UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Triennale in

Informatica

“Identificazione di biomarcatori molecolari per migliorare la diagnosi e monitorare la progressione del Parkinson.”

Relatore:

Prof. Michele Risi

Dott.ssa Maria Frasca

Candidato:

Gerardo Iuliano

Matricola: 05121/05757

Anno Accademico 2020/2021

Sommario

Introduzione

Il Parkinson è una malattia neurodegenerativa, ad evoluzione lenta ma progressiva, che coinvolge, principalmente, alcune funzioni quali il controllo dei movimenti e dell'equilibrio. La malattia fa parte di un gruppo di patologie definite "Disordini del Movimento" e tra queste è la più frequente. Le strutture coinvolte nella malattia si trovano in aree profonde del cervello, note come gangli della base (nuclei caudato, putamen e pallido), che partecipano alla corretta esecuzione dei movimenti (ma non solo).

La malattia di Parkinson si manifesta quando la produzione di dopamina nel cervello cala consistentemente in seguito alla perdita selettiva dei neuroni dopaminergici dalla zona substantia nigra (la perdita cellulare è di oltre il 60% all'esordio dei sintomi). Inoltre, la malattia è caratterizzata dalla presenza dei corpi di Lewy (ammassi proteici) all'interno dei neuroni celebrali. Dal midollo al cervello cominciano a comparire anche accumuli di una proteina chiamata alfa-sinucleina.

I principali sintomi motori della malattia di Parkinson sono il tremore a riposo, la rigidità, la bradicinesia (lentezza dei movimenti automatici) e, in una fase più avanzata, l'instabilità posturale (perdita di equilibrio); questi sintomi si presentano in modo asimmetrico (un lato del corpo è più interessato dell'altro). Il tremore non è presente in tutti i pazienti. All'esordio della malattia, spesso i sintomi non vengono riconosciuti immediatamente, perché si manifestano in modo subdolo, incostante e la progressione della malattia è tipicamente lenta.

Fattori tossici, esposizione lavorativa: il rischio di malattia aumenta con l'esposizione a tossine quali alcuni pesticidi (per esempio il Paraquat) o idrocarburi-solventi (per esempio la trielina) e in alcune professioni (come quella di saldatore) che espongono i lavoratori a metalli pesanti (ferro, zinco, rame). L'esposizione al fumo di sigaretta riduce probabilmente la comparsa di malattia di Parkinson. Il fumo sembra essere cioè un fattore protettivo.

Il Parkinson è difficile da diagnosticare nelle sue fasi iniziali e quando viene diagnosticato l'unico trattamento consiste nell'aumentare i livelli di dopamina inadeguati, ciò non comporta l'eliminazione di tutti i sintomi della malattia. Pertanto, è necessario trovare dei biomarcatori molecolari per migliorare l'accuratezza della diagnosi, monitorare la progressione e sviluppare interventi terapeutici. Sono stati identificati diversi geni causativi del morbo di Parkinson, tra cui a-sinucleina (SNCA), parkin (PARK2), UCHL-1(PARK5), PINK1 (PARK6), DJ-1 (PARK7), LRRK2 (PARK8) e ATP13A2 (PARK9). Queste molecole sono dei biomarcatori candidati per l'analisi di questa malattia. Tra questi, i livelli di DJ-1 e a-sinucleina nel liquido cerebrospinale umano e nel sangue sono i biomarcatori più frequentemente testati. Tuttavia, da soli questi due biomarcatori non sono soddisfacenti. Inoltre, i cambiamenti dei livelli di 8-idrossidesossiguanosina urinaria (Urinary 8-

OHdGe) e citochine pro-infiammatorie come fattore di necrosi tumorale α (TNF-alfa), interleuchina 6 (IL-6) e anche l'interleuchina 1b (IL-1b) sono stati studiati come biomarcatori per il Parkinson. Anche il livello dell'ormone proteico IGF2 (Insuline like growth factor 2) è significativamente più alto nei pazienti affetti da Parkinson rispetto quelli di controllo.

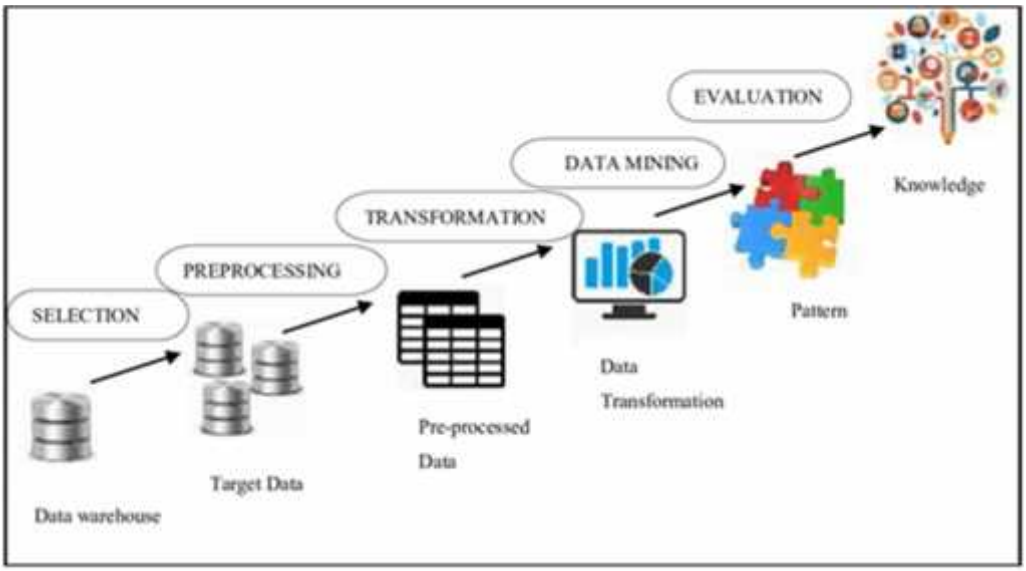
Nella presente analisi ci si propone l'identificazione dei biomarcatori molecolari utilizzando analisi bioinformatica computazionale dell'espressione genica. Pertanto, è necessario seguire un modello di analisi che permetta di estrapolare informazioni dai dati senza dipendere troppo dal campione di dati in analisi, che riesca quindi a gestire eventuali dati disuniformi o alterati senza che quest'ultimi influenzino l'analisi complessiva.

Per tale sviluppo si è intenzionati ad utilizzare l'ambiente R, in modo tale da poter analizzare facilmente i dati utilizzando le diverse ed innumerevoli funzioni a disposizione. Inoltre, in tale ambiente, è possibile eseguire algoritmi di Machine Learning, utili al fine dell'analisi in quanto questi utilizzano metodologie matematico computazionali per apprendere informazioni direttamente dai dati, senza modelli matematici ed equazioni predeterminate. Così come il Clustering, tecnica che consente di individuare dei gruppi significativi di dati in un dataset, consentendo di concentrare l'analisi sui campioni più consoni allo studio.

La struttura dell'elaborato si basa sul processo di "Knowledge Discovery in Database", ovvero un processo che unifica operazioni automatiche e scelte, decisioni e deduzioni umane, per estrarre conoscenza significativa e utilizzabile da masse informi di dati bruti ed eterogenei. Il processo di Knowledge Discovery si suddivide in cinque fasi (Fig. 1), le quali hanno una serie di dati in input e restituiscono in output i dati processati:

- selezione dei dati;
- pre-processamento dei dati;
- trasformazione dei dati;
- data-mining;
- valutazione;

***Fig. 1**: Struttura del processo Knowledge Discovery*



Il data set di partenza, ricavato da un data set dalla piattaforma GEO e dal database di TCGA, un programma di genomica del cancro, contiene dati reali genomici relativi ai pazienti sani e affetti dalla malattia. I segmenti di DNA analizzato (detti probe) sono stati fissati su un supporto, ovvero un DNA microarray, il quale è un utile strumento per identificare mutazioni presenti nei geni.

Il capitolo 1, quindi, corrisponderà alla fase di selezione ed illustrerà la struttura e il funzionamento dei DNA microarray per meglio comprendere come i dati siano stati estrapolati dal database originale ed organizzati in una matrice di dati. Per quest'analisi, così come spiegato precedentemente, si è pensato di applicare algoritmi di Machine Learning. A tal fine è doveroso preparare le informazioni in input eliminando il "rumore" o altri disturbi dei dati, applicando strategie per gestire i dati mancanti ed organizzando i dati per le successive analisi esplorative, unificandoli e consolidandoli in formati adatti all'analisi da effettuare, riducendone la varietà. Le operazioni appena elencate rappresentano le fasi 2-3 del KDD e saranno descritte nel capitolo 2. La fase di data mining verrà esposta nel capitolo 3, nel quale verranno illustrati nel dettaglio come sono stati costituiti e come lavorano gli algoritmi utilizzati per analizzare i dati e per scoprire il modello di previsione, ovvero il modello utile ad individuare i geni differenzialmente espressi.

Il capitolo 4, invece, definirà il processo di analisi dei pathway, documentando e interpretando i risultati ottenuti al termine dell'analisi.

Infine, nella conclusione, si discuterà dei risultati ottenuti dall'analisi.

Capitolo 1 - Selezione ed organizzazione dei dati

I Microarray sono nati a metà degli anni '90 e sono lo strumento più utilizzato per analizzare l'espressione genica in un campione biologico, comunemente definito profilo di espressione. Sono vetrini microscopici che contengono una serie ordinata di campioni, i quali consentono di differenziarne più tipi:

- campioni di DNA-> DNA microarray;
- campioni di RNA-> RNA microarray;
- campioni di proteine -> microarray proteici;
- campioni di tessuto-> microarray tissutali

In questo capitolo ci focalizzeremo sulle caratteristiche e sull'utilizzo dei DNA microarray. Il capitolo si apre con la descrizione dei tipi di DNA microarray, dedicando particolare importanza al loro rispettivo funzionamento e alla loro differenza. Sono, inoltre, spiegate alcune nozioni di base di biologia molecolare correlate al concetto di "espressione genica". Il capitolo si chiude con la presentazione della matrice dei dati in input utilizzati per il nostro caso di studio.

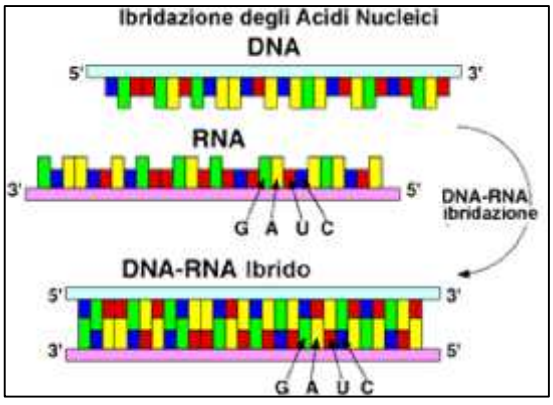
1.1 DNA Microarray

Un DNA Microarray (o gene chip) è costituito da un supporto solido come vetro, plastica, o silicio a cui sono ancorate delle sonde di DNA (o probe), in numero molto elevato per ogni gene e disposte in posizioni note. È lo strumento più utilizzato per analizzare l'espressione genica, ovvero per analizzare il processo di conversione delle informazioni contenute in un gene in prodotto genico funzionale (proteine o molecole di RNA). L'utilità di ciò sta nella possibilità di effettuare un confronto tra il livello di espressione genica in tipi cellulari diversi o in condizioni patologiche diverse, per determinare il ruolo che i geni hanno in queste. Si definiscono geni differenzialmente espressi (DEG) quei geni che in due condizioni biologiche differenti (es. tessuti sani e tessuti malati) hanno un livello di espressione significativamente diverso. I DNA microarray possono essere utilizzati solo dopo aver effettuato l'analisi del trascrittoma del genoma dell'organismo in esame. Per trascrittoma si intende l'insieme delle molecole di mRNA1 (o trascritti) presenti in una cellula. L'analisi del trascrittoma corrisponde all'analisi differenziale dell'espressione genica, ottenuta confrontando i profili trascrizionali di due o più individui, tessuti o tipi cellulari. Ad esempio, informazioni sui trascritti in soggetti sani e soggetti malati possono permettere di rilevare quali geni sono espressi in maniera significativamente diversa fra i due gruppi e quindi di evidenziare differenze che la condizione patologica comporta. Tale analisi può avvenire secondo due tecnologie:

1. ibridazione, si basa sulla proprietà dei nucleotidi di appaiarsi con i loro complementari fissati su un supporto;
2. sequenziamento, si intende l'identificazione della sequenza di DNA fornita in input alla strumentazione;

I DNA microarray sfruttano la tecnica di ibridazione inversa, consistente cioè nel fissare tutti i probe su un supporto e nel marcare invece l'acido nucleico target. Tale tecnica permette l'analisi dell'espressione genica monitorando in una sola volta gli RNA prodotti da migliaia di geni.

***Fig. 2**: Ibridazione di DNA e RNA*



1.2 Tipi di DNA microarray

Esistono due tipi di DNA microarray:

1. a singolo canale, i quali permettono di rilevare l'espressione assoluta di migliaia di geni in una certa condizione (es. sano o malato); Illumina human methylation 450k utilizza una tecnologia di microarray a singolo canale basata su microsfere di silice. Sulla superficie di ogni array, o BeadChip, possono essere analizzati contemporaneamente da centinaia di migliaia a milioni di genotipi per un singolo individuo. Queste minuscole perle di silice sono alloggiate in micropozzetti accuratamente incisi e rivestiti con più copie di una sonda oligonucleotidica mirata a un locus specifico nel genoma. Quando i frammenti di DNA passano sul BeadChip, ogni sonda si lega a una sequenza complementare nel DNA del campione, fermando una base prima del locus di interesse. La specificità allelica è conferita da una singola estensione di base che incorpora uno dei quattro nucleotidi marcati. Quando eccitato da un laser, l'etichetta del nucleotide emette un segnale. L'intensità di quel segnale trasmette informazioni sul rapporto allelico in quel locus. La qualità dei dati affidabile e l'eccezionale copertura di preziose regioni genomiche rendono gli array di genotipizzazione Infinium la piattaforma scelta dalle principali istituzioni per lo screening ad alto rendimento e programmi di ricerca su larga scala. La tecnologia Infinium produce una qualità dei dati e una velocità di chiamata eccezionali, oltre a una riproducibilità coerente.
2. a doppio canale, i quali permettono di confrontare migliaia di geni in due condizioni diverse (es. sano vs malato) studiando l'espressione relativa, ossia il rapporto tra le espressioni nelle due diverse condizioni. In questo caso, il processo è molto simile alla funzione di un microarray di DNA a canale singolo. La differenza sta nel fatto che il materiale biologico viene estratto da due tessuti in condizioni differenti, e per ogni spot, la misura dell'espressione è data dal rapporto tra i valori nelle due condizioni. Naturalmente, proprio perché ci sono tessuti derivanti da due diverse condizioni, si devono utilizzare due fluorofori;

Come si avrà modo di approfondire più avanti, ogni singolo esperimento di microarray è influenzato da molteplici fattori esterni e non riguardanti il fattore biologico di interesse. Nel singolo canale, quando si confrontano i valori di espressione di diversi esperimenti, potrebbero esserci differenze di intensità, ad esempio dovute a diversi parametri nella scansione. Ciò significa che i valori tra gli array devono essere corretti per renderli direttamente confrontabili. La tecnologia a doppio canale può confrontare due condizioni nello stesso esperimento; quindi, l'influenza di fattori esterni può essere ridotta. D'altra parte, tuttavia, i due fluorofori utilizzati sono noti per avere efficienze diverse, e quindi

il valore ottenuto deve essere comunque corretto in qualche modo. In definitiva, uno studio effettuato con microarray a doppio canale permette di dimezzare il numero di campioni necessari rispetto al singolo canale e, dunque, risulta essere meno costoso. Il singolo canale, però, ha maggiore riproducibilità, permette di ottenere valori assoluti e di effettuare confronti tra più di due condizioni; inoltre esistono metodi che permettono di correggere variazioni tra array ricavati da studi diversi e rendere dunque confrontabili esperimenti di batch diversi. Visti gli obiettivi di questa tesi, d'ora in avanti si farà riferimento solamente alla tecnologia a singolo canale.

1.3 Struttura della matrice dei dati osservati

La struttura della matrice dei dati osservati, o meglio dei dati derivanti da esperimenti di microarray hanno caratteristiche che rendono questo contesto molto differente da altre applicazioni statistiche. Come primo punto, si noti che le unità statistiche sono i chip stessi, dato che sono loro le unità elementari sulle quali si vuole rilevare un insieme di caratteri. Questi caratteri, o variabili, sono invece i geni, dei quali si vogliono misurare i valori di espressione. Inoltre, le condizioni esaminate sono state prelevate da due tipi di esperimenti:

- Profilo di metilazione mediante array (GSE51921) condotto dalla Clinica Fundacio/IDIBAPS, Barcellona.
- Profilo di metilazione mediante array di tasselli del genoma (GSE60821) condotto all'Università di Kyoto.

scaricati dal database Gene Expression Omnibus e dal database di TCGA, un programma di genomica del cancro.

***Fig. 3**: Matrice dei dati costituita da n numero totale di campioni e m numero totale di geni*

	Campione_1	…	Campione_n
Gene_1	$V_{1,1}$	…	$V_{1,n}$
…	…	…	…
Gene_m	$V_{m,1}$	…	$V_{m,n}$

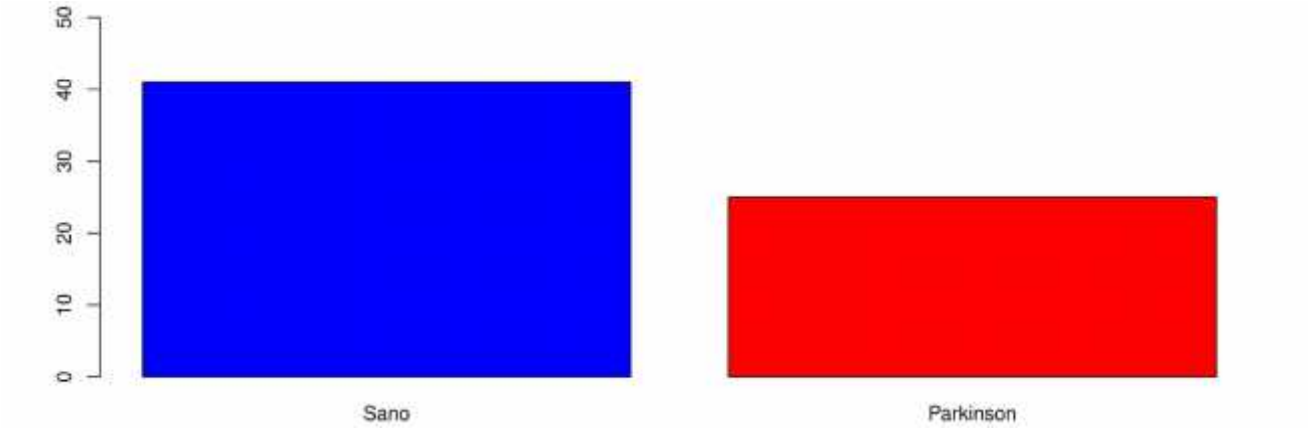
Ad ogni cella del data set è assegnato il valore del rapporto tra l'intensità di un gene a una data condizione rispetto alla condizione standard (i dati sono trasformati come log in base2). Tale valore corrisponde alla normalizzazione basata sui quantili del valore beta.

Il valore beta stima il livello di metilazione, distinguendo tre livelli:

1. valore beta <0.25 = Non metilato;
2. 0.25 < valore beta < 0.75 = Emimetilato;
3. valore beta > 0.75 = Metilato;

In particolare, la matrice dei dati in input utilizzati per il nostro caso di studio è costituita da 478177 righe e 66 colonne, le quali rappresentano i campioni analizzati. Il data set contiene dati ottenuti sia da individui affetti da Parkinson che individui di controllo, cioè sani.

***Fig. 4**: Distribuzione dei campioni.*



Capitolo 2 - Pre-processing Analysis

I dati derivanti dai microarray, come tutti i dati nei database di tutto il mondo, sono influenzati da rumore e valori mancanti, che ne rendono difficile l'analisi successiva e tutti i relativi processi di data mining. Infatti, per eseguire tale processo, generalmente, i dati dovrebbero essere preparati come segue:

1. le righe sono osservazioni e le colonne sono variabili;
2. qualsiasi valore mancante nei dati deve essere rimosso o stimato;
3. i dati devono essere ridimensionati;

Le fasi principali per processare i dati sono:

- data cleaning, ossia una pulizia dei dati eliminando o riempiendo i valori mancanti, togliendo il più possibile il rumore, identificando o rimuovendo gli outlier e risolvendo le inconsistenze;
- data transformation, come normalizzazioni e aggregazioni;
- data reduction, per ottenere una rappresentazione ridotta del data set;

In questo capitolo illustreremo nel dettaglio le fasi sopracitate e in particolare analizzeremo i diversi metodi esistenti, focalizzando l'attenzione su quelli applicabili al contesto dei microarray.

2.1 Riduzione del data set

Al termine del capitolo precedente è stato presentato il data set di partenza, il quale ricordiamo, è costituito da 478177 righe e 66 colonne.

***Fig. 5**: Piccolo sottoinsieme del dataset in input*

	GSM1255293	GSM1255294	GSM1255295	GSM1255296	GSM1255297	GSM1255298
cg00014152	0.95486271	0.87633898	0.84809061	0.967694994	0.93639018	0.94350283
cg00021786	0.95946451	0.90737040	0.88272538	0.957931287	0.95965681	0.96650827
cg00072839	0.04782258	0.08222626	0.07994500	0.057090583	0.02192519	0.01805974
cg00074638	0.53910664	0.58849741	0.53582295	0.586670127	0.54813833	0.55519468
cg00266918	0.05550467	0.12720488	0.08968033	0.101750547	0.04497212	0.05654001
cg00378717	0.02172863	0.01993563	0.01708322	0.011249343	0.02357298	0.02289420
cg00378950	0.81732824	0.71575724	0.80701199	0.887022516	0.82313653	0.86965480
cg00381376	0.01363728	0.42388250	0.03800582	0.016560446	0.41096552	0.03077243
cg00388637	0.61470832	0.66344294	0.67344147	0.673782405	0.63147257	0.63976255
cg00423014	0.65363867	0.70356887	0.64600539	0.679243697	0.72185369	0.71276596
cg00457389	0.04570490	0.11306660	0.03054833	0.035361414	0.02973752	0.03733634
cg00551957	0.87033117	0.86197749	0.82584846	0.916732335	0.91431617	0.89729698
cg00573298	0.04006960	0.49099339	0.02527568	0.023213290	0.38406737	0.04078687
cg00685229	0.44465151	0.48644154	0.34606815	0.468932122	0.54756811	0.28660727

Tra i diversi problemi statistici ritroviamo la grande dimensionalità del data set a cui si contrappongono campioni molto limitati. In letteratura si fa riferimento a questo problema con l'espressione "large p and small n". Oltre a comportare spesso tempi di elaborazione piuttosto lunghi, questa caratteristica espone al rischio di sovra parametrizzazione del modello. In secondo luogo, la maggior parte dei geni nel data set sono irrilevanti al fine dei processi di data mining e costituiscono rumore che interferisce con il potere discriminante degli altri geni. Questo accresce non solo i tempi di calcolo, ma anche la difficoltà di esecuzione. Per questo motivo, come operazione preliminare, si devono eliminare i geni irrilevanti per la nostra analisi ed individuare il gruppo di campioni di nostro interesse, in modo tale da migliorare l'accuratezza delle operazioni da svolgere successivamente. Si è deciso, quindi, di eliminare i geni relativi allo studio di altre problematiche. In particolare, sono stati eliminati i geni presenti nel data set "snps450k" e nel "cgcross". Gli SNPs (variazione del materiale genico a carico di un unico nucleotide), infatti, possono influenzare lo sviluppo delle patologie, in quanto sono variazioni dei fattori ereditari che si sono sviluppati naturalmente nel corso dell'evoluzione. Non sono in grado di predire esattamente il momento in cui si sviluppa un disturbo della salute, ma possono predire il rischio di sviluppare tale disturbo. Invece, la rimozione dei CpG (abbreviazione di " --C-- phosphate--G-- "), relativa ai cromosomi x e y, è stata effettuata per ridurre

la variabilità. Infatti, si sta eseguendo un'analisi soltanto sui geni autosomici e non su quelli legati al sesso e di conseguenza rimuovere tali informazioni aggiuntive migliora le prestazioni e il tempo d'elaborazione dell'analisi.

***Script 1**: Rimozione geni irrilevanti*

```
38 #Rimozione dei geni irrilevanti per il nostro studio
39
40 genes_useless = union(cgcross$V1,rownames(snps450k))
41 genes_useless = sort(genes_useless)
42 position_genes = which(rownames(Dataset_Parkinson2) %in% genes_useless)
43 Dataset_Parkinson_final = Dataset_Parkinson2[-position_genes,]
44
```

Dopo una prima scrematura il data set è composto da 433805 righe e 66 colonne.

2.2 Rimozione dei valori mancanti

I missing values non sono altro che dati che mancano all'interno di un data set. Vengono indicati dal codice NA (not available). I valori mancati possono produrre errori nell'analisi dei dati poiché interferiscono con i calcoli statistici e quindi con il corretto funzionamento dell'analisi. Le soluzioni perseguibili per risolvere questo problema sono:

1. rimpiazzare il dato mancante con un valore stimato;
2. cancellare il dato mancante in modo definitivo, anche per le future analisi;

Nel primo caso il valore stimato può essere:

- la media tra tutte le misurazioni effettuate;
- la media per quella data classe, ossia la media tra tutte le misurazioni derivanti dai campioni che appartengono alla stessa classe, ad esempio malato/sano;
- il valore più probabile, che può essere determinato con la regressione, alberi decisionali o tool basati sul formalismo Bayesiano;

Per quanto riguarda la cancellazione del dato, che nel caso del microarray significa eliminare dalle successive analisi un gene, la sua efficacia dipende dalla percentuale dei valori mancanti e dalla grandezza del data set. Infatti, se la percentuale diventa troppo alta si rischia di perdere delle informazioni importanti.

Nel nostro caso, come accennato nel paragrafo precedente, abbiamo un data set di grandi dimensioni, di conseguenza è stato possibile rimuovere i dati mancanti.

Tale operazione è stata eseguita nel seguente modo:

***Script 2**: Rimozione degli elementi nulli*

```
#Eliminazione elementi nulli

DS_Parkinson_omit = na.omit(Dataset_Parkinson_final)

```

Dopo la rimozione dei valori mancanti il data set è composto da 412414 righe e 66 colonne.

2.3 Rimozione del batch effect

Negli studi di espressione genica con l'utilizzo di microarray a singolo canale si assiste alla presenza di unwanted variation, letteralmente variazione non voluta. Oltre ai fattori biologici di interesse per gli sperimentatori, infatti, ci sono solitamente altri fattori, non biologici, che influenzano i valori di espressione. Questi possono essere, ad esempio, diversi parametri di scansione, diversa potenza del laser o diversi reagenti. Essenzialmente il rumore nei dati dei microarray è dovuto alla procedura di calcolo del valore di espressione. Un tipo molto forte di variazione non voluta è quella che si ricava dall'utilizzo di array ottenuti in condizioni sperimentali diverse. Questa è generalmente definita batch effect e si ha, ad esempio, quando i dati sono stati ricavati da due laboratori diversi, oppure con tecnologie di microarray diverse, o ancora semplicemente quando i campioni vengono ricavati in giorni diversi. Il batch effect può essere causato addirittura dal diverso momento del giorno nel quale si fanno le repliche (mattina/pomeriggio) e dal livello di ozono nell'atmosfera. Per campioni generati in uno stesso batch si intende, infatti, microarray processati nello stesso posto, in un breve periodo di tempo e utilizzando la stessa piattaforma. È quindi di fondamentale importanza ridurre tale variazione, in quanto si potrebbero ottenere delle misurazioni falsate. Come già detto all'inizio di questo capitolo, esistono diversi metodi per eseguire le operazioni di data cleaning. In questo caso, per eliminare il rumore dal data set in input, è stata utilizzata una funzione del pacchetto SVA. Il pacchetto SVA contiene funzioni per la rimozione di effetti batch e altre variazioni indesiderate nell'esperimento ad alto rendimento. Nello specifico, il pacchetto SVA contiene funzioni per l'identificazione e la costruzione di variabili surrogate per set di dati ad alta dimensione. Le variabili surrogate sono costruite direttamente da dati ad alta dimensionalità (come dati di espressione genica/ sequenziamento dell'RNA/ metilazione/ immagini del cervello) che possono essere utilizzate in analisi successive per regolare fonti di rumore sconosciute, non modellate o latenti. Il pacchetto SVA può essere utilizzato per rimuovere gli artefatti in tre modi:

- identificare e stimare variabili sostitutive per fonti sconosciute di variazione in esperimenti ad alto rendimento
- rimuovere direttamente il batch effect noto usando ComBat

- rimuovere il batch effect con sonde di controllo note.

In particolare, la funzione utilizzata è ComBat(), la quale per essere eseguita necessita di un fattore batch di riferimento. A tale scopo sono state utilizzate le informazioni contenute nel file "p_data" relative allo studio dei diversi campioni. Ci si è concentrati sulla colonna "gse.1", in quanto i 66 campioni provengono da 2 studi diversi. Tale fattore è stato passato come parametro alla funzione del pacchetto SVA, la quale ha eliminato la variazione indesiderata dal nostro data set.

***Script 3**: Rimozione del Batch Effect*

```
79   #Selezione del vettore di riferimento
80   batch <- p_data$gse.1
81   batch = batch[1:66]
82   batch_eff <- ComBat(DS_Parkinson_scale, batch)
83
```

È stata confrontata, infine, la variabilità nel data set dato in input alla funzione con il data set ottenuto in output.

***Fig. 6**: Rappresentazione dell'unwanted variation prima dell'eliminazione del batch effect*

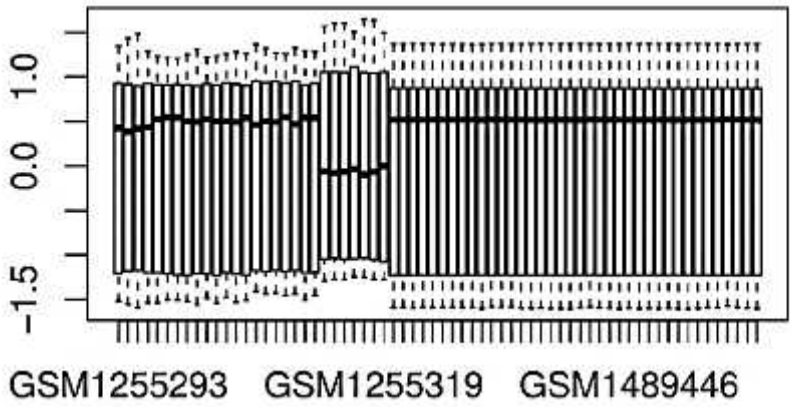
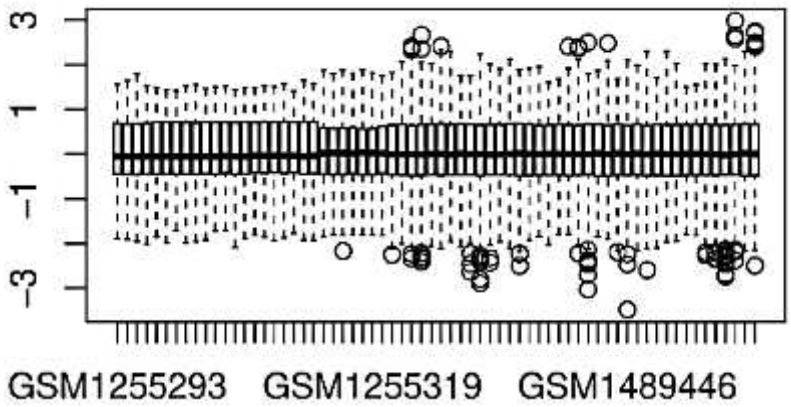


***Fig. 7**: Rappresentazione dell'unwanted variation dopo dell'eliminazione del batch effect*



I valori di espressione corretti (Fig. 6), per questi geni, hanno variabilità minore rispetto a quelli non corretti (Fig.7). Ciò indica che l'aggiustamento ha funzionato adeguatamente.

2.4 Normalizzazione

Il feature scaling (o ridimensionamento dei dati) è un metodo utilizzato per trasformare i dati grezzi in formato comprensibile, rendendo i dati analizzati più gestibili e allo stesso tempo preservandone il significato. Di seguito sono illustrati due tra i metodi di ridimensionamento dei dati più comuni:

- il ridimensionamento standard, noto anche come standardizzazione o normalizzazione del punteggio Z, è una procedura per "standardizzare" i dati in modo che abbiano media uguale zero e deviazione standard uguale a uno. La formula della standardizzazione è:

$$z = \frac{x - m}{d \qquad s_i}$$

  In statistica, una variabile standardizzata è una variabile quantitativa a cui è stata cambiata la scala di misurazione ottenendo dei numeri puri (detti anche punteggi z o punteggi standard). Questi nuovi valori sono detti anche adimensionali, in quanto sono svincolati dall'unità di misura della variabile di partenza, per renderla più facilmente confrontabile con le altre. È una procedura altamente consigliata in tutti quei casi in cui si effettua un confronto tra variabili che hanno diverse unità di misura/ordini di grandezza, ad esempio prima di un'analisi dei cluster. La standardizzazione gestisce i valori anomali, ma non produce dati normalizzati con la stessa scala;

- la normalizzazione quantile è una tecnica utilizzata per rendere due distribuzioni identiche. Questo tipo di ridimensionamento è spesso utilizzato nell'analisi dei dati dei microarray e in questo caso specifico con la normalizzazione quantile si può dare a tutti gli array la stessa distribuzione sostituendo i valori del data set originale con il quantile medio. Quindi, data una matrice X di n array (colonne) e ng geni (righe), si ordina ciascuna colonna di X per ottenere Xsort e successivamente si assegna ad ogni elemento della k-esima riga di Xsort la media della riga stessa, ottenendo X'sort. A questo punto si calcola Xnormalized, riordinando ogni colonna di X'sort secondo l'ordine originale;

In sintesi, la scelta del metodo di feature scaling adatto dipende dai valori desiderati e richiesti dai processi successivi. Tornando al nostro lavoro, si è scelto in questa fase di standardizzare il data set (Fig. 6), in quanto la fase successiva necessita di dati uniformi. Il codice è il seguente:

***Script 4**: Normalizzazione dei dati*

```
#Normalizzazione dei valori interni della mat rice

DS_Parkinson_scale = scale(DS_Parkinson_omit)

```

***Fig. 8**: Piccolo sottoinsieme del dataset dopo la normalizzazione*

	GSM1255293_healthy	GSM1255294_healthy	GSM1255295_healthy	GSM1255296_healthy	GSM1255297_park	GSM1255298_park
cg00014152	1.24234760	1.083033639	1.03587630	1.21983047	1.083339122	1.09127792
cg00021786	1.25566140	1.176673327	1.14316996	1.19200453	1.148597856	1.15476296
cg00266918	-1.33965301	-1.177535745	-1.31357244	-1.24605563	-1.416929156	-1.35635398
cg00378717	-1.45737301	-1.501228907	-1.53846810	-1.50597825	-1.476949906	-1.44920181
cg00378950	0.84443626	0.598463953	0.90862048	0.98991909	0.765682887	0.88748946
cg00381376	-1.48076273	-0.282288360	-1.47363260	-1.49084195	-0.390383114	-1.42746132
cg00457389	-1.38800547	-1.220198485	-1.49675506	-1.43726040	-1.459659476	-1.40934777
cg00551957	0.99778303	1.039696743	0.96697322	1.07439016	1.021431083	0.96376986
cg00573298	-1.40425151	-0.079776133	-1.51308901	-1.47188177	-0.465827606	-1.39982030
cg00685229	-0.23378292	-0.093511700	-0.51925696	-0.20161170	-0.007237274	-0.72146634
cg00697812	0.60181194	0.520341064	0.57249606	0.76039181	0.623891451	0.74848671
cg00751785	-1.40143634	-1.456493776	-1.50441968	-1.42094603	-1.472287059	-1.45620023
cg00812634	1.23442944	1.252192262	1.23690621	1.24090595	1.102669406	1.17683512

Capitolo 3 - Data Mining

Il data mining è un insieme di tecniche e metodologie che hanno lo scopo di estrarre informazioni utili da grandi quantità di dati, utilizzando metodi di Machine Learning, intelligenza artificiale, statistica e basi di dati. Costituisce la parte di modellazione del processo di “Knowledge Discovery in Database”.

Va specificato, inoltre, che le tecniche di data mining si suddividono in:

- supervisionate: l'algoritmo deve apprendere il comportamento di alcune variabili target. Un esempio sono gli alberi decisionali;
- non supervisionate: le variabili target non sono conosciute a priori. Un esempio è il clustering;

In questo capitolo illustreremo la tecnica del clustering, per poi passare alla tecnica della classificazione.

3.1 Clustering

Il clustering è una tecnica di analisi progettata per individuare gruppi significativi in un data set. La parola “significativi" in questo contesto ha un significato puramente topologico. Infatti, un cluster viene definito come un insieme di oggetti tale che gli oggetti all’interno di un cluster siano “simili”, mentre gli oggetti di cluster differenti siano “dissimili”. Le differenze sono valutate sulla base dei valori degli attributi che descrivono l’oggetto. Come funzionalità del data mining, il clustering può essere utilizzato per esaminare le distribuzioni dei dati, per osservare le caratteristiche di ciascuna distribuzione e per focalizzarsi su quelle di maggiore interesse. Alternativamente, esso può essere utilizzato come un passo di pre-processing per altri algoritmi, quali la classificazione e la caratterizzazione, che operano sui cluster individuati. In questo caso, infatti, il clustering viene utilizzato come operazione di pre-processing per ridurre ulteriormente la dimensionalità del data set e migliorare, così, l’efficienza e il tempo di esecuzione della classificazione. Nello specifico, per individuare il gruppo di campioni da eliminare sono stati applicati e comparati due algoritmi di clustering: il clustering gerarchico agglomerativo e il k-means clustering, entrambi oggetto di analisi in questa sezione.

3.1.1 Tipi di dati nel clustering

Si supponga che un insieme di dati da clusterizzare contenga n oggetti che possono rappresentare persone, cose, documenti, nazioni, ecc. Gli algoritmi di clustering tipicamente operano su una delle seguenti strutture dati:

- una matrice di dati (o struttura object-by-variable): questa rappresenta n oggetti, come ad esempio persone, con p variabili (chiamate anche misure o attributi), quali l'età, l'altezza, il peso, la razza e così via. La struttura è nella forma di una tabella relazionale, o matrice n × p (n oggetti per p variabili);

$$\begin{pmatrix} x_1 & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_n \end{pmatrix}$$

- una matrice di dissimilarità (o struttura object-by-object): questa memorizza il grado di dissimilarità di ciascuna coppia degli oggetti coinvolti. Essa è spesso rappresentata da una tabella n × n, come di seguito specificato:

$$\begin{pmatrix} 0 & 0 & \dots & 0 \\ d(2,1) & 0 & \dots & 0 \\ d(3,1) & d(3,2) & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ d(n,1) & d(n,2) & \dots & 0 \end{pmatrix}$$

  dove d(i, j) è la differenza o dissimilarità misurata tra gli oggetti i e j.
  Si noti che d(i, j) = d(j, i) e che d(i, i)= 0;

Molti algoritmi di clustering operano sulla matrice delle dissimilarità. Se i dati sono presentati sotto forma di una matrice di dati è necessario trasformare quest'ultima in una matrice di dissimilarità prima di applicare tali algoritmi.

3.1.2 Categorizzazione dei principali metodi di Clustering

La scelta dell'algoritmo da utilizzare in un dato contesto dipende dal tipo di dati disponibili, dal particolare scopo e dall'applicazione. In generale, i principali metodi di clustering possono essere classificati come di seguito specificato:

- metodo gerarchico. Dato un insieme di oggetti crea una decomposizione gerarchica;
- metodo di partizionamento. Dato un database di n oggetti, e k, il numero di cluster da costruire, un algoritmo di partizionamento organizza gli oggetti in k partizioni ($k \leq n$);

Alcune volte i due approcci vengono combinati adottando un approccio ibrido tra k-means clustering e clustering agglomerativo: il primo applicato per la sua efficienza in fase di esecuzione, il secondo per la sua qualità. In questa attività si è deciso di applicare entrambi gli algoritmi in modo da confrontarli e proporre una soluzione ottimale. Per poter utilizzare tali tecniche si è dovuto integrare al nostro lavoro il pacchetto CLUSTER.

3.2 Clustering Gerarchico

Un metodo di clustering gerarchico lavora raggruppando gli oggetti in alberi di cluster. Il raggruppamento gerarchico può essere suddiviso in due tipi principali, basandosi su come viene effettuata la decomposizione gerarchica.

1. nell'approccio agglomerativo, detto anche approccio "bottom up", ogni oggetto viene inizialmente considerato come un cluster a elemento singolo (foglia). Ad ogni passaggio dell'algoritmo, i due cluster più simili vengono combinati in un nuovo cluster più grande (nodi). Questa procedura viene ripetuta finché tutti i punti non sono membri di un unico grande cluster oppure fino a quando non si verifica una condizione di terminazione;
2. l'approccio divisivo, detto anche approccio "top-down", opera in modo inverso rispetto ai metodi gerarchici agglomerativi. Inizia, infatti, con tutti gli oggetti posti nello stesso cluster. Ad ogni passaggio dell'iterazione, il cluster più eterogeneo viene diviso in due. Il processo viene iterato finché ciascun oggetto si trova in un cluster differente o fino a quando non si verifica una determinata condizione di terminazione, legata al numero desiderato di cluster o alla distanza tra cluster;

In generale, partendo da una matrice dei dati mxn, l'algoritmo di clustering gerarchico deriva da questa una matrice di prossimità, ovvero un array di dimensione mxn che gode delle proprietà di non negatività, simmetria, identità e disuguaglianza triangolare. Nel nostro caso l'indice di prossimità utilizzato è la distanza euclidea, calcolata tra i vettori della matrice dei dati. Ricordiamo che la nostra matrice, dopo le fasi di pre-processing analysis, è costituita da 412414 righe e 66 colonne.

$$D_1 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

La distanza euclidea tra due generici punti di coordinate (x1; y1) e (x2; y2) viene calcolata come radice della somma dei quadrati delle differenze tra i punti di coordinata.

In seguito, l'algoritmo prevede l'unione iterata dei cluster, in base ad un determinato metodo applicato per calcolare la distanza tra i neo-cluster e le restanti unità statistiche. I metodi più comuni sono:

1. "complete": calcola tutte le differenze a coppie tra gli elementi nel cluster 1 e gli elementi nel cluster 2 e considera il valore più grande (cioè il valore massimo) di queste differenze come la distanza tra i due cluster. Tende a produrre grappoli più compatti;
2. "single": calcola tutte le dissomiglianze a coppie tra gli elementi nel cluster 1 e gli elementi nel cluster 2 e considera la più piccola di queste differenze come criterio di collegamento. Tende a produrre grappoli lunghi e "sciolti";
3. "averange": calcola tutte le differenze a coppie tra gli elementi nel cluster 1 e gli elementi nel cluster 2 e considera la media di queste differenze come la distanza tra i due cluster;
4. "Ward": riduce al minimo la varianza totale all'interno del cluster. Ad ogni passaggio la coppia di cluster con una distanza minima tra i cluster viene unita;

Nel caso in esame, il metodo applicato è stato quello di Average, utilizzato come dato di input, insieme alla matrice delle distanze, nella funzione hclust(). È stato scelto il metodo Average in quanto è un approccio robusto al rumore e agli outliers. Il codice utilizzato è il seguente:

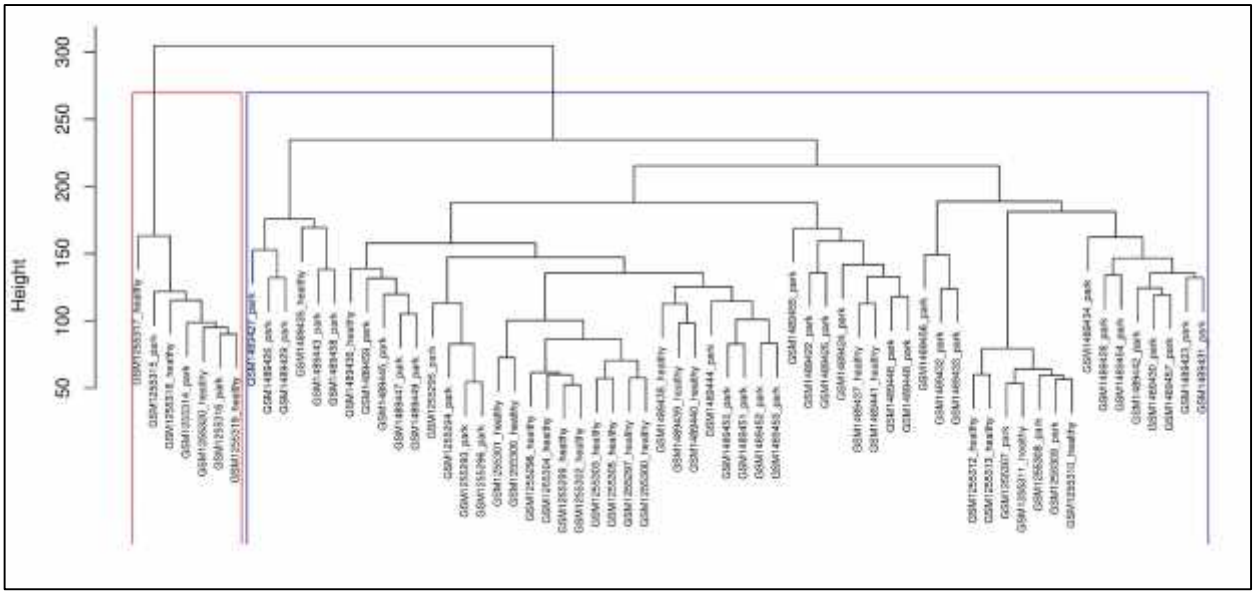
***Script 5**: Distanza euclidea e metodo Avarage per il clustering gerarchico*

```
96   #calcolo della matrice delle distanze e clustering gerarchico
97   d <- dist(t(batch_eff), method = "euclidean")
98   h_clust<- hclust(d, method = "ave")
99
```

L’applicazione dell’algoritmo gerarchico ha prodotto un dendrogramma, ovvero un grafo ad albero rappresentante i livelli di somiglianza, quindi di aggregazione dei campioni.
Infine, utilizzando la funzione rect.hclust() sono stati evidenziati i clusters individuati dall’algoritmo, visivamente delineati uno da un rettangolo di colore rosso e uno di colore blu.
Nello specifico, partendo dai 66 campioni iniziali, i due cluster risultano essere costituiti (Fig. 9):

1. Cluster costituito da 7 individui (evidenziato in rosso);
2. Cluster costituito da 59 individui (evidenziato in blu);

Fig. 9*: Dendrogramma rappresentate il raggruppamento prodotto dal clustering*



3.3 Clustering di Partizionamento

Mentre nel caso dei metodi gerarchici l'algoritmo cerca, ad ogni passo, la migliore scissione o aggregazione tra cluster, nei metodi non gerarchico l'algoritmo partiziona le n unità in un numero prestabilito k(<n) di gruppi, dove ciascuna partizione rappresenta un cluster ed ogni gruppo soddisfa i seguenti requisiti:

1. ciascun gruppo deve contenere almeno un oggetto;
2. ciascun oggetto deve appartenere esattamente ad un gruppo;

A differenza dei metodi gerarchici, l'assegnazione di un oggetto ad un cluster non è irrevocabile, infatti viene, successivamente, applicata una tecnica di rilocazione iterativa che tenta di migliorarlo spostando oggetti da un gruppo ad un altro. In questo modo, ad ogni passo, l'algoritmo rimette in discussione la partizione ottenuta. Di solito, scelta una partizione iniziale, si cerca di migliorarla in funzione del criterio di minimizzazione della varianza interna. Infatti, il processo prevede una prima fase nella quale vengono individuati i k poli provvisori, ciascuno dei quali rappresenta inizialmente la media o il centro di un cluster. Successivamente ciascuno degli oggetti rimanenti viene associato al cluster più simile basandosi sulla distanza tra l'oggetto e la media del cluster, in modo da formare i primi k cluster. Si individua, poi, il centroide di ogni gruppo e, per ogni gruppo, si calcola la varianza interna.

Calcolando la distanza euclidea tra ogni unità statistica ed il suo centroide si cerca di minimizzare l'errore quadratico, definito come:

$$E = \sum_{i-1}^{K} \sum_{p \in C_i} d_e(p, m_i)^2$$

dove m_i è il punto medio del cluster C_i.

Se l'algoritmo individua che tale unità è più vicina ad un altro dei k centroidi, allora viene riassegnata. Tale procedimento di riallocazione delle unità viene iterato fino a quando l'algoritmo converge, ovvero fino a quando non ci sarà più alcun spostamento. Il problema è che l'algoritmo potrebbe convergere ad un ottimo locale (e non globale); ciò significa che se partissimo da un diverso insieme di centri provvisori potremmo ottenere una partizione differente. Per questo motivo, viene definita una regola di arresto, ovvero un numero massimo di iterazioni dopo il quale si fa convergere l'algoritmo.

Il primo passo per applicare l'algoritmo k-means consiste nel normalizzare il data set in input. È stata utilizzata una funzione presente nel pacchetto LIMMA.

***Script 6**: Normalizzazione basata su quantili*

```
114   #Normalizzazione quantile
115
116   DS_normQuantile= normalizeQuantiles(t(batch_eff), ties = FALSE)
117
```

Come già spiegato in questa sezione, si devono prestabilire il numero dei k gruppi da formare. Nel nostro caso k è stato fissato al valore 2. È possibile controllare l'utilizzo della funzione nel box sottostante.

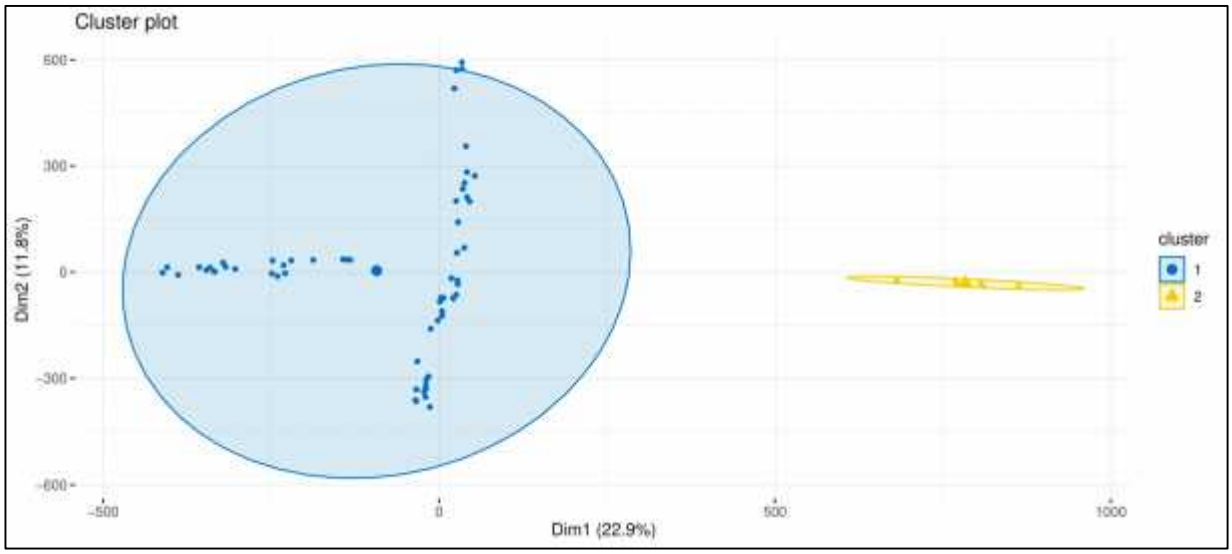
***Script 7**: Clustering k-means*

```
118   #Clustering k-means
119
120   k_means<- kmeans(DS_normQuantile,2)
121
```

La funzione kmeans restituisce un oggetto della classe "kmeans" ed è un elenco con i seguenti componenti:

- cluster = un vettore di numeri interi (da 1: k) che indica il gruppo a cui è allocato ogni punto;
- centers = una matrice di centri dei gruppi;
- totss = la somma totale dei quadrati;
- withinss = vettore della somma dei quadrati all'interno del gruppo;
- tot.withinss = somma totale dei quadrati all'interno del gruppo;
- betweenss = la somma dei quadrati tra gruppi;
- size = il numero di punti in ogni gruppo;
- iter = il numero di iterazioni;
- ifault = indicatore di un possibile problema dell'algoritmo;

***Fig. 10**: Cluster plot ottenuto da kmeans*



A questo punto è stato possibile fare una comparazione tra i due differenti metodi di analisi, evidenziando similarità e differenze nei risultati ottenuti. Prima di tutto una chiarificazione è necessaria: come sappiamo l'algoritmo di clustering gerarchico è stato applicato alla matrice risultante dalla pre-processing analysis, mentre l'algoritmo di clustering di partizionamento ha lavorato su una matrice ulteriormente normalizzata. I risultati ottenuti coincidono.

Ricordiamo che i campioni da eliminare sono quelli non adatti per la ricerca dei biomarcatori.

I campioni evidenziati dai clustering rappresentano degli outliers, ovvero dei valori anomali e aberranti distanti dagli altri valori disponibili. Tale insieme di campioni è stato ricavato dall'unione dei risultati ottenuti dalle due differenti esecuzioni dell'algoritmo ed è stato sottratto al data set.

In conclusione, il data set è stato pulito dai valori anomali diventando così composto da 412414 righe e 59 colonne.

3.4 Classificazione

La classificazione è una forma di analisi dei dati che può essere utilizzata per estrarre modelli che descrivono le classi dei dati; quindi, la classificazione predice le etichette di categorie che verranno applicate ai dati. La classificazione dei dati avviene in un processo a due passi:

1. il primo passo è costruire un modello per descrivere un set di dati in classi o concetti. Il modello è costruito analizzando le tuple del database descritte dagli attributi. Ogni tupla è assunta appartenente a una classe predefinita, determinata da un attributo specifico, che è chiamato attributo di etichetta di classe. Nel contesto della classificazione, le tuple di dati sono anche chiamate campioni o oggetti. L'insieme di tuple utilizzate per costruire il modello, formano il training data set. Le tuple singole che costituiscono il training set sono definite come training sample e sono selezionate in modo casuale dalla popolazione dei campioni. Tipicamente, il modello appreso è rappresentato sotto forma di regole di associazione, alberi decisionali o formule matematiche;
2. il secondo passo consiste nell'utilizzare il modello per la classificazione. Si stima, innanzitutto, l'accuratezza predittiva del modello o del classificatore. L'accuratezza di un modello su un dato test set è la percentuale di campioni del test set che sono correttamente classificati dal modello stesso. Per ogni campione di test, l'etichetta della classe conosciuta è confrontata con la predizione della classe per quel campione, fornita dal modello. Se l'accuratezza del modello è considerata accettabile, il modello può essere utilizzato per classificare future tuple di dati o oggetti per i quali la classe non è conosciuta;

Tramite la classificazione si intende quindi classificare i geni in base alla probabilità a posteriori di ogni gene di essere differenzialmente espresso.
In questa sezione andremo ad esaminare i due diversi modelli utili per la classificazione, quali il modello lineare e non lineare.

3.5 Modello lineare e modello non lineare

Nella statistica il modello lineare è una tecnica per l'analisi delle relazioni tra fenomeni. In generale il modello è costituito da un sistema di equazioni, lineari nei parametri, che descrivono o interpretano l'interazione fra due gruppi di variabili: il primo costituito da quelle determinate all'interno del sistema, dette anche variabili endogene o dipendenti, e il secondo costituito dalle variabili esogene o esplicative, il cui valore preesiste all'interazione, che sono determinate all'esterno del sistema.

L'insieme delle relazioni determina quindi simultaneamente i valori delle variabili endogene in funzione delle esogene. Le funzioni esprimenti tali relazioni si suppongono lineari nei parametri, ma ammettono non linearità nelle variabili; esse rappresentano, in linguaggio matematico, il sistema d'interdipendenze che secondo le conoscenze acquisite (teoria del fenomeno) caratterizzano gli eventi oggetto d'indagine statistica. In genere gli algoritmi di addestramento basati sulla classificazione lineare risentono del problema dell'underfitting, ovvero una classificazione basata su pochi parametri e che può soffrire di un'eccessiva discrepanza. In compenso sono semplici e veloci.
Per provare a catturare l'andamento non lineare delle variabili vengono adottati i modelli non lineari. La regressione non lineare è un metodo di stima di una curva interpolante un modello. Diversamente da quanto accade nel caso della regressione lineare, non esiste un metodo generale per determinare i valori dei parametri che garantiscono la migliore interpolazione dei dati. A tal fine, si ricorre a classi di algoritmi numerici di ottimizzazione, che a partire da valori iniziali, scelti a caso o tramite un'analisi preliminare, giungono a punti ritenuti ottimali. In genere, l'accuratezza del modello previsionale è più alta rispetto ai regressori lineari perché la stima è una curva o uno spazio curvo. Tuttavia, se il regressore si adatta eccessivamente ai dati di training si rischia di cadere nel problema dell'overfitting, ovvero troppi parametri nel modello e un'elevata variabilità della classificazione. Il modello è troppo complesso e sensibile ai dati di training.

3.5.1 Confronto tra modelli

La ricerca di un metodo o modello che sia in grado di fornire previsioni accurate è stata ed è tuttora un'attività oggetto di notevole interesse e impegno, a causa dell'utilità e della rilevanza che le previsioni rivestono in qualsiasi area di applicazione.
Lo scopo di questa sezione è quello di stabilire qual è il modello più adeguato a individuare i geni differenzialmente espressi. Innanzitutto, esaminiamo i criteri con i quali comparare e valutare diversi metodi di classificazione. Tali criteri sono:

- accuratezza: si riferisce all'abilità del modello di predire correttamente l'etichetta di classe di un nuovo dato;
- velocità: si riferisce ai costi computazionali coinvolti nella generazione e nell'utilizzo del modello;
- robustezza: l'abilità del modello a produrre predizioni corrette su dati rumorosi o con valori mancanti;
- scalabilità: si riferisce all'abilità nel costruire un modello efficiente data una gran quantità di dati;

- interpretabilità: si riferisce al livello di comprensione che il modello fornisce;

Seguendo le caratteristiche sopracitate possiamo affermare che il modello lineare è quello che più frequentemente viene utilizzato specie quando, oltre alla costruzione di previsioni "ottimali", è necessario simulare scenari. In questo caso f( ) e g( ) sono funzioni note, in qualche prefissata classe, a meno di un numero finito di parametri. In tali situazioni si parla anche di modelli per serie storiche. L'obiettivo delle serie storiche è quello di identificare le regolarità presenti nelle osservazioni passate e sviluppare un modello che sia in grado di rappresentare in modo semplificato il processo generatore dei dati. Il modello stimato viene poi impiegato con il fine ultimo di ricavare previsioni sull'andamento del fenomeno oggetto di studio. Nel contesto non lineare il modello è molto generico, di conseguenza è possibile che si presentino alcuni inconvenienti. Nello specifico tale modello è utile per uno smoothing dei valori osservati, lo è molto meno per costruire previsioni ed inoltre, in questo caso, le funzioni f( ) e g( ) risultano, in molti casi, di difficile interpretazione. D'altro lato, i modelli lineari non sono in grado di catturare particolari strutture come asimmetrie, curtosi, cicli limiti, presenti in alcuni fenomeni reali e quindi risulta indispensabile ricorrere a strutture dinamiche non lineari, le quali sono capaci di descrivere comportamenti molto complessi.

3.6 Geni differenzialmente espressi

Si definiscono geni differenzialmente espressi (DEG) quei geni che in due condizioni differenti (es. tessuti sani e tessuti cancerogeni) hanno un livello di espressione significativamente diverso. Rispetto ad una situazione di riferimento, i DEG possono dunque essere sovra o sotto espressi.

In questa sezione tratteremo e descriveremo l'implementazione di un modello lineare utile per la classificazione dei DEG. In particolare faremo riferimento alla funzione lmFit() [1] del pacchetto LIMMA. Questa funzione adatta modelli lineari multipli per quadrati minimi ponderati o generalizzati. Accetta dati da un esperimento che coinvolge una serie di microarray con lo stesso set di sonde. Un modello lineare viene adattato ai dati di espressione (per gene) per ciascuna sonda. I dati di espressione devono essere log-ratios per le piattaforme di array a doppio canale o valori log-expression per le piattaforme a singolo canale. I coefficienti dei modelli adattati descrivono le differenze tra le sorgenti di RNA ibridate agli array. I risultati del modello adattato per sonda sono memorizzati in una forma compatta adatta per ulteriori elaborazioni da parte di altre funzioni del pacchetto LIMMA. Per poter usufruire di tale funzione abbiamo bisogno di una design matrix a cui adattare il modello lineare. Il modello lineare è stato adattato tramite il seguente codice:

***Script 8**: Design matrix e modello lineare.*

```
#Design matrix
label<- as.factor(p_data_final$label_cell_line)
label = label[1:59]
model_limma<- model.matrix(~-1+label, data=p_data_final)
colnames(model_limma)<- c("Healthy","Park")

#Modello lineare
fit<- lmFit(DS_final, model_limma)

```

L'approccio richiede che siano specificate due matrici:

1. design matrix, la quale fornisce una rappresentazione dei diversi target di RNA che sono stati ibridati agli array;
2. contrast matrix, la quale permette di combinare i coefficienti definiti dalla design matrix in contrasto con i target di RNA;

La contrast matrix è stata creata tramite la funzione makeContrasts():

***Script 9**: Contrast matrix.*

```

#Matrice dei contrasti

contrasts_names<- "Healthy-Park"
contrasts<- makeContrasts(contrasts_names, levels= model_limma)

```

La matrice dei contrasti ha la seguente forma:

	Healthy-Park
Healthy	1
Park	-1

Dopo la creazione della matrice dei contrasti, è stata utilizzata la funzione contrasts.fit() la quale, dato un modello lineare adattato da dati di microarray, calcola i coefficienti stimati e gli errori standard per un dato insieme di contrasti. La funzione riorienta il modello adattato dalla design matrix originale ad un qualsiasi insieme di contrasti dei coefficienti originali. I coefficienti, le deviazioni standard non scalate e la matrice di correlazione vengono ricalcolate in termini di contrasti. L'idea di questa funzione è quella di adattare un modello full-rank usando lmFit o equivalente, quindi utilizzare contrasts.fit per ottenere coefficienti ed errori standard per qualsiasi numero di contrasti dei coefficienti del modello originale. Dopo ciò, è stato possibile eseguire il test di Ebayes per calcolare

alcune statistiche riguardanti i geni. Tale test utilizza un approccio Bayesiano empirico per il calcolo delle probabilità a posteriori di ogni gene di essere differenzialmente espresso. Questo metodo offre la possibilità di utilizzare informazioni raccolte da tutto l'insieme di geni per sfruttarle nell'inferenza di ogni singolo gene; in altre parole, sfrutta la struttura parallela dell'analisi. Per ogni gene si assume un modello lineare del tipo:

$$Y_g = Xa_g + \varepsilon_g$$

dove Y_g è il vettore dei valori di espressione, X è la matrice del disegno, a_g è il vettore dei parametri e ε_g è un termine d'errore non necessariamente normale a media zero. Dato un modello lineare di microarray, tale funzione calcola, tramite moderazione empirica di Bayes degli errori standard verso un valore comune dell'espressione differenziale nel seguente modo:

***Script 10**: Test Ebayes.*

```
#Statistiche sui geni

bayes<-eBayes(fit)

```

L'output della funzione eBayes() viene poi utilizzato come input della funzione topTable(), la quale estrae, in base alle statistiche calcolate con la prima funzione, una tabella dei geni top-ranked dal modello lineare adattato, ovvero una "classifica" dei geni che sono in particolar modo differenzialmente espressi. Il codice è il seguente:

***Script 11**: Classificazione geni.*

```
#classificazione dei geni per espressione differenziale
fit2<-eBayes(contrasts_fit)
CpG<- nrow(DS_final)
top_table<- topTable(fit2,number=CpG, coef=1)
top_table$score<- sign(top_table$logFC)*-log10(top_table$P.value)
top_table<- top_table[order(top_table$score, decreasing = TRUE),]

```

La tabella contiene varie statistiche riassuntive per i geni top-ranked e il contrasto selezionato:

- la colonna logFC fornisce il valore del contrasto; di solito questo rappresenta un cambiamento di log2 volte tra due o più condizioni sperimentali anche se a volte rappresenta un livello di espressione log2;
- la colonna AveExpr fornisce il livello medio di espressione log2 per quel gene su tutti gli array e canali nell'esperimento;
- la colonna t è la t-statistic moderata;
- la colonna P.Value è il p-value associato alla t-statistic e adj.P.Value è il p-value corretto per test multipli;
- la B-statistic (perdita o B) è la probabilità di log-odds che il gene sia differenzialmente espresso;
- la F-statistic moderata (F) combina la t-statistic per tutti i contrasti in un test globale di significatività per quel gene. La F-statistic verifica se uno qualsiasi dei contrasti è diverso da zero per quel gene, cioè se quel gene è espresso in modo differenziale su qualsiasi contrasto;

Dalle statistiche della top_table è stato calcolato uno score che indica quanto effettivamente i geni siano differenzialmente espressi. A questo punto è possibile scartare queste CpG differenzialmente espresse per poter, così, considerare soltanto quelle CpG con valore assoluto per lo score inferiore a 0.001, ossia quelle CpG che hanno valori simili. Questa operazione ha ridotto ulteriormente il data set, portandolo da 412414 righe a 273785 (top_table2).

***Script 12**: CpG con valore assoluto minore di 0.001.*

```
34
35 #CpG con valore assoluto per lo score di 0.001
36 index_CpG<- which(top_table[,7]<abs(0.001))
37 top_table2<- top_table[index_CpG,]
38
```

L’intera procedura di classificazione è stata effettuata due volte, una volta sull’intero data set producendo la tabella top_table2 e una volta solo su un sottoinsieme del data set, quello composto solo dai campioni affetti dalla malattia, producendo la tabella top_table3. In questo modo è possibile confrontare i dati passo dopo passo.

***Fig. 11**: Piccolo sottoinsieme della tabella top_table2*

	logFC	AveExpr	t	P.Value	adj.P.Val	B	score
cg24710104	4,125471755211 94e-06	0,818131413383744	0,000256479569207275	0,999796212433353	0,999832577563206	-6,79734192105917	8,65128348893377e-05
cg10641714	6,1278941201337e-06	-1,28244377085752	0,000249341565349068	0,999801883582203	0,999835824880302	-6,79734192287219	8,60492174513964e-05
cg16985952	8,154796700354 79e-06	0,689538380 61024	0,000241364251023568	0,999808222410628	0,999839739082445	-6,79734192483785	8,32959361323322e-05
cg09852601	7,834366467673 88e-06	0,032534420 6154621	0,000230257747526065	0,999817047157727	0,999846139636226	-6,79734192746813	7,94626790309291e-05
cg25870263	7,792535470979 58e-06	-0,243333993232256	0,000196199928786404	0,999844108026333	0,999870776807085	-6,7973419347614	6,77083016641321e-05
cg14189808	9,178779371010 75e-07	-0,408471233971945	1,63228511435865e-05	0,999985441455233	0,999987866173759	-6,79734193892263	6,32274168170627e-06
cg08872703	4,999675624527 37e-06	-0,452729663238337	0,000181642509139763	0,999855674721938	0,999879919289759	-6,79734193752341	6,26841954270337e-05
cg11373604	8,161973388154 39e-06	-0,73976174263054	0,00016343610632682	0,999870140741681	0,999891961115505	-6,79734194067821	5,64008214754793e-05
cg23666299	5,061654736460 14e-07	0,679260724909949	1,40201154924736e-05	0,999988860222 82	0,999988860222 82	-6,7973419539925	4,63797070573635e-06
cg12724384	3,772358171838 52e-06	0,294639398622589	0,000110106367000713	0,999912514244792	0,999931910907581	-6,79734194800349	3,79962428211137e-05
cg21752517	1,922782048356 4e-06	-0,955936662621203	7,26622956384614e-05	0,999942265883726	0,999954388870485	-6,79734195143997	2,50744188086189e-05
cg01885291	3,072217771377 12e-06	0,124253347173132	6,59951630714118e-05	0,99994756309879	0,999957261676061	-6,79734195190418	2,27365539385728e-05
cg23746347	5,345479248103 36e-06	-0,067395896616180 5	5,76919063504466e-05	0,999954160507322	0,999961434470629	-6,79734195241989	1,99082950200032e-05
cg11153172	1,094388646705 19e-06	-0,511036197311183	4,48059639158516e-05	0,999964399119647	0,999969248466413	-6,79734195308311	1,54615411120926e-05
cg19301979	5,576326082235 02e-05	0,956345196824874	0,00289233726873311	0,997701877203093	0,996239223431888	-6,79733775333287	0,000999210644085307

Capitolo 4 - Gene Set Enrichment Analysis

L'analisi dell'arricchimento del set di geni è un metodo per identificare classi di geni o proteine che sono sovra rappresentate in un ampio set di geni o proteine e possono avere un'associazione con i fenotipi della malattia. Il metodo utilizza approcci statistici per identificare gruppi di geni significativamente arricchiti o impoveriti. Le tecnologie di trascrittomica ei risultati della proteomica spesso identificano migliaia di geni che vengono utilizzati per l'analisi.

L'analisi dell'arricchimento del set di geni utilizza set di geni a priori che sono stati raggruppati in base al loro coinvolgimento nella stessa via biologica o in base alla posizione prossimale su un cromosoma. In GSEA, i microarray di DNA, o ora RNA-Seq, vengono ancora eseguiti e confrontati tra due categorie di cellule, ma invece di concentrarsi sui singoli geni in un lungo elenco, l'attenzione viene posta su un set di geni. Viene analizzato se la maggior parte dei geni nell'insieme ricade negli estremi di questo elenco: la parte superiore e inferiore dell'elenco corrispondono alle maggiori differenze di espressione tra i due tipi di cellule. Se il set di geni cade in alto (sovraespresso) o in basso (sottoespresso), si pensa che sia correlato alle differenze fenotipiche.

Per pathway si intende un set di geni in cui vengono annotate relazioni di ogni tipo intercorrenti fra due o più entità (proteine, complessi proteici, ormoni, etc.…).

La loro principale classificazione consta di due grandi famiglie di pathways:

1. i pathways di segnale che annotano i processi biologici che definiscono la trasduzione del segnale a livello cellulare;
2. i pathways metabolici che annotano le reazioni chimiche fra componenti finalizzate ai processi di anabolismo o catabolismo della cellula;

Lo scopo della GSEA è quello di individuare, sotto determinate condizioni, i geni differenzialmente espressi (ovvero i geni che reagiscono in maniera diversa a particolari stimoli) ed inoltre studiare le alterazioni di espressione genica di geni appartenenti ad uno specifico pathway per poter determinare se l’espressione differenziale è statisticamente significativa o è dovuta al caso. Infatti, le analisi dei pathway stanno giocando un ruolo sempre più importante nella comprensione del meccanismo biologico, della funzione cellulare e degli stati patologici.

In questo capitolo andremo ad estrarre informazioni rilevanti che consentono di comprendere i meccanismi sottostanti ad un lungo elenco di geni.

4.1 Analisi dei pathway

Nel precedente capitolo si è discusso delle tecniche di data mining e in particolare il capitolo si è chiuso con la classificazione del data set attraverso l'utilizzo di un modello lineare. I due data set risultanti, top_table2 e top_table3, sono costituiti entrambi dalle colonne:

- logFC;
- AveExpr;
- t;
- P.Value;
- Adj.P.Val;
- B;
- Score;

Rispettivamente top_table2 contiene 273785 geni, mentre top_table3 contiene 258568 geni.

Il primo passo, per eseguire l'analisi dei pathway sui due data set sopra descritti, è quello di ottenere un lungo elenco di geni. Il codice è il seguente:

***Script 13**: Dataset IlluminaHumanMethylation450*

```
13
14  #IlluminaHumanMethylation450 Dataset
15  ann <- getAnnotation(IlluminaHumanMethylation450kanno.ilmn12.hg19)
16
```

La funzione restituisce un data set contenente dati di annotazione che si basano sul file "HumanMethylation450_15017482_v.1.2.csv ", in cui ogni riga corrispondente a un loci di metilazione. Infatti, in questa sezione, oltre ad utilizzare il package "IlluminaHumanMethylation450k", tratteremo il pacchetto "missMethyl". Quest'ultimo fornisce funzioni per la normalizzazione e l'analisi della variabilità differenziale e della metilazione differenziale per i dati provenienti da IlluminaHumanMethylation450k. Il data set ottenuto dall'applicazione della funzione getAnnotation() è costituito da 485512 righe e 33 colonne. La colonna su cui ci siamo focalizzati, però, è "UCSC_RefGene_Name", in quanto tale sottoinsieme tenta di selezionare solo un singolo trascritto per gene, ovvero quello più interessante e significativo.

***Script 14**: Selezione del singolo trascritto per gene.*

```
16
17  Geni <- ann$UCSC_RefGene_Name
18  Geni <- as.data.frame(Geni)
19
```

Lo scopo prefissato per questo capitolo è quello di eseguire l'analisi dei pathway sui data set top_table2 e top_table3 risultanti dal capitolo precedente. Utilizzando il codice sottostante si è potuto ottenere una nuova tabella costituita dai geni presenti sia nel data set di partenza, sia nel data set "Geni", risultante dalla precedente operazione.

***Script 15**: Intersezione tra il dataset top_table2/top_table3 con Geni.*

```
35
36  idx2 <- which(Geni$CPG %in% row.names(top_table2))
37  Geni <- Geni[idx2,]
38  Geni2 <- as.data.frame(Geni)
39
```

```
139
140 idx2 <- which(Geni$CPG %in% row.names(top_table3))
141 Geni <- Geni[idx2,]
142 Geni3 <- as.data.frame(Geni)
143
```

***Fig. 12**: Piccolo sottoinsieme della tabella Geni2*

	col_geni	CPG
cg03630821	A1BG	cg03630821
cg03817621	A1CF	cg03817621
cg01723761	A2LD1	cg01723761
cg15384867	A2ML1	cg15384867
cg00241712	A4GALT	cg00241712
cg02299189	AAAS	cg02299189
cg02591564	AACS	cg02591564
cg00261325	AACSL	cg00261325
cg25159668	AADACL2	cg25159668
cg06339629	AADAT	cg06339629
cg16926284	AAGAB	cg16926284
cg01850785	AAK1	cg01850785
cg01851632	AAMP	cg01851632
cg04631389	AANAT	cg04631389
cg01439854	AARS	cg01439854
cg01524091	AARS2	cg01524091

	col_geni	CPG
cg03630821	A1BG	cg03630821
cg15384867	A2ML1	cg15384867
cg00241712	A4GALT	cg00241712
cg02299189	AAAS	cg02299189
cg00261325	AACSL	cg00261325
cg25159668	AADACL2	cg25159668
cg06339629	AADAT	cg06339629
cg16926284	AAGAB	cg16926284
cg01850785	AAK1	cg01850785
cg01851632	AAMP	cg01851632
cg04631389	AANAT	cg04631389
cg01439854	AARS	cg01439854
cg01524091	AARS2	cg01524091
cg04307927	AARSD1	cg04307927
cg08423350	AASDH	cg08423350
cg02453779	AASDHPPT	cg02453779

***Fig. 13**: Piccolo sottoinsieme della tabella Geni3*

A questo punto è stato possibile ricavare i parametri che consentono l'esecuzione del test di ontologia genica per dati di metilazione 450K.

***Script 16**: Ricavo dei parametri per il test di ontologia genica*

```
58
59  sigCpGs1 = rownames(top_table2)
60  sigCpGs1 = sigCpGs1[sigCpGs1 %in% rownames(ann)]
61  AnnotSig1 = ann[sigCpGs1,]
62
```

```
165
166 sigCpGs2 = rownames(top_table3)
167 sigCpGs2 = sigCpGs2[sigCpGs2 %in% rownames(ann)]
168 AnnotSig2 = ann[sigCpGs2,]
169
```

Il test verifica l'arricchimento dell'ontologia genica per CpG significativi dall'array Infinium HumanMethylation450k di Illumina, tenendo conto del diverso numero di sonde per gene presente sull'array. Questa funzione prende un vettore di caratteri di siti CpG significativi, mappa i siti CpG su Entrez Gene ID e verifica l'arricchimento del percorso GO o del percorso KEGG utilizzando un test ipergeometrico, tenendo conto del numero di siti CpG per gene sulla matrice 450K. La funzione sottostante ha permesso di verificare tutta la collezione di pathway da testare, GO (Gene Ontology) o KEGG (Kyoto Encyclopedia of Genes and Genomes), e i tassi di falsa scoperta, che sono stati calcolati utilizzando il metodo di Benjamini e Hochberg.

***Script 17**: Pathways GO e KEGG*

```
72
73  gst_KEGG1 <- gometh(sig.cpg=sigCpGs1, all.cpg=rownames(ann),plot.bias = TRUE,prior.prob = TRUE,collection = "KEGG",sig.genes = TRUE)
74  KEGG_enriched1 <- gst_KEGG1[gst_KEGG1$FDR<0.05,]
75  gst_GO1 <- gometh(sig.cpg=sigCpGs1, all.cpg=rownames(ann), collection = "GO",prior.prob=TRUE, anno = ann,sig.genes = TRUE)
76  GO_enriched1 <- gst_GO1[gst_GO1$FDR<0.05,]
77
```

```
181
182 gst_KEGG2 <- gometh(sig.cpg=sigCpGs1, all.cpg=rownames(ann),plot.bias = TRUE,prior.prob = TRUE,collection = "KEGG")
183 KEGG_enriched2 <- gst_KEGG2[gst_KEGG2$FDR<0.05,]
184 gst_GO2 <- gometh(sig.cpg=sigCpGs1, all.cpg=rownames(ann), collection = "GO",prior.prob=TRUE, anno = ann,sig.genes = TRUE)
185 GO_enriched2 <- gst_GO2[gst_GO2$FDR<0.05,]
186
```

In questo caso, il codice soprariportato ha restituito un data frame per i percorsi KEGG costituito da:

- Description, descrizione del pathway in fase di test;
- N, numero di geni nel percorso KEGG;
- DE, numero di geni differenzialmente metilati;
- P.DE, valore p per la sovra rappresentazione del termine percorso KEGG;
- FDR, falso tasso di scoperta;
- SigGenesInSet, geni presenti nel pathway;

***Fig. 14**: Rappresentazione di alcune righe di KEGG_enriched1*

	Description	N	DE	P.DE	FDR	SigGenesInSet
path:hsa05022	Pathways of neurodegeneration - multiple diseases	457	452	4.920136e-11	1.692527e-08	TANK,FRAT1,CASP12,PPIF,A
path:hsa05200	Pathways in cancer	521	511	2.742262e-09	4.716690e-07	AKT3,BCL2L11,FRAT1,RASGF
path:hsa05010	Alzheimer disease	365	361	5.132165e-09	5.884906e-07	AKT3,FRAT1,CASP12,PPIF,A
path:hsa01100	Metabolic pathways	1460	1392	1.611779e-08	1.386130e-06	NAT2,ADA,ACOT8,GNPDA1,
path:hsa05016	Huntington disease	288	284	8.411350e-07	5.787009e-05	PPIF,ACTR1B,ACTR1A,DNAL
path:hsa05014	Amyotrophic lateral sclerosis	348	341	1.052331e-06	6.033365e-05	TANK,POM121C,HDAC6,CA
path:hsa05012	Parkinson disease	249	246	2.194839e-06	1.078606e-04	PPIF,TRAP1,PSMD14,TUBA1
path:hsa05166	Human T-cell leukemia virus 1 infection	219	217	3.757207e-06	1.615599e-04	AKT3,CDK2,CDK4,CDKN1A,
path:hsa04010	MAPK signaling pathway	294	289	5.094193e-06	1.947114e-04	AKT3,PLA2G4B,RASGRP1,RA
path:hsa04020	Calcium signaling pathway	238	235	6.968306e-06	2.397097e-04	PPIF,TRDN,ADCY1,ADCY2,A
path:hsa04015	Rap1 signaling pathway	210	208	1.022491e-05	3.197607e-04	AKT3,RASGRP2,RAPGEF3,VA
path:hsa04120	Ubiquitin mediated proteolysis	141	141	1.438376e-05	4.123344e-04	UBA2,SAE1,HUWE1,STUB1,
path:hsa04550	Signaling pathways regulating pluripotency of stem cells	141	141	1.818717e-05	4.812605e-04	AKT3,APC2,PCGF3,LEFTY1,F
path:hsa04144	Endocytosis	248	244	2.036838e-05	5.004803e-04	PDCD6IP,ARPC5,ARPC4,ARP
path:hsa04151	PI3K-Akt signaling pathway	341	331	5.832014e-05	1.337475e-03	AKT3,BCL2L11,SGK2,LPAR6,

Mentre per i percorsi GO ha restituito un data frame costituito da:

- Ontology, ontologia a cui appartiene il termine GO;
    1. "BP"- processo biologico;
    2. "CC"- componente cellulare;
    3. "MF" - funzione molecolare;
- Term, descrizione del termine GO;
- N, numero di geni nel percorso GO;
- DE, numero di geni differenzialmente metilati;
- P.DE, valore p per la sovra rappresentazione del termine percorso GO;
- FDR, falso tasso di scoperta;
- SigGenesInSet, geni presenti nel pathway;

***Fig. 15**: Rappresentazione di alcune righe di GO_enriched1*

	ONTOLOGY	TERM	N	DE	P.DE	FDR	SigGenesInSet
GO:0005622	CC	intracellular anatomical structure	14605	13640	4.837698e-156	1.099122e-151	A1BG,NAT2,ADA,CDH2,AKT3,ME
GO:0005515	MF	protein binding	13473	12786	2.792522e-136	3.172165e-132	NAT2,ADA,CDH2,AKT3,MED6,NF
GO:0043229	CC	intracellular organelle	12368	11790	1.113575e-130	8.433107e-127	ADA,CDH2,AKT3,MED6,NR2E3,S
GO:0043226	CC	organelle	13834	13097	3.362636e-128	1.909894e-124	A1BG,ADA,CDH2,AKT3,MED6,NF
GO:0043231	CC	intracellular membrane-bounded organelle	11261	10751	8.523411e-125	3.872867e-121	ADA,CDH2,AKT3,MED6,NR2E3,S
GO:0043227	CC	membrane-bounded organelle	12759	12105	1.440740e-118	5.455363e-115	A1BG,ADA,CDH2,AKT3,MED6,NF
GO:0005737	CC	cytoplasm	11516	10966	2.600871e-115	8.441312e-112	A1BG,NAT2,ADA,CDH2,AKT3,DD
GO:0005488	MF	binding	15619	14837	6.420269e-108	1.823282e-104	NAT2,ADA,CDH2,AKT3,MED6,NF
GO:0071840	BP	cellular component organization or biogenesis	6458	6264	1.123990e-102	2.837325e-99	CDH2,AKT3,CDKN2B-AS1,ACOT
GO:0016043	BP	cellular component organization	6263	6071	2.516467e-96	5.717162e-93	CDH2,AKT3,CDKN2B-AS1,ACOT
GO:0044237	BP	cellular metabolic process	10742	10217	4.282762e-96	8.845502e-93	NAT2,ADA,CDH2,AKT3,MED6,NF
GO:0044260	BP	cellular macromolecule metabolic process	7971	7649	4.024576e-89	7.619529e-86	CDH2,AKT3,MED6,NR2E3,CDKN
GO:0032991	CC	protein-containing complex	5069	4928	3.284697e-63	5.740367e-60	CDH2,MED6,NR2E3,ABI1,KCNE3
GO:1901576	BP	organic substance biosynthetic process	5957	5758	1.668707e-61	2.707954e-78	ADA,MED6,NR2E3,NAALAD2,MI
GO:0009058	BP	biosynthetic process	6039	5834	3.960909e-61	5.999193e-78	ADA,MED6,NR2E3,NAALAD2,MI

Com’è possibile notare dagli script, sia dopo l’analisi dei percorsi KEGG che dei percorsi GO, i valori p risultanti sono stati corretti utilizzando l'aggiustamento del tasso di falsa scoperta (FDR).
In entrambi i casi, i geni differenzialmente espressi sono stati definiti come quelli con FDR < 0,05. Sono state escluse le sonde che mappano su più geni e le sonde che non si mappano su alcun gene. Se un gene veniva preso di mira da più sonde, è stato mantenuto il valore p più basso.

4.2 Risultati

In seguito al test sull'arricchimento dell'ontologia genica, i risultati hanno evidenziato diversi pathway interessanti. Analizzando i valori delle diverse colonne, molti pathway mostrano caratteristiche importanti. Partiamo dal FDR, questo dato risulta molto forte e permette quindi, di effettuare delle interpretazioni su dati statisticamente significativi. Inoltre, ordinando i pathway per numero di geni presenti nel percorso o per numero di geni differenzialmente metilati notiamo che nelle prime posizioni emergono sempre gli stessi pathway, che risultano quindi i più consistenti, importanti e significativi

.

***Fig. 16**: Principali pathway*

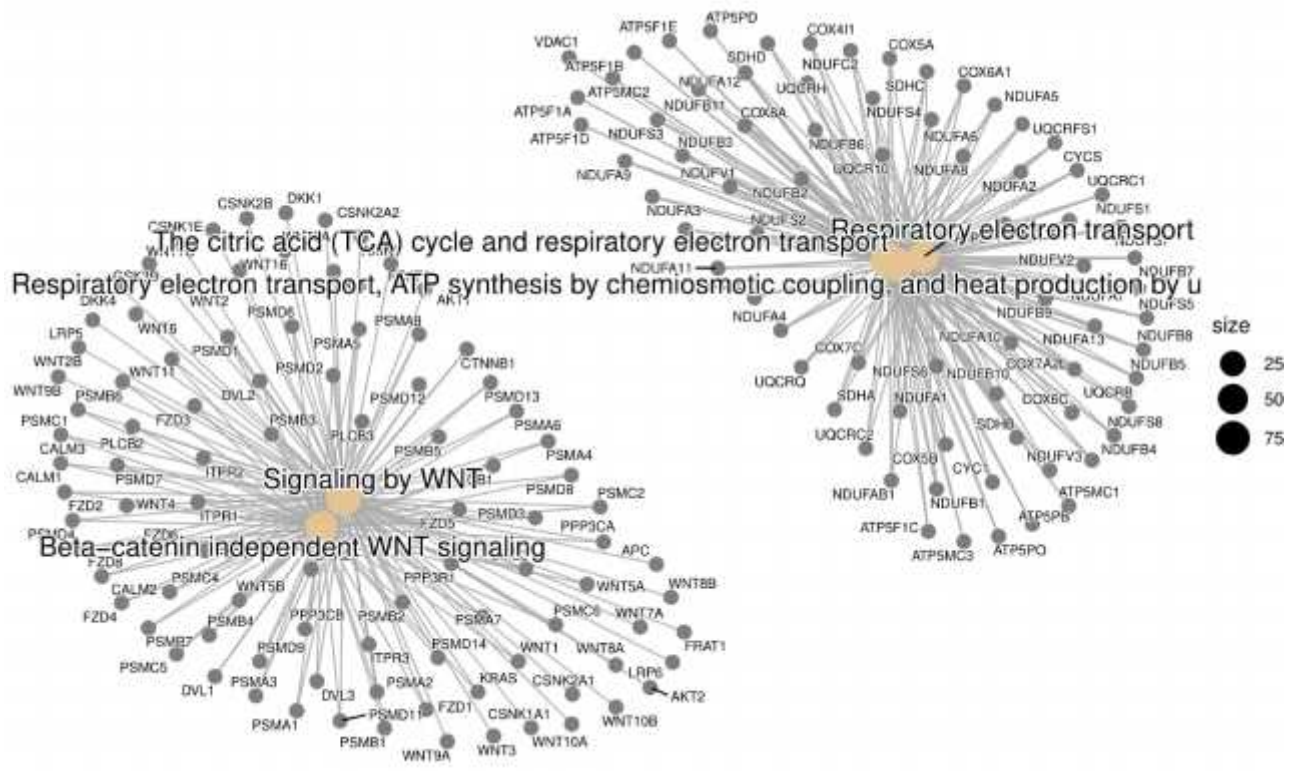
	Description	N	DE	P.DE	FDR
path:hsa05022	Pathways of neurodegeneration - multiple diseases	457	452	4.920136e-11	1.682687e-08
path:hsa05200	Pathways in cancer	521	511	2.742262e-09	4.689268e-07
path:hsa05010	Alzheimer disease	365	361	5.132185e-09	5.850691e-07
path:hsa01100	Metabolic pathways	1460	1392	1.611779e-08	1.378071e-06
path:hsa05016	Huntington disease	266	264	8.411350e-07	5.753364e-05
path:hsa05014	Amyotrophic lateral sclerosis	348	341	1.052331e-06	5.998287e-05
path:hsa05012	Parkinson disease	249	246	2.194839e-06	1.072335e-04

Conclusioni

Come potevamo aspettarci tra i risultati è presente il path “Parkinson Disease”, restano però interessanti anche: “Pathways of neurodegeneration – multiple diseases”, “Alzheimer disease”, “Metabolic pathways” e “Huntington disease”. Vediamo il perché.

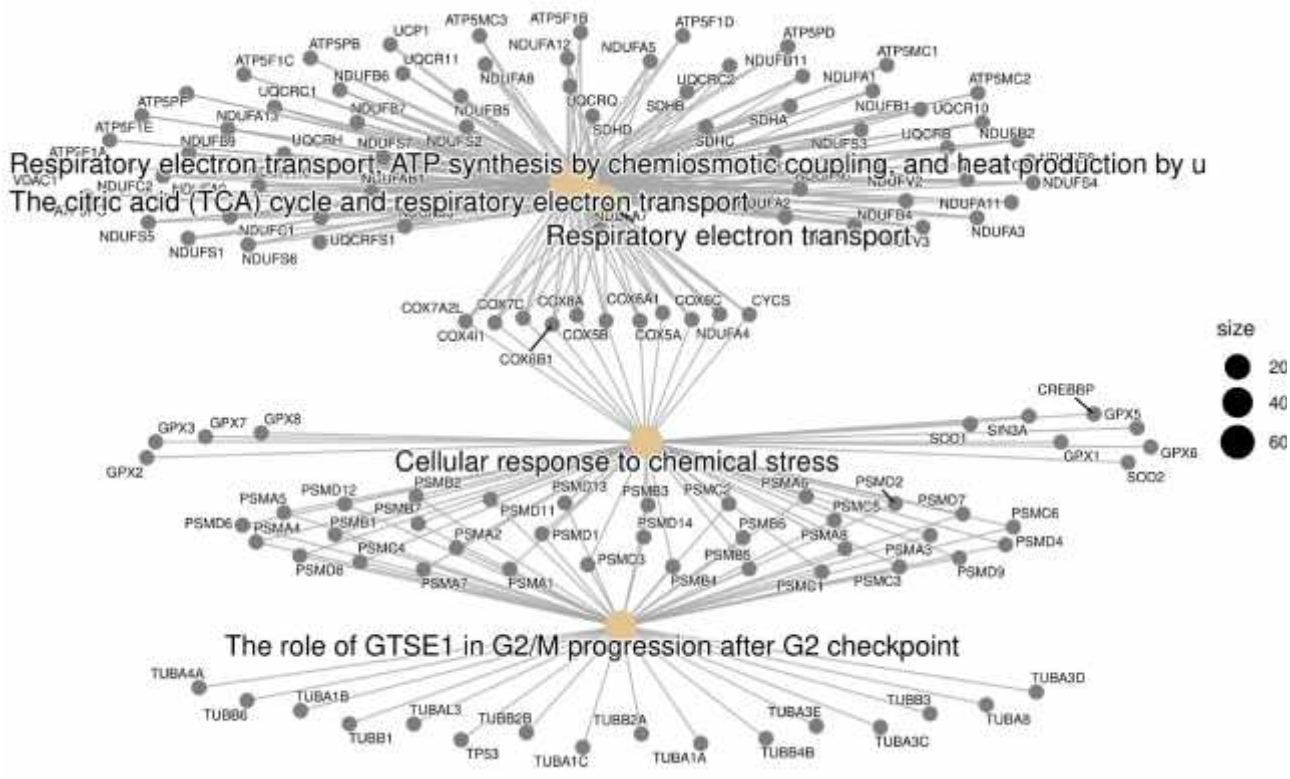
Secondo gli studi più recenti, pur interessando parti del cervello diverse, la malattia di Parkinson e l’Alzheimer potrebbero essere molto simili dal punto di vista biochimico. Difatti sono stati effettuati diversi studi sulle malattie neurodegenerative e sulla loro eziologia. Sia nell'Alzheimer che nel Parkinson, infatti, una proteina sticky (“appiccicosa”) forma degli aggregati tossici nelle cellule cerebrali. Nell'Alzheimer, l'incriminata si chiama Tau, e costituisce gli ammassi neurofibrillari; nel Parkinson, la proteina colpevole è l'alfa-sinucleina, che forma i corpi di Lewy nei neuroni. Due emergenze sociali da troppo tempo in attesa di una soluzione, trattate finora come problemi distinti e che invece potrebbero ottenere una cura proprio grazie alla loro somiglianza.

***Fig. 17**: Rete di collegamenti nel “Alzheimer disease” pathway*



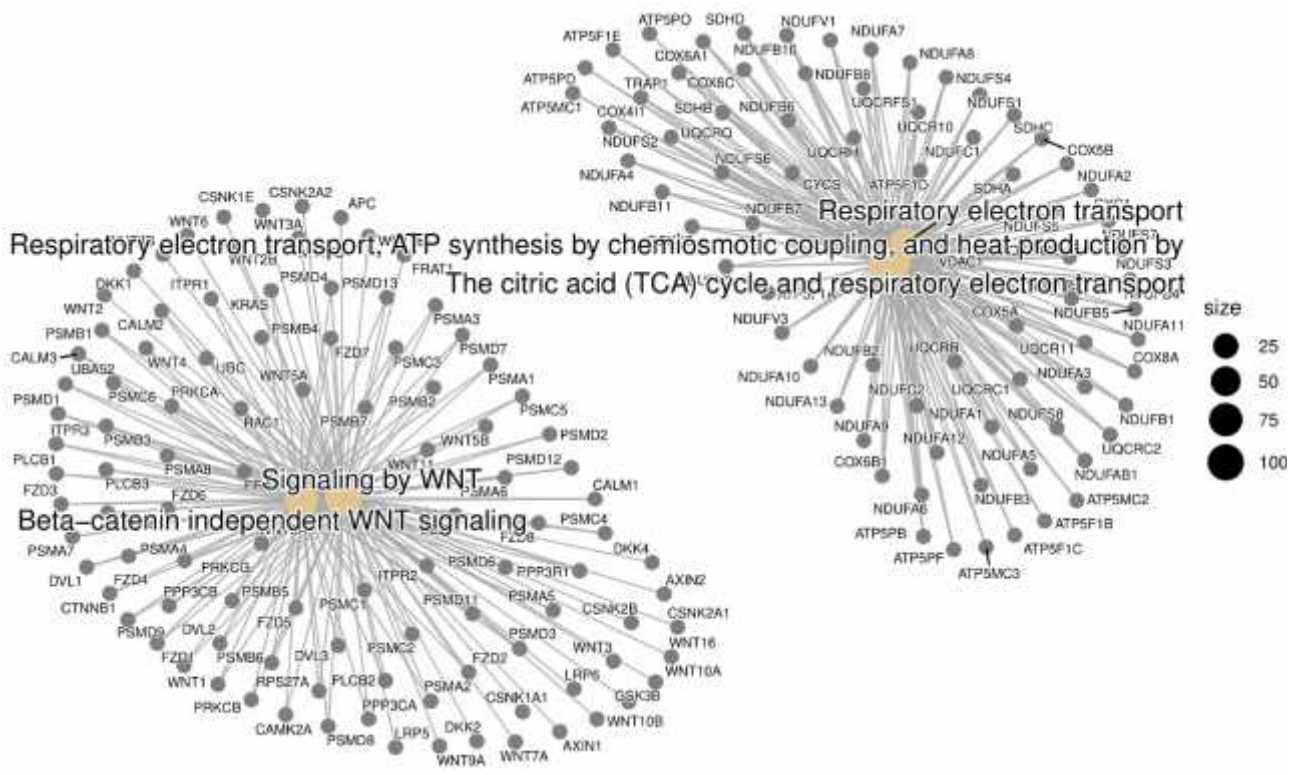
Dal punto di vista patologico, una caratteristica frequente di queste malattie è quindi l'accumulo e l'aggregazione di proteine anormali o mal ripiegate, come con l'amiloide-β (Aβ) e la Tau nella malattia di Alzheimer, l'α-sinucleina nella malattia di Parkinson e la proteina huntingtina nella malattia di Huntington (HD).

***Fig. 18**: Rete di collegamenti nel "Huntington disease" pathway*



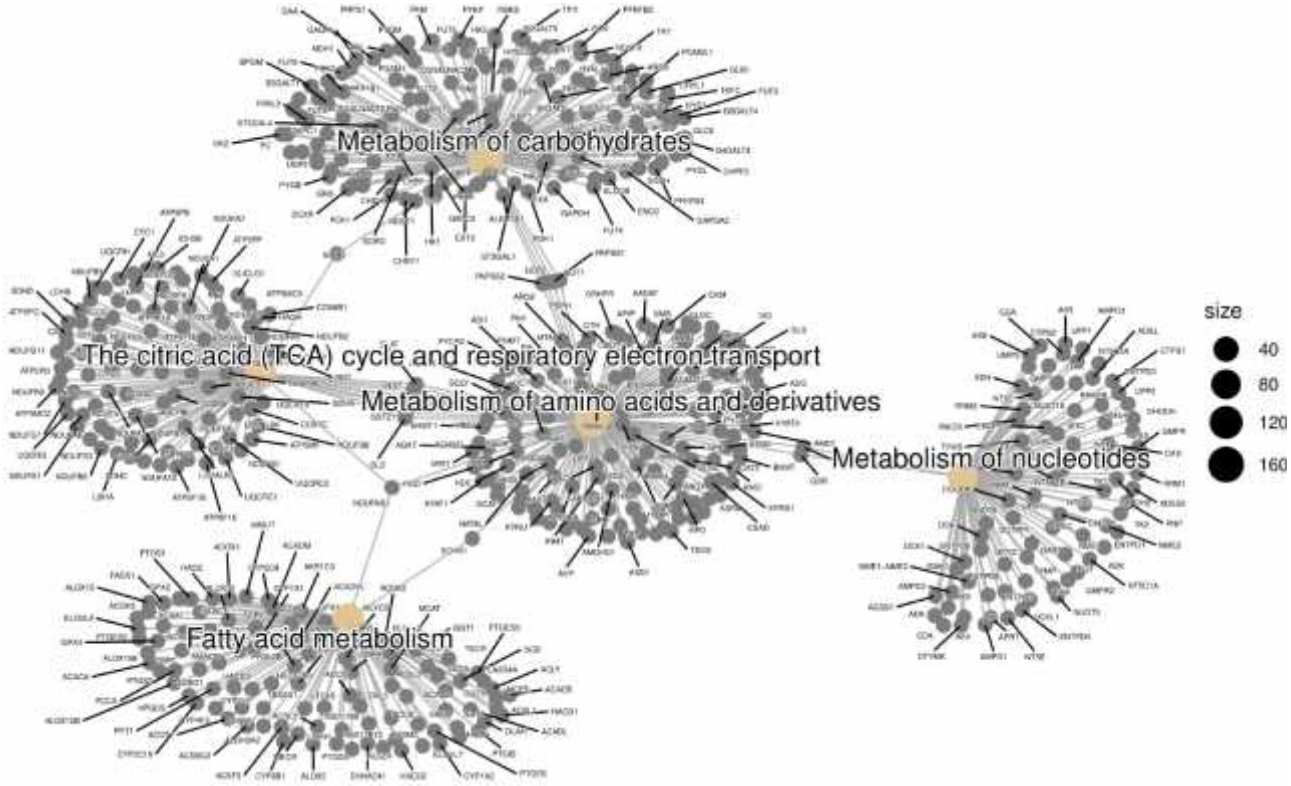
Nonostante si tratti di proteine differenti il loro "modus operandi", nel momento in cui entrano nel cervello, è lo stesso: invadono le vescicole, ovvero tanti comparti racchiusi in membrane, provocano la rottura di tali membrane ed entrano in contratto con il citoplasma, portando alla creazione di disfunzioni celebrali con effetti devastanti, e tristemente noti, sulla vita del paziente. Recentemente, l'analisi dei pathway è diventata un punto focale per lo studio dell'associazione genome-wide. Numerosi studi hanno dimostrato che i geni funzionanti nello stesso percorso possono influenzare collettivamente suscettibilità alle malattie neurodegenerative.

***Fig. 19**: Rete di collegamneti nel "Pathways of neurodegeneration – multiple diseases"*



Inoltre, numerosi studi evidenziano l'associazione tra manifestazioni neurologiche nel PD e nei disordini metabolici ereditari (IMD), che sono malattie genetiche caratterizzate da un'attività carente nelle vie del metabolismo intermedio che portano a manifestazioni multi-sistemiche. I disordini metabolici ereditari sono un gruppo complesso e diversificato di disordini genetici caratterizzati dall'interruzione delle funzioni biochimiche cellulari e da un'attività carente nelle vie del metabolismo intermedio. Il conseguente accumulo di sostanze tossiche o il deficit del prodotto finale porta a un quadro clinico multi-sistemico, dalla disfunzione epatica e dalla cardiomiopatia alla progressione verso l'encefalopatia, il coma o la morte se non trattata.

***Fig. 20**: Rete di collegamenti nel "Metabolic pathways"*



Tra le principali malattie metaboliche abbiamo il diabete di tipo 2 (T2D), una malattia multifattoriale complessa che coinvolge sia fattori genetici che di stile di vita, che portano a un graduale deterioramento della secrezione di insulina dalle cellule pancreatiche e ad un aumento cronico dei livelli di glucosio plasmatico. L'iperglicemia cronica compromette la funzione delle cellule β e quindi ci si può aspettare che contribuisca alla natura progressiva della malattia. Diversi studi hanno dimostrato che nei diabetici più giovani il rischio di contrarre il Parkinson è addirittura aumentato di ben quattro volte mentre nei diabetici con complicanze il rischio aumenta addirittura del 49%.

Insomma, sembra che queste malattie abbiano in comune molto più di quanto ci aspettassimo. Tante malattie neurodegenerative hanno in comune un'alterazione del metabolismo del calcio e conseguente disfunzione, degenerazione e morte delle cellule nervose.
Lo studio portato avanti fino ad ora non basta, ma aver identificato delle similarità tra diverse malattie ci permette di ampliare la ricerca su più fronti comuni, incrociare risultati e sviluppare nuove terapie che potrebbero essere applicate in modo trasversale.

Bibliografia

- Gene Expression Profiling Combined with Bioinformatics Analysis Identify Biomarkers for Parkinson Disease, Hongyu Diao, Xinxing Lil, Sheng Hu, Yunhui Liu.
- Debashis Ghosh, Arul M. Chinnaiyan, Mixture modelling of gene expression data from microarray experiments, Oxford University, 2001;
- Lorenzo Govoni, L'importanza del ridimensionamento dei dati nei problemi di machine learning; https://lorenzogovoni.com/ridimensionamento-dei-dati/
- Martina Collotta, Trascrittomica: la differenza che conta, Rivista Società Italiana di Medicina Generale, 2018; (p. 3).
- Che cosa sono gli SNPs?, Molecular Genetics Laboraties Grouup; http://www.nutrigenetica.it/che-cosa-sono-gli-snps
- Gordon Smyth [et al.], Linear Models for Microarray Dat, RDocumentation,2021;
- Paolo Pozzo, Variabile standardizzata: una guida pratica, Set 2020; https://paolapozzolo.it/variabile-standardizzata-una-guida-pratica
- Lorenzo Govoni, L'importanza del ridimensionamento dei dati nei problemi di machine learning; https://lorenzogovoni.com/ridimensionamento-dei-dati/
- Usama Fayyad, Gregory Piatetsky- Shapiro e Padhraic Smyth, The KDD process for extracting useful knowledge from volumes of data, Communications of the ACM, 1996;
- Una categorizzazione dei principali metodi di Clustering, Il Clustering, https://studylibit.com/doc/733356/13-il-clustering
- Debashis Ghosh, Arul M. Chinnaiyan, Mixture modelling of gene expression data from microarray experiments, Oxford University, 2001;
- Marco Frasca, Data Mining and Machine Learning in Lab. Lezione 7 Master in Data Science for Economics, Business and Finance, 2018;
- Alessandra Amendola, Cosimo Vitale, Modelli non lineari e previsioni in tempo reale, Quaderni di Statistica, 2000;

RDocumentation

- https://www.rdocumentation.org/packages/sva/versions/3.20.0/topics/ComBat
- https://www.rdocumentation.org/packages/limma/versions/3.28.1/topics/normalizeQuantiles
- https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/hclust
- https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/kmeans
- https://www.rdocumentation.org/packages/limma/versions/3.28.14/topics/lmFit
- https://www.rdocumentation.org/packages/limma/versions/3.28.14/topics/ebayes
- https://www.rdocumentation.org/packages/minfi/versions/1.18/topics/getAnnotation
- https://www.rdocumentation.org/packages/missMethyl/versions/1.6.2/topics/gometh
- https://www.rdocumentation.org/packages/DOSE/versions/2.10.6/topics/cnetplot
- https://www.rdocumentation.org/packages/ReactomePA/versions/1.16.2/topics/enrichPathway