

# Predizione delle regioni regolatorie attive in specifiche linee cellulari mediante metodi di deep learning

---

Andrea Pennati, Alessandro Beranti, Samuele Valente

1 febbraio 2022

## Sommario

La previsione della posizione delle regioni regolatorie nelle aree non codificanti del DNA rimane una sfida aperta nel campo della biologia computazionale e della genomica. Al fine di determinare le regioni regolatorie e classificarle come attive o inattive sono stati effettuati diversi studi mediante l'utilizzo di *Feed Forward Neural Network*, *Convolutional Neural Network* e *Multimodal Neural Network*. Lo scopo di questo studio è quello di riuscire a classificare attraverso tecniche di *machine learning* le regioni regolatorie come attive o inattive a partire da specifiche linee cellulari; in particolare sono state esaminate le linee K562, MCF-7 e A549.

## 1 Introduzione

Il genoma umano è composto da 3.2 miliardi di paia di basi di DNA contenenti all'incirca 20.000 geni codificanti per proteine, mentre la restante percentuale è ancora oggetto di studio. Le parti di DNA non codificanti, definite come regioni regolatorie, sono particolarmente importanti per il processo di trascrizione delle aree codificanti. Queste regioni hanno la capacità di incrementare o decrementare l'espressione di uno specifico gene all'interno dell'organismo. Esistono due tipologie di regioni: le *cis-regulatory regions* (CRR) e le *trans-regulatory regions* (TRR). In questo studio sono state esaminate le CRR. Esse sono composte da due elementi:

- *Promoter*<sup>1</sup>: sequenze di DNA che permettono alle proteine di legarsi in modo da iniziare la trascrizione di un settore di DNA. Solitamente queste sequenze sono lunghe 100-1000 paia di basi [4].
- *Enhancer*<sup>2</sup>: sequenze di DNA che svolgono funzioni aumentando notevolmente, fino a mille volte, la frequenza di trascrizione del gene che controllano. La loro struttura è simile a quella di un *promoter*, ma la loro densità è maggiore in quanto la loro lunghezza media è circa 100 paia di basi [1].

La classificazione di *promoter* ed *enhancer* è stata effettuata prendendo in esame le seguenti linee cellulari:

- **K562**: rappresenta la prima linea cellulare di un umano affetto da leucemia mieloide. Queste cellule sono del tipo eritroleucemia e la linea cellulare deriva da una paziente femmina di 53 anni con leucemia mieloide cronica <sup>3</sup> in crisi blastica,
- **MCF-7**: rappresenta una linea cellulare riguardante un tumore al seno. Questa linea cellulare è stata isolata nel 1970 da una donna bianca di 69 anni,
- **A549**: rappresenta una linea cellulare riguardante cellule epiteliali basali alveolari umane adenocarcinomiche e costituiscono una linea cellulare sviluppata per la prima volta nel 1972. Questa linea cellulare e' stata isolata da un tumore polmonare di un maschio caucasico di 58 anni.

## 2 Setup Sperimentale

L'obiettivo di questo lavoro è lo studio di due differenti *task*:

- *active enhancers vs inactive enhancers (AEvsIE)*,
- *active promoters vs inactive promoters (APvsIP)*

utilizzando il dataset *HG38*. Gli esperimenti sono stati eseguiti su macchina locale attraverso l'utilizzo di *Jupyter* <sup>4</sup> e di due GPU: *NVIDIA GTX 1060 6GB* e *NVIDIA RTX 2060 6GB*. Inoltre è stato usato *Google Colab* <sup>5</sup> per generare i grafici con *t-SNE* in quanto il carico computazionale risultava troppo oneroso per l'hardware a disposizione.

---

<sup>1</sup><https://www.genome.gov/genetics-glossary/Promoter>

<sup>2</sup><https://it.wikipedia.org/wiki/Enhancer>

<sup>3</sup>[https://en.wikipedia.org/wiki/Chronic\\_myelogenous\\_leukemia](https://en.wikipedia.org/wiki/Chronic_myelogenous_leukemia)

<sup>4</sup><https://jupyter.org/>

<sup>5</sup><https://colab.research.google.com/>

## 2.1 Task

L'obiettivo è determinare le regioni attive di *enhancer* e *promoter* nelle celle discusse in [Introduzione]. Ogni *task* è stato eseguito usando una *window size* di 256. Ogni *window* determina il numero di nucleotidi associati alla regione di *promoter* o *enhancer* in questione. Per ogni task sono stati usati tre differenti modelli:

- *Feed Forward Neural Network*,
- *Convolutional Neural Network*,
- *Multimodal Neural Network*.

Il modello *FFNN* è stato applicato solo ai dati epigenomici. Ai dati di sequenza, in quanto categorici, è stato invece applicato il modello *CNN* in quanto usualmente utilizzato per l'analisi e il riconoscimento di pattern all'interno di un'immagine. Il modello *MNN* è stato applicato a entrambi i tipi di dati.

## 2.2 Data Retrieval

Durante questo studio sono stati utilizzati sia dati epigenomici <sup>6</sup> sia dati sequenziali.

I dati epigenomici rappresentano termini formati da vettori numerici, i quali sono composti da misure di caratteristiche specifiche di una determinata regione. L'estrazione di questi dati è stata effettuata mediante l'utilizzo di *ENCODE*<sup>7</sup>: un'encyclopedia di elementi di DNA il cui scopo è quello di creare una serie di registri di candidati di CRR.

I dati sequenziali sono una rappresentazione letterale dei nucleotidi che compongono il DNA. Questi dati sono stati scaricati attraverso *UCSC Genome Browser*<sup>8</sup>: un motore di ricerca genomico messo a disposizione dalla *University Of California, Santa Cruz (UCSC)*. Questi dati sono stati utilizzati in combinazione con delle etichette in modo da permettere l'addestramento di modelli di *machine learning*. Le etichette sono state acquisite tramite *FANTOM5*<sup>9</sup>: un consorzio di ricerca internazionale focalizzato sull'annotazione dei genomi dei mammiferi.

Al fine di semplificare il processo di acquisizione dei dati sopra citati è stata utilizzata una pipeline messa a disposizione da AnacletoLAB<sup>10</sup>.

---

<sup>6</sup>[https://github.com/AnacletoLAB/epigenomic\\_dataset](https://github.com/AnacletoLAB/epigenomic_dataset)

<sup>7</sup><https://www.encodeproject.org/>

<sup>8</sup><https://genome.ucsc.edu/index.html>

<sup>9</sup><https://fantom.gsc.riken.jp/5/>

<sup>10</sup><https://github.com/AnacletoLAB/>

## 2.3 Data Pre-Processing

Durante la prima fase i dati acquisiti sono stati manipolati in modo da correggere e modificare eventuali mancanze o imperfezioni. Per migliorare e rendere più efficiente l'addestramento dei modelli di *machine learning* sono state adottate tecniche di riduzione della dimensionalità dei dati. Questo processo di manipolazione è stato applicato in maniera differente a seconda della tipologia di dati.

### 2.3.1 Dati Epigenomici

I dati epigenomici sono stati pre-processati attraverso:

- *NaN imputation*,
- *drop constant feature*,
- *data scaling*,
- *class balance*,
- *feature correlation visualization*,
- *feature selection*.

**NaN Imputation** I dati scaricati presentavano dei *record* con valori nulli. Attraverso l'utilizzo di una libreria che implementa il *K-nearest neighbors* [3] sono stati sostituiti i valori nulli con i valori generati dall'algoritmo; in particolare è stato utilizzato l'algoritmo con un valore  $k = 5$ . *KNN* è un algoritmo utilizzato nel riconoscimento di pattern per la classificazione di oggetti basato sulle caratteristiche degli elementi vicini a quello considerato.

**Drop Constant Feature** La presenza di *features* con valori uguali non porta a un incremento delle informazioni utili. Per risolvere questo problema sono stati messi a confronto i valori di ogni *feature*, in modo da evidenziare quelle uguali tra loro e di conseguenza eliminare quelle superflue.

**Data scaling** Il processo di scalatura consente di standardizzare i dati. Questa pratica è importante, non solo per riuscire a confrontare caratteristiche con unità di misura diverse. All'interno del progetto è stato usato *RobustScaler*<sup>11</sup> che scala le caratteristiche usando statistiche robuste rispetto agli outlier. Questo scaler rimuove la mediana e scala i dati in base all'intervallo inter-quantile; è utile perché i valori anomali possono spesso influenzare la media e la varianza del campione in modo negativo.

---

<sup>11</sup><https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html>

**Feature Selection con Pearson, Spearman e MIC** L’alta correlazione tra le *features* non porta a un incremento delle informazioni utili. L’utilizzo di tecniche basate su indici di correlazione lineari e non lineari ha permesso la rimozione di *features* altamente correlate tra loro. Gli indici utilizzati per la correlazione lineare sono stati:

- *indice di Pearson*<sup>12 13</sup>,
- *indice di Spearman*<sup>14</sup>.

Le *features* con  $p\_value > 0.01$  non presentano una correlazione statisticamente rilevante. A partire da questi valori è stata effettuata una valutazione attraverso l’analisi del *MIC (Maximal Information Coefficient)*<sup>15</sup>. Questo permette di evidenziare l’esistenza di una correlazione non lineare. Le *features* che hanno superato il valore soglia di 0.05 sono state eliminate in quanto non rilevanti.

**Features Selection** Applicando l’algoritmo *Recursive Feature Elimination* è stato ridotto ulteriormente il numero di feature delle varie linee cellulari. La cella *A549* è stata ridotta da 48 features a 25. La cella *MCF-7* da 117 a 80 e la cella *K562* da 429 a 200. Inizialmente si è provato ad applicare *boruta* ma è risultato troppo oneroso a livello computazionale per l’hardware in possesso. L’allenamento della rete e la conseguente valutazione delle prestazioni è stata fatta in primo luogo sui dati senza l’applicazione di RFE, successivamente applicando l’algoritmo per mostrare se ci fossero delle evidenti differenze nella valutazione delle prestazioni

**Class Balancing** I dati sono stati analizzati in modo da verificare che la proporzione tra etichette attive e inattive, sia dei *promoter* sia degli *enhancer*, fosse compresa 0.1 e 0.01. In tutte le linee cellulari analizzate non è stato necessario applicare un bilanciamento delle classi. In [1] viene riportata la proporzione.

**Feature Correlation Visualization** In [2], [3] e [4] viene mostrata una visualizzazione delle tre *features* maggiormente correlate ottenuta attraverso l’applicazione delle tecniche spiegate nei paragrafi precedenti.

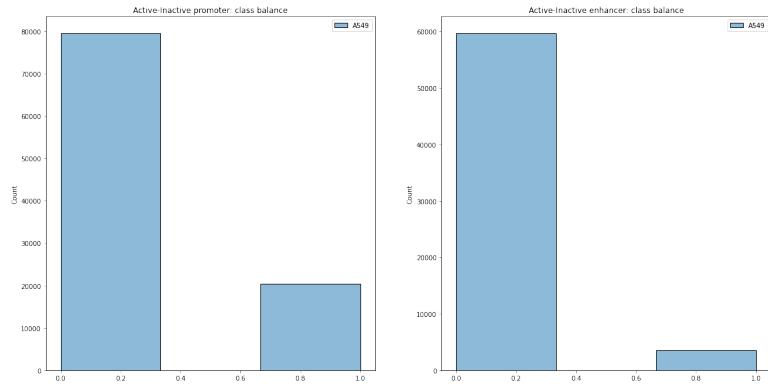
---

<sup>12</sup><https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html>

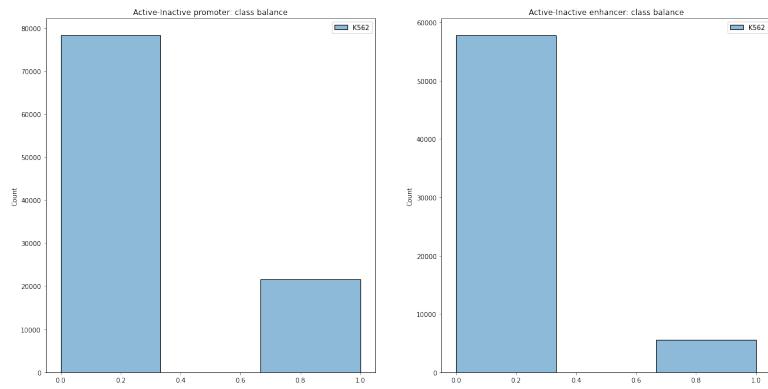
<sup>13</sup>[https://en.wikipedia.org/wiki/Pearson\\_correlation\\_coefficient](https://en.wikipedia.org/wiki/Pearson_correlation_coefficient)

<sup>14</sup>[https://en.wikipedia.org/wiki/Spearman%27s\\_rank\\_correlation\\_coefficient](https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient)

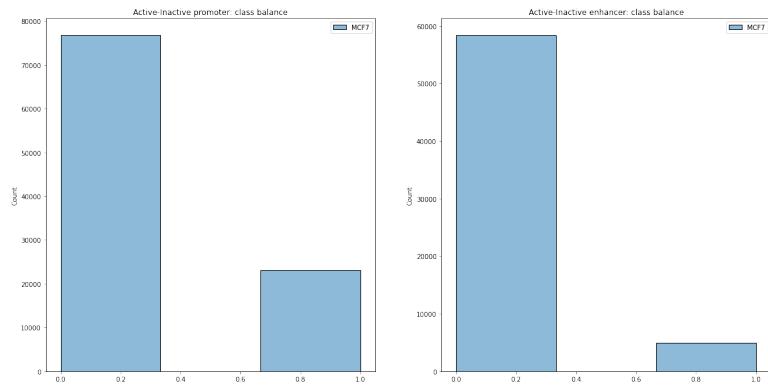
<sup>15</sup>[https://en.wikipedia.org/wiki/Maximal\\_information\\_coefficient](https://en.wikipedia.org/wiki/Maximal_information_coefficient)



(a) Class balance linea cellulare A549

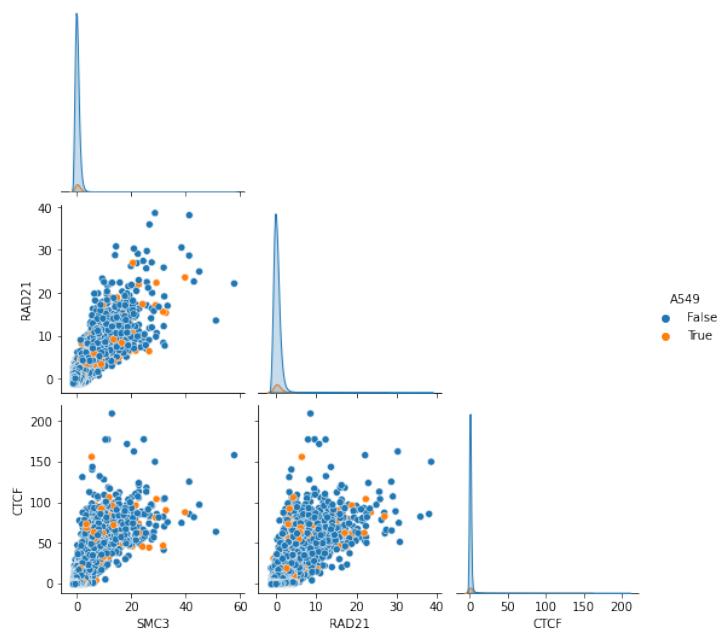


(b) Class balance linea cellulare K562

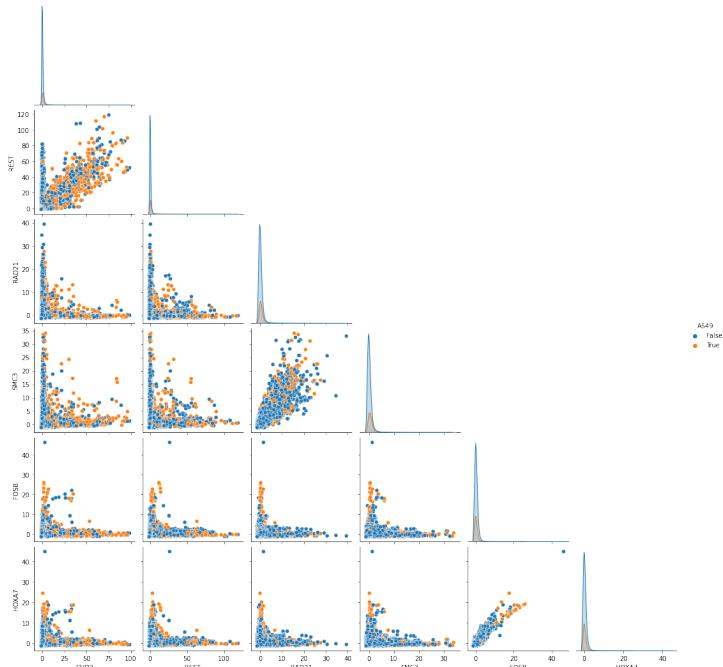


(c) Class balance linea cellulare MCF7

Figura 1: A sinistra le regioni inattive, a destra quelle attive

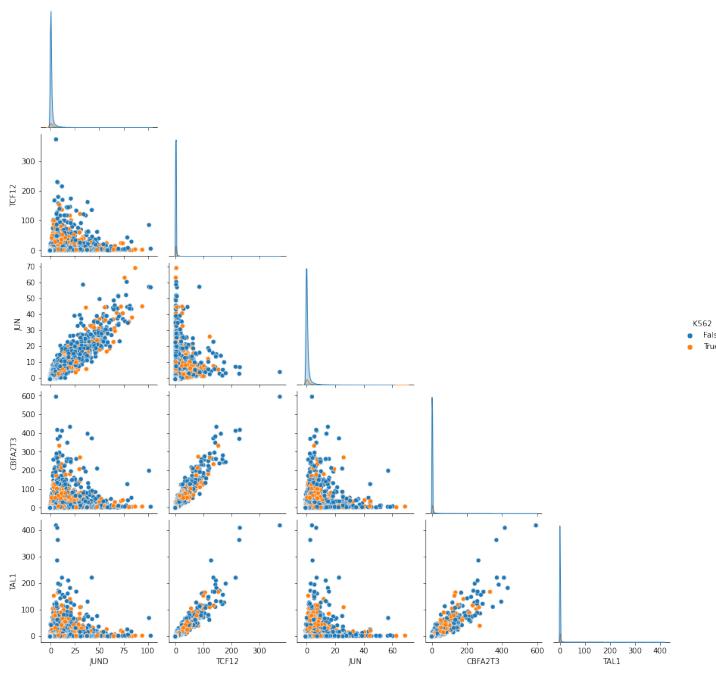


(a) enhancer

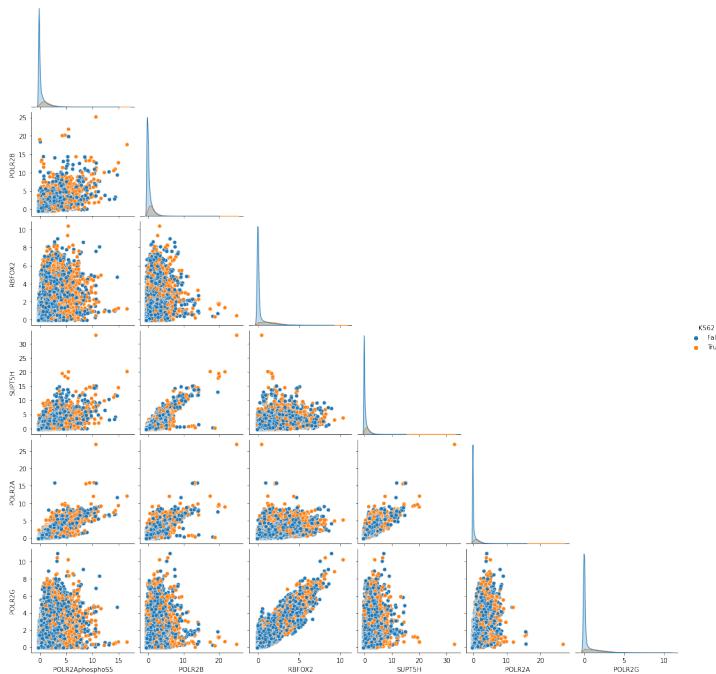


(b) promoter

Figura 2: Visualizzazione della correlazione delle features sulla linea cellulare A549

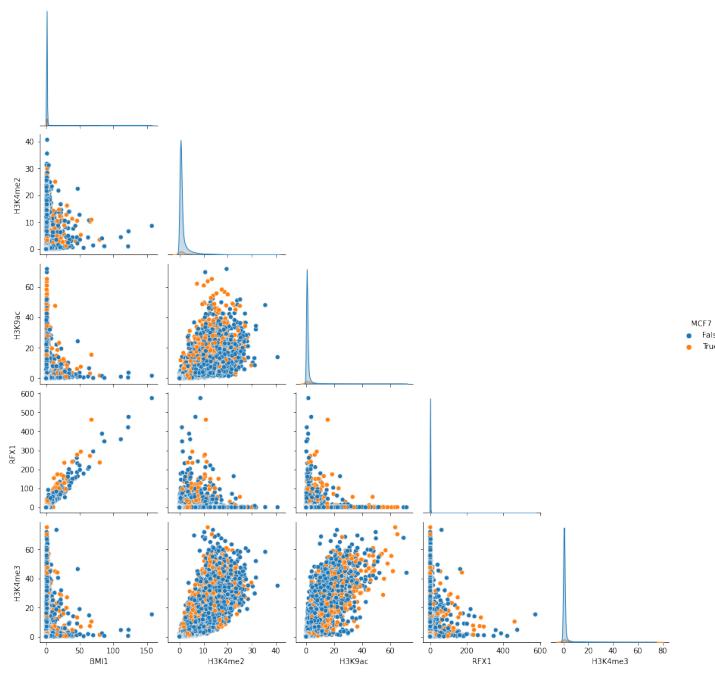


(a) enhancer

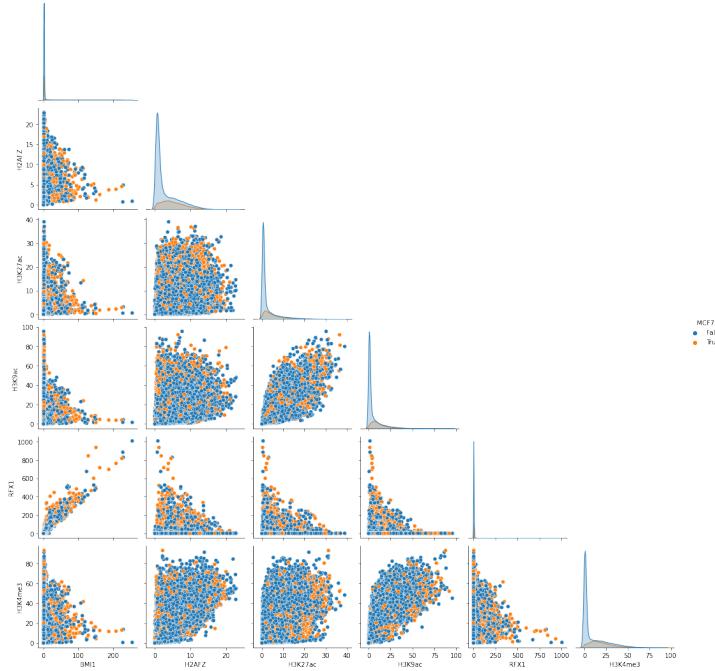


(b) promoter

Figura 3: Visualizzazione della correlazione delle features sulla linea cellulare K562



(a) enhancer



(b) promoter

Figura 4: Visualizzazione della correlazione delle features sulla linea cellulare MCF7

### 2.3.2 Dati Sequenziali

I dati sequenziali sono stati manipolati tramite la *One-Hot Encoding*<sup>16</sup>.

**One-Hot Encoding** dal momento che i dati sequenziali sono dati categ-  
rifici si è resa necessaria l'applicazione di una metodologia di conversione in  
dati numerici. La motivazione è data dal fatto che molti algoritmi di ap-  
prendimento automatico non possono operare direttamente su etichette ma  
richiedono che tutte le variabili di input e di output siano numeriche.

### 2.3.3 Principal Component Analysis (PCA) e t-distributed Stochastic Neighbor Embedding (t-SNE)

Con l'intento di capire la distribuzione dei dati sotto esame sono state utili-  
zzate due diverse tipologie di visualizzazione attraverso l'applicazione rispet-  
tivamente di PCA e t-SNE [7]. Come mostrato in [5] e [6] i dati epigenomici  
presentano una maggiore separazione tra dati attivi e inattivi, mentre i dati  
sequenziali non mostrano nessun tipo di *pattern* grafico.

## 2.4 Holdouts

L'operazione di holdout divide l'intero dataset in *train* e *validation sets*. In  
questo lavoro, a causa delle limitate risorse computazionali in nostro possesso,  
abbiamo usato solamente 2 holdouts per ogni *window size*. Ovviamente l'uso  
di più holdouts potrebbe portare a migliori risultati. In ogni holdout il  
dataset è stato diviso in *train data* ( 80% dell'intero dataset) e *test data*  
(20% dell'intero dataset) usando una *StratifiedRandomSplit*<sup>17</sup>.

## 2.5 Analisi dei risultati

I risultati sono stati elaborati valutando ogni modello su *train* e *test sets*  
usando tre metriche:

- *accuracy*: numero di osservazioni correttamente predette diviso il nu-  
mero totale di campioni,
- *AUPRC*: l'area sotto la curva *Precision Recall*. Metrica adatta nel caso  
in cui si tratta con dati sbilanciati.
- *AUROC*: misura la capacità di discriminazione del modello ed è calco-  
lata come l'area sotto la curva ROC. Più la metrica si avvicina a 1 più  
è alta la capacità di discriminare del modello.

All'interno del progetto è stato anche utilizzato il test di *Wilcoxon* in modo da  
verificare l'esistenza di differenze statistiche sulle performance tra i modelli  
utilizzati.

---

<sup>16</sup><https://en.wikipedia.org/wiki/One-hot>

<sup>17</sup>[https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.StratifiedShuffleSplit.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedShuffleSplit.html)

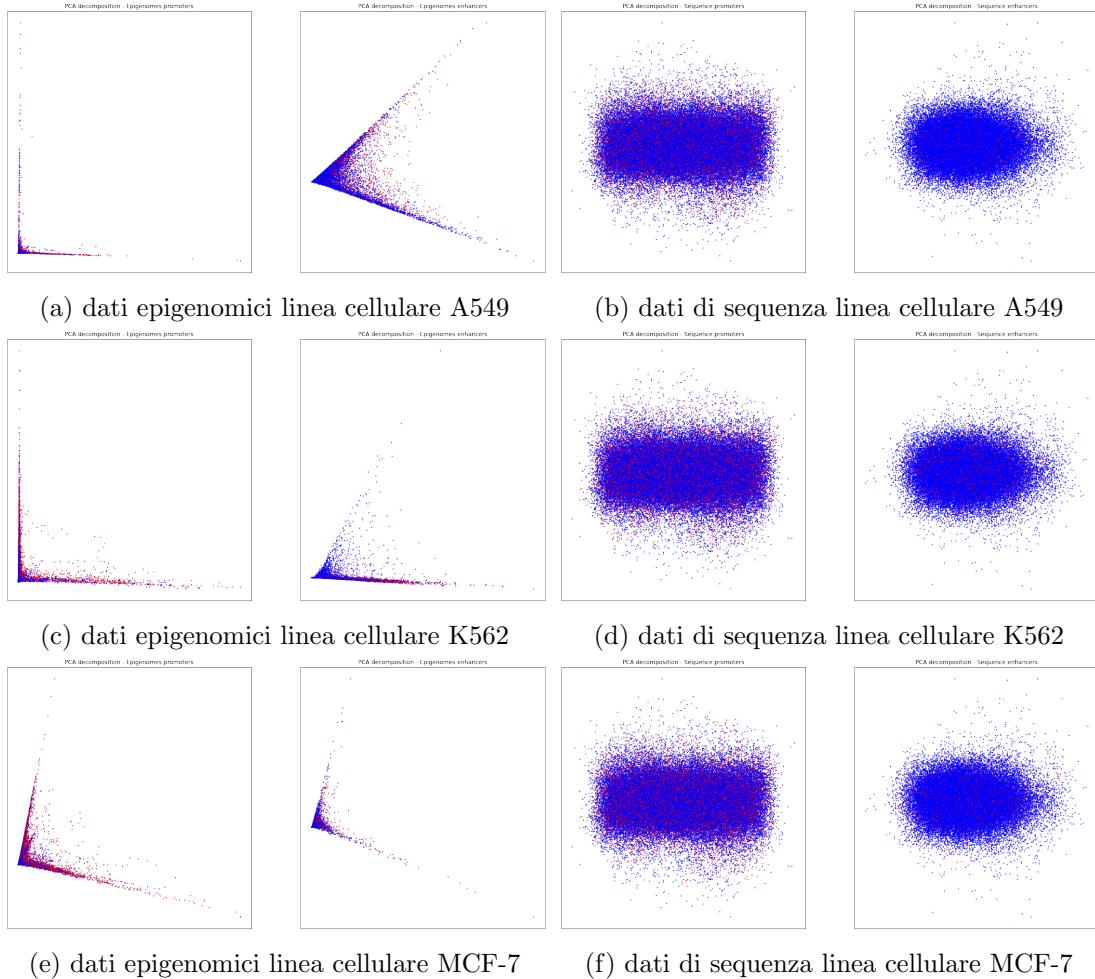


Figura 5: Visualizzazione dei dati ridotti tramite l'applicazione di PCA. In blu sono rappresentati le regioni attive, in rosso quelle inattive. A sinistra promoters, a destra enhancer.

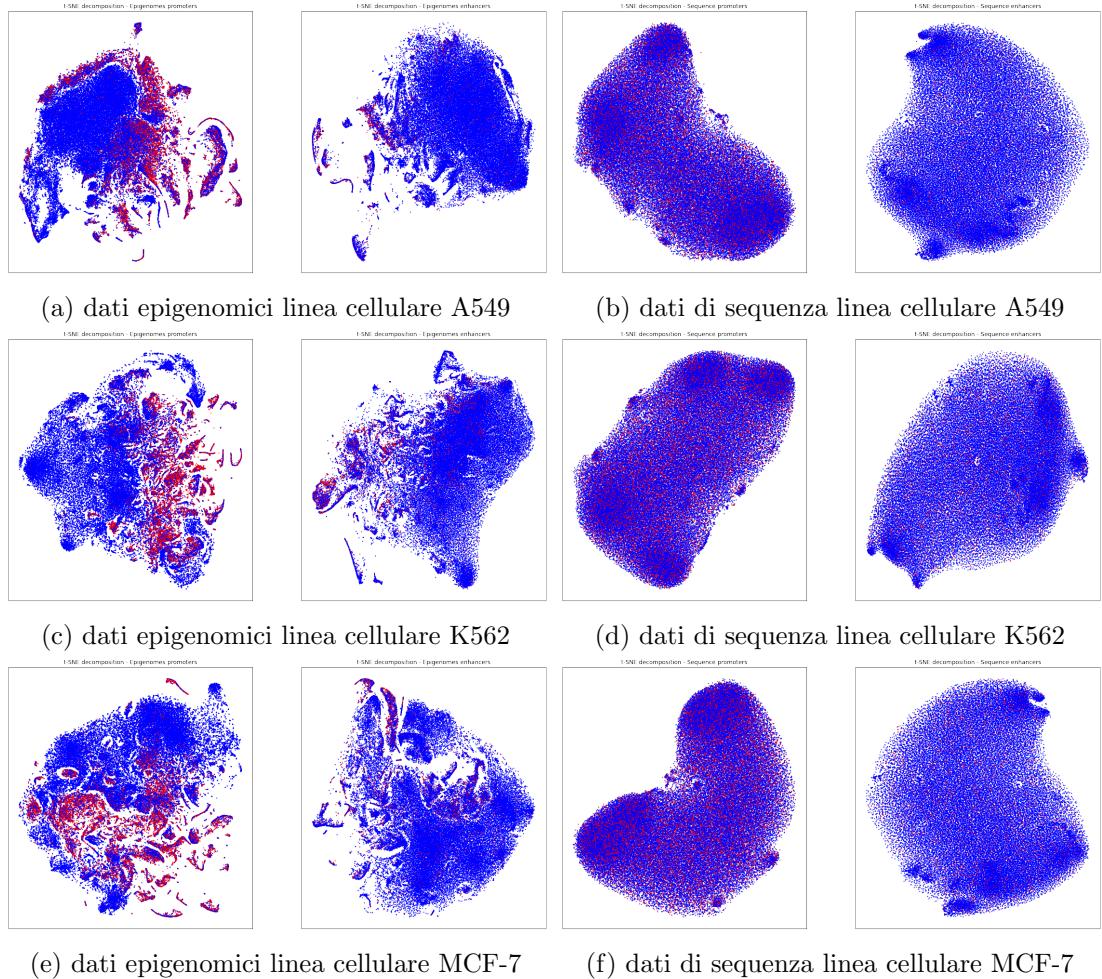


Figura 6: Visualizzazione dei dati ridotti tramite l'applicazione di t-SNE. In blu sono rappresentati le regioni attive, in rosso quelle inattive. La prima e terza colonna sono *promoters*, la seconda e l'ultima sono *enhancer*.

## 3 Modelli

In questo lavoro sono stati usati diversi classificatori per predirre l'attività delle regioni regolatorie nelle tre diverse linee cellulari. In particolare: *Feed Forward Neural Network*, *Convolutional Neural Network* e *Multimodal Neural Network*.

### 3.1 Feed-Forward Neural Network

Una *Feed Forward Neural Network* è una rete neurale nella quale possono essere presenti molti *layer* di diverse dimensioni. Ogni livello è collegato al successivo, ma diversamente dalle reti *Multi Layer Perceptron* (MLP), ci possono essere anche layer non densi. Ciò comporta che un nodo non è necessariamente collegato a ogni nodo del *layer* successivo. Questo tipo di reti sono state utilizzate per l'analisi dei dati epigenomici, dopo diverse prove siamo giunti alla struttura presentata nella Tabella 1. Si è voluto provare ulteriormente a migliorare i risultati della classificazione utilizzando la *Bayesian Optimization*. Quest'ultima è una tecnica di ottimizzazione che è stata utilizzata per fare *model selection* ovvero trovare la migliore combinazione di iperparametri per il modello in questione dato un input fissato. La struttura è visibile in Tabella 2 [5].

Layers	Units	Activation	Parameters
Dense	256	relu	
Dense	128	relu	
PRelu			
BatchNormalization			
Dropout			rate = 0.5
Dense	64	relu	
Dropout			rate = 0.5
Dense	32	relu	
Dense	16	relu	
PRelu			
BatchNormalization			
Dropout			rate = 0.5
Dense	1	sigmoid	

Tabella 1: Struttura per la Fixed Feed Forward Neural Netowrk

### 3.2 Convolutional Neural Network

Le *Convolutional Neural Network* (CNN) sono una famiglia di reti neurali utilizzate comunemente nell'analisi di immagini. Sono ispirate dall'organizzazione della corteccia visiva animale. In questo progetto il modello CNN

Layers	Units	Activation	Parameters
Dense	256, 128, 64	relu	
PRelu			
BatchNormalization			
Dropout			rate = 0.3
Dense	128, 64, 32	relu	
PRelu			
BatchNormalization			
Dropout			rate = 0.3
Dense	32, 16, 8	relu	
PRelu			
BatchNormalization			
Dropout			rate = 0.3
Dense	1	sigmoid	

Tabella 2: Struttura per la Feed Forward Neural Network usando la Bayesian Optimization.

è stato utilizzato usando come input dati di sequenza. Questi tipi di dati esprimono la posizione di ciascuna coppia di base azotata all'interno del DNA, quello che si vuole fare è cercare particolari pattern che possano far pensare che quella specifica area sia attiva o non attiva. Anche in questo caso l'approccio è stato quello descritto in [3.1]; in primo luogo è stato realizzato un modello *fixed* della rete, la sua struttura si può visualizzare in Tabella 3. Successivamente è stato realizzato un modello CNN utilizzando la *bayesian optimization* per fare *model selection* come detto precedentemente in [3.1]. La struttura è visibile in Tabella 4.

Layers	Units	Activation	Parameters
Convolutional	16	relu	kernel size = 3
MaxPooling			
Convolutional	32	relu	kernel size = 3
MaxPooling			
Convolutional	64	relu	kernel size = 3
MaxPooling			
Dropout			rate = 0.5
GlobalMaxPooling			
Dense	128	relu	
Dropout			rate = 0.5
Dense	1	sigmoid	

Tabella 3: Struttura per la Fixed Convolutional Neural Network

Layers	Units	Activation	Parameters
Convolutaional	64	relu	kernel size = 5
BatchNormalization			
Convolutional	64	relu	kernel size = 5
BatchNormalization			
Convolutional	64	relu	kernel size = 5
BatchNormalization			
MaxPooling			
Convolutional	32, 64, 128	relu	kernel size = 5,10
BatchNormalization			
Flatten			
Dense	10, 32, 64	relu	
Dropout			rate = 0.3
Dense	10, 32, 64	relu	
Dropout			rate = 0.3
Dense	1	sigmoid	

Tabella 4: Struttura per la Convolutional Neural Netowrk usando la Bayesian Optimization

### 3.3 MultiModal Neural Network

La *MultiModal Neural Network* è un modello che si ispira al modo che hanno comunemente le persone di imparare, cioè sfruttare più informazioni che provengono da varie fonti mettendole insieme per trarne una conclusione. Le *MMNN* ci consentono di combinare due modelli differenti in un singolo modello. Come in precedenza si è realizzato prima un modello *fixed*, concatenando *fixed* FFNN e *fixed* CNN. Successivamente è stata implementata una MMNN concatenando il modello Bayesian FFNN e il modello Bayesian CNN. La struttura viene illustrata in Tabella 5 [6, 2].

Layers	Units	Activation	Parameters
Concatenate(FFNN e CNN)			
Dense	64	relu	
Dense	64		
PRelu			
BatchNormalization			
Dropout			rate = 0.2
Dense	32		
PRelu			
BatchNormalization			
Dropout			rate = 0.2
Dense	1	sigmoid	

Tabella 5: Struttura per le Multimodal Neural Network.

## 4 Risultati

Inizialmente sono state usate un numero di epoche di un'unità di misura più grande rispetto ai risultati mostrati nelle Figure sottostanti. Siccome i tempi di computazione risultavano troppo elevati sulla linea cellulare con meno *features* (A549) è stato deciso di abbassare notevolmente il numero di epoche. Allo stesso modo è stato inizialmente utilizzato *Boruta* per fare *features selection*, ma i tempi di computazione risultavano troppo elevati. Si è allora deciso di utilizzare la *Recursive Feature Elimination* per abbassare il numero di *feature* a un valore fissato in modo da ridurre i tempi di calcolo.

Analizzando i risultati ottenuti delle tre linee cellulari non si notano particolari differenze di performance nei modelli. Di seguito analizziamo i *barplots* sottostanti, essi mostrano le performance dei modelli con e senza *feature selection* sia per gli *enhancer* sia per i *promoter*. Osservando i barplots possiamo trarne le seguenti conclusioni:

- le performance dei modelli con o senza *feature selection* sono praticamente le stesse,
- nelle *Multimodal Neural Network* è evidente come le *fixed* vadano in *overfitting* mentre si comporta meglio la versione che utilizza la *bayesian optimization* senza però restituire risultati soddisfacenti. Osservando la metrica *AUPRC* è evidente come utilizzando la *bayesian* il modello vada in *underfitting* nel caso degli *enhancer*.
- le *Feed Forward Neural Network* sono quelle che generalmente si comportano meglio rispetto agli altri modelli, non c'è un netto miglioramento tra la versione *fixed* e quella che utilizza la *bayesian optimization*,
- le *Convolutional Neural Network* sono il modello che si comporta peggio di tutti. La versione *fixed* va in *underfitting*, questo significa che il modello non è stato in grado di estrarre abbastanza informazioni rilevanti dal *dataset*, mentre la versione con la *bayesian optimization* va in *overfitting*,
- in generale tutti i modelli hanno risultati migliori nel caso dei *promoter*. In conclusione possiamo dire che predire lo stato di attività degli *enhancer* è più difficile che predire lo stato dei *promoter*.
- il test di *Wilcoxon* non evidenzia nessuna differenza statistica di performance tra i vari modelli.

## 5 Conclusioni

Lo scopo di questo studio è stato quello di cercare di predire quali regioni riguardanti *promoter* ed *enhancer* fossero attive o inattive. In conclusione, analizzando i risultati ottenuti, è possibile affermare che i modelli presentano risultati migliori utilizzando dati epigenomici rispetto a dati di sequenza. Inoltre è risultato più facile effettuare la classificazione tra i *promoter* rispetto agli *enhancer*.

## 6 Sviluppi futuri

Le prestazioni ottenute non sono del tutto soddisfacenti. Ci sono diversi miglioramenti che possono essere attuati per aumentare le prestazioni:

- utilizzare un altro tipo di *feature selection* come per esempio boruta,
- cambiare il numero di *feature* selezionate durante la *Recursive Feature Elimination*,
- aumentare il numero di epoch e il numero di holdout usati,
- provare a utilizzare la regressione anzichè la classificazione,
- estendere il numero di iperparametri utilizzati durante la *bayesian optimization*.

Tutti questi miglioramenti richiederebbero un aumento notevole del costo computazionale, che però potrebbe portare a un aumento di prestazioni nella previsione dell'attività delle regioni.

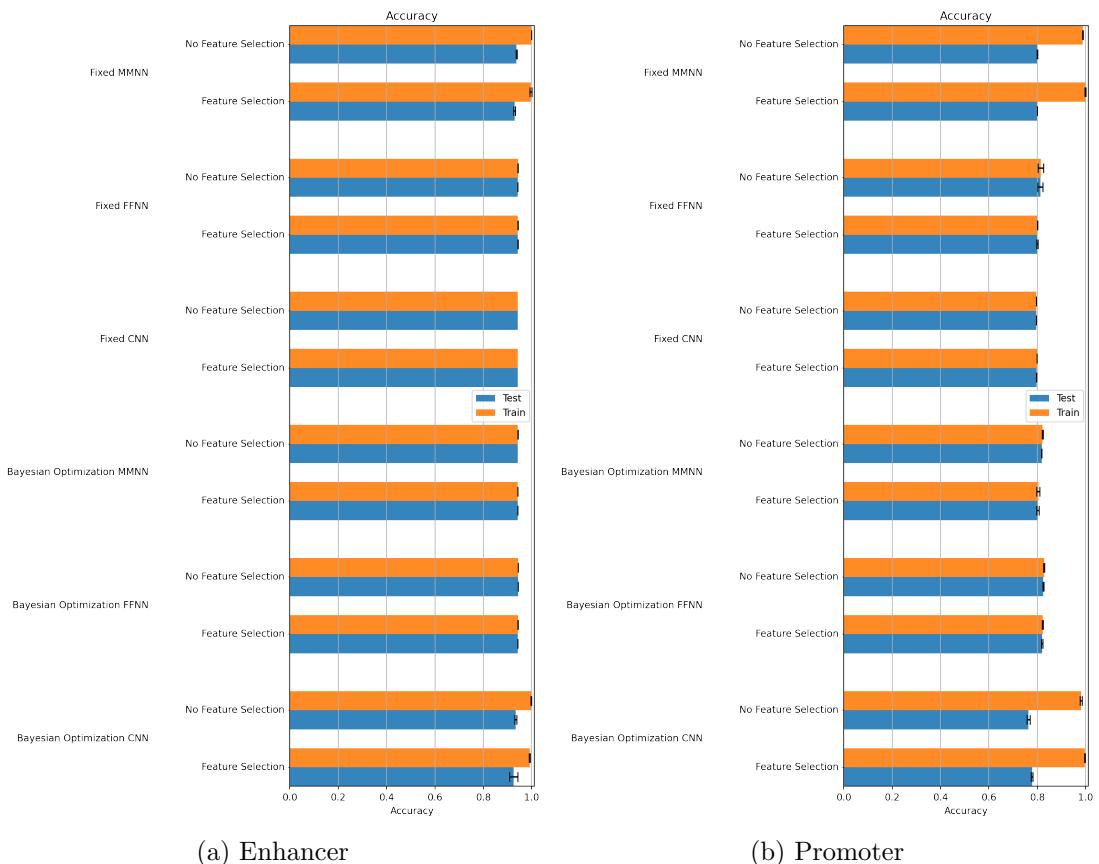


Figura 7: Linea cellulare A549

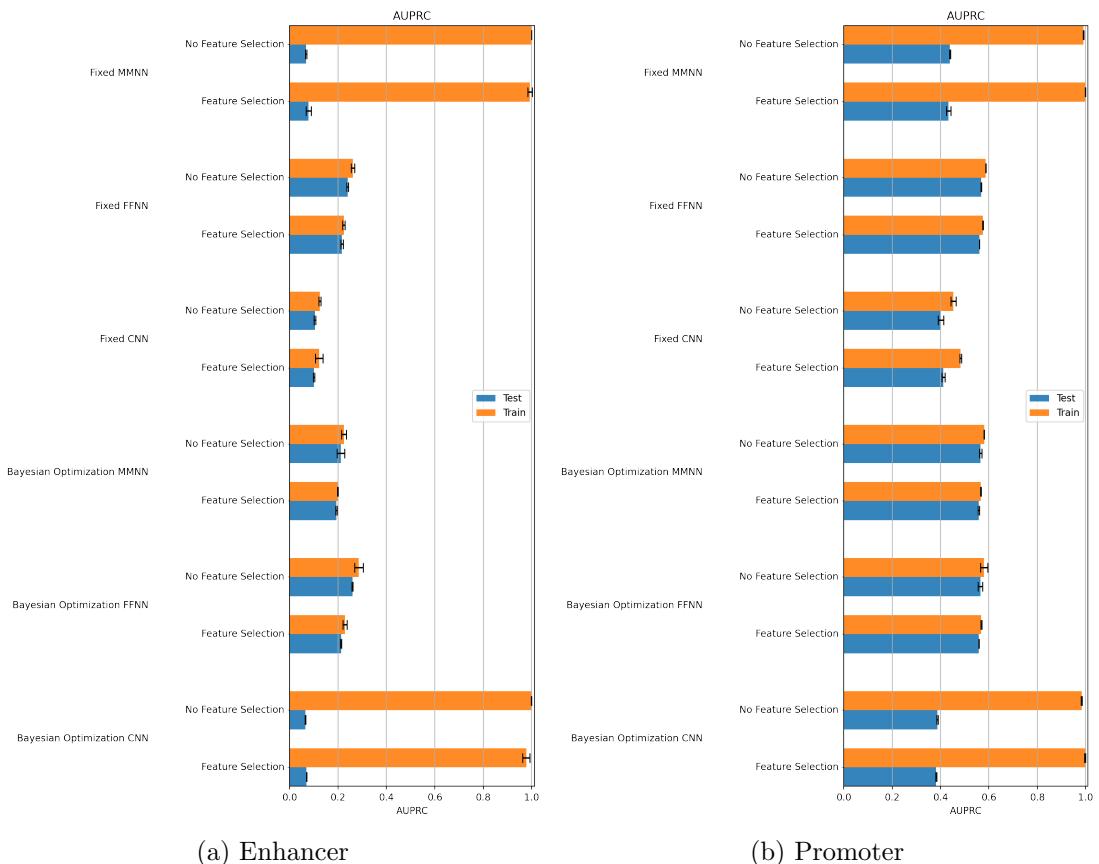


Figura 8: Linea cellulare A549

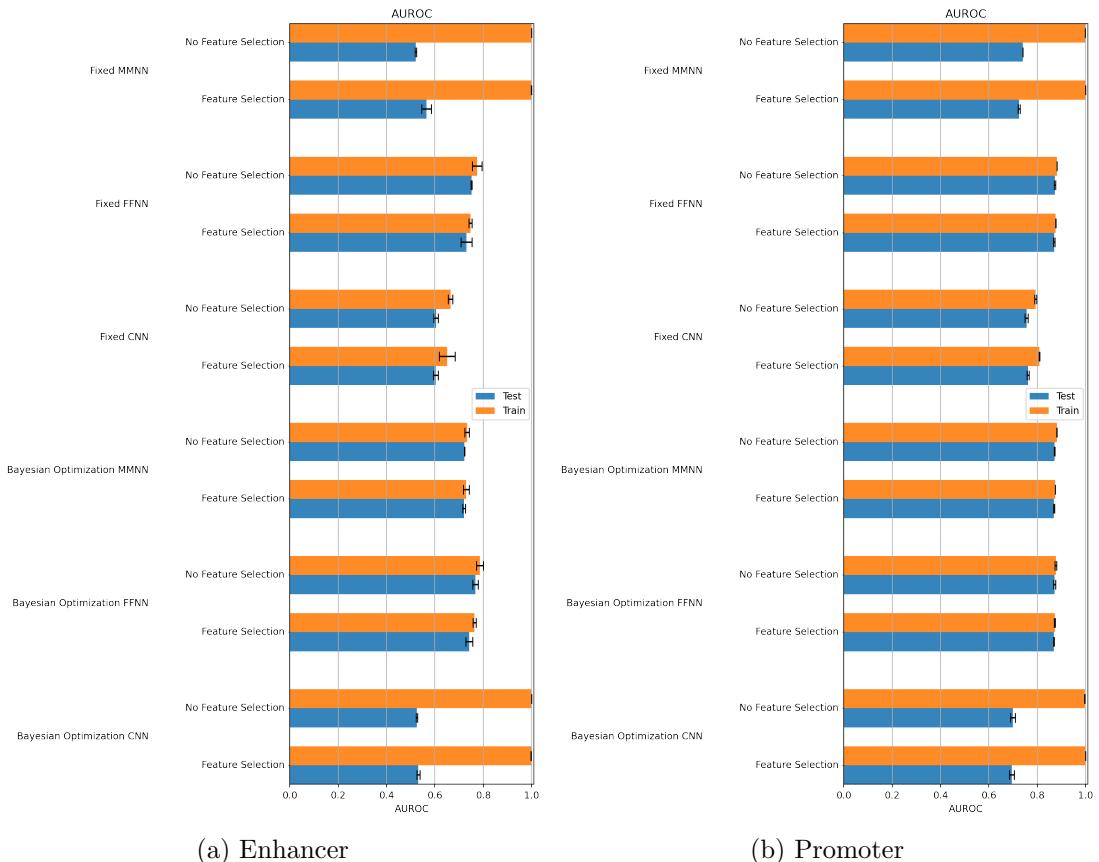


Figura 9: Linea cellulare A549

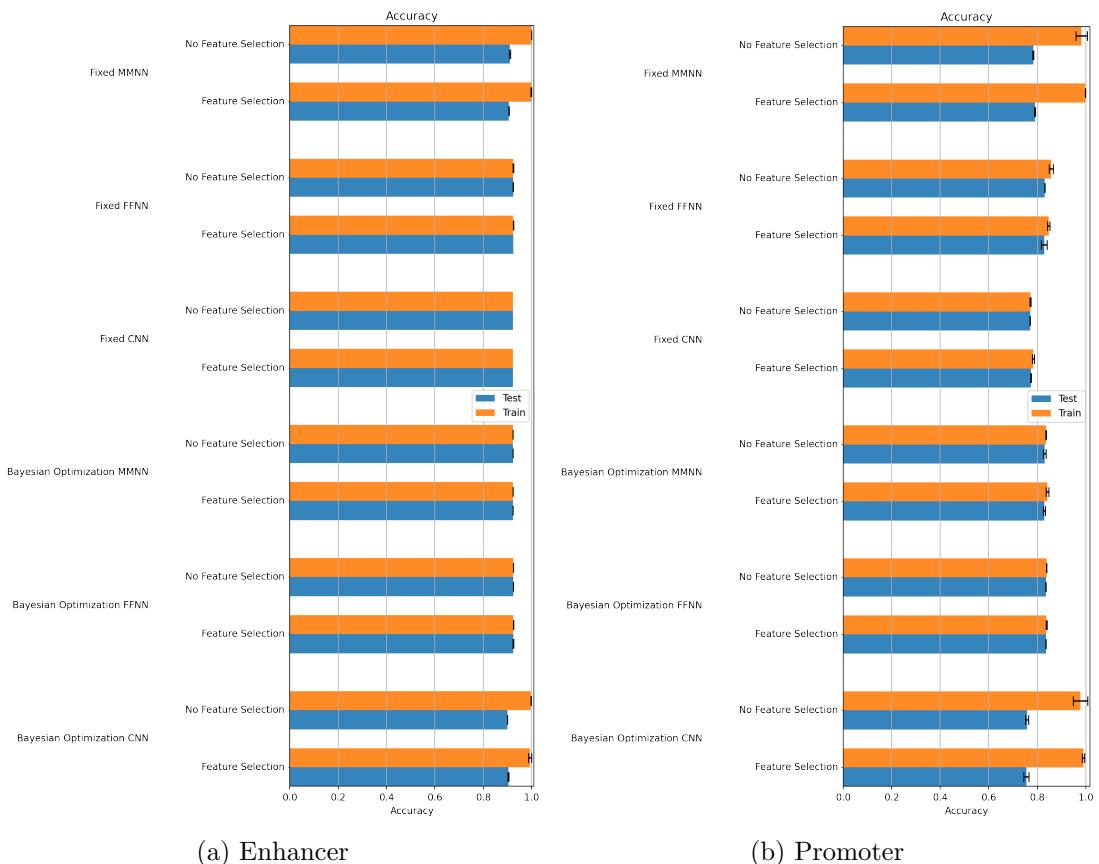


Figura 10: Linea cellulare MCF7

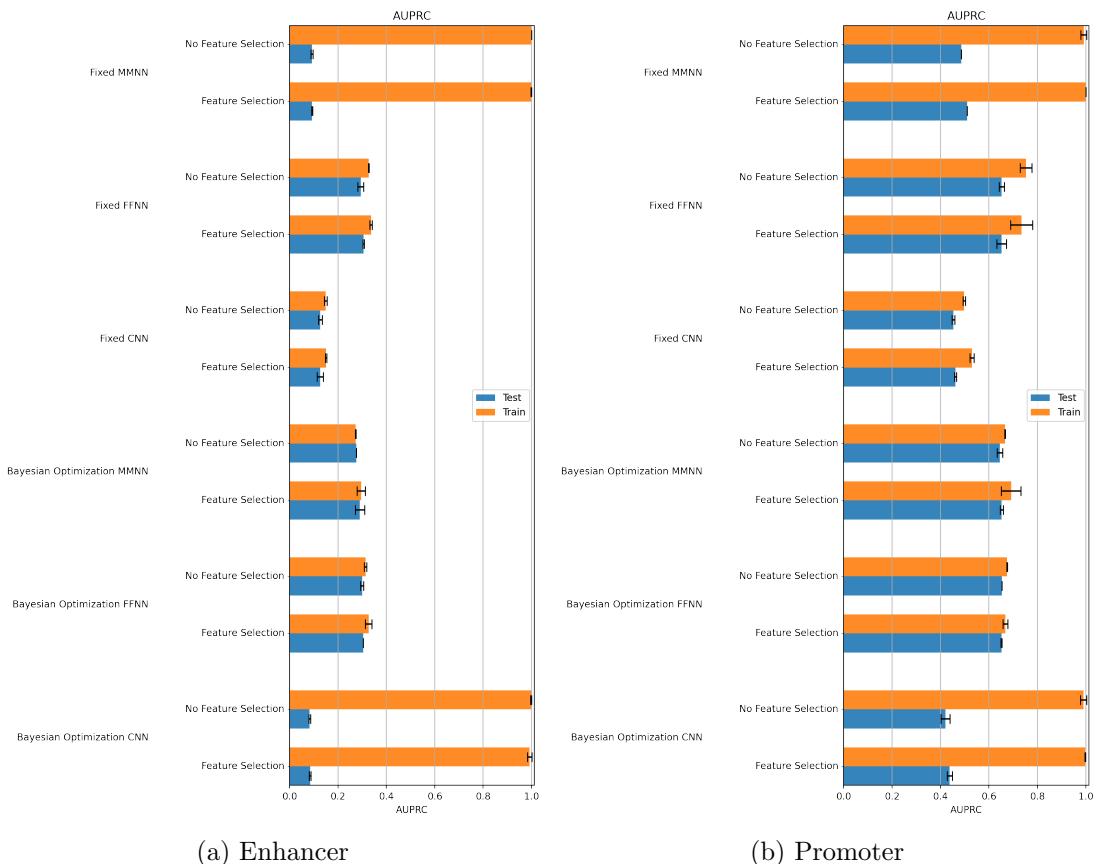


Figura 11: Linea cellulare MCF7

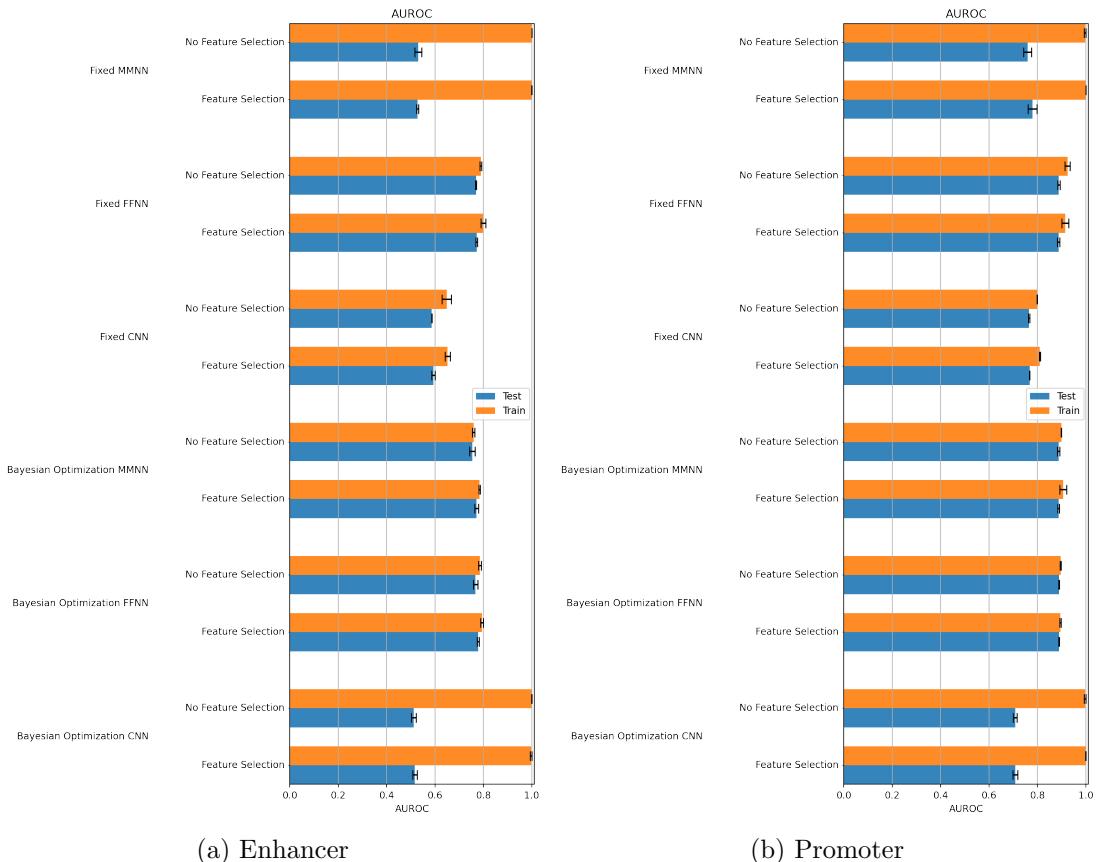


Figura 12: Linea cellulare MCF7

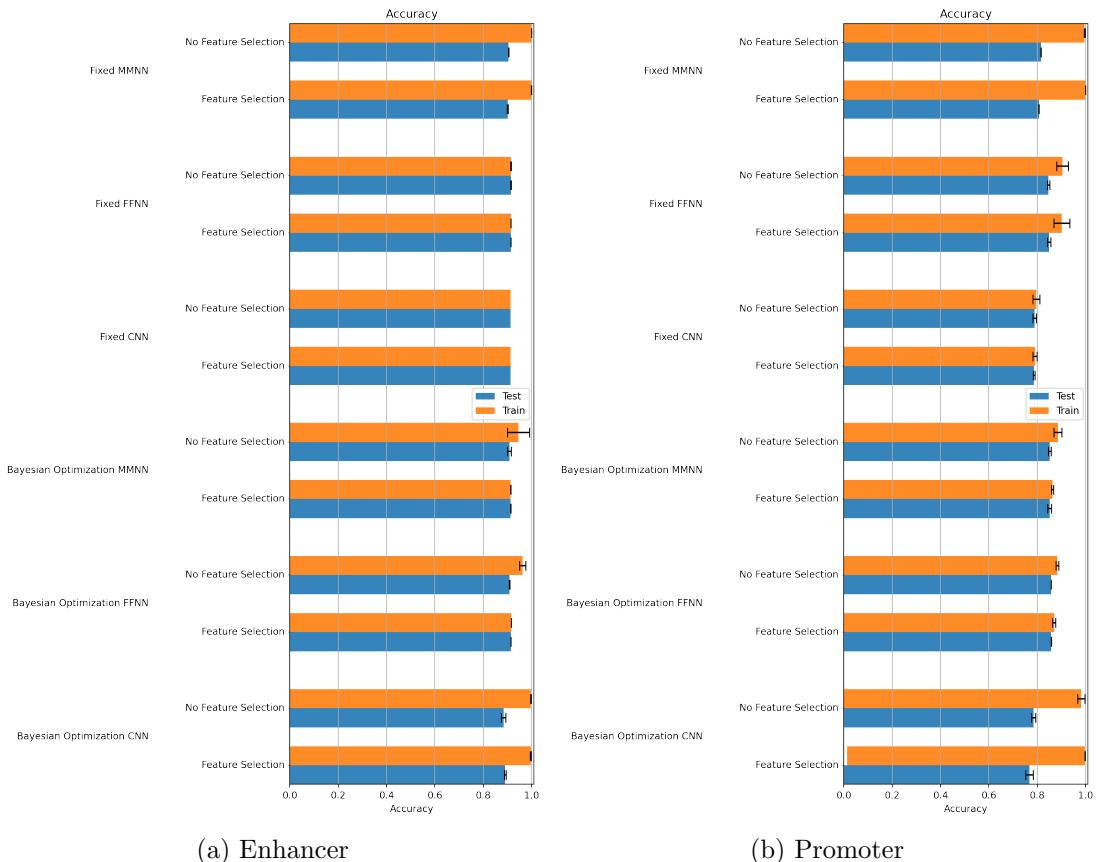


Figura 13: Linea cellulare K562

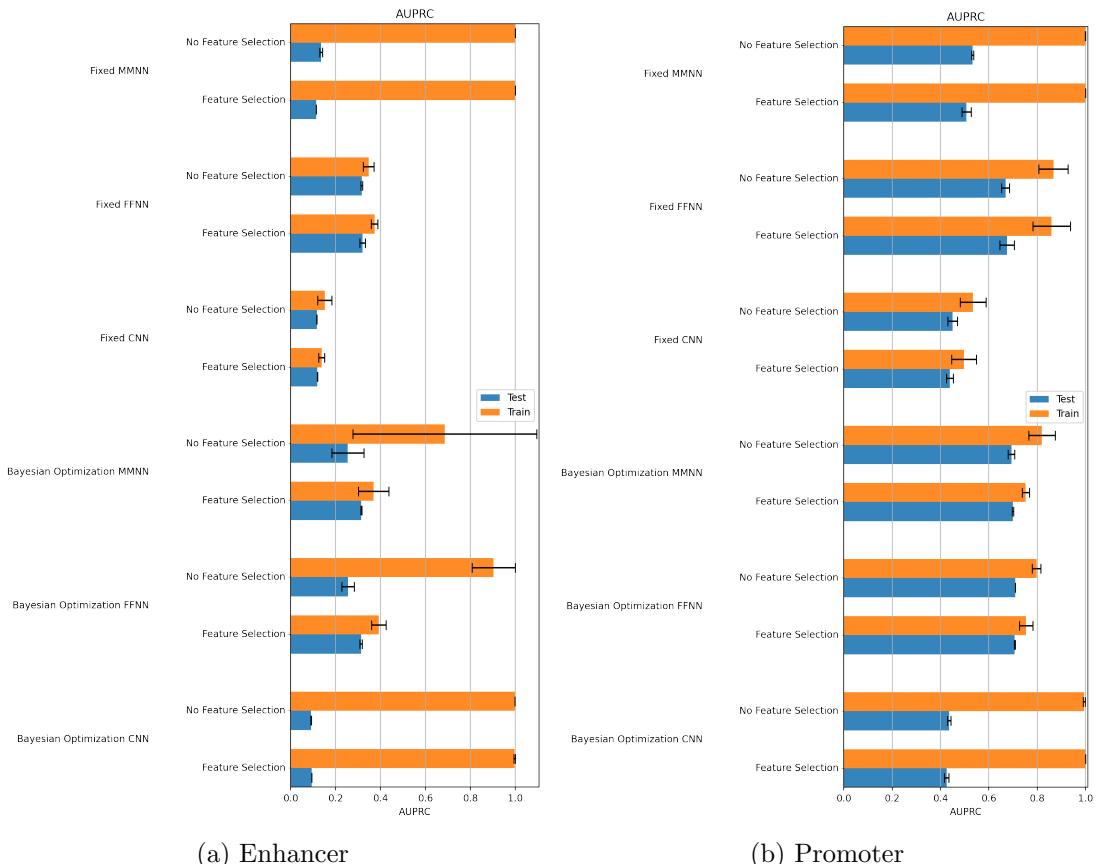


Figura 14: Linea cellulare K562

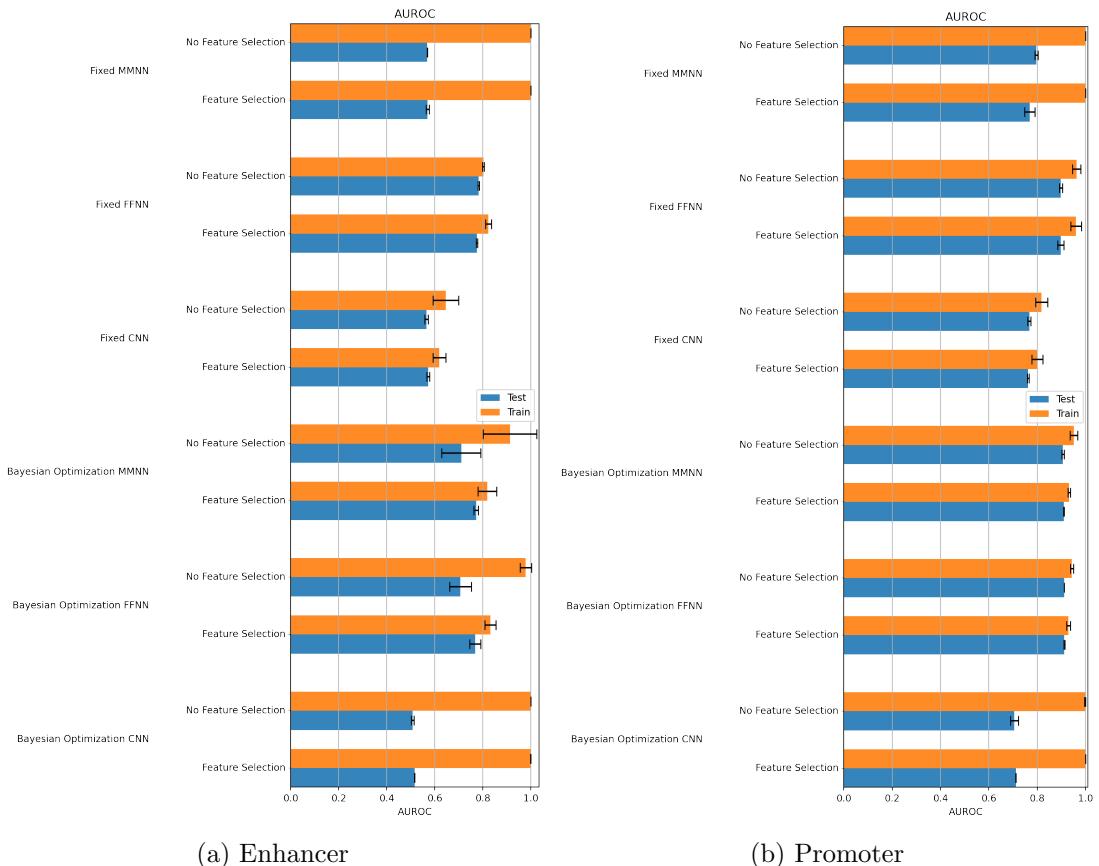


Figura 15: Linea cellulare K562

## Bibliografia

- [1] Elizabeth Blackwood e James Kadonaga. «Going the Distance: A Current View of Enhancer Action». In: *Science (New York, N.Y.)* 281 (ago. 1998), pp. 60–3. DOI: 10.1126/science.281.5373.60.
- [2] Luca Cappelletti et al. «Bayesian Optimization Improves Tissue-Specific Prediction of Active Regulatory Regions with Deep Neural Networks». English. In: *Bioinformatics and Biomedical Engineering - 8th International Work-Conference, IWBBIO 2020, Proceedings*. A cura di Ignacio Rojas et al. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 8th International Work-Conference on Bioinformatics and Biomedical Engineering, IWBBIO 2020 ; Conference date: 06-05-2020 Through 08-05-2020. Springer, gen. 2020, pp. 600–612. ISBN: 9783030453848. DOI: 10.1007/978-3-030-45385-5\_54.
- [3] Nicolo' Cesa-Bianchi. *The Nearest Neighbour algorithm*. URL: <https://cesa-bianchi.di.unimi.it/MSA/Notes/knn.pdf>.
- [4] Shaul Karni. «Analysis of Biological Networks : Transcriptional Networks-Promoter Sequence Analysis». In: 2007.
- [5] Rudolf Kruse et al. *Computational Intelligence*. 2016. DOI: 10.1007/978-1-4471-7296-3.
- [6] Yifeng Li, Wenqiang Shi e Wyeth Wasserman. «Genome-Wide Prediction of cis-Regulatory Regions Using Supervised Deep Learning Methods». In: *BMC Bioinformatics* accepted (mag. 2018). DOI: 10.1186/s12859-018-2187-1.
- [7] Laurens van der Maaten e Geoffrey Hinton. «Visualizing Data using t-SNE». In: *Journal of Machine Learning Research* 9.86 (2008), pp. 2579–2605. URL: <http://jmlr.org/papers/v9/vandermaaten08a.html>.