Progetto d'esame per il corso di Bioinformatica

Predizione di varianti genetiche patogeniche di malattie Mendeliane in regioni non codificanti del genoma umano con metodi di deep learning.

Le malattie genetiche Mendeliane sono causate da mutazioni a livello di un singolo gene: una variante a livello di singolo nucleotide (SNV Single-Nucleotide Variants) è cioè sufficiente a causare una patologia genetica di tipo Mendeliano. Delle 8000 circa malattie genetiche riconosciute come Mendeliane, conosciamo solo per la metà di esse il gene e la mutazione che ha causato la patologia.

Recentemente sono stati proposti metodi di Machine Learning per predire e/o effettuare il ranking di varianti genetiche potenzialmente causative di malattie genetiche Mendeliane, basati su metodi di Ensemble o Hyper-ensemble (Smedley et al., 2016, Schubach et al 2017).

L'obiettivo del progetto consiste nel valutare se metodi di deep learning possono essere efficaci per predire SNV associate a patologie Mendeliane, utilizzando un subset dei dati costruiti ed analizzati nei due paper citati.

In particolare ogni gruppo potrà scegliere liberamente un metodo di deep learning per applicarlo ai dati forniti, valutandone le capacita' predittive secondo le metodologie spiegate di seguito.

Dati.
I dati relativi alle varianti genetiche da analizzare/predire sono scaricabili dalla directory:
http://homes.di.unimi.it/valentini/ProgettoBioinformatica1718/data/

Tutti i dati si riferiscono esclusivamente alle regioni non coding: tutte le varianti nelle regioni codificanti sono state rimosse. I dati sono stati ottenuti da (Smedley et al. 2016), utilizzando tutte le SNV “positive”, cioè associate a patologie Mendeliane (406) ed estraendo casualmente un milione di esempi negativi (SNV “neutrali”) dagli oltre 14 milioni di esempi negativi disponibili.

Ad ogni SNV sono associate 26 feature numeriche descritte nei paper precedentemente citati che includono score di conservazione, annotazioni di trascrizione e regolazione, feature di sequenza e popolazione ed altre (per maggiori dettagli si vedano i Supplementary Material in Schubach et al. 2017). Tali feature costituiscono il vettore di input per la rete neurale. La variabile di output da predire è la patogenicità della SNV,

che può essere o di tipo dicreto (ad es: 0 e 1) oppure, meglio, può essere espressa come score a valori reali (ad es: valori reali nell' intervallo [0,1]).

I dati sono stati suddivisi in test set, costituito dalle SNV che giacciono sul cromosoma umano 19, e da un training set che include tutti le rimanenti SNV (cioe' le SNV che non sono sul cromosoma 19). Si tratta quindi di una versione di hold-out “chromosome-aware”.
I dati di training sono costituiti da 981388 SNV (esempi). Gli esempi “positivi”, cioè associati a malattie genetiche Mendeliane sono i primi 356, tutti gli altri esempi sono negativi.
I dati di test sono costituiti da 19018 esempi, di cui i primi 40 sono costituiti da SNV patogeniche.
I dati disponibili sono quindi molto sbilanciati.

Formato dei dati: sia il train (Mendelian.train.tsv.gz) che il test set (Mendelian.test.tsv.gz) sono file in testo piano gzipped con entry separate da carattere di tabulazione (\tab)
In entrambi i file la prima riga riporta in nomi delle 26 feature. Dalla seconda riga all'ultima sono riportate le 26 feature delle SNV precedute dal codice identificativo della SNV (che riporta il numero del cromosoma e la posizione genomica della variante). Il file Mendelian.train.tsv.gz ha quindi 981389 righe ed il file Mendelian.test.tsv.gz ha 190189 righe. Le primi 356 SNV del file di training riporta le SNV “positive” (cioè patogeniche), il resto sono tutte negative (cioè neutrali o benigne). Le primi 40 SNV del file di testing riporta le SNV “positive” (cioè patogeniche), il resto sono tutte negative (cioè neutrali o benigne).

Set-up sperimentale.
Il problema della detection di varianti genetiche associate a patologie genetiche Mendeliane può essere modellato come un problema di classificazione supervisionato. La rete neurale deve essere addestrata sul training set e le sue capacità di generalizzazione devono essere valutate sul test set.
Le metriche da utilizzare sono la AUROC (area sotto la curva ROC) e soprattutto l'AUPRC (l'area sotto la curva precision-recall). Valutate i risultati sia sul training set sia sul test set.

Modelli neurali.
Ogni gruppo può scegliere liberamente uno o più modelli di reti neurali per questo problema di predizione (Goodfellow et al. 2016). Potete utilizzare sia MLP deep (cioè reti neurali profonde completamente connesse), utilizzando opportuni algoritmi di learning, come ad es. Backpropagation con dropout o con discesa a gradiente stocastica, oppure architetture basate su deep convolutional networks. E' possibile esplorare diversi modelli (ad es: variando il numero di layer ed il numero di hidden neurons per ciascun

layer) per valutarne comparativamente le prestazioni.
Non è importante ottenere “ottimi risultati”, ma è importante invece descrivere con precisione l'architettura neurale utilizzata, possibilmente inserendo schemi/figure che ne chiariscano la struttura complessiva.

Implementazione.
In generale potete scegliere i linguaggi e le librerie che ritenete opportune per implementare i modelli.
Alcune possibilità potrebbero essere:
1) Keras: https://keras.io/
E' una libreria molto ad alto livello in Phyton (esiste anche un interfaccia in R) che richiede però anche Theano http://deeplearning.net/software/theano/install.html#install o Tensorflow https://www.tensorflow.org/install/

2) Caffe: http://caffe.berkeleyvision.org/
E' una libreria in C++ e si puo' installare con o senza supporto di GPU (cioè esiste la versione CPU-only). Il design della rete è effettuato ad alto livello con file di testo. Ha un'interfaccia anche in Phyton.

Un elenco di ulteriori librerie disponibili:
https://www.predictiveanalyticstoday.com/deep-learning-software-libraries/

Report finale.
Nel report dovete descrivere in dettaglio l'architettura neurale utilizzata, il tipo di algoritmo di learning utilizzato, i parameteri di learning della rete in modo indipendente dalla libreria software utilizzata.
Dovete inoltre allegare il codice documentato in modo che i risultati ottenuti siano riproducibili. Il codice non documentato non verrà accettato.
Riportate inoltre i risultati ottenuti, comparando i diversi modelli neurali che avete utilizzato negli esperimenti. Aggiungete eventualmente le vostre osservazioni riguardo alle difficoltà che avete incontrato ed ai possibili sviluppi del lavoro.
N.B.: Il report deve essere consegnato prima dell'esame orale e comunque almeno una settimana prima dell'orale stesso.

Bibliografia.

D. Smedley, M, Schubach, J. Jacobsen, S. Kohler, T. Zemojtel, M. Spielmann, M. Jager, H. Hochheiser, N. Washington, J. McMurry, M. Haendel, C. Mungall, S. Lewis, T. Groza, G. Valentini and P.N. Robinson A Whole-Genome Analysis Framework for Effective Identification of Pathogenic Regulatory Variants in Mendelian Disease, *The American Journal of Human Genetics*, 99:3, pp.595--606, September 2016.

doi.org/10.1016/j.ajhg.2016.07.005

M. Schubach, M. Re, P.N. Robinson and G. Valentini Imbalance-Aware Machine Learning for Predicting Rare and Common Disease-Associated Non-Coding Variants, *Scientific Reports, Nature Publishing*, 7:2959, 2017.

I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, MIT Press, 2016