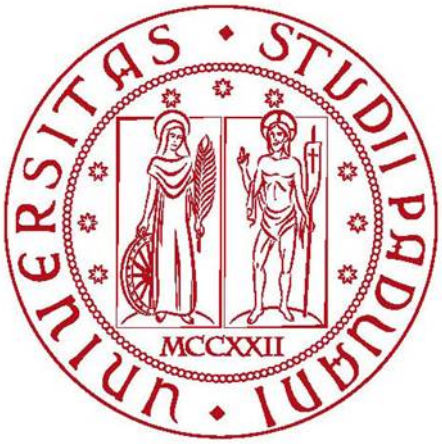

IDENTIFICAZIONE DI LIASI PER LA DEGRADAZIONE DELLA PARETE CELLULARE BATTERICA

Boschetti Leone, Trambaiollo Luca, Valentini Giacomo

Ottobre 2021

1 Introduzione

In questa relazione si propongono tre diversi metodi di codifica delle proteine al fine di identificare le liasi per la degradazione della parete cellulare batterica. Le proteine sono formate da lunghe sequenze dei 20 possibili amminoacidi (l'unità costitutiva della proteina), che si uniscono l'uno all'altro attraverso particolari legami, detti peptidici, a formare delle lunghe catene. La precisa sequenza degli amminoacidi nelle catene determina la forma, la funzione e le caratteristiche della proteina. I metodi proposti hanno l'obiettivo di trasformare una qualsiasi sequenza proteica in un array di lunghezza fissa. Gli array derivati dalle proteine, aventi tutti la stessa dimensione, rappresentano i pattern numerici forniti al classificatore SVM non-lineare della libreria LIBSVM, utilizzato nel nostro caso per sviluppare un nuovo predittore per l'identificazione delle liasi. A tal fine, si è suddiviso il data set di 375 pattern in un Training Set di 177, Validation Set di 94 e Test Set di 104. [Il file.mat contenente il Data set, suddiviso come descritto, è scaricabile al seguente indirizzo Dropbox: Clicca qua].
Solo un metodo dei 3 utilizzerà il Validation set per valutare gli iperparametri migliori. Per gli altri 2, non avendo iperparametri da impostare, non si utilizzerà il Validation set.

2 Metodi di rappresentazione delle proteine

I tre metodi di codifica di una proteina sviluppati sono:

- Metodo proprietà fisico/chimiche
- Metodo "soffuso"
- Metodo con Gaussiana

2.1 Metodo proprietà fisico/chimiche

Questo descrtittore ha lo scopo di codificare la proteina in un vettore di 9 elementi, ognuno rappresentante una diversa proprietà fisico/chimica dei vari amminoacidi componenti la proteina. I valori delle proprietà fisico/chimiche specifici ad ogni amminoacido sono salvati all'interno di una matrice. Le cui colonne corrispondono, nell'ordine, alle seguenti proprietà: Hydrophobicity, Hydrophilicity, Rigidity, Flexibility, Irreplaceability, Mass, pI, pK($\alpha - COOH$), pK($\alpha - NH_3^+$); mentre le righe coincidono con i 20 amminoacidi in ordine alfabetico. Quindi per ogni amminoacido costituente la proteina in esame si somma i valori delle 9 proprietà dello stesso al vettore di output:

$$vect(col) = vect(col) + matrice(n, col),\ 1 \leq col \leq 9,\ 1 \leq n \leq 20 \quad (1)$$

Il seguente metodo ha prestazioni temporali $\mathcal{O}(n)$ con n che sta ad indicare la lunghezza della proteina in esame. Questo algoritmo ha ottenuto un'accuratezza del 84.6154%, riuscendo a classificare correttamente 88 proteine sulle 104 totali del TestSet.

2.1.1 Pseudo-codice

Algorithm 1 Proprietà fisico/chimiche

```
1: function DESCRITTORE(proteina)
2:     mat ← Matrice proprietà fisico chimiche degli amminoacidi
3:     out ← Array di 9 elementi per descrivere la proteina
4:     a ← Stringa contenente i 20 amminoacidi
5:     len ← length(proteina)
6:     for n = 1, ..., 20 do
7:         for ammin = 1, ..., length(proteina) do
8:             if proteina(ammin) == a(n) then
9:                 for col = 1, ..., 9 do
10:                    out(col) ← out(col) + mat(n, col)
11:                end for
12:            end if
13:        end for
14:    end for
15: end function
```

2.2 Metodo "soffuso"

Questo descrittore non è finalizzato ad una rappresentazione univoca ed esatta della disposizione degli amminoacidi nella proteina. Punta invece a fornire una descrizione "soffusa" del posizionamento di questi, tramite il calcolo di media, varianza e numero di occorrenze che caratterizzano la distribuzione di ognuno dei 20 amminoacidi. In particolare restituisce un vettore di lunghezza fissa pari a 60, in cui i primi 20 elementi rappresentano la media di ognuno dei 20 amminoacidi, i secondi 20 la varianza e gli ultimi 20 le varie occorenze all'interno della proteina:

$$vect = [media \;\; varianza \;\; occorrenze] \tag{2}$$

Il seguente metodo ha prestazioni temporali $\mathcal{O}(n)$ con n che sta ad indicare la lunghezza della proteina in esame. Questo algoritmo ha ottenuto un'accuratezza del 84.6154%, riuscendo a distinguere correttamente 88 proteine sulle 104 totali del TestSet. Nota: l'alto numero di dimensioni rispetto al numero di pattern presenti nel Training Set potrebbe causare una bassa

affidabilità nel riscontro delle prestazioni (curse of dimensionality) verificate sul Test Set.

2.2.1 Pseudo-codice

Algorithm 2 Metodo "soffuso"

```
function DESCRITTORE(proteina)
    alf ← Stringa contenente la codifica alfabetica dei 20 amminoacidi
    media ← Inizializzazione array di 20 elementi rappresentanti la media di ogni amminoacido
    varianza ← Inizializzazione array di 20 elementi rappresentanti la varianza di ogni amminoacido
    occorrenze ← Inizializzazione array di 20 elementi rappresentanti il numero di occorrenze di ogni amminoacido
    for n = 1, ..., 20 do
        lis ← Inizializzazione array vuoto
        for amminoacido = 1, ..., length(proteina) do
            if proteina(amminoacido) == alf(n) then
                lis(end + 1) ← amminoacido
            end if
        end for
        if length(lis) > 0 then
            media(n) ← sum(lis)/length(lis)
            varianza(n) ← sum((lis − media(n))^2)/length(lis)
            occorrenze(n) ← length(lis)
        else
            media(n) ← 0
            varianza(n) ← 0
            occorrenze(n) ← 0
        end if
    end for
    vettore ← [ media varianza occorrenze ]
end function
```

2.3 Metodo con Gaussiana

Questo descrittore è sviluppato a partire dal metodo "Amino-Acid Composition (AS)" illustrato in "An Empirical Study of Different Approaches for Protein Classification" (Nanni L., Lumini A., Brahnam S., 2014). Si è pensato di modificare quest'ultimo in modo che tenga conto anche della disposizione degli amminoacidi all'interno della sequenza proteica. Come primo approccio, si è fatto ciò sommando le posizioni all'interno della proteina per ognuno dei 20 amminoacidi invece di contare le sole occorrenze. Questo però lascia ancora un' importante incertezza sulla disposizione univoca degli amminoacidi all'interno della proteina, in quanto differenti combinazioni di posizione possono dare lo stesso risultato di somma. Si è cercato quindi di contrastare questo fenomeno sommando il valore di una funzione esponenziale, di variabile la posizione dell'amminoacido nella sequenza proteica, invece che la sola posizione. In conclusione si è preferito adottare una curva gaussiana al posto di una funzione esponenziale, avendo osservato che le prestazioni erano migliori e che si evitava l'esplosione numerica conseguente all'utilizzo di una funzione esponeziale. La funzione gaussiana è in funzione della posizione dell'amminoacido normalizzata sulla lunghezza della proteina (posizione/lunghezza proteina). Si ha quindi un vettore di lunghezza 20, dove ogni elemento è la somma dei valori della gaussiana per un dato amminoacido:

$$vect(n) = vect(n) + gauss(am/lung, media, sigma) * lung,\ 1 \leq n \leq 20 \quad (3)$$

La funzione "gaussiana" sviluppata riceve in ingresso la posizione relativa dell'amminoacido e restituisce il corrispondente valore della funzione gaussiana caratterizzata da una certa media ("media") e deviazione standard ("sigma"). Il seguente metodo ha prestazioni temporali $\mathcal{O}(n)$ con n che sta ad indicare la lunghezza della proteina in esame. Per testare questo metodo, a differenza dei due precedenti, è stato necessario utilizzare anche il Validation set, per impostare i valori dei due iperparametri (media, deviazione standard) in modo da ottenere le migliori prestazioni.
Questo algoritmo ha ottenuto un'accuratezza del 84.043% (classificando correttamente 79/94 pattern) come prestazione migliore sul Validation set assegnando agli iperparametri i valori: media = 15, deviazione = 15 (Figura 2.3). Come si può osservare nella Figura 2.3 c'è un' ampia area di valori per cui si ottiene la massima prestazione. I valori scelti per media e deviazione standard riportati sopra sono stati presi approssimativamente nel mezzo di

questa area. Per lo stesso valore degli iperparametri si è ottenuta in seguito un'accuratezza del 87.5% sul Test set.

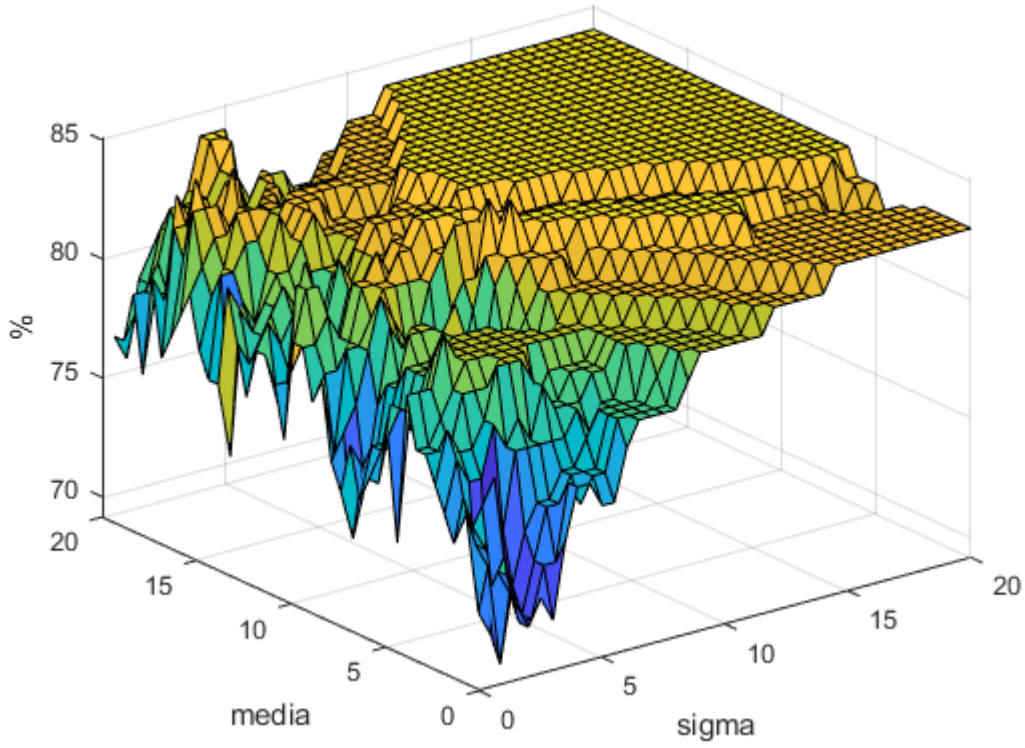

Figure 1: andamento delle prestazioni al variare degli iperparametri sul Validation set

2.3.1 Pseudo-codice

Algorithm 3 Metodo gaussiana

```
    function DESCRITTORE(proteina)
2:      alf ← Stringa contenente la codifica alfabetica dei 20 amminoacidi
        som ← Inizializzazione array di 20 elementi
4:      lung ← length(proteina)
        for n = 1, ..., 20 do
6:          for amminoacido = 1, ..., length(proteina) do
                if proteina(amminoacido) == alf(n) then
8:                  som(n) ← som(n)+gaussiana(amminoacido/lung, media, sigma)*
    lung
                end if
10:         end for
        end for
12: end function
```

Algorithm 4 Gaussiana

```
    function GAUSSIANA(percentuale, media, sigma)
2:      perc ← percentuale * 20
        peso ← ((1/sqrt(2 * pi * (sigma^2))) * exp(−((perc − media)^2)/(2 *
    (sigma^2)))) * 70
4: end function
```

3 Valutazione delle prestazioni

Per fornire un metodo di valutazione delle prestazioni più intuitivo e di più facile comprensione, abbiamo utilizzato i seguenti criteri: la sensibilità (Sn), la specificità (Sp), il coefficiente di correlazione di Matthews (MCC), l'accuratezza complessiva (OA), e l'accuratezza media (AA), che sono state definite come:

$$Sn = \frac{TP}{TP + FN} \tag{4}$$

$$Sp = \frac{TN}{TN + FP} \tag{5}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TN + FN) \times (TP + FN) \times (TN + FP)}} \tag{6}$$

$$OA = \frac{TP + TN}{TP + FN + TN + FP} \tag{7}$$

$$AA = \frac{1}{2} \times (\frac{TP}{TP + FN} + \frac{TN}{TN + FP}) \tag{8}$$

Dove TP (True Positive) è il numero di liasi correttamente classificate, TN (True Negative) è il numero di non-liasi correttamente classificate, FP (False Positive) il numero di non-liasi che sono state classificate come liasi, FN (False Negative) il numero di liasi che sono state classificate come non-liasi.

4 Risultati

Metodo	Sn(%)	Sp(%)	MCC	OA(%)	AA(%)
Metodo proprietà fisico/chimiche	11.76	98.85	0.2345	84.62	55.31
Metodo "soffuso"	58.82	89.66	0.4639	84.62	74.24
Metodo con Gaussiana	35.29	97.70	0.4579	87.50	66.50

Table 1: Prestazioni dei metodi da noi sviluppati; su Data set diviso in Training, Validation e Test

Metodo	Sn(%)	Sp(%)	MCC	OA(%)	AA(%)
Metodo proprietà fisico/chimiche	8.82	100	0.2709	83.42	54.41
Metodo "soffuso"	41.18	92.16	0.3716	82.89	66.67
Metodo con Gaussiana	-	-	-	-	-
PAAC [1]	76.47	93.16	0.678	90.13	84.82

Table 2: Prestazioni dei metodi da noi sviluppati e PAAC [1]; su Data set diviso in soli Training e Test come in [1]

5 Conclusione

Come riportato nella Tabella.1 (1), il metodo "soffuso", tra i 3 sviluppati, ottiene le migliori prestazioni in 3 dei 5 criteri considerati (Sn, MCC, AA). Mentre gli altri due metodi si dimostrano migliori a identificare correttamente un maggior numero di non-liasi (Sp), ma per l'applicazione dei nostri metodi è più interessante avere una sensibilità (Sn) elevata che una specificità (Sp) elevata. Questo perché il nostro fine è l'identificazione di liasi, che potranno essere successivamente sottoposte a verifica sperimentale. Si verificherà quindi quali tra queste siano Falsi Positivi (FP). Ma le liasi scartate perché classificate come non-liasi (Falsi Negativi (FN)) rischiano di non essere più recuperate e andar perse. Quindi l'interesse principale è evitare di

scartare liasi identificandole erroneamente come non-liasi, applicando nel contempo un'importante scrematura tra tutte le proteine esaminate. Si osserva poi che il metodo fisico/chimico ha ottenuto prestazioni piuttosto comparabili tra il Data set nuovo (TR, VA, TE) e quello originale (preso dall'articolo [1]), mentre il metodo "soffuso" ha riscontrato un forte calo nella sensibilità (Sn). Anche altri parametri (MCC, AA) del metodo "soffuso" sono peggiorati notevolmente, questo è dovuto alla forte correlazione che questi due hanno con la sensibilità. Questa accentuata variabilità nelle prestazioni potrebbe essere dovuta, oltre ad una maggior difficoltà intrinseca del data set, all'elevato numero di dimensioni che caratterizza il metodo "soffuso" (curse of dimensionality), conferendogli una maggiore incertezza nelle prestazioni ottenute. Infine, si fa notare che gli iperparametri nel metodo della Gaussiana sono stati impostati valutando solo l'accuratezza complessiva (OA). Si potrebbe allora valutare gli iperparametri da impostare su altri criteri in base ai nostri scopi. Come la sensibilità (Sn), più interessante ai nostri fini. Si può osservare nelle due tabelle (Tabella 1 e Tabella 2) come nessuno dei tre metodi da noi sviluppati si avvicina alle prestazioni ottenute con PAAC [1] (a parte per Sp). Anche se per il metodo con Gaussiana non è possibile il confronto diretto con PAAC [1], è possibile un confronto indiretto attraverso il confronto con gli altri due metodi da noi sviluppati (Tabella 1).

6 Bibliografia

[1] Research article - Identification of Bacterial Cell Wall Lyases via Pseudo Amino Acid Composition - Xin-Xin Chen, Hua Tang, Wen-Chao Li, Hao Wu, Wei Chen, Hui Ding, and Hao Lin - 2016