CRISPRLearner

A deep learning-based system to predict CRISPR/Cas9 sgRNA on-target cleavage efficiency

UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

SERLAB
Software Engineering Research
UNIVERSITÀ DEGLI STUDI DI BARI

Relatore
Prof. Giovanni Dimauro

Laureando
Pierpasquale Colagrande

Sommario

- Introduzione
- Descrizione del problema
- Concetti chiave
- Panoramica del sistema
- Discussione dei risultati
- Conclusione e sviluppi futuri

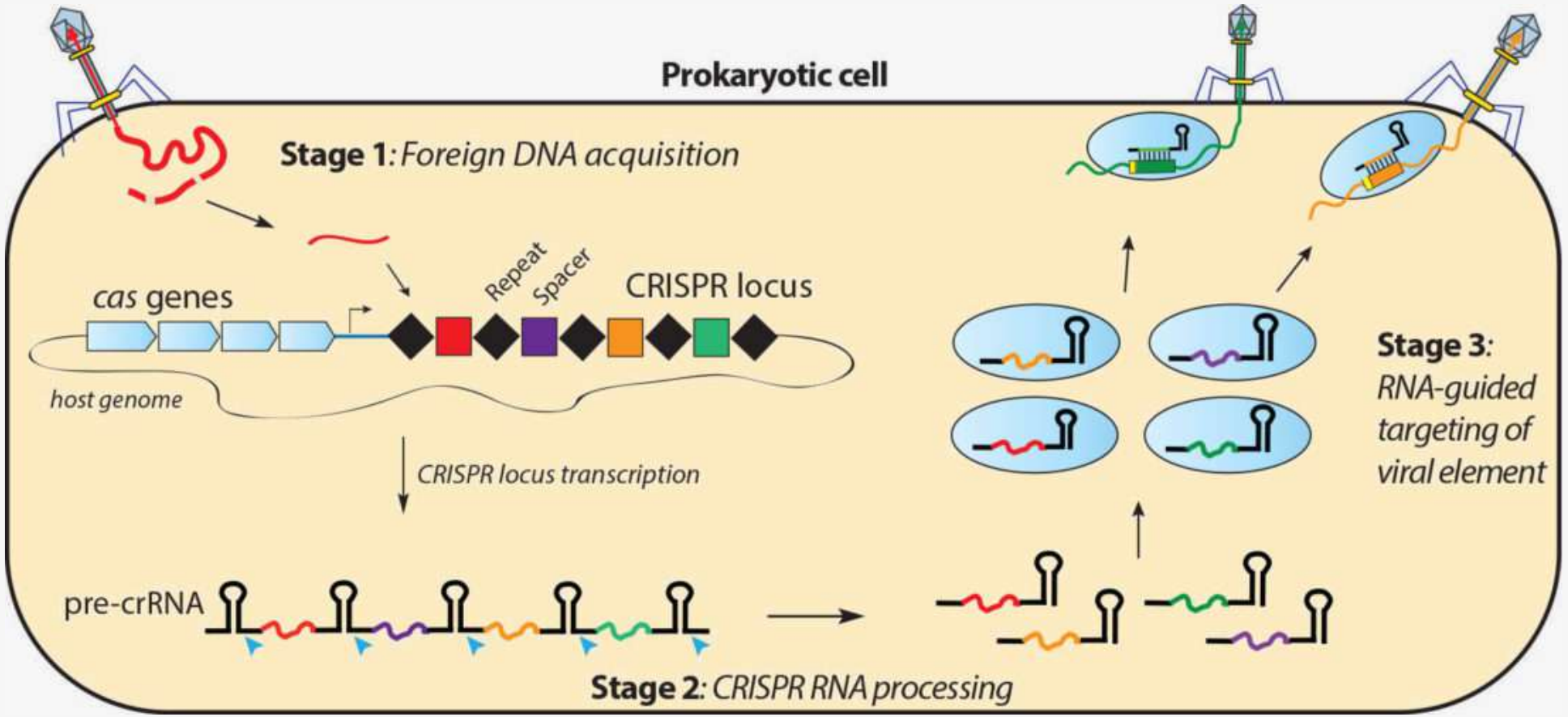
Introduzione

CRISPR è un sistema immunitario adattivo inizialmente scoperto in batteri e archei

Il batterio dedica una parte del suo codice genetico alla memorizzazione di sequenze prelevate da DNA di virus che avevano, in passato, attaccato la cellula

Quanto uno di questi virus attacca nuovamente il batterio, esso utilizza la corrispondente sequenza virale, in accoppiata con una proteina chiamata Cas9 ed un'ulteriore sequenza, per cercare, trovare e tagliare il DNA virale all'interno della cellula, eliminando in questo modo la minaccia

La proteina Cas9 è in fatti in grado di despiralizzare e tagliare un filamento di DNA se accoppiata con una sequenza di RNA, che la guida nel punto del taglio



Funzionamento del sistema immunitario CRISPR

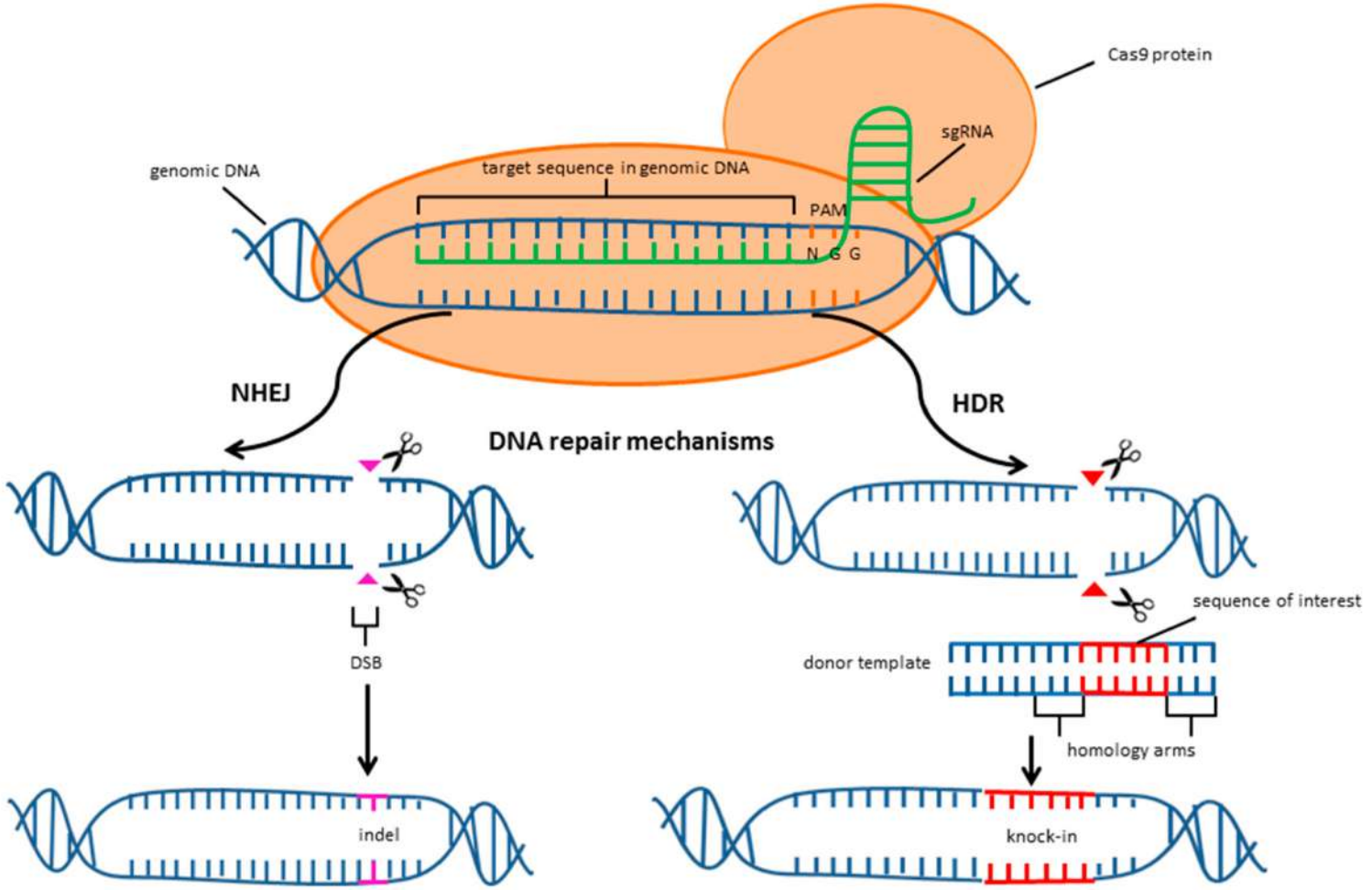
Introduzione

Jennifer Doudna e Emmanuelle Charpentier hanno reingegnerizzato la endonucleasi Cas9 per poter disattivare una qualsiasi DNA specificato da una singola sequenza di RNA accoppiata alla proteina stessa, detta sgRNA (single guide RNA)

L'sgRNA guida la proteina nel punto preciso del DNA obiettivo, mentre la proteina Cas9 despiralizza e taglia tale DNA

Modificando la sequenza sgRNA, è possibile quindi prendere di mira e disattivare una qualsiasi sequenza di DNA

Si viene a creare quindi un potente sistema di editing genomico chiamato CRISPR/Cas9



Editing genomico con CRISPR/Cas9

Descrizione del **problema**

Non tutte le sequenze sgRNA progettate per disattivare uno specifico DNA obiettivo sono egualmente efficienti

Per questo motivo, sono stati sviluppati diversi sistemi per prevedere e quantificare l'efficacia di un sgRNA, misura che può essere estratta solo con la sperimentazione diretta

Prevedere tale efficienza evita al ricercatore il processo di estrazione di quest'ultima mediante sperimentazione diretta, evitando quindi di procedere per tentativi modificando la sequenza in relazione ai risultati ottenuti tramite sperimentazione

I sistemi basati su deep learning si sono dimostrati superiori rispetto ai concorrenti

Concetti **chiave**

Il machine learning è una branca dell'informatica che studia come migliorare le prestazioni di un algoritmo in un particolare compito

Tale algoritmo migliora le sue prestazioni autonomamente, imparando con l'esperienza

Il problema del machine learning è quello di non poter elaborare i dati in forma grezza, ma sono necessarie delle caratteristiche, dette feature, che vengono essere estratte manualmente a partire dai dati da uno più soggetti competenti nel dominio applicativo

Nel deep learning, invece, questa fase viene automatizzata, rappresentando i dati con feature map tridimensionali su più livelli di astrazione

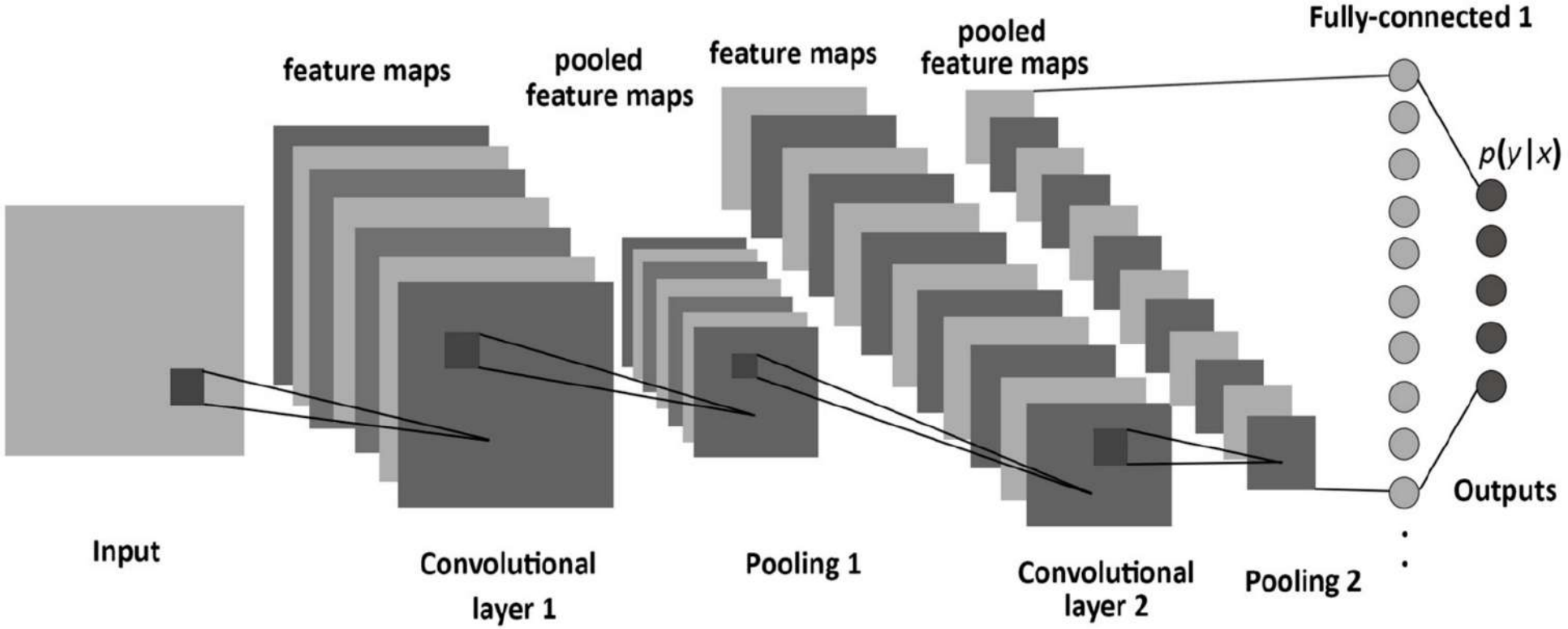
Concetti **chiave**

Le reti neurali sono strutture artificiali ispirate alle reti neurali biologiche, composte da neuroni artificiali ed interconnessioni fra essi

Come all'interno di una rete neurale biologica, un neurone artificiale raccoglie dei dati di ingresso da altri neuroni, li elabora e li restituisce ad altri neuroni come dati d'uscita

Le reti neurali trovano applicazione nell'ambito della computer vision sotto forma reti neurali convoluzionali, particolari strutture che analizzano le immagini individuando gli elementi che le compongono

Le reti neurali hanno però bisogno di essere addestrate mediante una procedura detta training, che serve a tutti gli effetti ad allenare la rete neurale a svolgere bene il compito per cui è stata progettata, per poi testarla per verificarne le potenzialità



Una rete neurale convoluzionale

Panoramica del **sistema**

Il sistema realizzato consente di calcolare l'efficienza di sequenze sgRNA fino ad una lunghezza massima di 30 basi a partire da 10 differenti modelli, addestrati su 10 differenti dataset estratti dal dataset Haeussler, ognuno riguardante una tipologia di genoma o cellula

Tuttavia sono stati necessari alcuni accorgimenti nella gestione dei dati dei vari dataset e nella preparazione e rappresentazione di questi ultimi per la fase di addestramento

Il sistema è anche stato progettato nell'ottica della futura espansione da parte di coloro che volessero addestrare altri modelli utilizzando un loro dataset

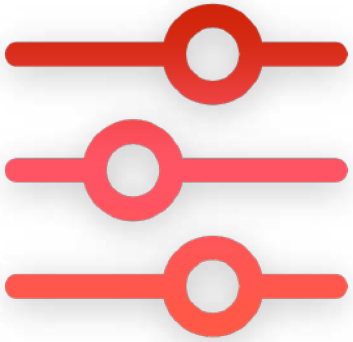
Il sistema presenta un'interfaccia grafica command-line che mostra un menù e consente all'utente di scegliere fra 5 diverse operazioni

- Addestrare 10 modelli a partire dai 10 dataset contenuti nell'Haeussle
- Addestrare un modello a partire da un proprio dataset
- Prevedere l'efficienza di una sequenza sgRNA utilizzando uno dei modelli addestrati
- Spiegazione dei comandi
- Uscita dal programma

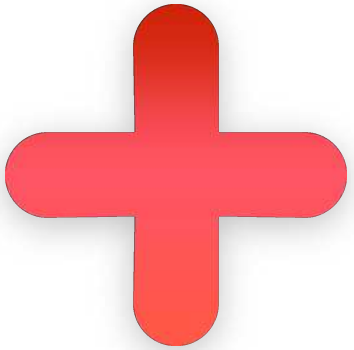
Nel caso in cui l'utente scelga la prima o la seconda opzione, il sistema fornirà la possibilità di salvare i modelli addestrati

La terza opzione apparirà nel menù soltanto se il sistema rileva che sono stati salvati dei modelli addestrati, consentendo quindi all'utente di scegliere quale di essi utilizzare in relazione alla tipologia di cellula di cui ci si sta occupando

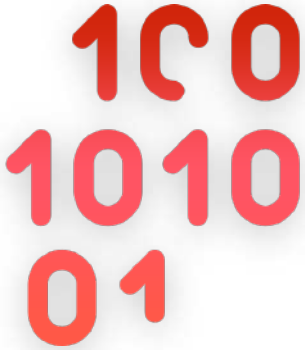
Estrazione dei dataset



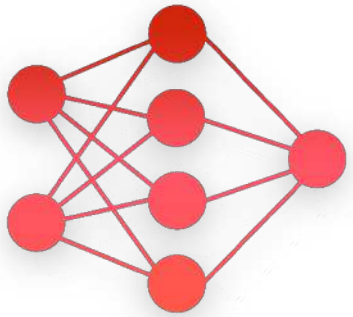
Normalizzazione dei dati



Data augmentation



Data encoding per il training



Training della rete neurale

La fase di estrazione dei dataset viene eseguita quando l'utente esegue il primo comando mentre viene saltata quando il l'utente esegue il secondo comando

Il dataset Haeussler contiene diversi dataset mischiati fra di loro, per questo motivo occorre una fase di estrazione di quelli di nostro interesse

Presenta diverse colonne che caratterizzano ogni sequenza, ma quelle utilizzate sono state *dataset*, contenente il nome del dataset a cui appartiene la sequenza, *seq*, che contiene la sequenza sgRNA, *modFreq*, che contiene il valore di efficienza e *longSeq100Bp*, che contiene la sequenza estesa composta da 100 basi

Utilizzando i nomi dei dataset contenuti all'interno della colonna *dataset*, vengono creati 10 file corrispondenti ai 10 dataset da estrarre, estratti a partire dai loro nomi, utilizzando le altre 3 colonne indicate sopra per estrarne le informazioni importanti

La lunghezza delle sequenze in termini di numero di basi e la scala di misurazione dell'efficienza sono differenti da dataset a dataset

Occorre quindi standardizzarli e normalizzarli ad una misura comune

La fase di estrazione dei dati dal dataset Haeussler è accompagnata da quella di standardizzazione della lunghezza, poiché la rete neurale accetta sequenze di 30 basi mentre tutti i dataset presentano sequenze di lunghezza inferiore a tale misura (di solito intorno alle 22-23 basi)

Per standardizzare la lunghezza delle sequenze, si è individuata la sequenza contenuta nella colonna *seq* all'interno della sequenza contenuta nella colonna *longSeq100Bp*, estraendola insieme alle basi mancanti a partire dalla base antecedente la sequenza *seq* del numero di posizioni corrispondente al numero di basi mancanti alla sequenza *seq* per raggiungere una sequenza di lunghezza 30 basi

seq

GTACTCCAGCGCGCGGCTCACGG

longSeq100Bp

CTTGATGCGGCGCGGGGTGCGCTTGGAGTTGTACTCCAGCGCGCGGCTCACGG
CCCAGTAGCGGTCCAGGCTGATGGCGCACAGGTGCACGATGGACGAG

Viene trovata la sequenza *seq* (di lunghezza 23) in *longSeq100Bp* per estrarre la sequenza di 30 basi a partire dalla settima posizione antecedente *seq* in *longSeq100Bp*

CTTGATGCGGCGCGGGGTGCGCTTGGAGTTGTACTCCAGCGCGCGGCTCACGG
CCCAGTAGCGGTCCAGGCTGATGGCGCACAGGTGCACGATGGACGAG

Sequenza estratta

TGGAGTTGTACTCCAGCGCGCGGCTCACGG

Per ogni sequenza di ogni dataset viene effettuata tale operazione di standardizzazione ed ogni sequenza viene salvata nel file corrispondente al dataset a cui appartiene insieme alla relativa efficienza *modFreq*

Dopo tale operazione, quindi, saranno stati creati 10 file corrispondenti ai 10 dataset estratti, con le sequenze standardizzate

Tali file saranno quindi composti da due colonne, quella della sequenza e quella della relativa efficienza

Tale operazione di standardizzazione della lunghezza, così come la fase di estrazione del dataset, non viene effettuata nel caso in cui l'utente decida di addestrare un modello basandosi sul proprio dataset, poiché si da per scontato che tale dataset sia necessariamente nella stessa forma dei file estratti dal dataset, quindi con solo due colonne contenenti la sequenza e la relativa efficienza, dove la sequenza deve necessariamente essere già una sequenza di 30 basi

Per standardizzare le efficienze all'interno di un dataset in un intervallo [0,1] il sistema utilizza uno scaler MinMax, funzione che sfrutta i valori minimo e massimo di efficienza del dataset

La funzione in questione è $f_{nk} = \frac{f_k - f_{min}}{f_{max} - f_{min}}$, dove f_{max} e f_{min} sono rispettivamente il valore massimo e il valore minimo di efficienza del dataset, f_k è il valore originale di efficienza e f_{nk} è il valore di efficienza standardizzato

Tale procedura viene eseguita per ogni dataset, ottenendo quindi 10 file composti da una colonna con le sequenze standardizzate a 30 basi e una colonna con le rispettive efficienze standardizzate

Questa procedura viene effettuata anche nel caso in cui l'utente fornisca un suo dataset per addestrare un suo modello

All’interno della sequenza di 30 basi, viene selezionata una sotto sequenza di 23 basi estratta a partire dall’ottava posizione all’interno della sequenza di 30 basi

Nelle prime due basi della sotto sequenza vengono generate mutazioni alternando ogni base azotata (A, C, G e T), ottenendo quindi 16 sequenze tra le quali sarà presente la sequenza originale, basandoci sul principio che delle mutazioni nelle prime 2 posizioni all’interno di una sequenza sgRNA di 23 basi non hanno alcun effetto sull’efficienza della sequenza

Questa procedura viene effettuata per ogni sequenza di ogni dataset, andando così a generare 10 file che avranno una colonna con le sequenze aumentate (tra cui saranno comprese quelle originali) e le rispettive efficienze

Viene effettuata questa procedura anche quando un utente carica un suo dataset per addestrare un modello

TGGAGTTGTACTCCAGCGCGCGGCTCACGG

TGGAGTTAAACTCCAGCGCGCGGCTCACGG
TGGAGTTACACTCCAGCGCGCGGCTCACGG
TGGAGTTAGACTCCAGCGCGCGGCTCACGG
TGGAGTTATACTCCAGCGCGCGGCTCACGG
TGGAGTTCAACTCCAGCGCGCGGCTCACGG
TGGAGTTCCACTCCAGCGCGCGGCTCACGG
TGGAGTTCGACTCCAGCGCGCGGCTCACGG
TGGAGTTCTACTCCAGCGCGCGGCTCACGG
TGGAGTTGAACTCCAGCGCGCGGCTCACGG
TGGAGTTGCACTCCAGCGCGCGGCTCACGG
TGGAGTTGGACTCCAGCGCGCGGCTCACGG
TGGAGTTGTACTCCAGCGCGCGGCTCACGG
TGGAGTTTAACTCCAGCGCGCGGCTCACGG
TGGAGTTTCACTCCAGCGCGCGGCTCACGG
TGGAGTTTGACTCCAGCGCGCGGCTCACGG
TGGAGTTTTACTCCAGCGCGCGGCTCACGG

Le sequenze vengono codificate per essere utilizzate come input per la rete neurale mediante una tecnica di one-hot encoding, utilizzando 4 canali per le 4 possibili basi azotate

La matrice corrispondente sarà quindi una matrice a 4 righe e 30 colonne, ma se la sequenza da codificare presenta una lunghezza inferiore alle 30 basi, all'inizio della matrice verranno aggiunte tante colonne di zero quante sono le basi mancanti

La fase di encoding viene effettuata sulle sequenze dei 10 dataset quando l'utente sceglie di addestrare i 10 modelli sui 10 dataset estratti dall'Haeussler, sul dataset fornito dall'utente quando egli vuole addestrare un suo modello e sulla sequenza fornita dall'utente quando egli vuole prevederne l'efficienza utilizzando un modello addestrato

Per questo motivo, i risultati ottenuti dalla codifica non vengono salvati su file come avviene per le fasi precedenti

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
								G	C	C	G	A	G	T	A	C	T	G	G	C	C	G	C	G	C	G	G	C	G	G
A channel	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C channel	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	1	1	0	1	0	1	0	0	1	0	0
G channel	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	1	1	0	0	1	0	1	0	1	1	0	1	1
T channel	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Encoding di una sequenza di 23 basi

La rete neurale convoluzionale viene successivamente addestrata utilizzando l'80% dei dati per il training e il restante 20% per il testing

La rete neurale convoluzionale presenta:

- un layer di input che accetta come input una matrice 4x30x1
- un layer di convoluzione che effettua 50 convoluzioni con un filtro 4x4, producendo 50 feature map di dimensione 1x27, seguito da un layer di attivazione ReLU
- un layer di pooling che effettua un max pooling con un kernel di dimensione 1x2, producendo 50 feature map di dimensione 1x13
- un layer flatten che combina le feature map in un unico vettore di 650 elementi
- due fully connected layer composti entrambi da 128 unità, ognuno dei due seguito da un layer di attivazione ReLU
- fra questi due layer, un layer di dropout con un dropout rate di 0.3, per
- un layer fully connected composto da un unico nodo seguito da layer di regressione lineare, che calcola l'output

Discussione dei **risultati**

I modelli sono stati valutati utilizzando l'errore quadratico medio come funzione di perdita e il coefficiente di Spearman

I risultati ottenuti sono stati piuttosto soddisfacenti e superiori rispetto ai risultati dei sistemi concorrenti, nonostante i modelli costruiti siano stati 10 e non sia stato costruito un unico modello generalizzato

Si è scelto questo approccio perché un modello generalizzato avrebbe portato dei risultati inferiori

Inoltre, il sistema è stato dotato della possibilità di addestrare e salvare nuovi modelli su nuovi dataset, per consentirne l'espandibilità a più tipologie di cellule e genomi

Discussione dei **risultati**

Dataset	CRISPRLearner	DeepCas9	sgRNA Designer (rule set I)	SSC	sgRNA Scorer	sgRNA Designer (rule set II)
Chari	0.98	0.49	0.25	0.29	0.48	0.38
Wang/Xu	0.99	0.61	0.34	0.49	0.32	0.48
Doench mouse-EL4	1.00	0.59	0.58	0.4	0.4	0.7
Doench A375	1.00	0.38	0.27	0.29	0.24	0.54
Hart	0.99	0.41	0.29	0.29	0.21	0.38
Moreno-Mateos	1.00	0.23	0.04	0.17	0.14	0.12
Gandhi	1.00	0.32	0.24	0.15	0.25	0.42
Farboud	0.98	0.57	0.3	0.55	0.6	0.54
Varshney	1.00	0.3	0.14	0.17	0.28	0.22
Gagnon	1.00	0.25	-0.07	0.18	0.2	0.1

Comparazione di CRISPRLearner con altri sistemi concorrenti (Spearman)

Conclusione e sviluppi **futuri**

Preferire più modelli specializzati piuttosto che un modello generalizzato si è rivelato essere una strategia vincente per ottenere alti risultati, così come la fase di aumento dei dati

Il sistema può essere espanso aggiungendo e addestrando ulteriori modelli per ulteriori tipologie di cellule e genomi, rendendolo così capace di prevedere efficienze rispetto ad un numero maggiore di cellule

Inoltre, la rete e il funzionamento generale possono essere riadattati per aggiungere la possibilità di prevedere anche il profilo off-target, ovvero la gravità delle mutazioni causate dal taglio e dalla conseguente riparazione effettuata dalla cellula

Il sistema può anche essere reso disponibile online in modo da creare un sistema più accessibile, soprattutto per quanto riguarda l'espansione e l'aggiunta di modelli addestrati

Grazie per l’attenzione

UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

SERLAB
Software Engineering Research
UNIVERSITÀ DEGLI STUDI DI BARI