen, Jacob Engelbrecht (1997), A Neural Network Method for Identification of Prokaryotic and Eukaryotic Signal Peptides and Prediction of their Cleavage Sites. International Journal of Neural System, Vol. 8, Nos. 5 & 6 pp.581-599.

[3] Erik L.L. Sonnhammer , Gunnar Von Heijne And Anders Krogh (1998), A Hidden Markov Model for Predicting Transmembrane Helices in Protein Sequences. In J. Glasgow Et Al., eds., Proc. Sixth Int. Conf. On Intelligent Systems For Molecular Biology, pp.175-182.

[4] Henrik Nielsen, Jacob Engelbrecht, Soren Brunak and Gunnar Von Heijne (1997), Identification of Prokaryotic and Eukaryotic Signal Peptides and Prediction of their Cleavage Sites. Protein Engineering, Vol.10 No.1 pp.1-6

[5] Anders Krogh, Bjorn Larsson, Gunnar Von Heijne And Erik L.L. Sonnhammer (2001). Predicting Transmembrane Protein Topology with a Hidden Markov Model : Application to Complete Genomes. J Mol. Biol., Vol.305, pp567-580