CANCRO AL SENO

Il cancro al seno è una delle forme più comuni di cancro nella popolazione femminile con oltre 1.300.000 casi e 450.000 decessi ogni anno in tutto il mondo. Esistono diversi sottotipi di cancro al seno, che differiscono per la loro biologia e le loro caratteristiche cliniche.

La classificazione dei tumori al seno si basa sulla biologia del tumore, sulle sue caratteristiche istologiche e sulle sue proprietà genetiche.

Esistono diversi tipi di **tumori della mammella**:

- carcinoma duttale
- carcinoma lobulare

Le altre forme meno frequenti e con prognosi favorevole sono:

- carcinoma intraduttale in situ
- carcinoma tubulare
- carcinoma papillare
- carcinoma mucinoso
- carcinoma cribriforme

Tra questi le forme più comuni di tumore invasivo al seno con prognosi sfavorevole sono:

- Carcinoma Lobulare: si tratta della *tipologia più comune* di neoplasia alla mammella e colpisce nel 70-80% dei casi.
- Carcinoma Duttile: rappresenta il 15% delle neoplasie del seno.

Questa classificazione aiuta a identificare il trattamento più appropriato per ogni paziente e a prevedere la prognosi. Clinicamente, questa malattia eterogenea è classificata in tre gruppi terapeutici di base.

1. Tumori sensibili all'ormone: Questo gruppo comprende i tumori che producono recettori per gli ormoni estrogeni e progesterone. Questi tumori sono spesso trattati con terapie ormonali, come l'inibizione dell'aromatasi o l'uso di antagonisti del recettore degli estrogeni. Il gruppo positivo al recettore degli estrogeni (ER) è il più numeroso e diversificato, con diversi test genomici per aiutare a prevedere i risultati per i pazienti con pronto soccorso sottoposti a terapia endocrina.
2. Tumori HER2-positivi: Questo gruppo comprende i tumori che presentano un elevato livello di espressione del recettore HER2. Questi tumori sono spesso trattati con terapie mirate, come l'utilizzo di anticorpi monoclonali diretti contro HER2.
3. Tumori triplo-negativi: Questo gruppo comprende i tumori che non presentano recettori per gli ormoni estrogeni e progesterone e non presentano un elevato livello di espressione del recettore HER2. Questi tumori sono spesso trattati con terapie standard come la chemioterapia. Hanno un'incidenza aumentata nei pazienti con mutazioni germinali *BRCA1* o di origine africana.

ANALISI DELL'ESPRESSIONE GENICA

Gli approcci sperimentali per studiare su larga scala il profilo trascrizionale, definendo quali geni vengono trascritti e a quale livello in una determinata condizione, sono principalmente due: approcci basati sull'utilizzo di DNA microarray e metodi basati sul sequenziamento con tecnologie NGS dell'RNA a seguito della sua conversione in cDNA.

La **tecnica dell'RNA-seq** permette il sequenziamento degli RNA messaggeri, permette di identificare le molecole di RNA e di quantificare la loro espressione in un campione biologico. Gli RNA cellulari, dopo essere stati estratti dalla cellula, vengono retrotrascritti in DNA (cDNA), sequenziato attraverso tecniche NGS. Vengono così ottenute delle *sequenze di DNA di lunghezza variabile* in base alla tecnologia di sequenziamento utilizzata, definite **reads**. Tali sequenze (reads) vengono poi mappate su un genoma o un trascrittoma di riferimento per identificare i geni espressi nel campione in esame.

L'idea alla base dell'utilizzo del sequenziamento NGS per la stima del livello di espressione genica è che l'abbondanza di un dato RNA in un campione sia direttamente proporzionale al numero di read sequenziate che provengono dall'RNA stesso.

Il livello di espressione di un gene viene quindi stimato sulla base delle read che possono essere assegnate ai suoi trascritti, ovvero che provengono dai suoi esoni. Difatti, ogni gene può esprimere diversi trascritti alternativi sia codificanti che non codificanti proteine grazie al processo dello splicing alternativo che permette la produzione di trascritti multipli da uno stesso locus genico.

Il *totale delle read allineate su un gene* (o un trascritto, o un esone), detto **count**, è una unità di misura dell'espressione del gene stesso. Il vantaggio della tecnologia RNA-Seq sta nella possibilità di essere utilizzata anche quando non è nota la sequenza del gene in esame. Le reads sono sottoposte a controllo di qualità e pre-processing.

Il campo di interesse è quello dell'analisi dell'espressione differenziale, cioè l'identificazione di geni che presentano significative differenze del loro livello di espressione tra due o più condizioni sperimentali (interne o esterne alla cellula). Si valuta cioè, se le differenze osservate tra i count delle diverse condizioni sperimentali siano o meno statisticamente significative.

The Cancer Genome Atlas (TCGA) Breast Cancer

Nell'era del Next Generation Sequencing (NGS) una grande quantità di dati biologici viene sequenziata, analizzata e archiviata in molti database pubblici, la cui interoperabilità è spesso richiesta per consentire una maggiore accessibilità. Questa abbondanza di dati ci permette di effettuare analisi sul corredo genetico di soggetti umani, studiando la predisposizione a malattie come il cancro.

Il *The Cancer Genome Atlas* (TCGA) è il più grande sforzo di caratterizzazione del singolo genoma fino ad oggi, ha raccolto informazioni molecolari su varianti genomiche, trascrittomiche ed epigenomiche di 33 tipi di tumori provenienti da più di 11.000 pazienti, inclusi 10 tumori rari. Utilizzando una serie di strumenti che comprendono le tecnologie omiche all'avanguardia, il progetto ha catalogato tutte le mutazioni somatiche, l'espressione genica differenziale, la metilazione e l'espressione proteica in molti tipi di tumori.

Uno dei tumori più studiati nel progetto TCGA è il cancro al seno, che rappresenta una delle principali cause di morte tra le donne. I progressi nelle tecnologie di sequenziamento di nuova generazione negli ultimi anni hanno consentito uno studio approfondito delle mutazioni somatiche in oltre 1.000 campioni di cancro al seno. Uno dei risultati più importanti del progetto TCGA sul cancro al seno è la classificazione delle subtype di cancro al seno basate sulla biologia intrinseca del tumore. Questa classificazione ha permesso di identificare sottotipi di cancro al seno con prognosi e risposte terapeutiche diverse, aprendo la strada a un trattamento personalizzato basato sul profilo genetico del tumore di ogni paziente.

Un aspetto importante del progetto TCGA è stata l'identificazione di geni significativamente mutati (SMG) nel cancro al seno. Questi SMG sono geni che sono mutati a livello di coppia di basi del DNA ad un tasso più elevato di quanto ci si aspetterebbe in base al tasso di mutazione di fondo. Questi includono geni comunemente mutati come *PIK3CA, TP53, MAP3K1, MAP2K4, GATA3, MLL3* e *CDH11*.

Il progetto TCGA sul cancro al seno ha identificato numerosi nuovi bersagli terapeutici, inclusi geni che sono sovraespressi nei tumori al seno e che possono essere bloccati da farmaci specifici. Queste scoperte hanno aperto la strada a nuove terapie mirate che possono essere più efficaci e con meno effetti collaterali rispetto alle terapie attuali.

Nel 2014 è stato lanciato dall'NCI il Genomic Data Commons (GDC), un database interattivo con interfaccia grafica per organizzare e distribuire in modo facile tutti i dati generati dall'NCI ed altri consorzi con lo scopo di dare alla comunità scientifica una delle più grandi risorse genomiche sul cancro al fine di potenziare la ricerca in campo terapeutico e farmacologico.

TCGA's Breast Cancer Project May Yield Important Therapeutic Benefits, but It's Too Early to Be Sure (cancernetwork.com)

ACQUISIZIONE DEI DATI CLINICI E DEI COUNTS

Il portale dati Genomic Data Commons (GDC) fornisce un accesso web ai dati degli studi di genomica del cancro. Le principali funzionalità di interesse del portale dati GDC sono: un accesso aperto e granulare alle informazioni di tutti i dataset presenti in GDC; una ricerca avanzata con filtraggio assistito dalla visualizzazione; strumenti di visualizzazione dei dati per supportare l'analisi e l'esplorazione dei dati.

L'interfaccia è accessibile anche mediante una API che fornisce un accesso programmatico alle funzionalità GDC permettendo di ottenere e scaricare dati di interesse con formato di comunicazione JSON e protocollo comunicativo HTTP. La comunicazione con l'API GDC comporta l'esecuzione di chiamate agli endpoint API. Ogni endpoint API GDC rappresenta funzionalità specifiche e nel nostro progetto abbiamo utilizzato come endpoint: "files" per accedere ai file di interesse e "genes" per accedere ai geni ritenuti maggiormente correlati a patologie tumorali dalla comunità scientifica.

Attraverso Postman abbiamo creato filtri su file JSON che attraverso l'ambiente di programmazione MATLAB ci hanno permesso di scaricare le informazioni di interesse dal portale.

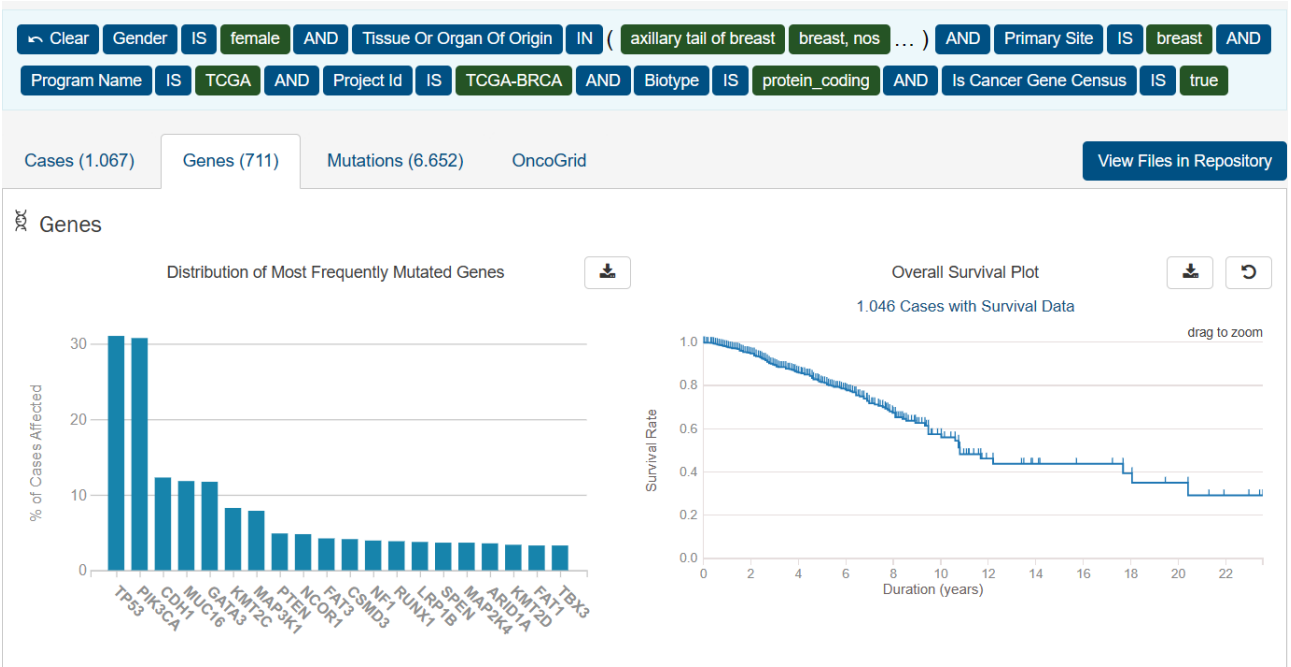
Il primo filtro realizzato per scaricare i dati relativi ai "counts" considera file ad accesso aperto del progetto "TCGA_BRCA" che abbiano come workflow "STAR-Counts" e come approccio sperimentale la tecnica dell'RNA-Seq. La tecnica STAR-counts è una metodologia di analisi delle sequenze di RNA che utilizza l'algoritmo di allineamento STAR (RNA-Seq Alignment to a Reference Transcriptome) per mappare le sequenze di RNA reads al trascritto di riferimento e quindi contare il numero di volte che ogni trascritto è rappresentato nei dati di sequenza. Questa tecnica ha elevata accuratezza e velocità rispetto gli altri algoritmi di allineamento.


```
{
    "filters":{
        "op": "and",
        "content": [
            {
                "op": "=",
                "content": {
                    "field": "cases.project.project_id",
                    "value": "TCGA-BRCA"
                }
            },
            {
                "op": "=",
                "content": {
                    "field": "analysis.workflow_type",
                    "value": "STAR - Counts"
                }
            },
            {
                "op": "=",
                "content": {
                    "field": "access",
                    "value": "open"
                }
            },
            {
                "op": "in",
```

```
                "content": {
                    "field": "cases.samples.sample_type",
                    "value":["metastatic","primary tumor"]
                }

                {
                "op": "in",
                "content": {
                "field": "cases.demographic.gender",
                "value":"female"
            }

                }
        ]
    },

"fields":"file_id,file_name,cases.case_id,cases.diagnoses.primary_diagnosis,cases.diagnoses.ajcc_pathologic_stage,cases.diagnoses.ajcc_pathologic_n,cases.diagnoses.ajcc_pathologic_m,cases.diagnoses.ajcc_pathologic_t,cases.diagnoses.tumor_stage,cases.diagnoses.age_at_diagnosis,cases.diagnoses.morphology,
cases.demographic.gender,cases.demographic.age_at_index,cases.demographic.race,cases.diagnoses.treatments.treatment_or_therapy",
    "size": 1000

}
```

Attraverso il primo filtro abbiamo ottenuto una tabella contenente sulle righe i geni e sulle colonne i pazienti con oltre 60.000 trascritti e una tabella contenente sulle righe i pazienti e sulle colonne le features cliniche.

Il secondo filtro JSON scaricato direttamente dal portale GDC, contenente i geni implicati in patologie tumorali. A causa dell'elevato numero di geni codificanti proteine presenti nel progetto TCGA-BRCA (circa 20.096 geni) si è resa utile una selezione dei 711 geni considerati implicati con i tumori definiti dal *Catalogue of Somatic Mutations in Cancer* (COSMIC) *Cancer Gene Census* (CGC). Il Cancer Gene Census (CGC) è un elenco curato da esperti dei geni per catalogare quei geni che contengono mutazioni che sono state implicate causalmente in un cancro e spiegare come la disfunzione di questi geni guida il cancro.



Le tabelle ottenute che verranno utilizzate nelle successive elaborazioni sono:

Data_table: contiene sulle righe 711 geni e sulle colonne 973 pazienti tali per cui tutti i campi richiesti nel primo filtro fossero presenti. Ogni cella della matrice contiene il numero di read (ovvero i counts) che mappano sugli esoni del gene in ciascuno dei campioni. Un numero di conteggi elevato indica che più reads sono associate a quel gene e suggerisce un livello più elevato di espressione di quel gene.

data_table

711x973 table

	1 ea645243-df49-4466-a255-9f	2 001cef41-ff86-4d3f-a140-a64	3 0685edd2-ce1c-4e0e-8dda-3	4 b2aac45b-2073-4c7a-adb9-7	5 0741b5db-4405-42ba-b63a-c	6 bb8d42d3-ad65-4d88-ae1d-f	7 045c13ef-3db7-4adf-b0a3-
1 LASP1	11270	18309	27106	18998	19239	5677	83
2 HOXA11	55	8	10	71	58	7	
3 CREBBP	8666	10279	8017	7937	4844	16751	6C
4 ETV1	841	387	4334	1365	960	941	7
5 GAS7	4414	1475	3975	2928	4831	1819	1C
6 CD79B	571	178	221	35	44	10	
7 PAX7	1	2	0	2	0	5553	7
8 BTK	584	398	1609	398	614	519	1
9 BRCA1	658	1104	735	321	434	3070	
10 WAS	705	348	2046	417	613	222	2
11 WWTR1	16909	2325	9343	4342	3948	6810	6E
12 CD74	80004	61515	314340	113657	111697	62183	394
13 BIRC3	4174	1504	3134	587	1998	1528	E
14 FAS	807	323	2357	632	670	622	8
15 BCLAF1	8429	7746	9780	7810	7905	23379	9C
16 ANK1	72	62	77	20	57	41	
17 RABEP1	2391	13837	2989	8572	3349	8044	12
18 ZCCHC8	1583	1534	1657	1249	832	2718	11
19 CUL3	5291	4477	4987	3731	3855	7779	21

Clinical_data: oltre ai dati relativi alla trascrittomica per ogni singolo paziente è stato necessario ottenere anche i dati clinici in merito a dati demografici, sulla diagnosi e sulla tipologia di trattamento.

clinical_data

973x10 table

	1 age	2 age_at_diag	3 ajcc_pathologic_stage	4 ajcc_pat_n	5 ajcc_pathologic_t	6 ajcc_pat_m	7 race	8 primary_diag	9 diag	10 treatments
1 ea645243-df49-4466-a255-9f3d4321e357	67	24647	'Stage IA'	'N0'	'T1c'	'MX'	'black or af...	'Infiltrating duc...	'8500/3'	'no'
2 001cef41-ff86-4d3f-a140-a647ac4b10a1	60	22279	'Stage IA'	'N0 (mol+)'	'T1c'	'M0'	'white'	'Infiltrating duc...	'8500/3'	'no'
3 0685edd2-ce1c-4e0e-8dda-393139af4223	31	11354	'Stage I'	'N0 (i-)'	'T1c'	'M0'	'white'	'Secretory carci...	'8502/3'	'yes'
4 b2aac45b-2073-4c7a-adb9-769a4fdcc111	71	26221	'Stage IIB'	'N1'	'T2'	'M0'	'white'	'Infiltrating duc...	'8500/3'	'no'
5 0741b5db-4405-42ba-b63a-c6ee4f341480	79	28940	'Stage IIA'	'N0'	'T2'	'M0'	'white'	'Infiltrating duc...	'8500/3'	'no'
6 bb8d42d3-ad65-4d88-ae1d-f9aadfc7962d	69	25230	'Stage IIA'	'N0 (i-)'	'T2'	'M0'	'white'	'Lobular carcin...	'8520/3'	'yes'
7 045c13ef-3db7-4adf-b0a3-23338f0479f3	49	18014	'Stage IIA'	'N1'	'T1c'	'M0'	'black or af...	'Infiltrating duc...	'8500/3'	'no'
8 cea9d8f9-e18c-4947-a461-5f712e3c1e6d	37	13817	'Stage IA'	'N0'	'T1c'	'M0'	'white'	'Infiltrating duc...	'8500/3'	'yes'
9 0dca98b0-f43e-45b6-9a02-00092c78678c	72	26588	'Stage IIA'	'N0'	'T2'	'M0'	'white'	'Infiltrating duc...	'8500/3'	'no'
10 2fdfd287-d13c-4910-9788-73987d45908a	73	26845	'Stage IA'	'N0 (i-)'	'T1'	'M0'	'white'	'Infiltrating duc...	'8500/3'	'yes'
11 0fe1419e-a005-407c-8ae7-15c4c1579539	51	18788	'Stage I'	'N0'	'T1c'	'M0'	'white'	'Infiltrating duc...	'8500/3'	'yes'
12 6cdc0d53-f813-4101-81e5-9bee68270536	46	17152	'Stage I'	'N0'	'T1c'	'M0'	'white'	'Infiltrating duc...	'8500/3'	'no'
13 b63391a0-73f8-4544-9e94-f6529245ca2a	78	28495	'Stage IIA'	'N1a'	'T1c'	'M0'	'white'	'Lobular carcin...	'8520/3'	'yes'
14 7d9d3522-ec3b-4efe-8c8e-c0e675276ef5	67	24493	'Stage IIA'	'N1'	'T2'	'M0'	'white'	'Infiltrating duc...	'8500/3'	'yes'
15 0bc5744c-5fa3-45bb-87d0-70a02068b392	30	11204	'Stage IIA'	'N0'	'T2'	'M0'	'black or af...	'Infiltrating duc...	'8500/3'	'yes'
16 a4903de8-6cf5-4541-8ec7-065beace8b44	61	22642	'Stage IIIC'	'N3'	'T2'	'M0'	'white'	'Lobular carcin...	'8520/3'	'yes'
17 757df7b0-4774-4493-98bf-999ded9ac86e	63	23229	'Stage IIB'	'N1'	'T2'	'M0'	'white'	'Infiltrating duc...	'8500/3'	'no'
18 67c73260-a242-4bba-87c5-d2302556dff7	50	18535	'Stage IIIA'	'N1mi'	'T3'	'M0'	'white'	'Lobular carcin...	'8520/3'	'yes'
19 35bd694d-1dd2-466f-ab27-03320614b40e	36	13458	'Stage IIA'	'N1mi'	'T1c'	'MX'	'black or af...	'Infiltrating duc...	'8500/3'	'yes'

Tra tutte le features presenti abbiamo selezionato «age at diagnosis» e «race» che verranno utilizzate per il classificatore.

L'output scelto per il nostro classificatore riguarda la primary diagnosis del tumore al seno, in particolare la classe 1 è rappresentata dal Infiltrating ductal carcinoma (IDC) e la classe 2 dal lobular carcinoma. Sono entrambi tipi di cancro al seno, ma differiscono nel loro modo di crescita e presentazione:

- Infiltrating Ductal Carcinoma: è il tipo più comune di cancro al seno, che origina dalle cellule dei dotti lactiferi che portano il latte al capezzolo. IDC si diffonde attraverso i tessuti del seno infiltrandosi in profondità, dando luogo a tumori duri e spesso con una superficie irregolare.
- Lobular Carcinoma: questo tipo di cancro origina dalle cellule dei lobuli, che sono le unità produttrici di latte nel seno. LC si diffonde in modo diverso da IDC, infiltrandosi nei lobuli e poi nelle aree circostanti. Tumori di LC tendono ad essere più morbidi e avere una forma più regolare rispetto a IDC.

Le differenze principali tra IDC e LC riguardano il luogo di origine, la struttura e la modalità di diffusione del tumore. È importante sottolineare che entrambi i tipi di cancro al seno possono essere aggressivi e richiedere un trattamento tempestivo; pertanto, è fondamentale che ogni persona sia consapevole dei sintomi e si sottoponga regolarmente a controlli medici.

Target: è stato scelto come output la "primary diagnosis" distinguendo due casi: "*Infiltrating duct carcinoma*" e "*Lobular carcinoma*". È stata creata la table target con valori logici 0 e 1 rispettivamente per Lobular carcinoma e Infiltrating duct carcinoma.

target

973x1 table

	1 primary_diag
1 ea645243-df49-4466-a255-9f3d4321e357	1
2 001cef41-ff86-4d3f-a140-a647ac4b10a1	1
3 0685edd2-ce1c-4e0e-8dda-393139af4223	0
4 b2aac45b-2073-4c7a-adb9-769a4fdcc111	1
5 0741b5db-4405-42ba-b63a-c6ee4f341480	1
6 bb8d42d3-ad65-4d88-ae1d-f9aadfc7962d	0
7 045c13ef-3db7-4adf-b0a3-23338f0479f3	1
8 cea9d8f9-e18c-4947-a461-5f712e3c1e6d	1
9 0dca98b0-f43e-45b6-9a02-00092c78678c	1
10 2fdfd287-d13c-4910-9788-73987d45908a	1
11 0fe1419e-a005-407c-8ae7-15c4c1579539	1
12 6cdc0d53-f813-4101-81e5-9bee68270536	1
13 b63391a0-73f8-4544-9e94-f6529245ca2a	0
14 7d9d3522-ec3b-4efe-8c8e-c0e675276ef5	1
15 0bc5744c-5fa3-45bb-87d0-70a02068b392	1

PREPROCESSING DEI DATI

Si è reso utile un preprocessing dei dati al fine di:

1. **Eliminare le righe contenenti *valori NaN* dalla tabella clinical_data**

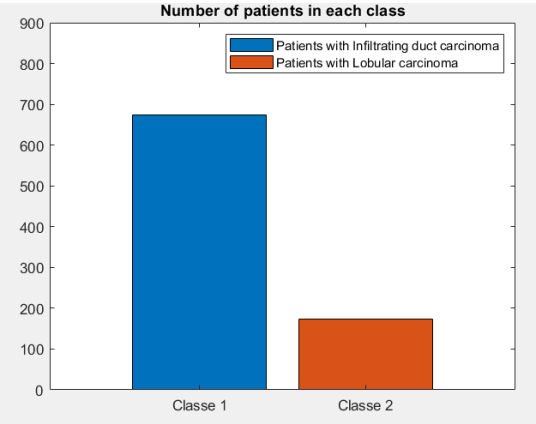
```
% elimino le righe con valori NaN
missing_values = ismissing(clinical_data);
sum_missing_values = sum(missing_values, 2);
missing_rows = find(sum_missing_values > 0);
clinical_data(missing_rows, :) = [];
data_table(:, missing_rows) = [];
```

Eliminare righe con valori NaN (Not a Number) delle features cliniche è importante perché questi valori rappresentano una mancanza di informazioni sulle caratteristiche del campione. I valori NaN possono causare problemi nell'analisi e nella interpretazione dei risultati, poiché possono distorcere i risultati o

generare risultati non significativi. Inoltre, l'uso di valori NaN nelle analisi può causare errori nell'utilizzo di alcuni algoritmi statistici e nella selezione delle features più rilevanti. Il numero di pazienti si riduce a 770.

Su 770 pazienti totali abbiamo che:

- 609 pazienti appartenenti alla classe Infiltrating duct carcinoma, NOS
- 161 pazienti appartenenti alla classe Lobular carcinoma, NOS



2. **Eliminare le righe contenenti un certo numero di *counts nulli* dalla tabella data_table:**

```
%Elimino le righe che hanno i counts nulli
geneData = table2array(data_table);
mask = geneData > 0;
sum_mask = sum(mask,2);
idx = sum_mask >= size(geneData,2)*80/100;
data_table(~idx,:) = [];
```

Eliminare le righe con counts nulli (0) in un'analisi di espressione genica differenziale è necessario perché questi dati non rappresentano alcuna informazione utile sulla quantità di espressione del gene. I counts nulli possono essere il risultato di una scarsa qualità delle sequenze di RNA, di un'assenza di trascrizione del gene nelle condizioni studiate o di un'inefficienza nella sequenza delle librerie. Questi dati possono causare problemi nell'analisi e nella interpretazione dei risultati, pertanto è importante rimuoverli.

I geni da 711 diventano 682 poiché abbiamo eliminato le righe dei geni dove più dell'80% dei pazienti presentava counts pari a zero poiché possono compromettere i risultati influendo sulle stime dell'espressione genica.

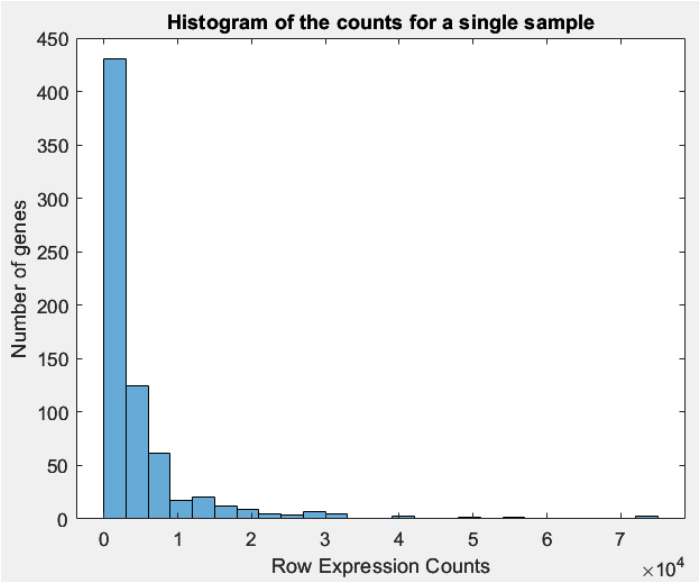
data_table

682x783 table

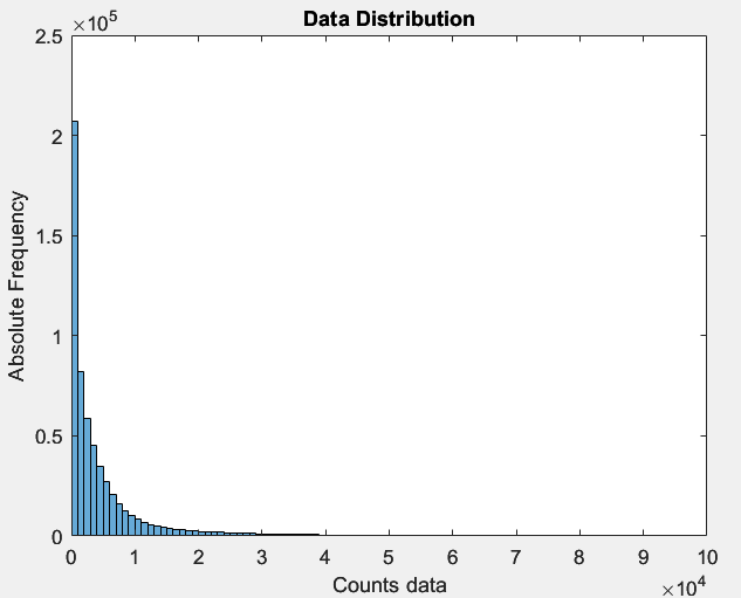
	1 ea645243-df49-4466-a255-9f	2 001cef41-ff86-4d3f-a140-a64	3 b2aac45b-2073-4c7a-adb9-7	4 0741b5db-4405-42ba-b63a-	5 cbb8d42d3-ad65-4d88-ae1d-f
1 LASP1	11270	18309	18998	19239	5677
2 HOXA11	55	8	71	58	7
3 CREBBP	8666	10279	7937	4844	16751
4 ETV1	841	387	1365	960	941
5 GAS7	4414	1475	2928	4831	1819
6 CD79B	571	178	35	44	10
7 BTK	584	398	398	614	519
8 BRCA1	658	1104	321	434	3070
9 WAS	705	348	417	613	222
10 WWTR1	16909	2325	4342	3948	6810
11 CD74	80004	61515	113657	111697	62183
12 BIRC3	4174	1504	587	1998	1528
13 FAS	807	323	632	670	622
14 BCLAF1	8429	7746	7810	7905	23379
15 ANK1	72	62	20	57	41
16 RABEP1	2391	13837	8572	3349	8044
17 ZCCHC8	1583	1534	1249	832	2718
18 CUL3	5291	4477	3731	3855	7779
19 FLT4	1919	561	1400	314	315
20 CDH1	8028	18879	14400	19819	23293
21 TNC	16620	12564	16747	10237	5173
22 EPHA3	268	215	286	418	992

CARATTERISTICHE DEI DATI DI CONTEGGIO RNA-SEQ

Per avere un'idea di come sono distribuiti i conteggi di RNA-seq, tracciamo un istogramma dei conteggi per un singolo campione:



Tracciamo anche un istogramma dei conteggi per tutti i campioni in analisi:



Entrambi i grafici illustrano alcune **caratteristiche comuni** dei dati di conteggio dell'RNA-seq:

- un basso numero di conteggi associati a una grande percentuale di geni
- una lunga coda destra a causa della mancanza di qualsiasi limite superiore per l'espressione
- Ampia gamma dinamica

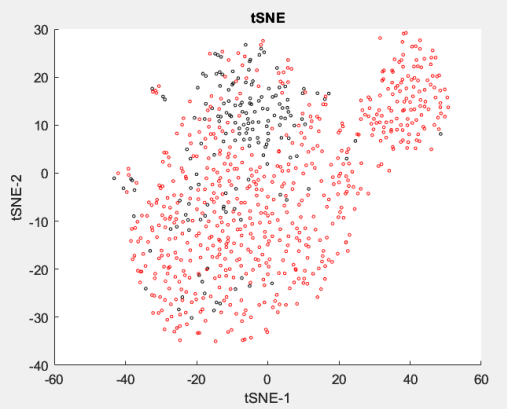
Osservando la forma dell'istogramma, vediamo che *non è distribuito normalmente*.

VISUALIZZAZIONE DEI DATI DA ANALIZZARE

Per visualizzare i dati è stata sfruttata la tecnica di riduzione di dimensionalità t-SNE (*t-distributed Stochastic Neighbor Embedding*). La riduzione di dimensionalità ci permette di visualizzare i dati in maniera compatta in 2 o 3 dimensioni.

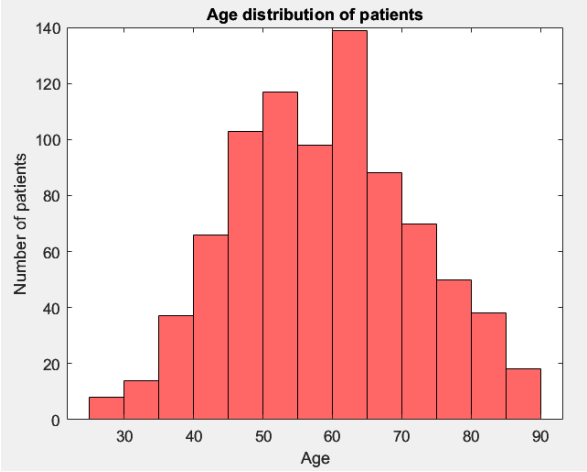
La t-SNE è una tecnica di riduzione della dimensionalità adatta per incorporare dati dimensionali per la visualizzazione in uno spazio a bassa dimensione. Oggetti simili sono modellati da punti vicini e oggetti dissimili sono modellati da punti distanti con alta probabilità. Questo algoritmo si distingue in due fasi:

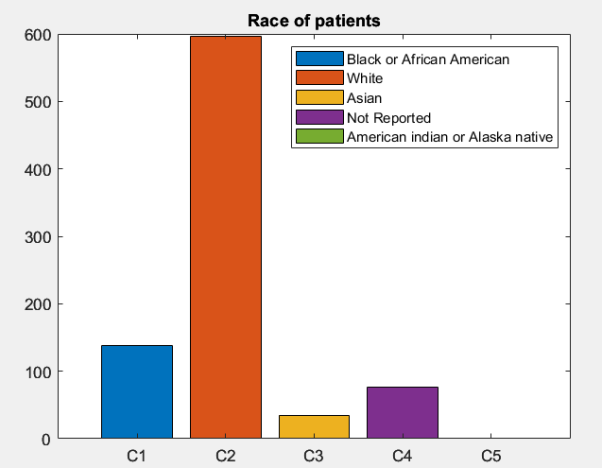
1. Si costruisce una distribuzione di probabilità su coppie di oggetti ad alta dimensione in modo tale che oggetti simili abbiano un'alta probabilità di essere scelti, mentre punti dissimili hanno una probabilità estremamente piccola di essere scelti.
2. Si definisce una distribuzione di probabilità simile sui punti nella mappa a bassa dimensione e si minimizza la divergenza di Kullback-Leibler tra le due distribuzioni rispetto alle posizioni dei punti nella mappa.

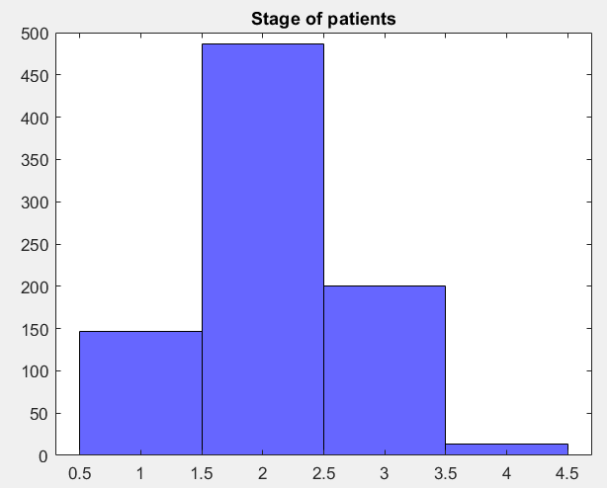


L'algoritmo "exact" ottimizza la divergenza Kullback-Leibler delle distribuzioni tra lo spazio originale e lo spazio incorporato.

Per le features cliniche, i dati relativi ai pazienti risultano distribuiti nel seguente modo:







NORMALIZZAZIONE

I conteggi iniziali presenti nella Matrice dei Conteggi non sono valori direttamente interpretabili come livelli di espressione e non possono essere utilizzati direttamente per le analisi successive. Questo perché:

1. Geni diversi producono RNA di lunghezza differente, come pure uno stesso gene può produrre trascritti alternativi di lunghezza differente. A parità di espressione, un RNA più lungo produrrà più frammenti di un RNA più corto per cui confrontare direttamente i conteggi iniziali potrebbe portare alla conclusione errata che un gene sia espresso più di un altro, quando invece la differenza nel conteggio delle reads è dovuta unicamente alla differenza di lunghezza dei rispettivi RNA.
2. Il conteggio non offre una stima assoluta del livello di espressione, ma relativa: il conteggio delle reads di un gene dovrà quindi essere confrontato con il numero totale di reads che sono stati attribuite a tutti i geni studiati nello stesso paziente.

È necessaria, dunque, la normalizzazione dei dati. I conteggi iniziali non vengono normalizzati confrontando i conteggi di tutti i geni in una condizione (normalizzazione per colonne) ma vengono normalizzati confrontando i conteggi dello stesso gene in tutti gli esperimenti (normalizzazione per righe).

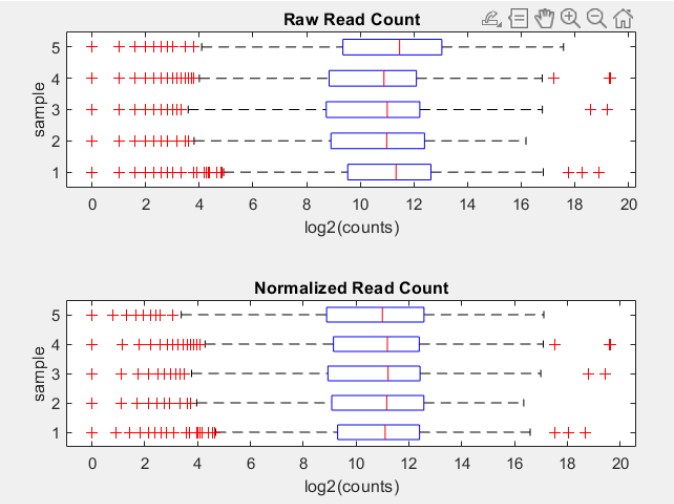
I counts delle reads nei dati di RNA-Seq sono linearmente correlati all'abbondanza di trascrizioni. Tuttavia, i counts delle reads per un dato gene dipendono non solo dal livello di espressione del gene, ma anche dal numero totale di letture sequenziate (*depth of coverage*) e dalla lunghezza della trascrizione del gene (*length of coverage*). Pertanto, per dedurre il livello di espressione di un gene dal conteggio delle letture, dobbiamo tenere conto della profondità del sequenziamento e della lunghezza del trascritto del gene.

Normalizzazione Con Size Factors

L'obiettivo della Normalizzazione con Size Factor è quello di portare tutti i valori dei conteggi su una scala comune, rendendoli comparabili. Questo viene fatto dividendo i conteggi di ciascun campione per i corrispondenti size-factor.

La procedura seguita è:

1. Per ogni gene si calcola un campione di "preudoreference" pari alla media geometrica di tutti i campioni.
2. Si calcola il rapporto di ciascun campione con quello di "preudoreference"
3. Si calcola il size factor per ogni campione prendendo la mediana dei rapporti dei conteggi osservati rispetto a quelli di un campione di "preudoreference"
4. Infine, si calcolano i valori dei counts normalizzati dividendo il valore grezzo di ogni campione per il size factor di quel campione

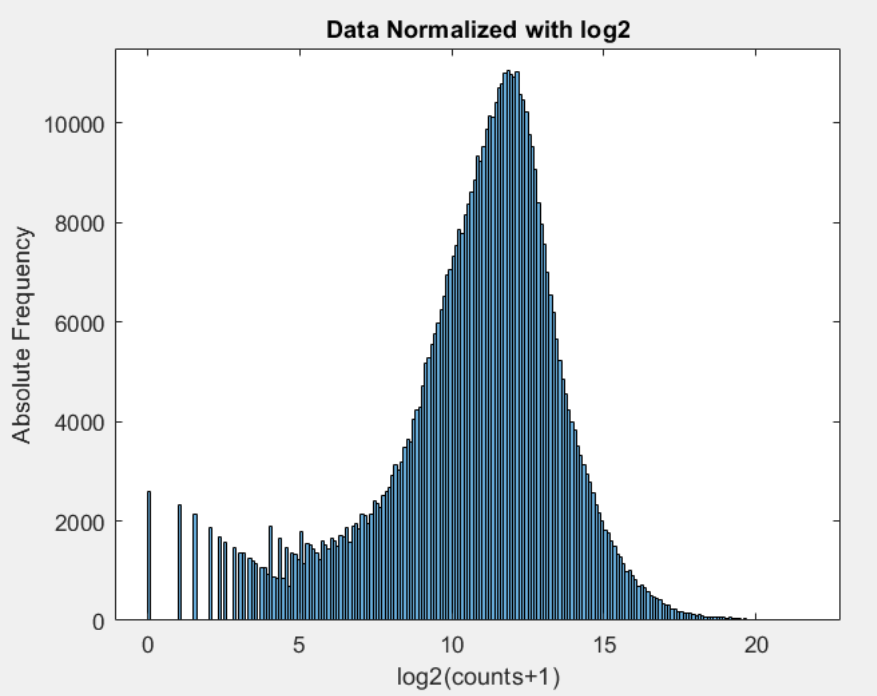


Normalizzazione Con Log2

I dati di espressione genica possono avere una vasta gamma di valori e spesso presentano una distribuzione asimmetrica. La normalizzazione con il log2 prima di eseguire una analisi di componenti principali (PCA) aiuta a rendere i dati più uniformi e omogenei, rendendo la PCA più affidabile.

Questa operazione è utile perché:

- Modifica la scala dei dati: I counts dei geni spesso hanno una scala molto ampia, con alcuni geni che sono molto più espressi di altri. Questo rende più facile la comparazione tra i geni e tra i campioni.
- Questa trasformazione ha l'effetto di "appiattire" i valori più elevati e di "espandere" i valori più bassi.
- Rende i dati più stabili: I dati normalizzati con log2 sono meno sensibili alle variazioni nei counts dei geni più espressi rispetto ai dati non normalizzati.

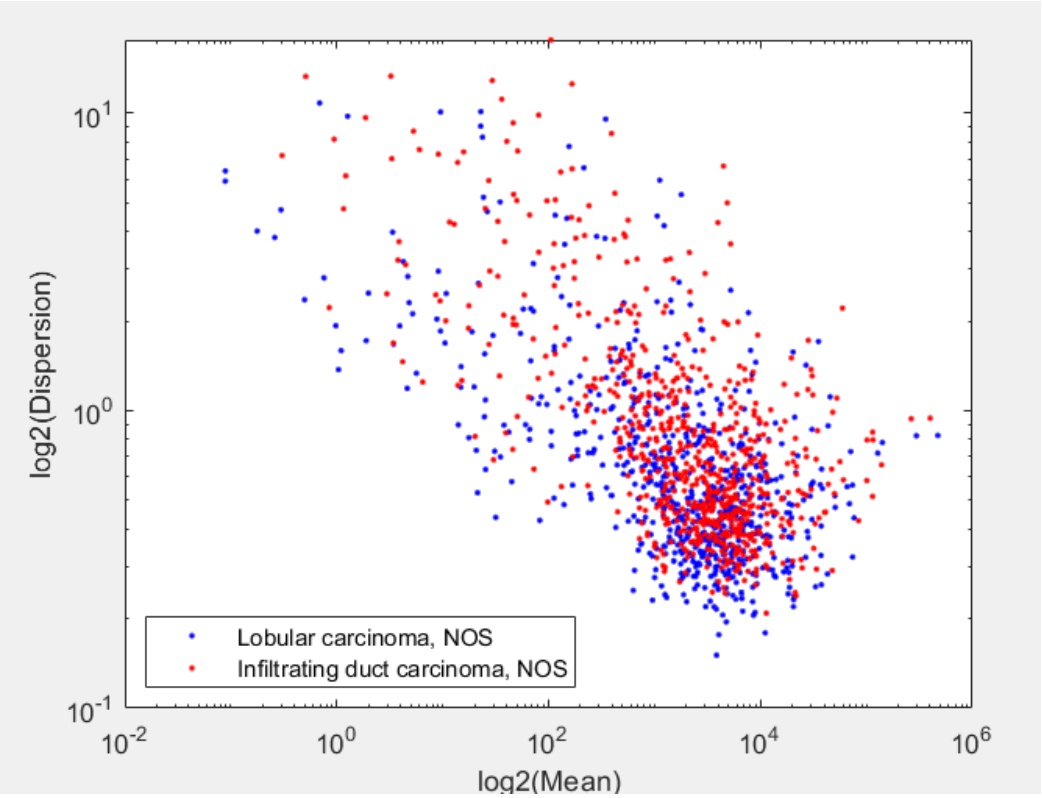


WORKFLOW 1

Media, Dispersione E Fold Change Dei Conteggi Normalizzati Con Size Factor

In un'analisi di espressione genica differenziale, la media, la deviazione standard e il fold change vengono solitamente calcolati sui counts normalizzati con size factor. Questo perché i counts grezzi possono essere influenzati da molte variabili, come la quantità di RNA estratto e la variazione di efficienza di amplificazione tra i campioni. La media, la deviazione standard e il fold change calcolati sui counts normalizzati forniscono informazioni più affidabili sulla variazione di espressione tra i geni e i campioni di interesse.

È possibile tracciare i valori di dispersione empirica rispetto alla media dei conteggi normalizzati in una scala logaritmica:



È possibile osservare la differenza dell'espressione genica tra due condizioni, calcolando il Fold Change (FC) per ciascun gene, ovvero il rapporto tra i conteggi nel gruppo con “Lobular Carcinoma” rispetto ai conteggi nel gruppo “Infiltrating dust Carcinoma”.

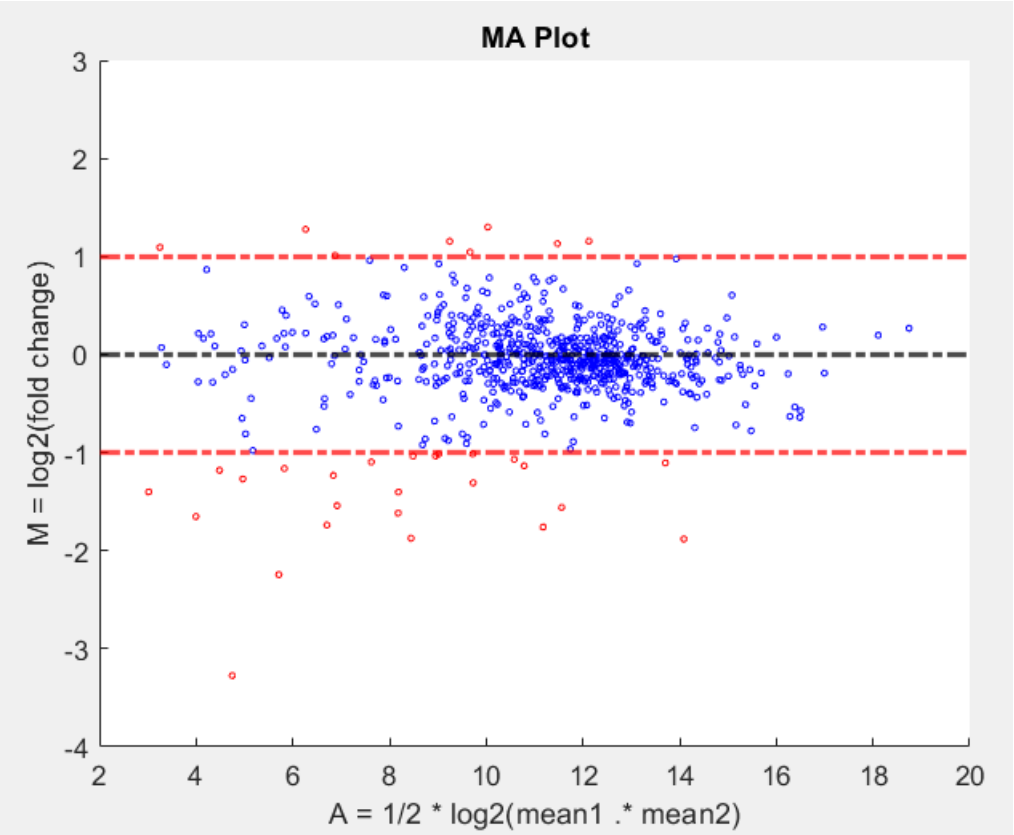
Generalmente questi rapporti sono considerati nella scala log2 che consente di quantificare e comparare i cambiamenti nell'espressione genica in modo preciso e uniforme. Per esempio, un $FC = 1/2$ o $FC = 2/1$ corrisponde nella scala logaritmica a $log2FC = -1$ o $log2FC = +1$. Questo significa che un aumento di 2 volte nell'espressione genica (ad esempio, se l'espressione di un gene passa da 1 a 2 unità) o una riduzione di 2 volte viene tradotto in un $log2FC$ di 1 o -1.

```
% compute the mean and the log2FC
meanBase = (mean1 +  mean2) / 2;
foldChange = mean1 ./ mean2;
log2FC = log2(foldChange);
```

L'MA plot (Mean-Difference Plot) è un tipo di grafico utilizzato per rappresentare la differenza di espressione di geni in due campioni differenti, facendo un confronto tra il rapporto di espressione logaritmico (M) mostrato sulle ordinate e la deviazione standard logaritmica (A) mostrata sulle ascisse.

Questo tipo di grafico consente di visualizzare in modo semplice e intuitivo la differenza di espressione tra i geni e di identificare quelli che hanno una differenza significativa di espressione tra due classi. Inoltre, l'MA plot è utile per valutare la qualità dei dati e la bontà dell'analisi, in quanto mostra la distribuzione dei dati e la presenza di eventuali outlier.

I dati rappresentati in rosso corrispondono ai geni differenzialmente espressi nelle due classi, in particolare i geni up-regolati sono quelli in cui il $log2FC > 1$ e i geni down-regolati sono quelli in cui il $log2FC < -1$.



Modellazione Dei Conteggi

L'obiettivo è quello di identificare quali geni variano la propria espressione in modo statisticamente significativo nelle condizioni studiate. Per far ciò si utilizzano solitamente **test statistici** in cui la significatività statistica deve riflettere, per quanto possibile, la significatività biologica della variazione.

È necessario valutare se la variazione delle espressioni di un gene in due o più condizioni è superiore alla variabilità attribuibile unicamente a fattori sperimentali. I test statistici confrontano per ciascun gene il valore di espressione medio in ciascuna delle condizioni, utilizzando la varianza risultante dalle repliche per valutare la variabilità sperimentale o biologica dell'espressione.

I dati di conteggio in generale possono essere modellati con varie distribuzioni:

1. **Distribuzione binomiale:** È una distribuzione di probabilità discreta che descrive il numero di successi in un esperimento di Bernoulli, ovvero la variabile aleatoria $S_n = X_1 + X_2 + \cdots + X_n$ che somma n variabili aleatorie indipendenti di uguale distribuzione di Bernoulli. Esempi di casi di distribuzione binomiale sono i risultati di una serie di lanci di una stessa moneta o di una serie di estrazioni da un'urna (con reintroduzione), ognuna delle quali può fornire due soli risultati: il *successo* con probabilità p e il *fallimento* con probabilità $q = 1 - p$. La *distribuzione binomiale negativa* è una distribuzione di probabilità discreta con due parametri, p e n, che descrive il numero di fallimenti precedenti il successo n-esimo in un processo di Bernoulli di parametro p. In questa distribuzione la varianza σ^2 è funzione della media μ e dipende da un parametro φ detto dispersione: $\sigma^2 = \mu + \varphi\mu^2$.
2. **Distribuzione di Poisson:** È una distribuzione di probabilità discreta che esprime le probabilità per il numero di eventi che si verificano successivamente ed indipendentemente in un dato intervallo di tempo, sapendo che mediamente se ne verifica un numero λ (valore medio). Quando il numero di casi è molto grande (ad esempio persone che acquistano i biglietti della lotteria), ma la probabilità di un evento è molto piccola (probabilità di vincita) si utilizza la distribuzione. È appropriato per i dati in cui media == varianza.

Con i dati RNA-Seq, viene rappresentato un numero molto elevato di RNA e la probabilità di estrarre una particolare trascrizione è molto piccola.

Se le proporzioni di mRNA rimanessero esattamente costanti tra le repliche biologiche per un gruppo campione, potremmo aspettarci una distribuzione di Poisson (dove media == varianza). Tuttavia, ci aspettiamo sempre una certa variabilità tra le repliche. Poiché stiamo modellando delle variabili provenienti da individui diversi, a causa dell'eterogeneità biologica, la varianza di questi dati è maggiore della media. Questi dati sono quindi da considerarsi overdispersi, e quindi è necessario usare la distribuzione binomiale negativa come modello.

La distribuzione che è stata dimostrata meglio modellare la loro variabilità sperimentale è la distribuzione binomiale negativa.

Inferenza dell'espressione differenziale con un modello binomiale negativo

La distribuzione che meglio si adatta ai dati RNA-seq, dato questo tipo di variabilità tra le repliche, è la distribuzione binomiale negativa che è una buona approssimazione per i dati in cui la media < varianza, come nel caso dei dati di conteggio RNA-Seq.

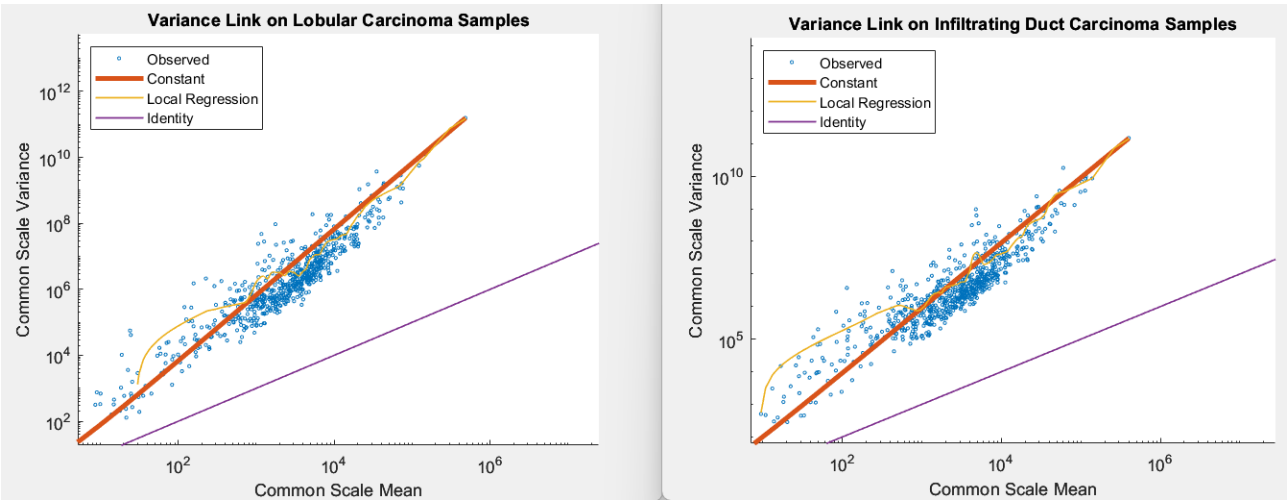
In Matlab test = nbintest(X,Y,name, value) esegue un test di ipotesi che due campioni indipendenti di dati di conteggio a lettura breve, in ogni riga di X e Y, provengano da distribuzioni con medie uguali sotto i presupposti che:

1. I counts sono modellati utilizzando la distribuzione binomiale negativa.
2. La varianza e la media dei dati in ciascuna riga sono collegate tramite una funzione di regressione lungo tutte le righe.

test è un oggetto NegativeBinomialTest con p-Values a due lati archiviati nella proprietà pValue. Con 'VarianceLink' specifico la relazione tra media e varianza; le opzioni sono:

- **'Local Regression'** (Default) → La varianza è la somma del termine del rumore shot (media) e di una funzione regolare non parametrica regredita localmente della media. Questa opzione è l'impostazione predefinita e si usa quando i dati sono sovradispersi e hanno più di 1000 righe (geni).
- **'Constant'** → La varianza è la somma del termine del rumore shot (media) e di una costante moltiplicata per la media al quadrato. Questo metodo utilizza tutte le righe nei dati per stimare la costante. Si utilizza questa opzione quando i dati sono distribuiti in modo eccessivo e hanno meno di 1000 righe.
- **'Identity'** → La varianza è uguale alla media. I conteggi sono quindi modellati individualmente dalla distribuzione di Poisson per ogni riga di X e Y. Si utilizza questa opzione quando i dati hanno pochi geni e la regressione tra la varianza e la media non è possibile a causa del numero molto ridotto di campioni o repliche. Questa opzione non è consigliata per i dati con dispersione eccessiva.

La **linea Identity** rappresenta il modello di Poisson, dove la varianza è identica alla media. Si osservi che i dati sembrano essere sovradispersi (ovvero, la maggior parte dei punti si trova al di sopra della linea Identity). La **linea Constant** rappresenta il modello binomiale negativo, dove la varianza è la somma del termine del rumore di tiro (media) e una costante moltiplicata per la media al quadrato. Le opzioni "Local Regression" e "Constant linkage" sembrano adattarsi meglio ai dati sovradispersi.



p-Value

Nei test di verifica d'ipotesi, il **valore *p*** dal punto di vista statistico è la probabilità, per una ipotesi supposta vera (ipotesi H0), che la variazione di espressione di un gene sia dovuta al caso e dal punto di vista biologico che la variazione sia dovuta unicamente a fattori sperimentali.

In altri termini, il valore *p* aiuta a capire se la differenza tra il risultato osservato e quello ipotizzato è dovuta alla casualità introdotta dal campionamento, oppure se tale differenza è statisticamente significativa, cioè difficilmente spiegabile mediante la casualità dovuta al campionamento.

Quando si effettua un test d'ipotesi si fissa un'ipotesi nulla e un valore soglia α che indica il livello di significatività del test. Nel nostro studio poniamo **α=0.01**, ovvero siamo disposti ad accettare una probabilità dell'1% di commettere un errore di tipo I, ovvero individuare un False Positive.

Calcolato il *p-value* relativo ai dati osservati è possibile comportarsi come segue:

- se valore $p > \alpha$ accetto l'ipotesi nulla;
- se valore $p \leq \alpha$ l'evidenza empirica è fortemente contraria all'ipotesi nulla che quindi va rifiutata. In tal caso si dice che i dati osservati sono statisticamente significativi.

Al termine del test statistico a ciascun gene sarà assegnato un valore di probabilità. L'output di nbintest include un vettore di p-value. Un valore p-value indica la probabilità che si verifichi un cambiamento nell'espressione così forte come quello osservato sotto l'ipotesi nulla, cioè le condizioni non hanno alcun effetto sull'espressione genica.

La soglia dei valori P per determinare quali Fold Changes sono più significativi di altri non è appropriata per questo tipo di analisi dei dati, a causa del problema dei test multipli. Durante l'esecuzione di un gran numero di test simultanei, la probabilità di ottenere un risultato significativo semplicemente a causa del caso aumenta con il numero di test. Per tenere conto di test multipli, eseguire una correzione (o aggiustamento) dei valori P in modo che la probabilità di osservare almeno un risultato significativo dovuto al caso rimanga al di sotto del livello di significatività desiderato.

Aggiustamento del p-value

Per andare ad individuare i geni differenzialmente espressi, in modo significativo, non è sufficiente considerare il p-value del test statistico, in quanto questo non tiene conto dell'elevato numero di test effettuati, per cui andremmo a commettere un elevato numero di FP (accetto H1 e rigetto l'ipotesi H0 che tuttavia è vera).

La probabilità di commettere un FP aumenta con l'aumentare del numero di test effettuati. Difatti aumenta la probabilità di osservare un evento raro e quindi il rischio di identificare casualmente relazioni significative che non esistono veramente.

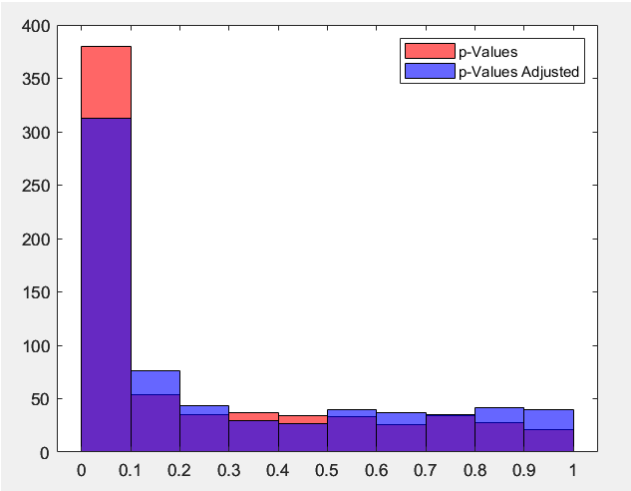
La Family Wise Error Rate (FWER) e la False Discovery Rate (FDR) sono due metodi utilizzati per gestire il problema dei falsi positivi in test multipli. Per questo motivo, nel caso di totale indipendenza tra i test, è possibile effettuare due tipi di correzioni del p-value:

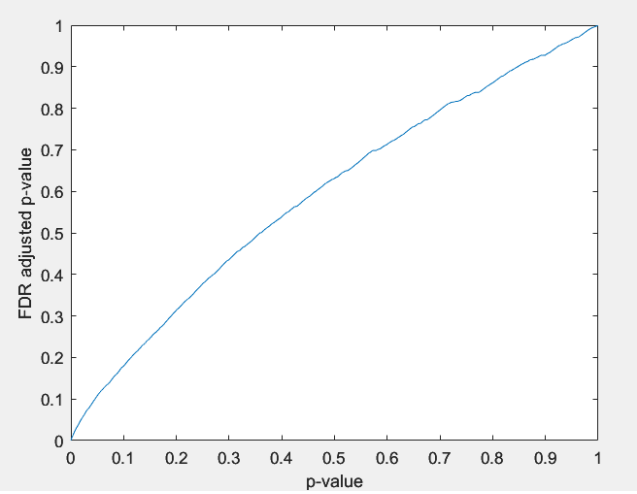
- **Family wise error rate**, è la probabilità di commettere almeno un falso positivo su tutti gli n test effettuati. È una correzione molto stringente in quanto riduce il potere di un test, dove il potere di un test è legato alla capacità del test di non commettere errori sui falsi negativi.
    - Un metodo basato sul calcolo del FWER è la Correzione di Bonferroni dove vado a dividere alfa per il numero dei test (n), per cui rigetto l'ipotesi nulla quando il $pValue < \frac{alpha}{n}$. Si va a ridurre la probabilità di rifiutare l'ipotesi nulla.
- **False Discovery rate**: rappresenta la proporzione di falsi positivi rispetto a tutti i risultati positivi, ovvero è la proporzione di risultati che vengono identificati come significativi, ma che in realtà non lo sono. Il FDR è un modo di quantificare l'errore di tipo I (falso positivo) in un'analisi statistica multipla, in cui vengono effettuati molti test indipendenti su un insieme di variabili. Il FDR fornisce una stima del tasso di questi falsi positivi rispetto al numero totale di risultati significativi.
    - Un esempio di questa correzione è la Correzione di Benjamini – Hochberg, la quale per un certo valore di significatività alfa, si trova il K più grande tale che rigetto l'ipotesi nulla quando il $pValue < \frac{alpha}{n} \cdot K$. Il valore K rappresenta il numero di risultati significativi accettabili.

La FWER viene utilizzata in un contesto in cui è importante mantenere un basso livello di errore di tipo I, come ad esempio nei trial clinici. Per la finalità del nostro studio preferiamo sbagliare sui FP perché è sempre meglio ritenere un paziente malato quando in realtà è sano piuttosto che ritenerlo sano quando invece è malato e quindi non fornire le cure necessario.

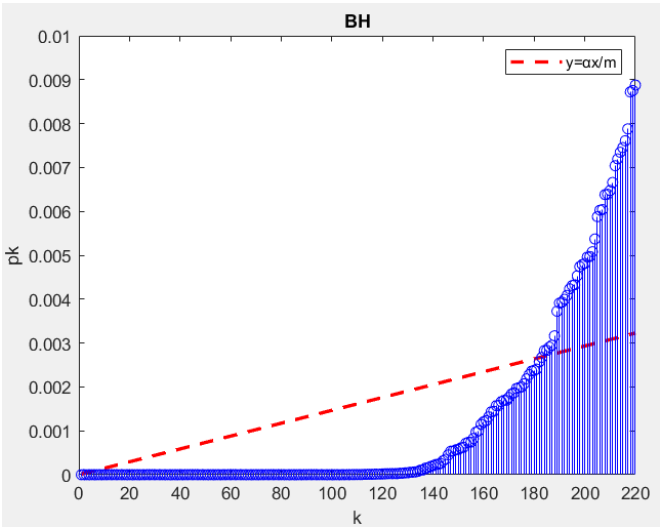
Al fine di limitare gli errori sui falsi negativi optiamo per la correzione di Benjamini – Hochberg che non va a ridurre di molto il potere del test. L'aggiustamento di Benjamini-Hochberg è un metodo statistico che fornisce un P-value aggiustato che risponde alla seguente domanda: quale sarebbe la frazione di falsi positivi se tutti i geni con P-value aggiustato al di sotto di una data soglia fossero considerati significativi?

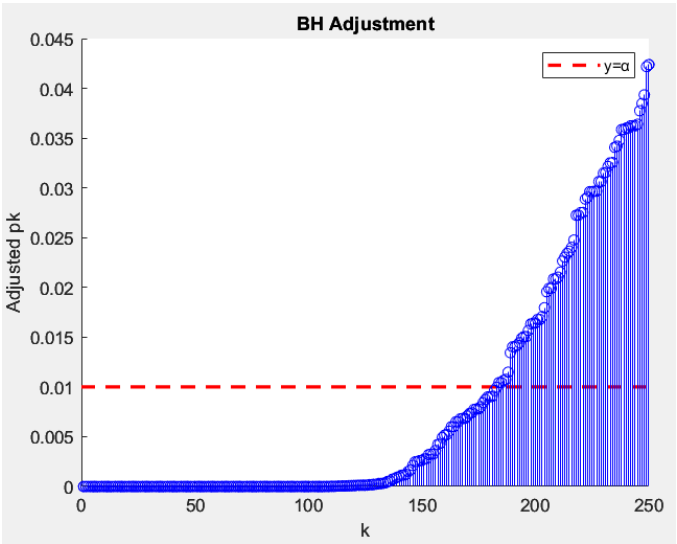
Fissando un livello di significatività $\alpha = 0.01$ allora ho un valore di *false discovery rate* del' 1%. Si identificano i geni espressi in modo significativo considerando tutti i geni con valori P aggiustati al di sotto di questa soglia.





Con la Correzione di Benjamini – Hochberg viene determinato il valore K tale che tutti i p-value sotto la curva di coefficiente angolare α/m corrispondono alle ipotesi verificate. Ovvero tutti i p-Values al di sotto della linea rossa sono associati ai geni significativamente statistici, ovvero i geni per i quali le differenze osservate non sono imputabili al caso ma sono significative. Rifiutando più ipotesi nulle abbiamo più errori di tipo I (falsi positivi) e meno di tipo II (falsi negativi).





Selezione geni significativi e individuazione dei geni up-regolati e down-regolati

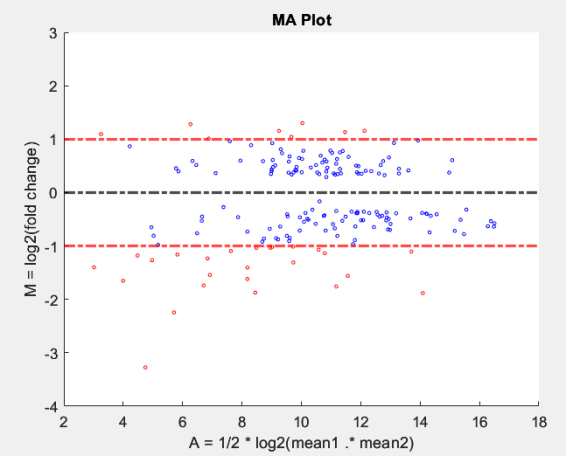
Una volta applicata la correzione di Benjamini - Hochberg andiamo quindi a selezionare i geni significativi come quei geni che hanno un p-value aggiustato minore di 0.01 andando così a ridurre significativamente il numero di features.

In seguito, si individuano i geni differenzialmente espressi andando a confrontare, per ciascun gene significativo, il log2FC prima con il valore 1 e poi con il valore -1 per identificare i geni differenzialmente espressi. Si considera la scala log2, in modo da rilevare raddoppi o dimezzamenti mediante la variazione di +1 o -1 (nella scala logaritmica) del *fold change* (FC) definito come il rapporto tra le medie dei livelli di espressione genica nei due gruppi sperimentali. I geni sono differenzialmente espressi quando:

- Se il log2FC > 1 allora il gene è up-regolato
- Se il log2FC < - 1 allora il gene è down-regolato

Dalla nostra analisi abbiamo ottenuto:

```
There are 183 significant genes on 682
There are 8 Up-regulated genes
There are 25 Down-regulated genes
```



A questo punto costruiamo la matrice normCounts_sig che presenta 183 geni e 770 pazienti:

normCounts_sig

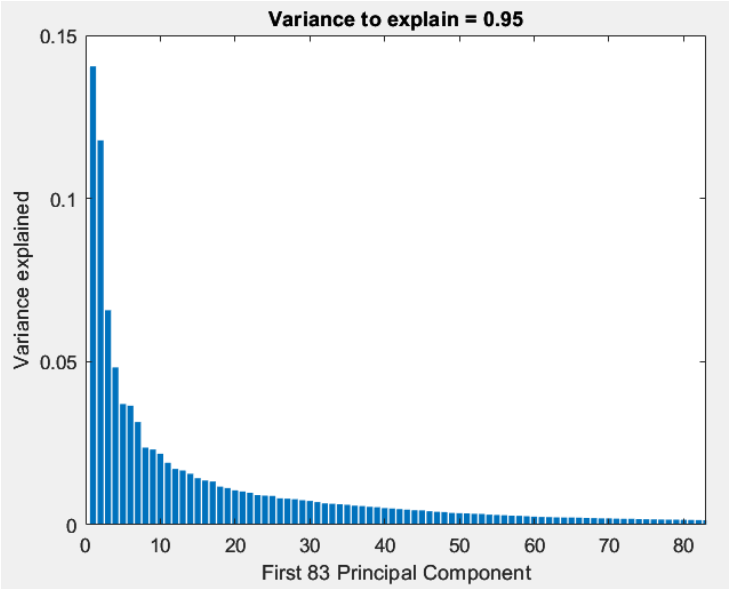
183x770 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	46.6377	9.0177	81.5853	71.5644	5.0441	22.8975	114.0353	222.8866	34.2578	490.2078	36.2393	42.3864	7.1830	82.3896
2	3.7429e+03	1.6626e+03	3.3645e+03	5.9608e+03	1.3108e+03	1.1362e+03	5.4473e+03	9.8222e+03	3.2447e+03	1.0935e+03	5.5729e+03	5.3419e+03	1.8508e+03	4.1591e+03
3	484.1840	200.6446	40.2181	54.2902	7.2059	90.4997	37.3675	30.3936	24.4699	200.6620	100.7452	24.7254	57.4640	152.2642
4	557.9563	1.2444e+03	368.8573	535.4992	2.2122e+03	50.1565	1.2885e+03	109.7548	865.2552	1.4827e+03	781.3185	729.9886	812.8764	467.2218
5	61.0530	69.8875	22.9818	70.3305	29.5442	10.9036	32.2134	62.4758	13.7031	99.6576	39.8632	35.3220	46.6895	80.3037
6	1.6272e+03	632.3688	1.6087e+03	387.4349	226.9856	927.8949	382.6946	645.8647	708.6479	451.1528	1.0118e+03	1.0208e+03	357.9529	1.2202e+03
7	6.8074e+03	2.1281e+04	1.6547e+04	2.4454e+04	1.6785e+04	1.2174e+04	3.8680e+04	3.5042e+04	2.7008e+04	1.3295e+04	1.5732e+04	2.3433e+04	4.3952e+04	5.8658e+04
8	526.5819	509.5021	2.0500e+03	404.7091	792.6481	162.4634	400.7341	401.8714	394.4546	202.0087	1.1452e+03	720.5694	506.4016	657.0306
9	1.6959	2.2544	9.1927	4.9355	2.1618	9.8132	12.2411	54.0331	6.8516	1.3467	1.4496	7.0644	21.5490	6.2574
10	2.3686e+04	2.6738e+03	5.0640e+03	5.7350e+03	1.3951e+03	7.7372e+03	1.2117e+04	2.1951e+03	4.6512e+03	472.7004	8.1509e+03	8.1358e+03	1.7155e+03	3.7711e+03
11	339.1832	227.6979	149.3814	193.7175	498.6477	1.0893e+03	507.0382	535.2657	143.8829	750.1256	83.3503	136.5785	417.8113	275.3271
12	3.3918	20.2899	1.1491	132.0240	7.2059	1.0904	3.2213	6.7541	2.9364	0	1.4496	3.5322	1.1972	12.5149
13	2.7626e+03	2.6456e+03	3.1945e+03	1.8867e+05	1.1344e+04	4.2688e+03	1.9193e+03	1.0538e+04	3.9103e+03	3.4436e+03	4.7850e+03	4.2069e+03	865.5517	1.2880e+03
14	1.5678e+04	7.2818e+03	8.8170e+03	8.0535e+03	8.2046e+03	1.2583e+04	6.9336e+03	6.0568e+03	5.6144e+03	4.2732e+03	1.3099e+04	5.5255e+03	8.5621e+03	9.5853e+03
15	1.9808e+03	1.8419e+03	3.3956e+03	3.4487e+03	4.8532e+03	5.5292e+03	3.1588e+03	4.6274e+03	2.1475e+03	3.5890e+03	2.1874e+03	1.4694e+03	2.9055e+03	2.4832e+03
16	1.1575e+04	1.9129e+03	388.3918	2.2284e+03	466.9418	244.2403	1.1358e+03	1.0439e+04	1.6923e+03	43.0952	167.4254	1.1232e+03	1.4522e+03	882.2982
17	215.3813	1.2084e+03	1.5363e+03	1.1956e+03	453.9712	665.1186	1.6429e+03	1.3669e+03	2.5429e+03	107.7380	3.7225e+03	1.8438e+03	1.7048e+03	815.5523
18	9.8497e+04	4.7554e+04	4.6055e+04	9.6949e+04	1.4121e+05	1.5236e+05	3.6593e+04	1.1024e+05	4.7933e+04	1.6982e+05	6.2608e+04	4.6576e+04	2.1910e+05	5.1220e+04
19	7.2271e+03	3.1122e+03	5.2042e+03	1.0177e+04	7.0430e+03	1.7849e+04	6.6598e+03	1.2743e+04	1.4056e+04	8.5827e+03	4.2741e+03	4.6401e+03	1.2686e+04	3.4082e+03
20	4.4501e+03	7.7395e+03	5.4892e+03	5.9016e+03	9.2812e+03	1.0862e+04	1.0252e+04	5.8432e+03	8.3472e+03	1.0548e+04	7.0609e+03	8.4114e+03	6.6898e+03	7.6393e+03
21	47.4856	1.6852e+03	1.2605e+03	76.4999	925.9571	433.9626	1.3523e+03	98.7793	1.2587e+03	216.8227	1.7424e+03	605.1841	1.9155e+03	723.7766
22	922.5783	72.1419	114.9088	294.8947	45.3971	230.0656	98.5729	991.1702	39.1518	1.8908e+03	99.2956	45.9186	312.4606	141.8352
23	42.3979	55.2336	59.7526	99.9434	38.9118	31.6204	83.1104	74.2955	1.1687e+03	70.0297	78.2768	38.8542	33.5207	92.8186
24	302.7210	217.5529	475.7225	570.0476	366.7799	148.2887	738.9743	408.6255	277.9779	41.7485	750.8775	421.5095	155.6317	488.0799

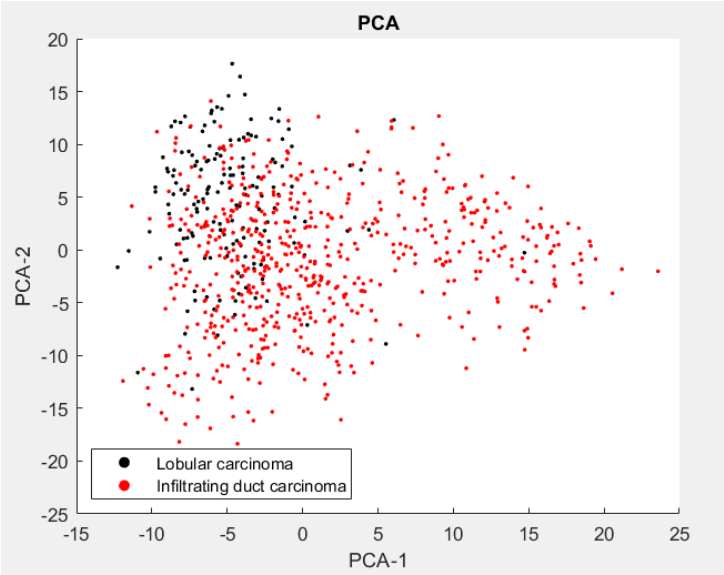
PCA

A questo punto è possibile scegliere se ridurre ulteriormente il campione implementando la PCA oppure se mantenere il campione contenente i geni significativi. Nel caso il cui viene scelto di ridurre il campione è necessario porre la variabile `doPCAgenisig=true` mentre se si sceglie di non ridurre poniamo la variabile `doPCAgenisig= false.`

La PCA (Principal Component Analysis) è una tecnica di analisi multivariata utilizzata per la riduzione delle dimensioni di un dataset. L'obiettivo principale del PCA è trovare un insieme di variabili chiamate Principal Components che rappresentino il massimo della varianza del dataset originale. Ogni Principal Component spiega una percentuale di varianza del dataset e si sceglie il numero minimo di PC che ci permette di preservare la percentuale di varianza fissata a 95%.



Il numero di Principal Component che spiegano il 95% di varianza è pari a 83.



Bilanciamento

A questo punto è possibile scegliere se bilanciare il campione ed effettuare la classificazione oppure se non bilanciare il campione ed effettuare la classificazione. Nel caso il cui viene scelto di bilanciare è necessario porre la variabile `doBilanciamento=true` mentre se si sceglie di non bilanciare poniamo la variabile `doBilanciamento=false`.

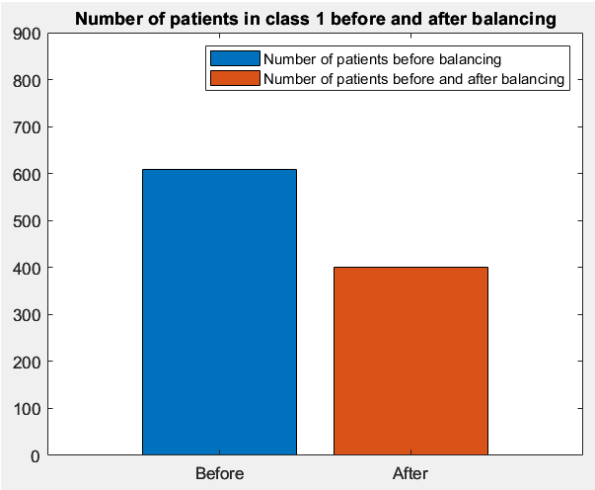
Su 770 pazienti totali abbiamo che:

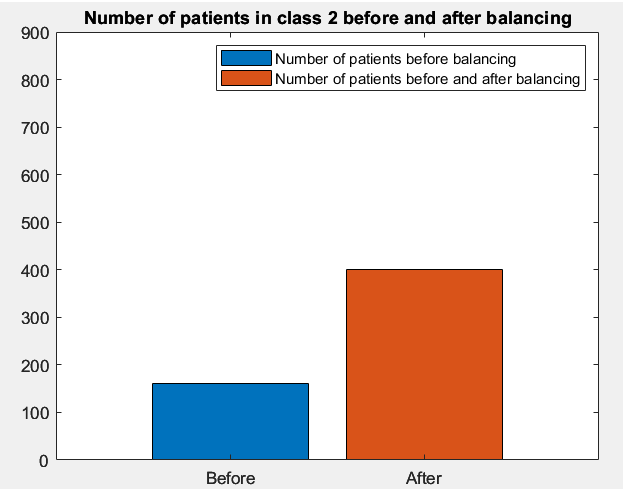
- 609 pazienti appartenenti alla classe Infiltrating duct carcinoma, NOS
- 161 pazienti appartenenti alla classe Lobular carcinoma, NOS

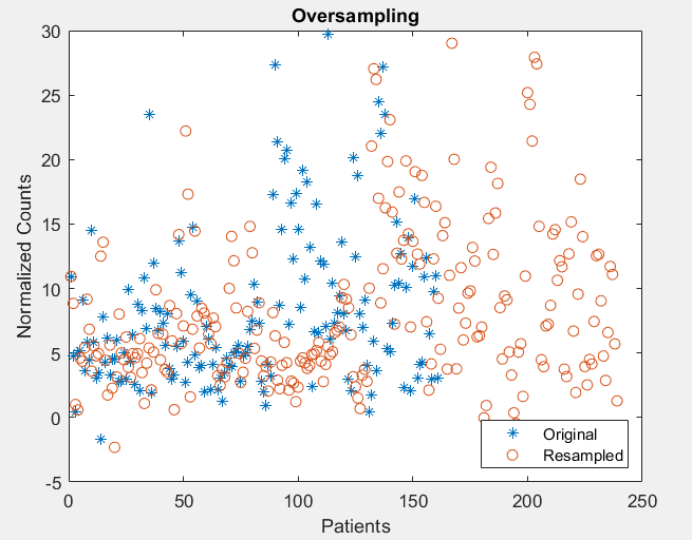
Per bilanciare le due classi è necessario effettuare due operazioni:

- Riduzione dei pazienti della classe Infiltrating duct carcinoma, NOS da 609 a 400
- Creazione di 239 campioni fittizi per la classe Lobular carcinoma

In questo modo entrambe le classi presentano un numero di pazienti pari a 400 per un totale di 800 pazienti.







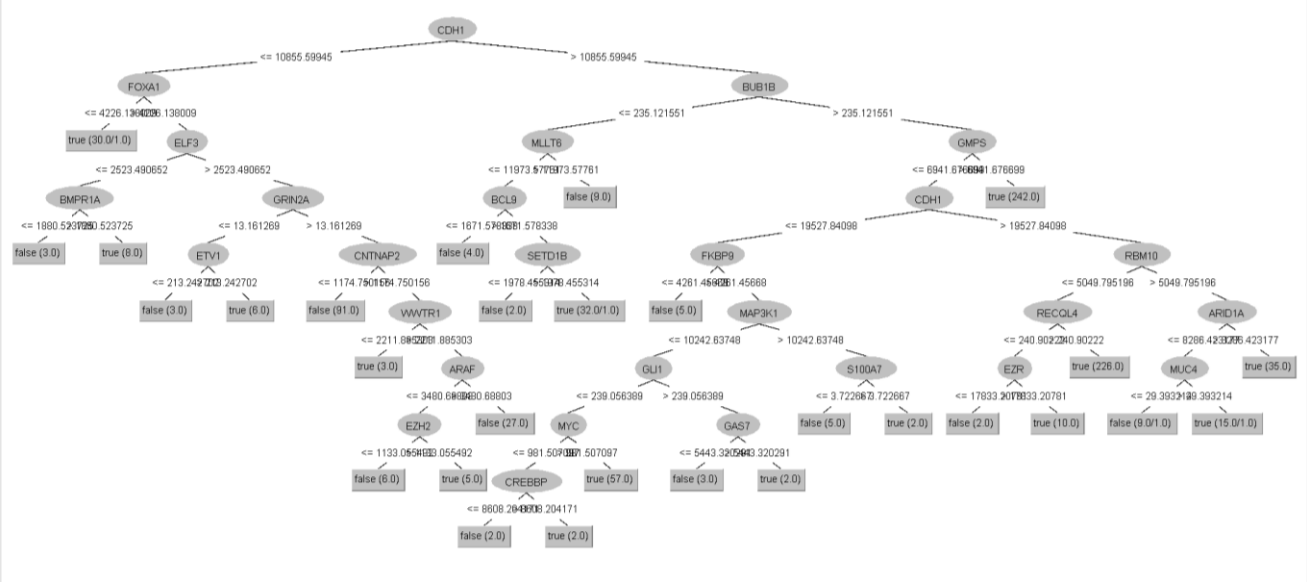
Abbiamo effettuato il bilanciamento anche a livello della clinical_data:

1. Eliminazione indici dei pazienti dalla classe Infiltrating duct carcinoma che sono stati esclusi nel passo 1) del bilanciamento del dataset
2. Aggiunta di campioni fittizi creati per la classe Lobular carcinoma.

WEKA

WEKA (Waikato Environment for Knowledge Analysis) è un insieme di algoritmi di machine learning che consente agli utenti di esplorare, preparare, visualizzare e analizzare dati. WEKA include anche alcuni algoritmi per la costruzione di alberi filogenetici, che sono un importante strumento nella biologia molecolare per la comprensione delle relazioni evolutive tra le specie.

In Weka abbiamo posto in input la matrice dei conteggi normalizzata con il size factor e abbiamo selezionato come output la colonna 'primary diag' che contiene valori di verità true/false. Selezionando come algoritmo "J48 Algorithm for Decision Tree" abbiamo ottenuto il seguente albero decisionale:



Il nodo radice è rappresentato dal gene *CDH1* e la discriminazione avviene sulla base dei valori di conteggio. Si hanno 28 nodi e 29 foglie totali. Andando a confrontare i geni ottenuti nel grafo con quelli significativi ottenuti nel workflow 1 (199 geni significativi) grazie all'nbintest otteniamo 14 geni in comune: *CDH1*, *CNTNAP2, EZH2, GLI1, GAS7, MAP3K1, S100A7, RECQL4, MUC4, CDH1, BUB1B, ELF3, GMPS, GRIN2A.*

Il nodo radice, ovvero il nodo più discriminate, è rappresentato dal gene CDH1. A seguito dell'analisi genica differenziale del workflow1, dopo aver individuato i geni significativi ponendo, per ogni gene, `padj < 0.01` e ordinando in ordine crescente i geni significativi in base al valore del padj, vediamo che anche in questo caso il gene CDH1 è posto in cima alla lista.

I risultati dell'albero decisionale mostrano che le istanze classificate correttamente sono circa l'87% (valore di accuratezza) mentre quelle classificate erroneamente sono circa il 12%. La AUC (area sotto la curva ROC) è pari all'81%.

```
Correctly Classified Instances         741               87.5887 %
Incorrectly Classified Instances       105               12.4113 %
Kappa statistic                          0.6193
Mean absolute error                      0.125
Root mean squared error                  0.3414
Relative absolute error                 38.3727 %
Root relative squared error             84.6327 %
Total Number of Instances              846
```

I nostri positivi sono quelli che appartengono alla classe 'Infiltrating duct carcinoma, NOS' mente i negativi sono quelli che appartengono alla classe 'Lobular carcinoma, NOS'.

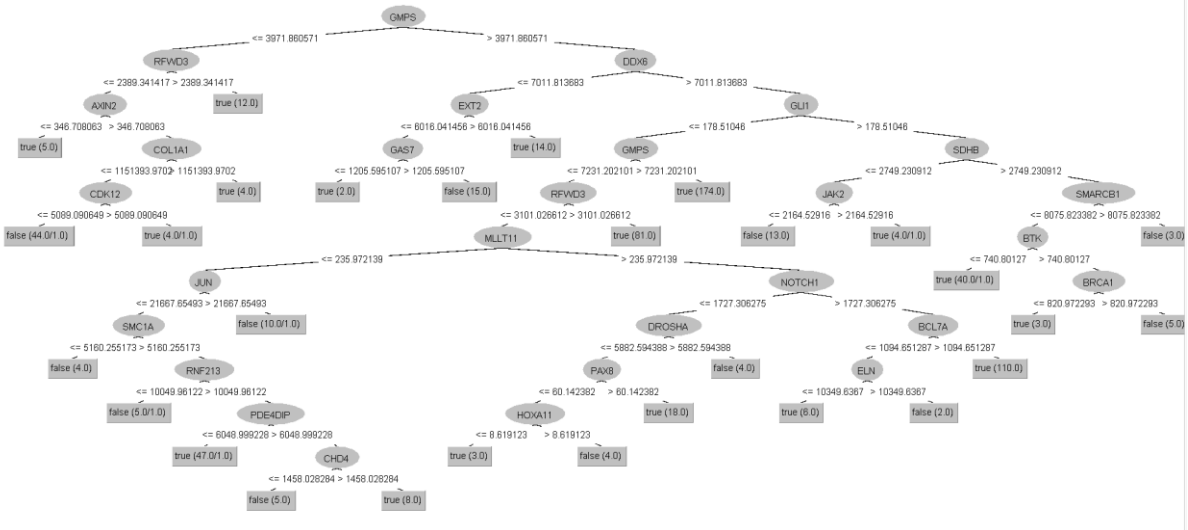
```
=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,921    0,301    0,923      0,921    0,922      0,619    0,813     0,915     true
               0,699    0,079    0,695      0,699    0,697      0,619    0,813     0,581     false
Weighted Avg.  0,876    0,255    0,876      0,876    0,876      0,619    0,813     0,846

=== Confusion Matrix ===

   a   b   <-- classified as
 620  53 |   a = true
  52 121 |   b = false
```

Su 683 pazienti appartenenti alla classe 1 vengono classificati correttamente 620 pazienti mentre su 173 pazienti appartenenti alla classe 2 vengono classificati correttamente 121 pazienti. È visibile come l'algoritmo sbaglia maggiormente sui falsi positivi che sono 52 su 173 totali piuttosto che sui falsi negativi che sono 53 su 683 totali.

Andando a rimuovere dalla matrice il gene CDH1 vediamo che le prestazioni della rete risultano peggiori rispetto al caso precedente:

```
Correctly Classified Instances         520               80.1233 %
Incorrectly Classified Instances       129               19.8767 %
Kappa statistic                          0.2968
Mean absolute error                      0.1999
Root mean squared error                  0.4402
Relative absolute error                 68.3804 %
Root relative squared error            115.2877 %
Total Number of Instances              649

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,888    0,600    0,873      0,888    0,880      0,297    0,673     0,879     true
               0,400    0,112    0,434      0,400    0,416      0,297    0,673     0,316     false
Weighted Avg.  0,801    0,514    0,795      0,801    0,798      0,297    0,673     0,780

=== Confusion Matrix ===

   a   b   <-- classified as
 474  60 |   a = true
  69  46 |   b = false
```

L'albero che si ottiene è il seguente:



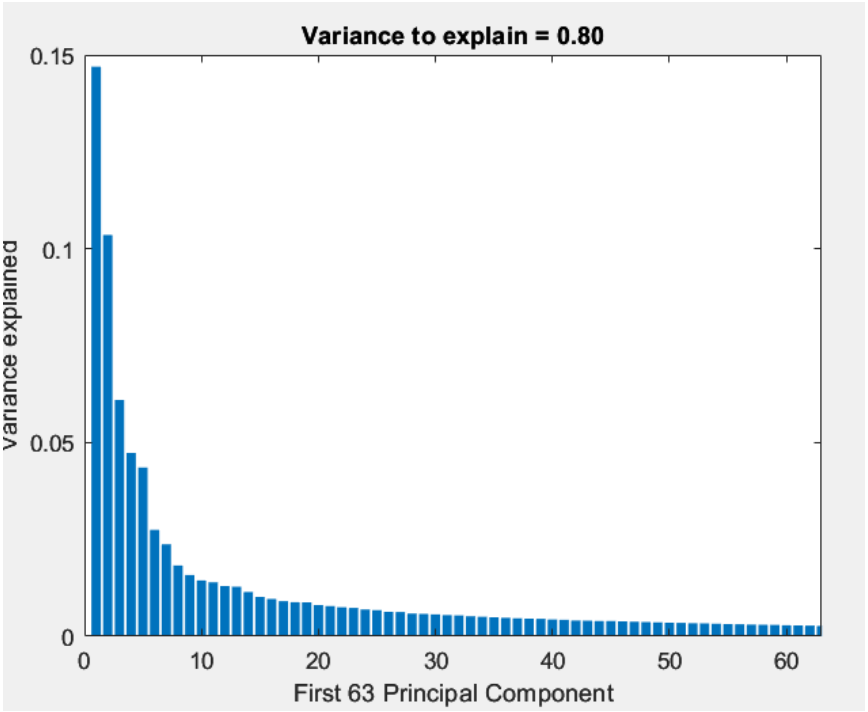
Il nodo radice è rappresentato dal gene GMPS che è presente sempre nella lista di geni significativi trovati nel workflow1.

WORKFLOW 2

Analisi delle componenti principali (PCA)

Partendo dai dati normalizzati prima con il Size Factor e in seguito con il log2 effettuiamo la PCA, ovvero una tecnica di riduzione della dimensionalità che consente di eseguire un mapping dallo spazio iniziale ad uno spazio di dimensione inferiore. L'obiettivo non è quello di mantenere alcune "dimensioni" e cancellarne altre ma "combinare" le dimensioni in modo opportuni. La PCA privilegia le dimensioni che rappresentano al meglio i pattern.

Utilizza una *trasformazione ortogonale* per convertire un set di osservazioni di variabili probabilmente correlate in un set di valori di variabili lineari non correlate chiamate componenti principali. Questa trasformazione è definita in modo tale che la prima componente principale abbia la massima varianza possibile e ogni componente successiva abbia a sua volta la massima varianza possibile sotto il vincolo che è ortogonale alle componenti precedenti. I vettori risultanti sono un insieme di basi ortogonali non correlato.



La PCA è utilizzata per ridurre il numero di features. Ogni Principal Component spiega una percentuale di varianza del dataset e si sceglie il numero minimo di PC che ci permette di preservare la percentuale di varianza fissata a 80%.

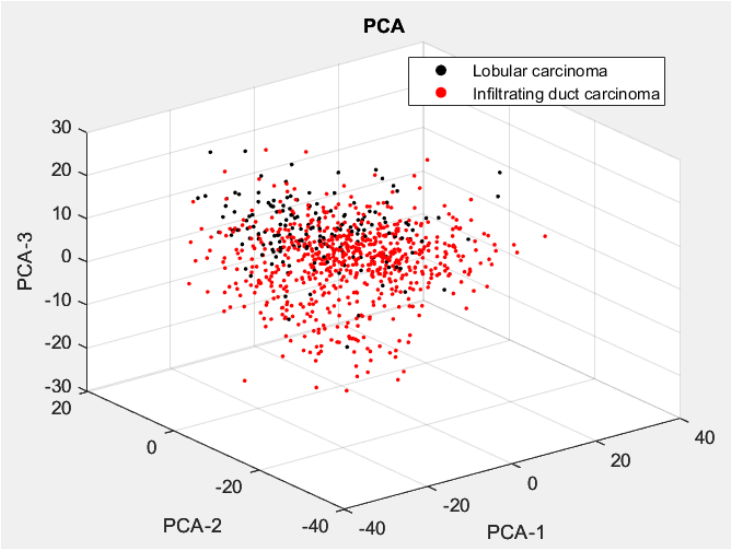
Il numero di Principal Component che spiegano l'80% di varianza è pari a 63.

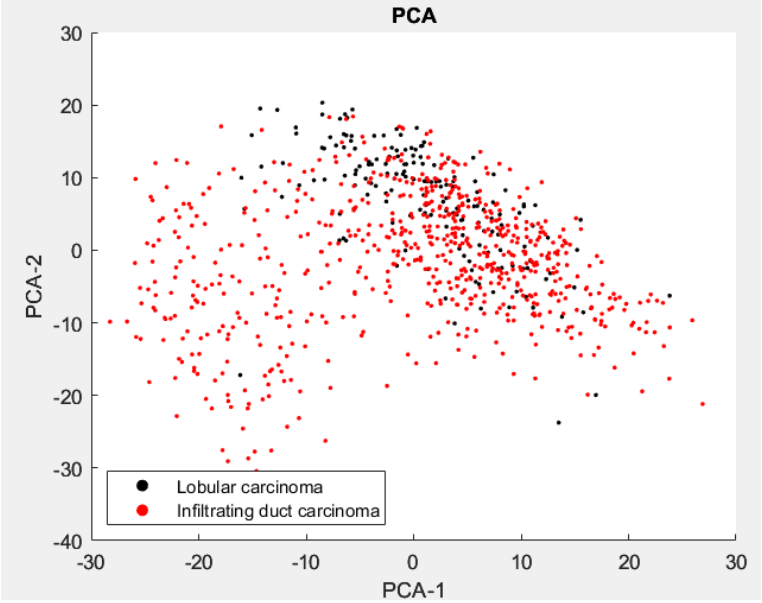
I risultati della PCA sono:

- **Coeff**: restituisce i coefficienti della componente principale.
- **Score**: è la rappresentazione della matrice X nelle coordinate delle componenti principali.
- **Latent**: varianze delle componenti principali, cioè gli autovalori della matrice di covarianza di X
- **Expleined**: percentuale della varianza totale spiegata da ogni componente principale.
- **Mu**: stimare le medie della variabile in X

Viene restituita la matrice x_pca_reduced che rappresenta i dati originali trasformati in base alle componenti principali ottenute dalla PCA. Le colonne della matrice x_pca_reduced rappresentano le componenti principali calcolate dalla PCA, che descrivono la maggior parte della variazione nei dati originali, mentre le righe rappresentano i pazienti e restano invariate.

Queste componenti principali possono essere utilizzate per visualizzare la distribuzione dei campioni in uno spazio bidimensionale o tridimensionale.





Bilanciamento

A questo punto è possibile scegliere se bilanciare il campione ed effettuare la classificazione oppure se non bilanciare il campione ed effettuare la classificazione.

Nel caso il cui viene scelto di bilanciare è necessario porre la variabile `doBilanciamento=true` e seguire il procedimento descritto nel workflow 1 (sezione Bilanciamento) mentre se si sceglie di non bilanciare poniamo la variabile `doBilanciamento=false`.

CLASSIFICATORE BINARIO

RETE NEURALE ARTIFICIALE (ANN)

Una rete neurale artificiale (Neural Network) è un algoritmo di machine learning che viene spesso utilizzato per la classificazione binaria, ovvero una forma di classificazione in cui un elemento viene classificato in una di due categorie: ad esempio, positivo o negativo.

Una rete neurale artificiale per la classificazione binaria è composta da più strati di neuroni artificiali, ognuno dei quali elabora i dati in ingresso e produce un output. Gli strati di neuroni sono fully connected, ovvero tutti i neuroni sono connessi a tutti i neuroni degli strati precedenti e successivi.

I neuroni artificiali elaborano i dati utilizzando una funzione di attivazione non lineare, che è progettata per rendere la rete neurale capace di elaborare non linearità complesse.

La rete neurale viene allenata utilizzando un insieme di dati di addestramento. Durante il processo di addestramento, la rete neurale fa previsioni sulle etichette di classe (positivo o negativo) per i dati di addestramento, e la precisione delle previsioni viene valutata utilizzando una funzione di perdita. La rete neurale viene quindi ottimizzata modificando i pesi e i bias dei neuroni in modo che la funzione di perdita venga minimizzata.

Per creare una ANN in ogni workflow utilizziamo la funzione **patternnet** che restituisce una rete neurale di riconoscimento del modello con una dimensione del livello nascosto di hiddenSizes, una funzione di addestramento, specificata da trainFcn, e una funzione di perdita specificata da performFcn:

- hiddenLayers= [num1 num2] specifica il numero di neuroni per ogni layer nascosto e nel nostro caso i valori degli hiddenLayers verranno calcolati mediante l'algoritmo genetico per i due workflow.
- trainFcn=`'traingdx'` è la funzione di addestramento della rete fissata a "discesa del gradiente del tasso di apprendimento variabile"
- performFcn=`'crossentropy'` è la funzione di perdita della rete

Per la configurazione della rete fissiamo i seguenti parametri:

- net.layers{end}.transferFcn = `'logsig'` è la funzione di attivazione dell'ultimo livello
- net.divideFcn = `'dividerand'`;
- net.divideParam.trainRatio = 0.75;
- net.divideParam.valRatio = 0.25;
- net.divideParam.testRatio = 0.00;

Utilizziamo la funzione **configure** per configurare gli input e gli output della rete neurale.

- net.trainParam.epochs = `500` è il numero massimo di epoche per il train
- net.trainParam.lr = `1e-3` è il learning rate
- net.trainParam.max_fail = `100` è il numero massimo di errori di validazione

ALGORITMO GENETICO

L'algoritmo genetico si basa sull'idea della competizione tra individui di una popolazione in un ambiente dalle risorse limitate che quindi causa la selezione naturale del più "adatto" tra gli individui, secondo una funzione di fitness. L'individuo è di tipo binario, nel nostro progetto abbiamo applicato l'algoritmo genetico per ottimizzare il numero di neuroni degli strati nascosti nella rete ANN considerando come funzione di fitness l'accuratezza media calcolata sul test set per 10 cross validazioni.

Abbiamo considerato due layer nascosti dunque il cromosoma corrisponderà a una sequenza di bit che codifica per due geni che indicano il numero di neuroni nascosti per il primo e il secondo layer.
Ogni gene è rappresentato da 7 bit in modo da far variare il numero di neuroni per layer da 2 a 129.

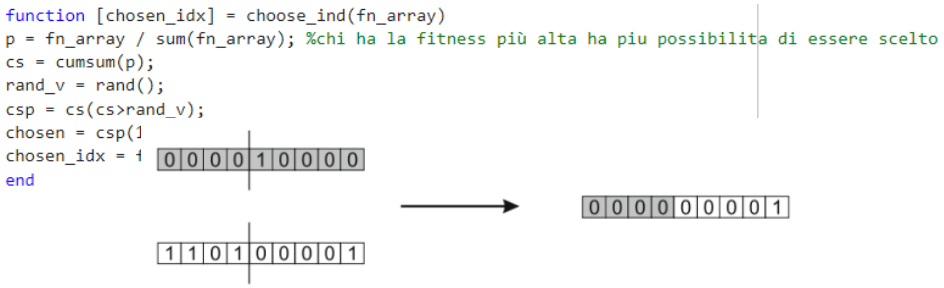
Abbiamo creato la popolazione iniziale di 10 individui in modo casuale e per ogni epoca, per un totale di 40 epoche avvengono i seguenti passaggi bioispirati :

1. **Selezione della specie** → Calcolo della funzione di fitness per ciascun individuo della popolazione e mantenimento nella popolazione successiva dei due individui più fittanti. La funzione di fitness viene calcolata a seguito dell'addestramento di una rete neurale sulla base dei dati di train e test passanti come argomento alla funzione:

   ```
   function [fitness] = compute_fn_ANN(ind, I_min, I_max,x_train,x_test,t_train,t_test)
   ```

   calcolata la matrice di confusione delle 10 cross validazioni, l'accuratezza è calcolata come media dell'accuratezza di ogni iterazione.

2. **Crossover** → è stato implementato il crossover a singolo punto del cromosoma selezionando 2 genitori sulla base della loro probabilità cumulativa derivata dalla funzione di fitness.



A questo punto, dopo aver individuato il punto di crossover (numero random da 2 a 13), con un meccanismo a croce vengono scambiate le porzioni dei cromosomi genitori.

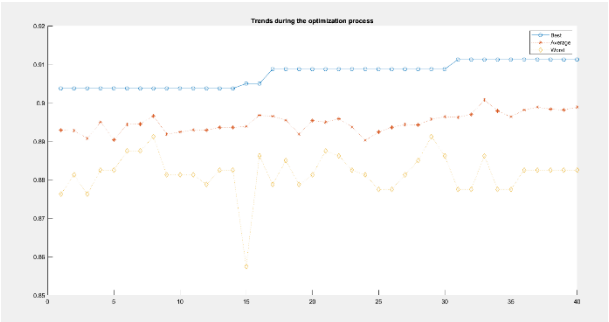
3. **Mutazione** → ottenuto l'individuo figlio su questo viene applicata la una modifica casuale di una variabile binaria nel caso in cui la probabilità di mutazione generata randomicamente superi la soglia impostata, nel nostro caso abbiamo deciso che pm= 0.2.

Alla fine delle 40 epoche si ottiene l'individuo migliore. In figura un esempio di addestramento in cui vengono plottati per ciascun epoca l'individuo migliore, medio e peggiore.

Abbiamo applicato l'algoritmo per k=10 cross-validazioni ottenendo i seguenti hidden layer per:

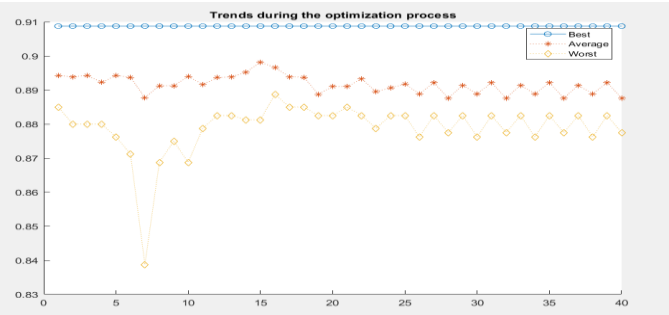
workflow1

```
miglior individuo trovato all'epoca 31
hiddenlayer=     57     76
```



workflow2

```
miglior individuo trovato all'epoca 1
hiddenlayer=     55     98
```



REGRESSIONE LOGISTICA

La regressione logistica è una tecnica statistica utilizzata per modellare la relazione tra una variabile binaria dipendente e una o più variabili indipendenti. La tecnica è utilizzata principalmente per problemi di classificazione binaria, dove l'obiettivo è prevedere se un'osservazione appartiene a una classe o all'altra sulla base di una serie di predittori o variabili indipendenti. La regressione logistica utilizza una funzione logistica per trasformare il risultato in una probabilità compresa tra 0 e 1.

Nell'**analisi di regressione**, si ipotizza l'esistenza di una funzione $f: \mathbb{R}^n \to \mathbb{R}$ che esprime la relazione tra la variabile dipendente y e le n variabili esplicative x_j:

$$y = f(x_1, x_2, \dots, x_n)$$

Per l'analisi di regressione lineare, la classe di ipotesi F sarà composta da funzioni lineari: la relazione f tra la variabile dipendente e la variabile indipendente è lineare.

$$y = \theta_0 + x_1\theta_1 + x_2\theta_2 + \cdots + x_n\theta_n$$

In forma matriciale:

$$Y = X\theta$$

Relazione ipotizzata per la **Regressione Logistica**:

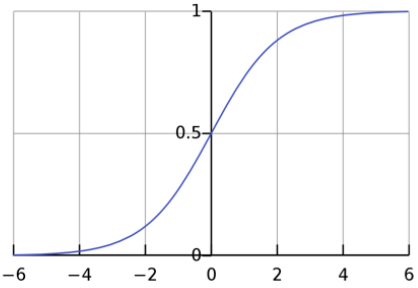
$$h_\theta(\boldsymbol{x}) = g(\theta_0 + x_1\theta_1 + x_2\theta_2 + \cdots + x_n\theta_n) = g(X\theta)$$

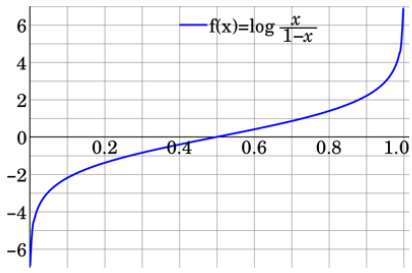
Dove $g(z)$ è una funzione logistica (una sigmoide) con funzione inversa $f(x)$

Nota che $g(z)$: $\mathbb{R}\to$ (0,1), mentre $f(x)$:(0,1) $\to\mathbb{R}$.

$$g(z) = \frac{1}{1 + e^{-z}}$$

$$f(x) = \log\left(\frac{x}{1 - x}\right)$$



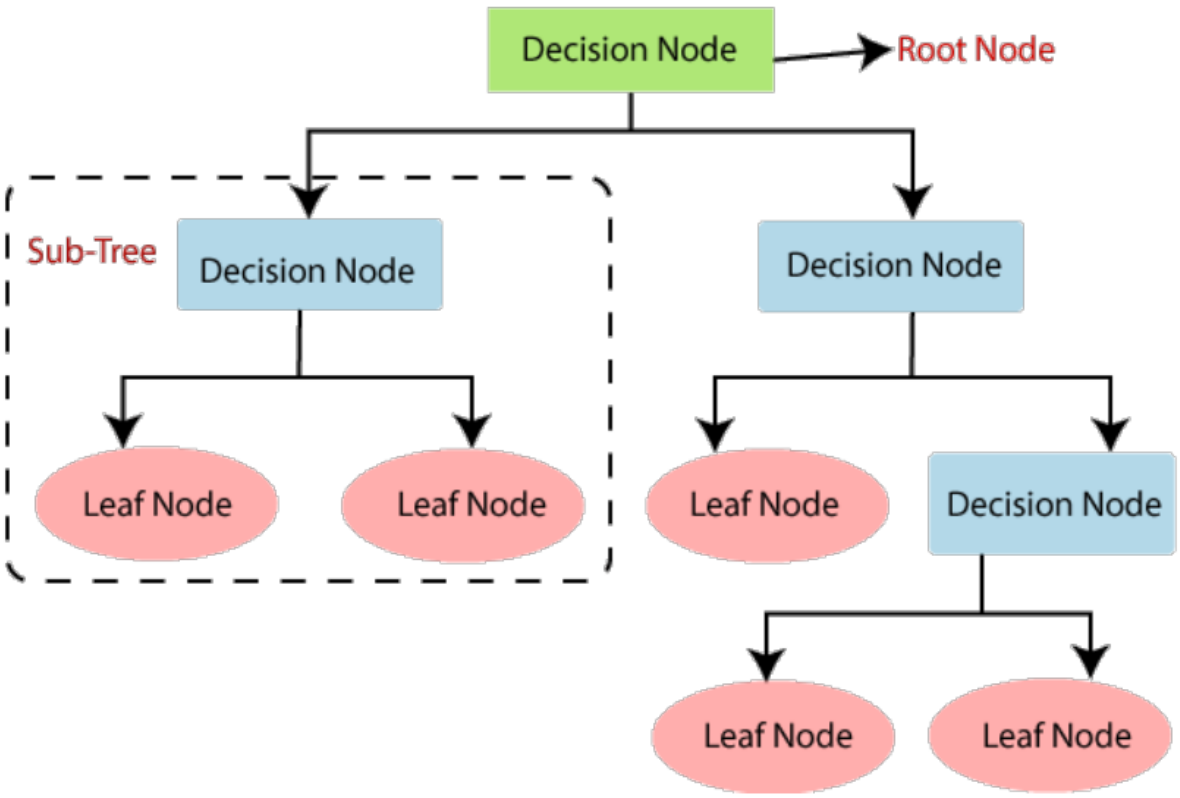


La funzione logistica restituisce un valore compreso tra 0 e 1 che viene utilizzato per assegnare una probabilità all'appartenenza di un'osservazione a una determinata categoria. Questa probabilità viene quindi utilizzata per prevedere la classe di una nuova osservazione.

ALBERO DECISIONALE

Un albero decisionale è uno strumento di supporto alle decisioni che utilizza un modello ad albero delle decisioni e delle loro possibili conseguenze, compresi i risultati degli eventi casuali, i costi delle risorse e l'utilità. È un modo per visualizzare un algoritmo che contiene solo istruzioni di controllo condizionale.

Un albero decisionale è una struttura simile a un diagramma di flusso in cui ogni nodo interno rappresenta un "test" su un attributo (ad esempio, se il lancio di una moneta esce testa o croce), ogni ramo rappresenta il risultato del test e ogni nodo foglia rappresenta un'etichetta di classe (decisione presa dopo aver calcolato tutti gli attributi). I percorsi dalla radice alla foglia rappresentano regole di classificazione (o regressione).



CROSS VALIDAZIONE

La cross-validazione k-folds è un metodo di validazione dei modelli che consiste nel dividere il dataset in k parti o "pieghe" uguali e ogni classificatore binario è stato allenato e valutato su ciascuna di queste pieghe. La media delle prestazioni su tutte le pieghe fornisce una stima più precisa dell'efficacia del classificatore nell'identificare differenze significative nell'espressione genica.

L'algoritmo ha $k - 1$ passaggi in cui, ad ogni passaggio, selezioniamo pieghe diverse per il test set e le pieghe rimanenti che lasciamo per il train set. Alla fine, i risultati delle k valutazioni vengono aggregati per ottenere una stima più precisa della performance del modello.

Fissando k=10 è necessario dividere il set di dati in k parti di uguale numerosità e, ad ogni passo, la k-esima parte dell'insieme di dati viene considerata come il set di dati di test e la restante parte costituisce i dati di train.

La procedura generale è la seguente:

1. Dividere il set di dati in k gruppi:

```
for i = 1:k
    idx = training(cv,i);
    idx_train = find(idx);
    idx_test= find(~idx);
    x_train{i} = Xs(:,idx_train);
    x_test{i} = Xs(:,idx_test);
    t_train{i} = tp(:,idx_train);
    t_test{i} = tp(:,idx_test);
end
```

2. Ad ogni iterazione:

```
for i = 1:k
    % RETE ANN
    net=init(net);
    [trained_net,tr] = train(net,x_train{i},t_train{i});
    y_pred_ANN{i} = trained_net(x_test{i});
    y_pred_bin_ANN{i} = double(y_pred_ANN{i} > 0.5);

    cm_ANN=confusionmat(t_test{i},y_pred_bin_ANN{i});

    accurracy_ANN(i) = (cm_ANN(1,1)+cm_ANN(2,2))/sum(cm_ANN,'all');
    precision_ANN(i)= (cm_ANN(2,2)/(cm_ANN(2,2)+cm_ANN(1,2)));
    recall_ANN(i)= (cm_ANN(2,2)/(cm_ANN(2,2)+cm_ANN(2,1)));
    miss_rate_ANN(i)= (cm_ANN(2,1)/(cm_ANN(2,2)+cm_ANN(2,1)));
    C_ANN(1,1) = C_ANN(1,1) + cm_ANN(1,1);
    C_ANN(2,2) = C_ANN(2,2) + cm_ANN(2,2);
    C_ANN(2,1) = C_ANN(2,1) + cm_ANN(2,1);
    C_ANN(1,2) = C_ANN(1,2) + cm_ANN(1,2);
    [fpr_ANN{i}, tpr_ANN{i}, th_ANN{i} ,auc_ANN(i)] = perfcurve (t_test{i},y_pred_ANN{i},1);
end
```

```
for i = 1:k
     % REGRESSIONE LOGISTICA
     t_train{i}= logical(t_train{i});
     t_test{i}= logical(t_test{i});
     LR= fitclinear(x_train{i}',t_train{i}','Learner','logistic');
     [y_pred_LR{i}, score_LR{i}] = LR.predict(x_test{i}');

     cm_LR=confusionmat(t_test{i},y_pred_LR{i});

     accurracy_LR(i) = (cm_LR(1,1)+cm_LR(2,2))/sum(cm_LR,'all');
     precision_LR(i)= (cm_LR(2,2)/(cm_LR(2,2)+cm_LR(1,2)));
     recall_LR(i)= (cm_LR(2,2)/(cm_LR(2,2)+cm_LR(2,1)));
     miss_rate_LR(i)= (cm_LR(2,1)/(cm_LR(2,2)+cm_LR(2,1)));
     C_LR(1,1) = C_LR(1,1) + cm_LR(1,1);
     C_LR(2,2) = C_LR(2,2) + cm_LR(2,2);
     C_LR(2,1) = C_LR(2,1) + cm_LR(2,1);
     C_LR(1,2) = C_LR(1,2) + cm_LR(1,2);
     y_pred_bin_LR= score_LR{i};
     [fpr_LR{i}, tpr_LR{i}, th_LR{i} ,auc_LR(i)] = perfcurve (t_test{i},y_pred_bin_LR(:,2)',1);
end
```

```
for i = 1:k
    % DECISION TREE
    DT = fitctree(x_train{i}',t_train{i}');
    [y_pred_DT{i}, score_DT{i}] = predict(DT,x_test{i}');

    cm_DT=confusionmat(t_test{i},y_pred_DT{i});

    accurracy_DT(i) = (cm_DT(1,1)+cm_DT(2,2))/sum(cm_DT,'all');
    precision_DT(i)= (cm_DT(2,2)/(cm_DT(2,2)+cm_DT(1,2)));
    recall_DT(i)= (cm_DT(2,2)/(cm_DT(2,2)+cm_DT(2,1)));
    miss_rate_DT(i)= (cm_DT(2,1)/(cm_DT(2,2)+cm_DT(2,1)));
    C_DT(1,1) = C_DT(1,1) + cm_DT(1,1);
    C_DT(2,2) = C_DT(2,2) + cm_DT(2,2);
    C_DT(2,1) = C_DT(2,1) + cm_DT(2,1);
    C_DT(1,2) = C_DT(1,2) + cm_DT(1,2);
    y_pred_bin_DT= score_DT{i};
    [fpr_DT{i}, tpr_DT{i}, th_DT{i} ,auc_DT(i)] = perfcurve (t_test{i},y_pred_bin_DT(:,2)',1);

end
```

METRICHE PER VALUTARE LE PRESTAZIONI DEL CLASSIFICATORE

MATRICE DI CONFUSIONE

La matrice di confusione mostra il numero di campioni che sono stati classificati correttamente e in modo errato dal modello. È composta da quattro elementi: veri positivi (VP), falsi positivi (FP), veri negativi (VN) e falsi negativi (FN).

Può essere utilizzata per calcolare diverse metriche di performance del modello, come l'accuratezza, la precisione, il recall e il FNR. Queste metriche possono aiutare a valutare la capacità del modello di prevedere correttamente le classi e a identificare eventuali debolezze del modello.

$$Accuratezza = \frac{TP + TN}{P + N} \qquad Precisione = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN} \qquad FNR = \frac{FN}{TP + FN}$$

CURVA ROC

La curva ROC è un grafico comunemente utilizzato nell'analisi delle prestazioni di un classificatore binario, che rappresenta la relazione tra la "vera positività" (TPR) e la "falsa positività" (FPR) alla variazione del punto di taglio (threshold) del classificatore.

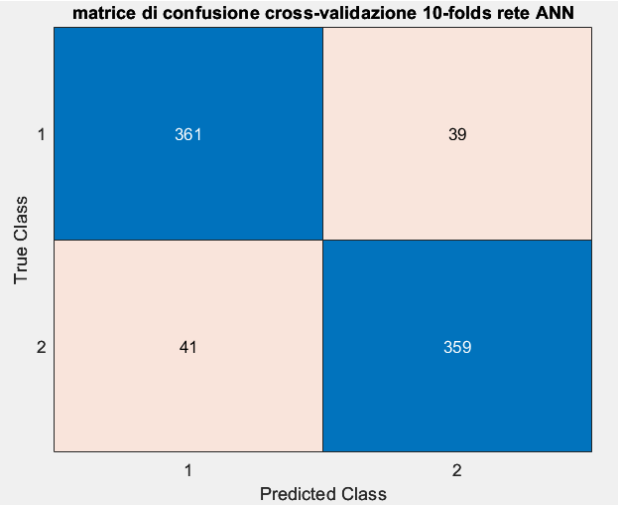
L'area sotto la curva ROC (AUC) è una misura numerica della capacità del classificatore di distinguere tra due classi. Più precisamente, l'AUC rappresenta la probabilità che un classificatore assegnerà una maggiore probabilità a un'osservazione casuale positiva rispetto a un'osservazione casuale negativa. In altre parole, quanto più alta è l'AUC, tanto più è probabile che il classificatore identifichi correttamente le osservazioni positive. L'AUC varia tra 0 e 1, dove un valore di 0.5 indica una performance casuale del classificatore, mentre un valore di 1 indica una performance perfetta.

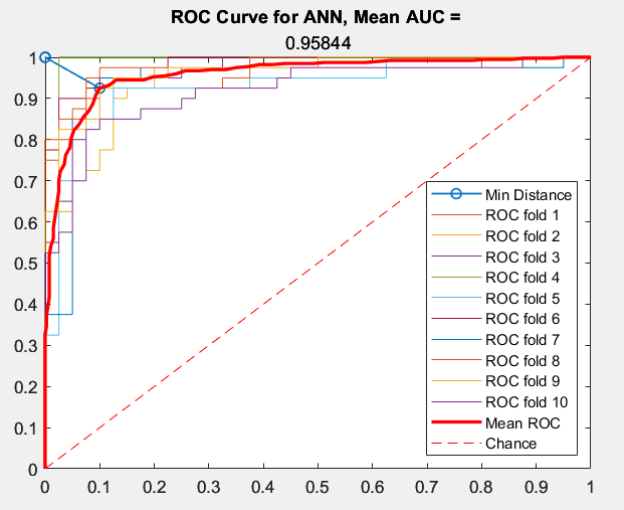
Il punto che ha la minima distanza dal punto (0,1) sulla curva ROC (Receiver Operating Characteristic) rappresenta la configurazione di soglia ottimale per un classificatore. Il punto (0,1) rappresenta il massimo valore di TPR con un FPR pari a zero, il che significa che non ci sono falsi positivi nella classificazione. Il punto più vicino a (0,1) sulla curva ROC rappresenta la configurazione di soglia che ottiene la massima TPR possibile per un determinato livello di FPR. Questo punto indica la configurazione ottimale per il classificatore per equilibrare la precisione e la sensibilità.

RISULTATI WORKFLOW 1

RISULTATI RETE NEURALE ARTIFICIALE

La matrice di confusione e le curve ROC che otteniamo per il **campione bilanciato** sono:



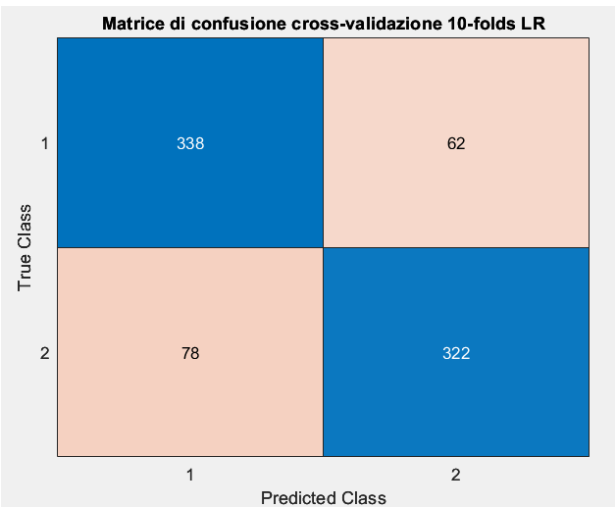


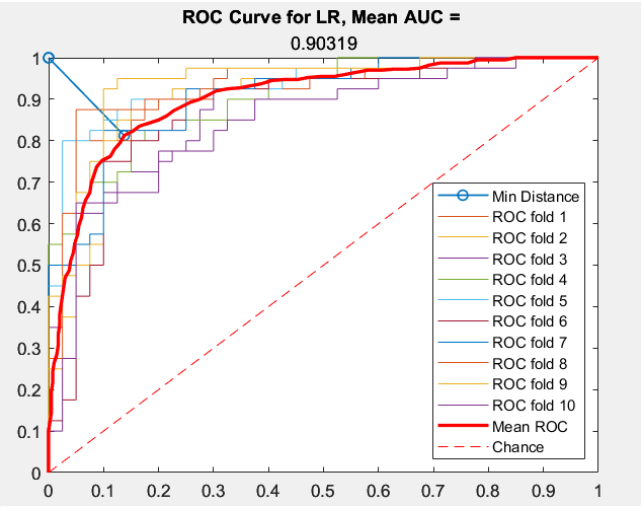
I risultati sono:

```
Risultati rete neurale artificiale con cross-validazione 10-folds:
Accuratezza media: 90.00
Precisione media: 90.37
Miss Rate media: 10.25
Recall media: 89.75
Area media sottesa alla curva roc: 95.84

Il best cut-off per l'ANN è: 0.45
Si ottiene in corrispondenza della cordinata x: 0.10 e della coordinata y:0.93
```

RISULTATI REGRESSIONE LOGISTICA

La matrice di confusione e le curve ROC che otteniamo per il **campione bilanciato** sono:



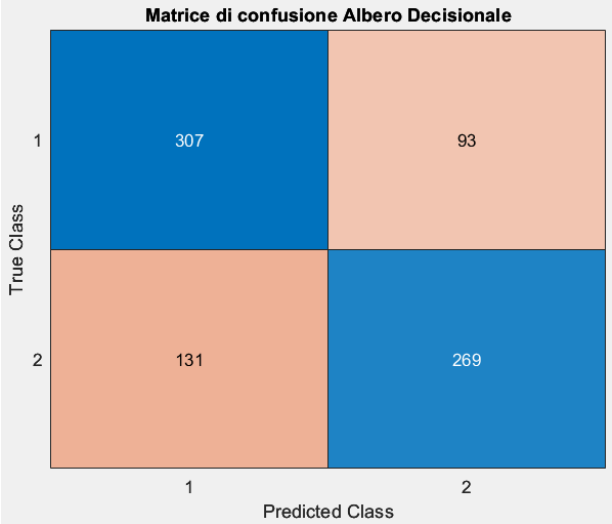


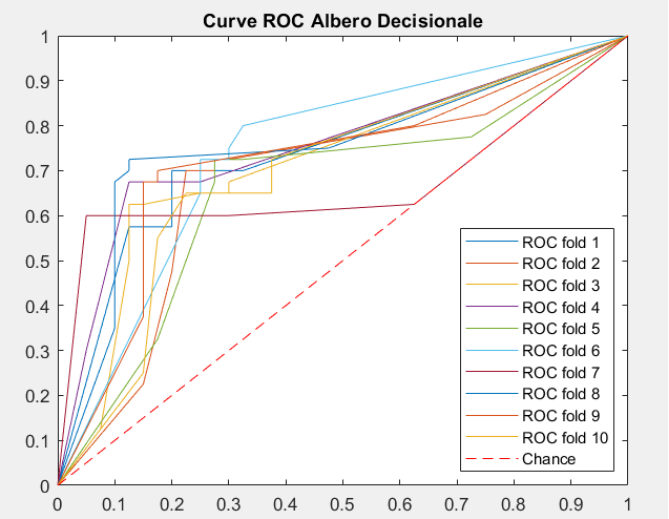
I risultati sono:

```
Risultati regressione logistica con cross-validazione 10-folds:
Accuratezza media: 82.50
Precisione media: 83.99
Miss Rate media: 19.50
Recall media: 80.50
Area media sottesa alla curva roc: 90.32

Il best cut-off per la LR è: 0.52
Si ottiene in corrispondenza della cordinata x: 0.14 e della coordinata y:0.81
```

RISULTATI ALBERO DECISIONALE

La matrice di confusione e le curve ROC che otteniamo per il **campione bilanciato** sono:





I risultati sono:

```
Risultati Albero Decisionale con cross-validazione 10-folds:
Accuratezza media Albero Decisionale: 72.00
Precisione media Albero Decisionale: 74.67
Miss Rate media Albero Decisionale: 32.75
Recall media Albero Decisionale: 67.25
Area media sottesa alla curva roc Albero Decisionale: 71.34
```

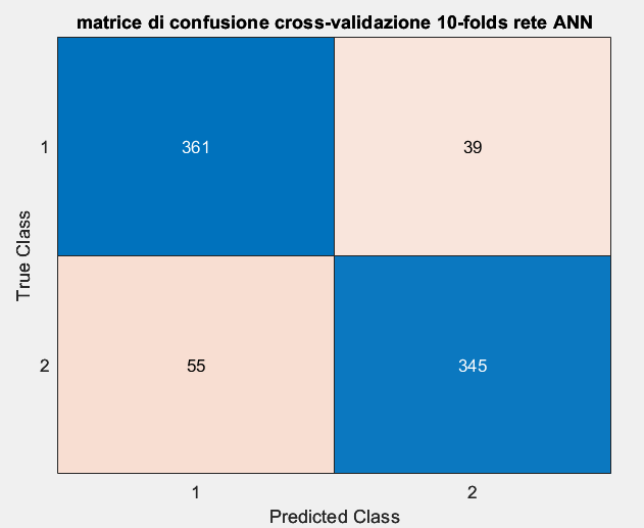
Confrontando i risultati del workflow 1 otteniamo:

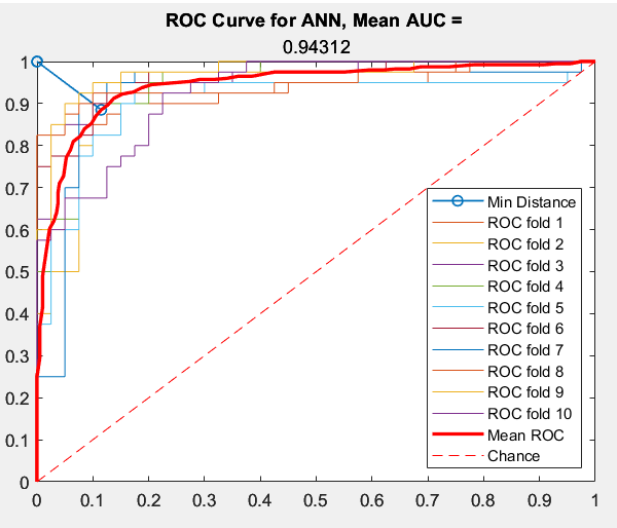
Workflow 1	**RETE ANN**	**REGRESSIONE LOGISTICA**	**ALBERO DECISIONALE**
ACCURATEZZA MEDIA	**90.00**	82.50	72.00
PRECISIONE MEDIA	**90.37**	83.99	74.67
MISS RATE MEDIA	**10.25**	19.50	32.75
RECALL MEDIA	**89.75**	80.50	67.25
AUCROC	**95.84**	90.32	71.34

RISULTATI WORKFLOW 2

RISULTATI RETE NEURALE ARTIFICIALE

La matrice di confusione e le curve ROC che otteniamo per il **campione bilanciato** sono:



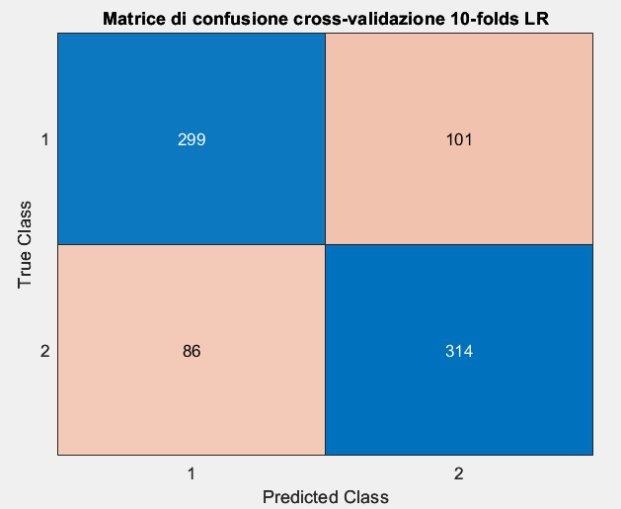


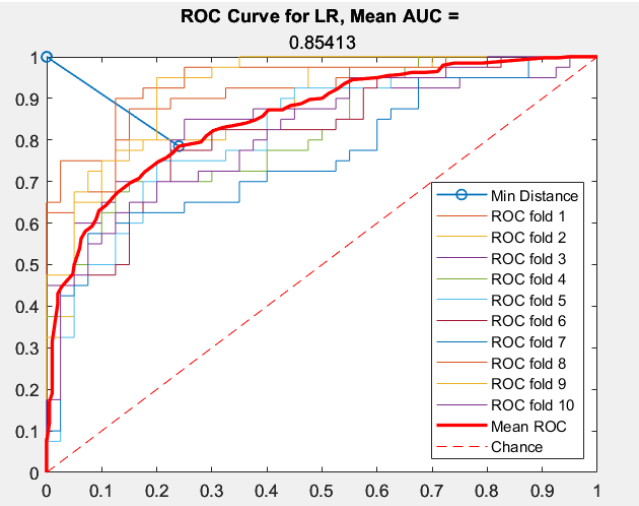
I risultati sono:

```
Risultati rete neurale artificiale con cross-validazione 10-folds:
Accuratezza media: 88.25
Precisione media: 90.02
Miss Rate media: 13.75
Recall media: 86.25
Area media sottesa alla curva roc: 94.31

Il best cut-off per l'ANN è: 0.44
Si ottiene in corrispondenza della cordinata x: 0.11 e della coordinata y:0.89
```

RISULTATI REGRESSIONE LOGISTICA

La matrice di confusione e le curve ROC che otteniamo per il **campione bilanciato** sono:

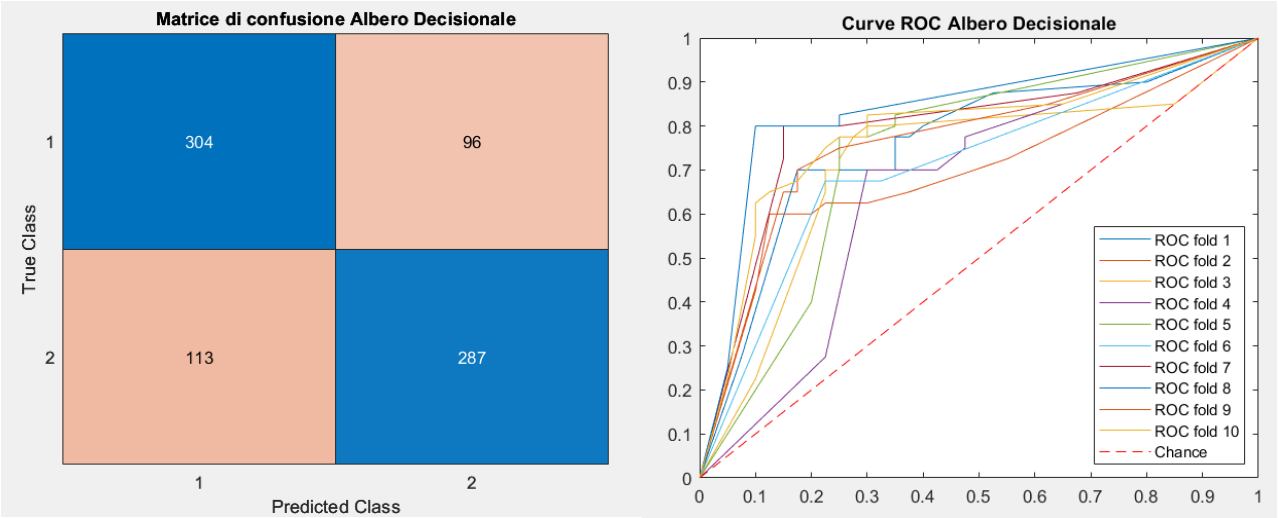




I risultati sono:

```
Risultati regressione logistica con cross-validazione 10-folds:
Accuratezza media: 76.62
Precisione media: 76.04
Miss Rate media: 21.50
Recall media: 78.50
Area media sottesa alla curva roc: 85.41

Il best cut-off per la LR è: 0.51
Si ottiene in corrispondenza della cordinata x: 0.24 e della coordinata y:0.79
```

RISULTATI ALBERO DECISIONALE



La matrice di confusione e le curve ROC che otteniamo per il **campione bilanciato** sono:

I risultati sono:

```
Risultati Albero Decisionale con cross-validazione 10-folds:
Accuratezza media Albero Decisionale: 73.88
Precisione media Albero Decisionale: 75.16
Miss Rate media Albero Decisionale: 28.25
Recall media Albero Decisionale: 71.75
Area media sottesa alla curva roc Albero Decisionale: 74.58
```

Confrontando i risultati del workflow 2 otteniamo:

Workflow 2	RETE ANN	REGRESSIONE LOGISTICA	ALBERO DECISIONALE
ACCURATEZZA MEDIA	**88.25**	76.62	73.88
PRECISIONE MEDIA	**90.02**	76.04	75.16
MISS RATE MEDIA	**13.75**	21.50	28.25
RECALL MEDIA	**86.25**	78.50	71.75
AUCROC	**94.31**	85.41	74.58

CONCLUSIONI

Nel workflow 1, in seguito alla ricerca dei geni statisticamente significativi, alla riduzione della dimensionalità con PCA e al bilanciamento del campione, la modellazione con ANN realizzata con cross validazione k-folds (con k=10) mostra prestazioni migliori rispetto alla modellazione con regressione logistica e con albero decisionale in termini di Accuratezza media, Precisione media, Miss Rate media, Recall media e AUCROC.

Nel workflow 2, in seguito alla riduzione della dimensionalità con PCA e al bilanciamento del campione, la modellazione con ANN realizzata con cross validazione k-folds (con k=10) mostra prestazioni migliori rispetto alla modellazione con regressione logistica e con albero decisionale in termini di Accuratezza media, Precisione media, Miss Rate media, Recall media e AUCROC.

Tra i due workflow, le prestazioni migliori sono ottenute della rete ANN del workflow 1.

L'ANN è un modello di apprendimento automatico più complesso e flessibile rispetto alla regressione logistica e all'albero decisionale. La capacità di apprendere modelli non lineari complessi rende l'ANN una scelta ottima per l'analisi di dati biologici, dove le relazioni tra i geni possono essere molto complesse. In secondo luogo, l'ANN è particolarmente adatto per la classificazione di grandi quantità di dati. In un'analisi genica differenziale, si possono avere a disposizione migliaia di geni e molti campioni, e l'ANN può gestire efficacemente questa grande quantità di dati.

Inoltre, la PCA riduce la dimensionalità del dataset, il che può migliorare le prestazioni del modello riducendo la complessità del problema. Infine, il bilanciamento del campione può avere un effetto significativo sulle prestazioni del modello, poiché i modelli di apprendimento automatico spesso funzionano meglio quando le classi sono bilanciate.

L'utilizzo di un algoritmo genetico per la scelta degli hidden layers si è rivelata un'opzione valida per questo tipo di problema complesso dell'analisi genica differenziale e per il dataset di grandi dimensioni, in cui non è chiaro quale sia la migliore architettura della rete neurale da utilizzare.

Workflow 1	RETE ANN	REGRESSIONE LOGISTICA	ALBERO DECISIONALE
ACCURATEZZA MEDIA	90.00	82.50	72.00
PRECISIONE MEDIA	90.37	83.99	74.67
MISS RATE MEDIA	10.25	19.50	32.75
RECALL MEDIA	89.75	80.50	67.25
AUCROC	95.84	90.32	71.34

Workflow 2	RETE ANN	REGRESSIONE LOGISTICA	ALBERO DECISIONALE
ACCURATEZZA MEDIA	88.25	76.62	73.88
PRECISIONE MEDIA	90.02	76.04	75.16
MISS RATE MEDIA	13.75	21.50	28.25
RECALL MEDIA	86.25	78.50	71.75
AUCROC	94.31	85.41	74.58