CLASSIFICAZIONE DI PRODOTTI ITTICI FRESCHI/DECONGELATI TRAMITE IDENTIFICAZIONE ROBUSTA DI MARCATORI METABOLICI

RELATORE: Ch.ma Prof.ssa Barbara Di Camillo

CORRELATORE: Dott. Roberto Piro

LAUREANDO: Alberto Giraldo

A.A. 2018-2019

UNIVERSITÀ DEGLI STUDI DI PADOVA
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
TESI DI LAUREA MAGISTRALE IN BIOINGEGNERIA

CLASSIFICAZIONE DI PRODOTTI ITTICI FRESCHI/DECONGELATI TRAMITE IDENTIFICAZIONE ROBUSTA DI MARCATORI METABOLICI

RELATORE: Ch.ma Prof.ssa Barbara Di Camillo

CORRELATORE: Dott. Roberto Piro

LAUREANDO: *Alberto Giraldo*

Padova, 9 dicembre 2019

Indice

Capitolo 1

Introduzione

Razionale

La possibile vendita di prodotti scongelati per freschi è una problematica fortemente avvertita dal consumatore, e nel contempo una pratica potenzialmente penalizzante soprattutto per la produzione nazionale e locale. Tale tipologia di frode si presta ad essere applicata soprattutto a prodotti ittici costituiti da pesci con carni magre (sogliola, orata, dentice) di notevole pregio commerciale che subiscono minori alterazione dal congelamento, e a cefalopodi, solitamente venduti scongelati, per i quali il requisito di freschezza è particolarmente apprezzato anche perché sinonimo di "prodotto locale".

Oltre ad essere una frode in commercio, il prodotto decongelato rappresenta un potenziale pericolo per il consumatore in quanto la sua vita commerciale ridotta potrebbe portare ad un più rapido deperimento e dunque ad un aumento del rischio batteriologico o chimico.

Una delle più comuni malattie causate dal pesce e dai prodotti ittici (in particolare lo sgombro, ma anche tonno, sardine, etc.) mal conservati o conservati troppo a lungo è l'intossicazione da istamina o sindrome sgombroide. L'istamina è una sostanza di origine naturale (prodotta anche dal nostro organismo), inodore e incolore. Il livello di tale sostanza nei prodotti ittici dipende dalla quantità di amminoacidi presenti e *aumenta con il diminuire della freschezza del*

prodotto. La maggior parte viene prodotta dalla proliferazione di batteri a temperature superiori ai 6-10°C. Se assunta in dosi elevate può essere pericolosa e provocare, nei casi più gravi, shock istaminico seguito da ipotensione e collasso cardio-circolatorio [1].

Le metodologie analitiche attualmente in uso per distinguere le differenti tipologie di prodotto e rilevare le frodi sono visive e strumentali, ma caratterizzate da una scarsa sensibilità e da una difficile valutazione ed interpretazione, spesso delegata all'esperienza dell'operatore. Inoltre la loro efficacia dipende dalla tipologia di prodotto ittico e dal processo di congelamento e scongelamento utilizzato. Le metodologie di analisi più diffuse, in gran parte applicate in passato, sono:

- Pesci: valutazione delle pinne; GR Torrymeter; prova dell'emolisi; opacamento del cristallino; attività enzimatica della b-idrossiacil-CoA-deidrogenasi (HADH) o della succinico-deidrogenasi (SDH); NIR; prova istologica
- Molluschi cefalopodi: valutazione della sepiomelanina; brillantezza dei colori della livrea, cristallino limpido e trasparente; attività enzimatica HADH
- Crostacei: valutazione dell'integrità di arti ed antenne; attività enzimatica HADH

Scopo del lavoro

Lo scopo di questo progetto è la realizzazione di un classificatore automatico robusto, semplice da usare e rapido, basato su dati metabolici da spettrometria di massa. In particolare il classificatore sfrutterà in parte algoritmi utilizzati in genomica, appositamente adattati a questa specifica applicazione.

Originalità della proposta

Allo stato attuale gli Istituti Zooprofilattici Sperimentali (IZS) non dispongono di metodi rapidi ed efficaci per la distinzione tra prodotti freschi e prodotti con-

[1] https://www.salepepesicurezza.it/pesce-istamina/

gelati/decongelati, che permettano di rispondere alle richieste di monitoraggio espresse sia dagli organismi di controllo locali che dal Ministero della Salute e dalla Comunità Europea. L'unico ad ora disponibile è il metodo istologico, che però presenta diversi limiti. Infatti è un procedimento lungo (un esame può richiedere anche una settimana), richiede una preparazione specifica del campioni e la sua efficacia dipende dal particolare prodotto ittico (non funziona, ad esempio, su cefalopodi e molluschi) [13].

L'IZS delle Venezie ha già una importante esperienza sull'argomento, avendo avuto modo di utilizzare nel tempo i metodi sopra indicati, in particolare il test istologico.

Metodologia applicata

I campioni analizzati in questo lavoro sono forniti dall'IZSVe in collaborazione con il Centro di Referenza Nazionale per le malattie dei pesci, molluschi e crostacei. Il database con i risultati dello spettrometro di massa è stato fornito dal dott. Roberto Piro, responsabile del Laboratorio di Chimica sperimentale dell'IZSVe. Per maggiori dettagli sulla raccolta e trattamento dei campioni biologici si rimanda al capitolo 3.1.

Per l'analisi preliminare del database è stato utilizzato MetaboAnalyst, un insieme di 12 moduli (basati su R) che permettono l'analisi statistica, funzionale e integrativa di dati metabolici. Per maggiori informazioni si rimanda al sito ufficiale[2] che ospita una istanza del tool usabile gratuitamente e si consiglia la lettura del paper di presentazione della nuova versione di MetaboAnalyst [14].

R è un linguaggio di programmazione object-oriented e un ambiente di sviluppo specifico per l'analisi statistica dei dati, distribuito con licenza GNU/GPL e disponibile per i principali sistemi operativi. Per maggiori informazioni su R si

[2]https://www.metaboanalyst.ca/

rimanda al sito ufficiale[3]. Anche il classificatore vero e proprio è stato sviluppato interamente in R.

[3]https://www.r-project.org/about.html

Capitolo 2

Spettrometria di massa

2.1 Concetti base

La spettrometria di massa è una tecnica analitica distruttiva basata sulla ionizzazione di una molecola (in fase gassosa o liquida) e la sua successiva frammentazione in ioni di diverso rapporto massa/carica (m/z). E' proprio grazie al diverso rapporto m/z che gli ioni vengono discriminati e rivelati da un detector. L'esperimento di spettrometria consiste principalmente in 3 fasi:

- Ionizzazione delle molecole in fase gassosa
- Separazione dei diversi ioni prodotti
- Rivelazione degli ioni prodotti

In figura 2.1 è riportato lo schema generico di uno spettrometro.

La spettrometria di massa non richiede nessuna conoscenza a priori del campione né una particolare pre-selezione del campione (cosa necessaria, ad esempio, per tecniche basate su label radioattive)[1]. Il risultato dell'esperimento è lo spettro di massa, che rappresenta l'abbondanza relativa degli ioni in funzione del loro rapporto m/z. Ogni molecola avrà una "firma" caratteristica nello spettro che dipende sia dal settaggio delle varie parti dello spettrometro (soprattutto lo io-

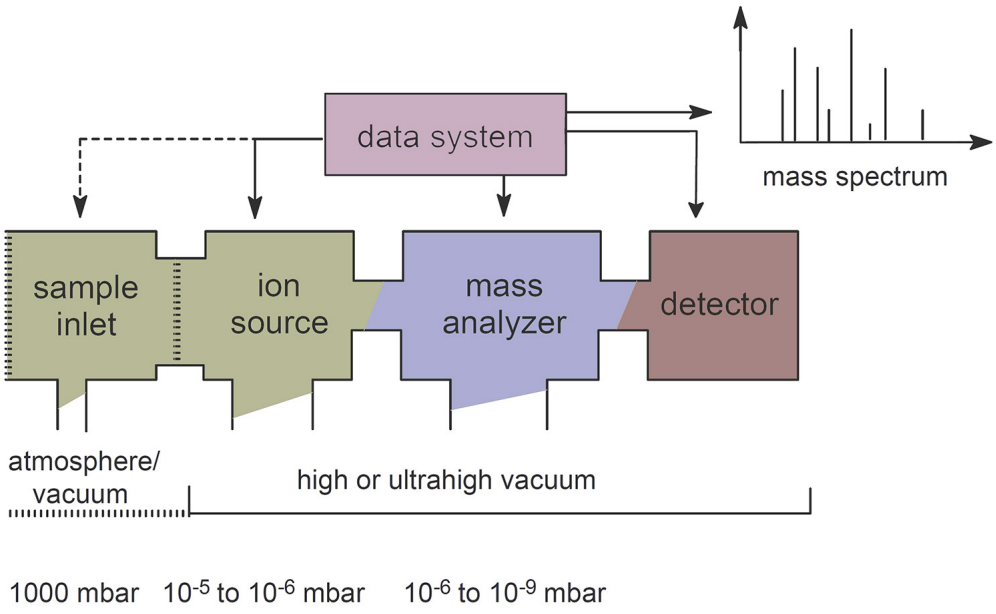


Figura 2.1: Schema di funzionamento spettrometro di massa
Fonte: http://ms-textbook.com/for-instructors/

nizzatore) sia dalla natura stessa della molecola. Nel seguito verranno descritte accuratamente le tre fasi principali di un esperimento di spettrometria.

2.1.1 Ionizzazione

Il campione viene introdotto nella camera di ionizzazione che molto spesso è sottovuoto. Questo può avvenire sia allo stato solido, usando una sonda, che allo stato liquido o gassoso, usando un sistema di valvole. Alcuni metodi di analisi (utili quando si devono analizzare miscele di prodotti) utilizzano come ingresso allo spettrometro l'uscita di un gascromatografo o un cromatografo liquido ad alta penetrazione, ottenendo rispettivamente le tecniche GC-MS e HPLC-MS. Esistono varie tecniche di ionizzazione, le più diffuse sono riportate schematicamente di seguito.

Ionizzazione elettronica (EI)

Tecnica usata prevalentemente quando si devono analizzare molecole organiche "leggere" (peso molecolare minore di 600) che possono essere facilmente rese in

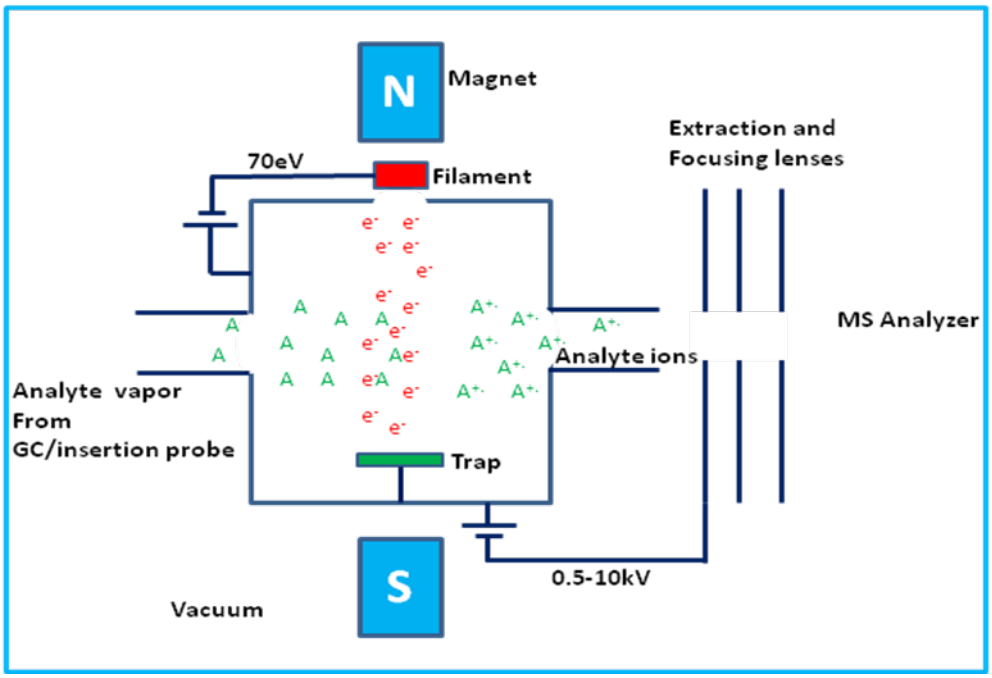


Figura 2.2: Schema di ionizzazione elettronica
Fonte: https://www.chem.pitt.edu/facilities/mass-spectrometry/mass-spectrometry-introduction

fase gassosa senza decomposizione (volatili). Ciò avviene per desorbimento termico, per questo devono essere anche termicamente stabili [9]. Le molecole in fase gassosa entrano nella sorgente ionica dove sono bombardate con elettroni, causando una ionizzazione *hard*[1]. In figura 2.2 è riportato uno schema di ionizzazione elettronica.

Ionizzazione chimica (CI)

Tecnica particolarmente utile quando non si osserva alcuno ione molecolare nello spettro di massa EI di un composto e per confermare il peso molecolare di un composto. Anche in questo caso va usata con composti piccoli (peso < 600Da), volatili e termostabili [11]. In figura 2.3 è riportato lo schema di funzionamento della ionizzazione chimica.

Un gas reagente G subisce ionizzazione elettronica diventando ione radicalico G^+. Questo reagisce con molecole neutre di gas e diventa gas reagente proto-

[1] un metodo di ionizzazione è detto hard se causa la frammentazione della molecola

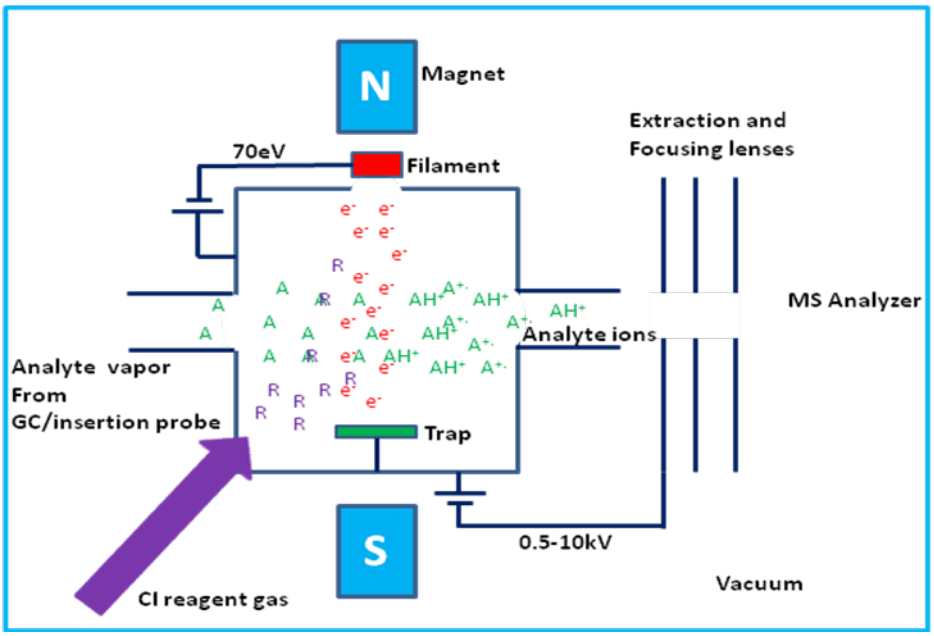


Figura 2.3: Schema di ionizzazione chimica
Fonte: https://www.chem.pitt.edu/facilities/mass-spectrometry/mass-spectrometry-introduction

nato (acido) GH^+. Quando viene introdotto il campione, le molecole neutre del campione M ricevono il protone dagli ioni GH^+ e diventano ioni MH^+:

$$GH^+ + M \rightarrow MH^+ + G \tag{2.1}$$

Dall'equazione si può apprezzare una particolarità della ionizzazione chimica, ovvero che nello spettro di massa lo ione molecolare sarà rappresentato con una unità di massa in più (avendo acquisito un protone).

A differenza del metodo EI, il processo CI è una ionizzazione *soft*[2] e produce abbondanti ioni quasi molecolari. Questo è dovuto al fatto che nella CI le molecole del campione sono ionizzate tramite reazioni ione-molecola con un trasferimento di energia nettamente inferiore rispetto a quella degli elettroni nella EI.

In generale, le molecole di gas reagente sono presenti in quantità elevate, circa 100: 1 rispetto alle molecole del campione. Come gas reagenti si usano spesso idrogeno, metano, isobutano e ammoniaca, caratterizzati da alta affinità protonica.

[2]Un metodo di ionizzazione è detto soft se non causa la frammentazione delle molecole

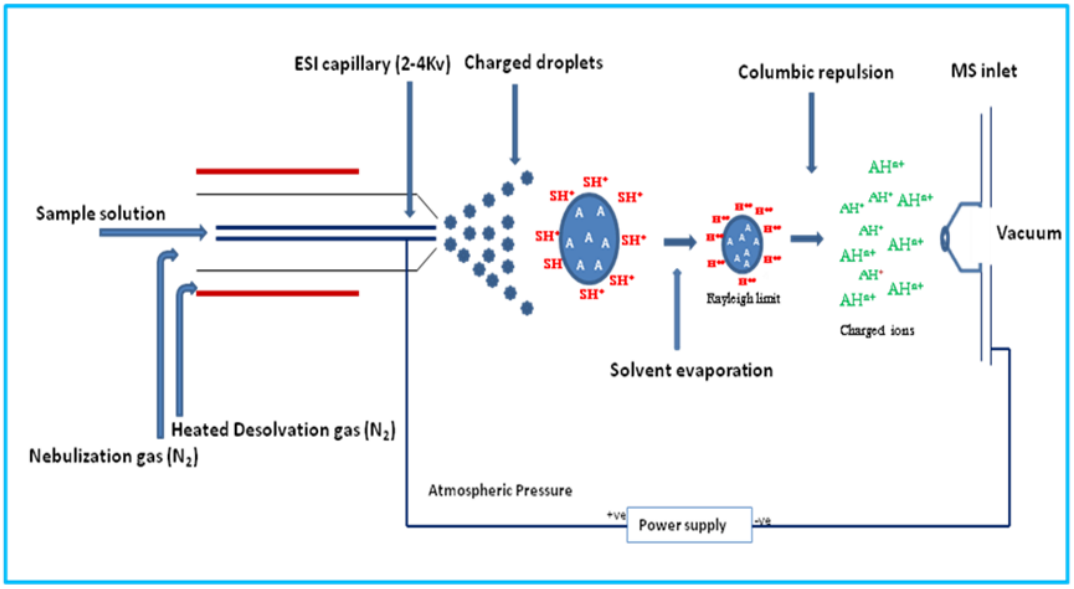


Figura 2.4: Schema di ionizzazione elettrospray

Fonte: https://www.chem.pitt.edu/facilities/mass-spectrometry/mass-spectrometry-introduction

Ionizzazione elettrospray (ESI)

La tecnica ESI consiste nello spruzzare la soluzione del campione attraverso un ago altamente carico chiamato capillare ESI che è a pressione atmosferica (vedi figura 2.4).

ESI può produrre ioni a carica singola o multipla. Il numero di cariche dipende da diversi fattori come le dimensioni, la composizione chimica della molecola dell'analita, la composizione del solvente, la presenza di co-soluti e i parametri dello strumento. Per le piccole molecole ($<$ 2000 Da) l'ESI genera in genere ioni con una,due o tre cariche, mentre per le molecole grandi ($>$ 2000 Da) l'ESI può produrre una serie di ioni a carica multipla.

ESI è molto adatto per una vasta gamma di composti biochimici tra cui peptidi e proteine, lipidi, oligosaccaridi, oligonucleotide, composti bioorganici e polimeri sintetici [9].

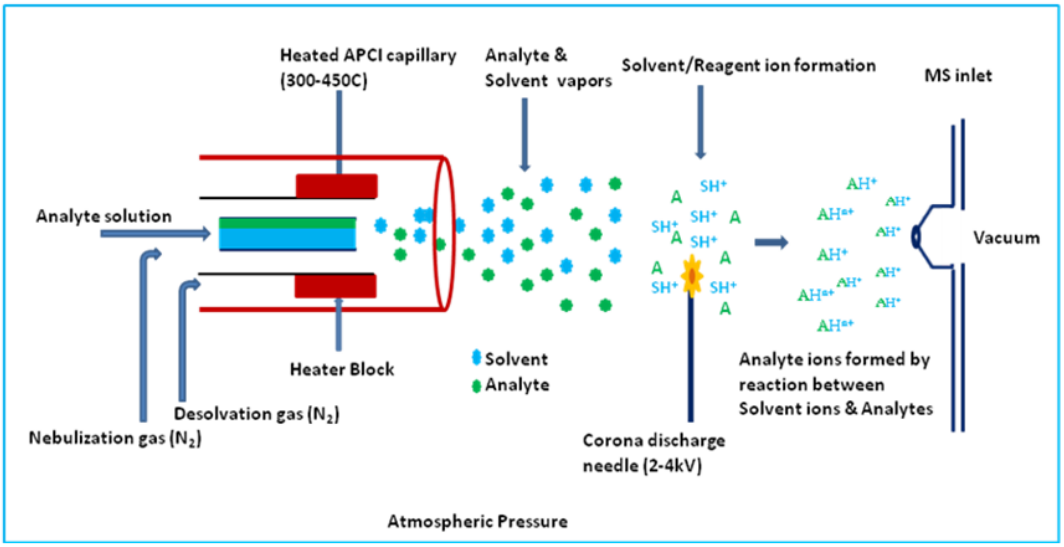


Figura 2.5: Schema di ionizzazione ACPI

Fonte: https://www.chem.pitt.edu/facilities/mass-spectrometry/mass-spectrometry-introduction

Ionizzazione chimica a pressione atmosferica (ACPI)

La particolarità di questo metodo di ionizzazione é che genera ioni direttamente dalla soluzione ed è in grado di analizzare composti relativamente non polari. Simile all'elettrospray, l'effluente liquido di APCI viene introdotto direttamente nella sorgente di ionizzazione attraverso la sonda APCI (vedi figura 2.5).

La soluzione del campione subisce la nebulizzazione e viene rapidamente riscaldata in un flusso di gas di azoto (gas di desolvatazione) prima di uscire dalla sonda. Gli ioni solvente/reagente si formano nella punta dell'ago tramite effetto corona. Questi ioni reagiscono con molecole di analiti formando ioni di analiti protonati o deprotonati caricati singolarmente [9].

In generale, il trasferimento di protoni avviene in modalità positiva per produrre $[A + H]^+$ ioni. Nella modalità ioni negativi, si verifica il trasferimento di elettroni o la perdita di protoni per produrre ioni M^-. o $[A - H]^-$, rispettivamente. Grazie alla presenza dei solventi e all'alta pressione del gas la frammentazione durante la ionizzazione è ridotta e si generano principalmente ioni quasi-molecolari. La carica multipla non viene generalmente osservata perché il processo di ionizzazione è più energetico di ESI.

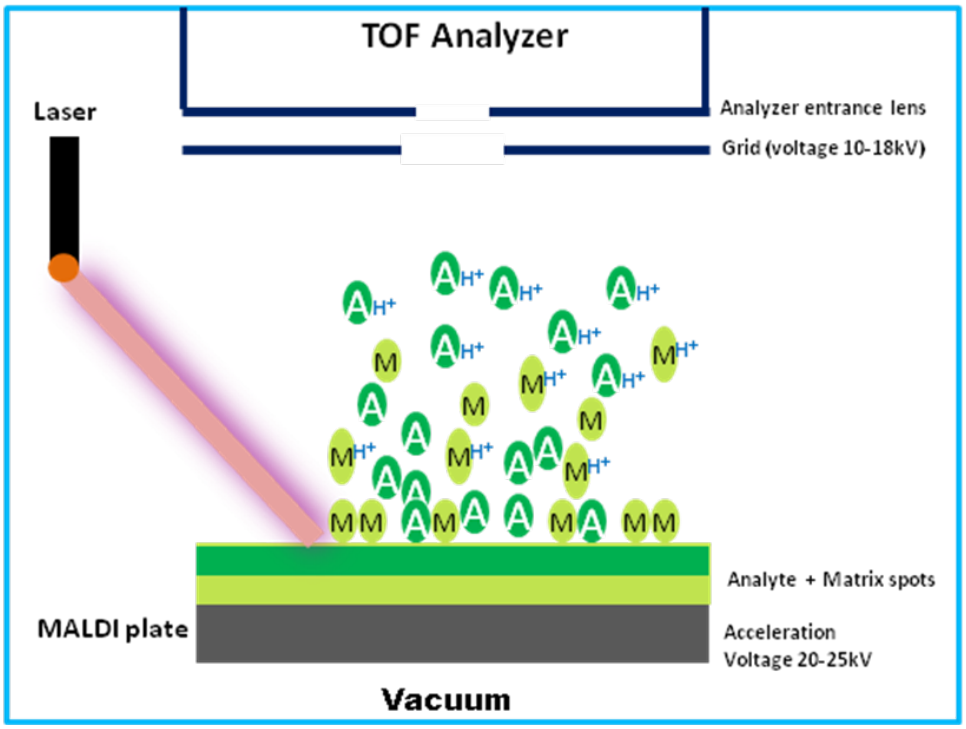


Figura 2.6: Schema ionizzazione MALDI

Fonte: https://www.chem.pitt.edu/facilities/mass-spectrometry/mass-spectrometry-introduction

Ionizzazione MALDI

Il desorbimento/ionizzazione laser assistito da matrice (MALDI) è una tecnica che consente di generare ioni intatti in fase gassosa di composti ad alto peso molecolare come macromolecole organiche. MALDI è uno dei recenti sviluppi delle tecniche di ionizzazione "soft" nel campo della spettrometria di massa (è usata con successo a partire dal 1990 circa) . Può desorbire ioni molecolari di analiti intatti con masse relative fino a 300KDa. Dopo aver fatto adsorbire il campione su di una matrice realizzata in vari materiali, specialmente organici (tipo glicerolo, acido caffeico, ecc.), questa viene portata in soluzione e successivamente bombardata con un fascio laser (spesso un laser ad azoto). In figura 2.6 è riportato lo schema del processo. Questo causa il de-adsorbimento del campione che viene rilasciato in forma "clusterizzata", ovvero complessato con la matrice. La matrice smorza gli effetti del fascio laser proteggendo l'analita che viene ionizzato e vaporizzato successivamente tramite l'energia in eccesso ceduta dalla matrice

stessa. Vengono così ottenuti ioni quasi molecolari generalmente a singola carica, come quelli creati dall'acquisizione o dalla perdita di un protone. Molto spesso la tecnica MALDI viene abbinata a spettrometri dotati di analizzatore a tempo di volo (TOF).

2.1.2 Separazione dei diversi ioni

Una volta ottenuti gli ioni, questi vengono separati tramite l'analizzatore in base al loro rapporto m/z. Di seguito sono riportate brevi descrizioni degli analizzatori più diffusi.

Analizzatore magnetico

Grazie ad un campo magnetico è possibile far compiere agli ioni traiettorie circolari, il cui raggio dipende dal rapporto m/z dello ione. Cambiando le traiettorie degli ioni mediante variazioni del campo magnetico applicato, ioni con diverso rapporto m/z possono essere focalizzati sul rivelatore. Come si vede in figura 2.7 è costituito da un tubo lungo circa 1 metro, piegato con un raggio di curvatura r' ed immerso in un campo magnetico di intensità B.

Quando gli ioni che escono dalla camera di ionizzazione entrano nel tubo analizzatore, per effetto del campo magnetico B subiscono una deviazione (deflessione) dalla loro traiettoria rettilinea che viene curvata. All'uscita della camera di ionizzazione il fascio di ioni è accelerato attraverso un potenziale V di 6000 - 8000 volt. Gli ioni vengono espulsi, attraverso una fenditura di uscita, con circa la stessa energia cinetica pari a:

$$E_{cinetica} = \frac{1}{2}mv^2 = zV \tag{2.2}$$

con z la carica degli ioni, V il potenziale applicato e m e v rispettivamente massa e velocità dello ione. L'accelerazione che gli ioni subiscono è proporzionale al potenziale V delle piastre acceleratrici. Quando gli ioni entrano nel campo magnetico B subiscono una forza centripeta che tende a far loro percorrere una

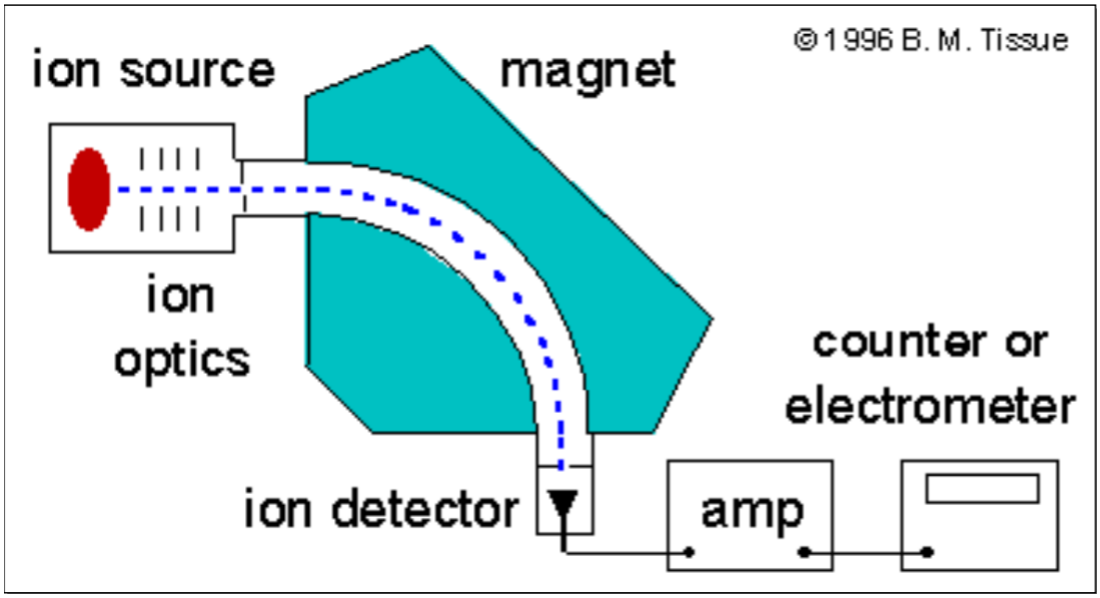


Figura 2.7: Schema analizzatore magnetico
Fonte: http://www.ecs.umass.edu/cee/reckhow/courses/772/slides/772l21p.pdf

traiettoria circolare di raggio r :

$$F_{centripeta} = Bzv$$

bilanciata da quella centrifuga

$$F_{centrifuga} = mv^2/r$$

Eguagliando e semplificando le due equazioni appena trovate si ottiene

$$Bz = \frac{mv}{r} \tag{2.3}$$

La nuova traiettoria curvilinea ha un raggio di curvatura r che è direttamente proporzionale alla quantità di moto dello ione mv e inversamente proporzionale al campo magnetico B. Ricavando v^2 dall'eq. 2.2 e sostituendolo nell'eq 2.3 elevata al quadrato si ottiene l'equazione fondamentale dell'analizzatore magnetico [10]:

$$\frac{m}{z} = (\frac{B^2}{2V}) \tag{2.4}$$

Dall'equazione 2.4 si capisce la forza dell'analizzatore magnetico: per ogni coppia B e V esiste un solo rapporto $\frac{m}{z}$ per cui $r = r'$. Nella pratica V è costante, mentre scansionando per valori di campo magnetico diversi (a parità di campo elettrico accelerante) è possibile far uscire ioni diversi a tempi diversi [6].

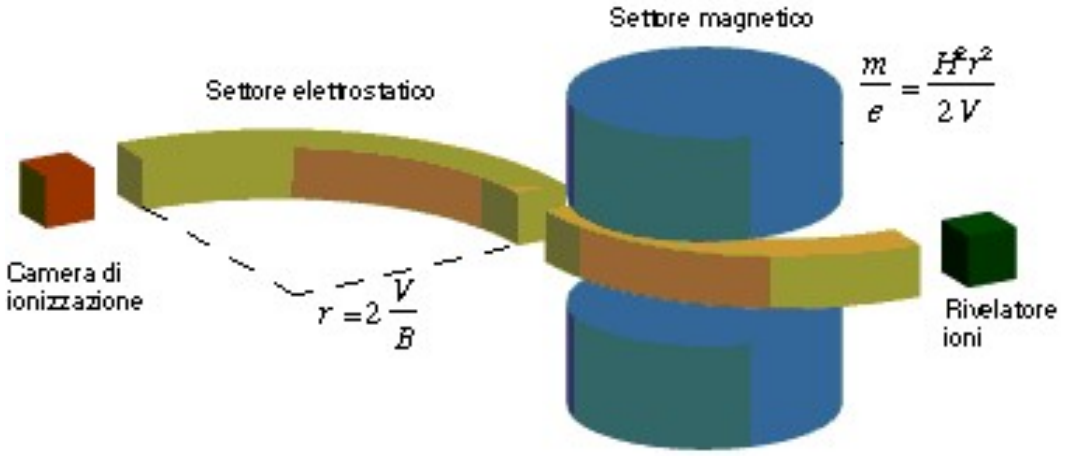


Figura 2.8: Analizzatore a doppia focalizzazione
Fonte: https://www.vialattea.net/content/799/

Analizzatore a doppia focalizzazione

E' l'analizzatore che consente di ottenere le risoluzioni migliori[3]. Aggiungendo prima dell'analizzatore magnetico un analizzatore elettrostatico (ESA) il percorso degli ioni positivi viene focalizzato ulteriormente in direzione dal campo elettrico statico (vedi figura 2.8). Questi analizzatori possono separare ioni che hanno la stessa massa nominale ma che hanno diversa formula bruta. Lo svantaggio è che sono molto costosi, quindi il loro impiego non è molto diffuso. Nel settore elettrostatico (ES) gli ioni non vengono separati in funzione del rapporto massa/carica, ma solo focalizzati in base alla loro **energia traslazionale**. Gli ioni generati nella camera di ionizzazione possono essere dotati di energia cinetica iniziale diversa da 0. L'energia cinetica totale di uno ione dopo accelerazione nel campo V sarà quindi la somma di due componenti: energia cinetica iniziale ed energia cinetica guadagnata durante l'accelerazione. Il settore elettrostatico si limita ad uniformare le energie translazionali degli ioni che hanno uguale m/z, compensando differenze di velocità iniziale; Se così non fosse, nel settore successivo, quello magnetico, ioni con ugual rapporto m/z ma differente energia traslazionale

[3]In termini di **massa esatta**, ovvero con precisione fino alla quarta cifra decimale

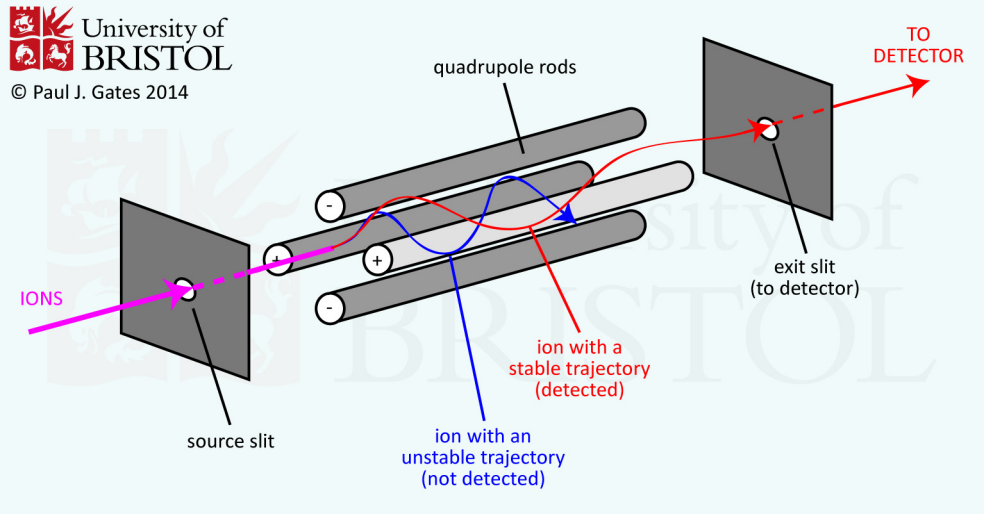


Figura 2.9: Schema analizzatore quadrupolo
Fonte: http://www.chm.bris.ac.uk/ms/quadrupole.xhtml

seguirebbero traiettorie diverse, diminuendo la risoluzione dello strumento.

Analizzatore a filtro di massa quadrupolo

E' costituito schematicamente da quattro barre metalliche parallele. In questo dispositivo la separazione degli ioni dipende dal moto degli ioni risultante dall'applicazione di una combinazione di campi elettrici continui (DC) e alternati a radiofrequenza (RF). Alle barre opposte del quadrupolo è applicata una differenza di potenziale, generata da una corrente continua ed alternata. In particolare a una coppia di barre (opposte) viene applicato un potenziale $U + V\cos(\omega t)$, mentre alle restanti due viene applicato un potenziale $-(U + V\cos(\omega t))$, dove U è un voltaggio continuo (DC) e V un voltaggio alternato (AC). Si veda per maggior chiarezza le figure 2.9 e 2.10.

Gli ioni (positivi), accelerati dalle piastre acceleratrici, entrano nel tunnel delimitato dalle barre e vengono respinti dai poli positivi ed attratti dai negativi. Il potenziale elettromagnetico oscillante delle barre, fa in modo che quando le due sbarre verticali hanno potenziale positivo quelle orizzontali l'hanno negativo, e viceversa. A causa dell' oscillazione del potenziale, quando gli ioni entrano in questo campo percorrono una traiettoria a zig zag oscillando nelle direzioni x e

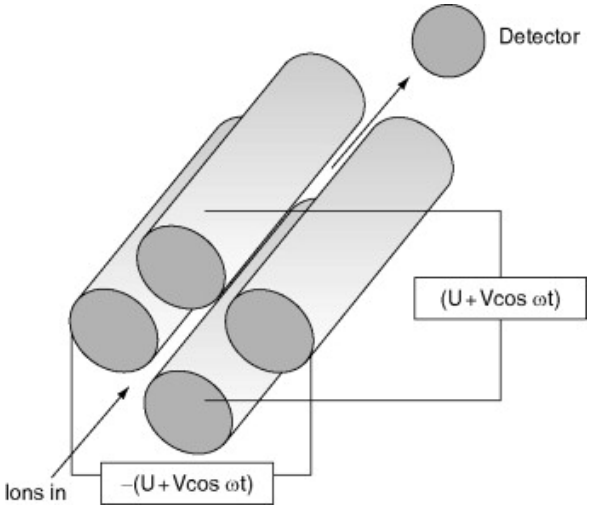


Figura 2.10: Potenziali applicati al quadrupolo

Fonte:
https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/quadrupole-mass-analyzer

y (determinate dalle perpendicolari alle barre). L'ampiezza di tali oscillazioni dipende dalla frequenza del potenziale applicato e dalle masse degli ioni.

Le oscillazioni che gli ioni subiscono nel loro transito potranno essere:

- stabili (moto diventa sinusoidale), permettendo così allo ione di uscire dal quadrupolo ed entrare nel sistema di rivelazione
- instabili e porteranno alla collisione dello ione con le barre del quadrupolo (gli ioni si scaricano su una delle barre)

A determinati valori della tensione applicata, solo ioni aventi un certo rapporto m/z usciranno dal quadrupolo stesso. Variando nel tempo la tensione applicata, tutti gli ioni saranno messi in condizione di uscire (a tempi diversi) dal quadrupolo.

Analizzatore a trappola ionica (IT)

Opera su un principio simile a quello del quadrupolo, tuttavia non funziona da filtro. Anziché permettere agli ioni di attraversare il campo quadrupolare, la IT trattiene tutti gli ioni al suo interno. La trappola ionica è costituita da tre

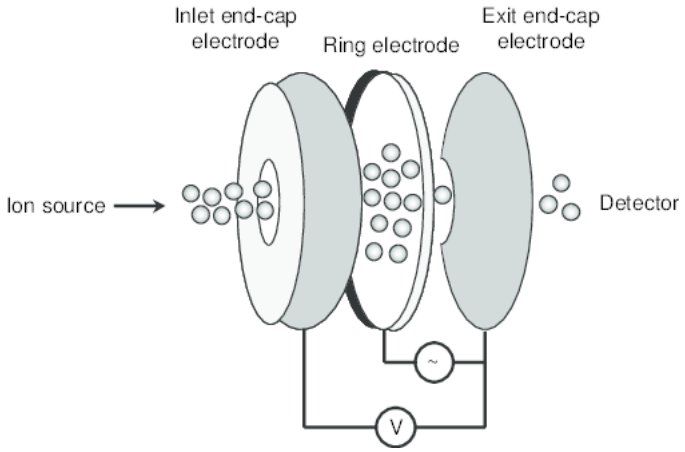


Figura 2.11: Disposizione degli elettrodi nell'analizzatore IT
Fonte: https://www.researchgate.net/figure/Quadrupole-ion-trap-mass-analyzer_fig2_236737006

elettrodi in acciaio le cui superfici interne sono di forma iperbolica. Gli elettrodi sono disposti in una geometria "a panino" :

- L 'elettrodo ad anello nel centro (ring electrode) a cui è applicato un potenziale alternato V con frequenza angolare Ω (RF);
- i due elettrodi laterali (end-cup), uno di entrata e uno di uscita degli ioni, sopra e sotto al ring electrode che sono a potenziale di terra o con tensione applicata AC o DC.

I tre elettrodi formano insieme una cavità (o trappola) in cui gli ioni vengono immagazzinati e dove avviene l'analisi di massa (vedi fig. 2.11)

Applicando all'elettrodo ad anello un voltaggio RF di appropriata grandezza e frequenza vengono intrappolati gli ioni che coprono un certo intervallo di massa. Il campo generato esercita sugli ioni nella trappola una compressione che impedisce loro di muoversi (vedi fig 2.12).

All'interno della trappola è presente elio ad una pressione di circa 10^{-2} Bar che collide con gli ioni causando una diminuzione della loro energia cinetica e quindi una riduzione dell'ampiezza delle loro oscillazioni (raffreddamento o "cooling"). In questo modo anche l'elio contribuisce a focalizzare tutti gli ioni verso il centro della trappola ionica, aumentando la sensibilità e la risoluzione.

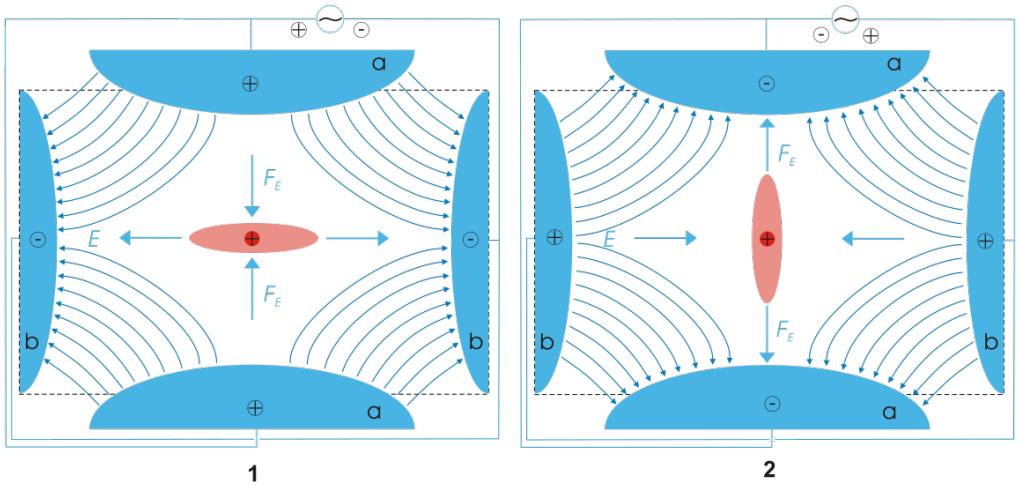
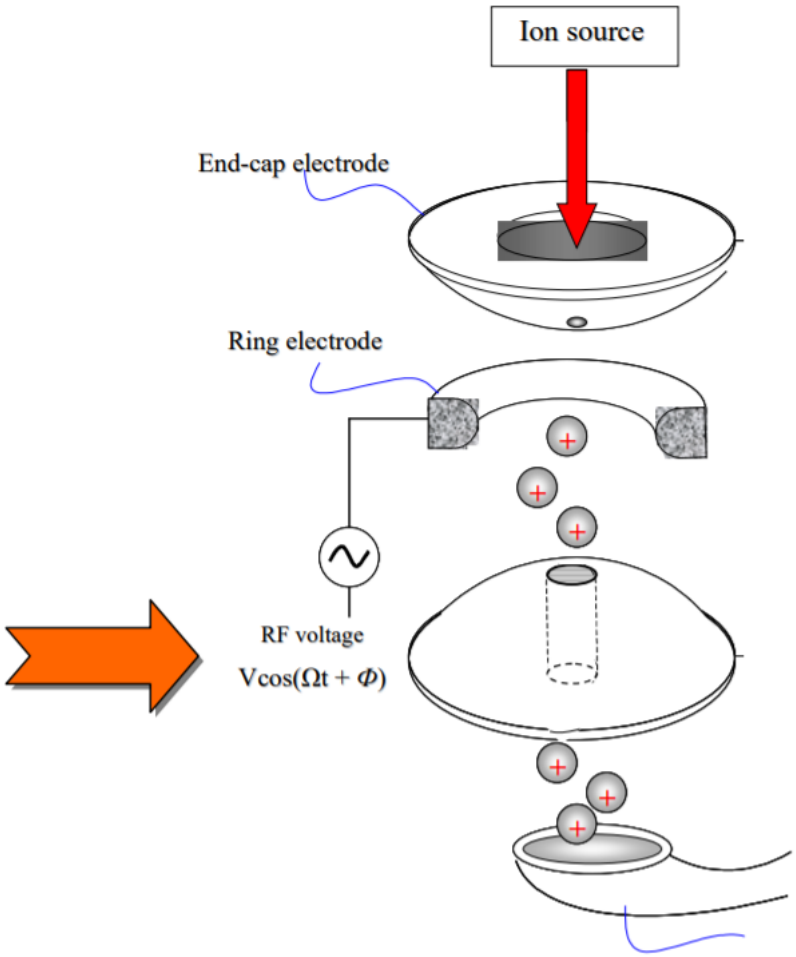


Figura 2.12: Potenziale nell'analizzatore IT
Fonte: https://en.wikipedia.org/wiki/Quadrupole_ion_trap

Una volta intrappolati gli ioni vengono espulsi progressivamente rendendo instabile la loro traiettoria tramite un processo detto "mass-selective axial instability mode" (o ad instabilità massa-selettiva). Quando si vuole registrare uno spettro di massa viene aumentata l'ampiezza del voltaggio RF applicato al ring electrode. Si ha come conseguenza che gli ioni di m/z sempre più elevato diventano instabili dal momento che il loro moto all'interno della trappola aumenta in ampiezza e può farli uscire dai confini fisici del sistema. A questo punto essi sono espulsi dal sistema in sequenza di massa attraverso i fori praticati nell'elettrodo terminale (figura 2.13). Gli ioni sono espulsi facendo una scansione della RF applicata all'elettrodo centrale e applicando ai due laterali una RF supplementare oscillante (AC). Poco prima dell'espulsione, il moto assiale dell'ione entra in risonanza con il potenziale oscillante (AC) applicato, eccitando lo ione per risonanza; la traiettoria assiale dello ione eccitato diviene sempre più ampia consentendo allo ione stesso di sfuggire all'effetto della nuvola di ioni nella trappola. In questo modo gli ioni della stessa specie possono essere ben raggruppati prima di uscire sequenzialmente dalla trappola, garantendo una migliore risoluzione spettrale e sensibilità.



Fonte:
https://people.unica.it/michelabegala/files/2010/06/MS-Lezione-III-analizzatori.pdf

Figura 2.13: Uscita degli ioni dalla ion trap

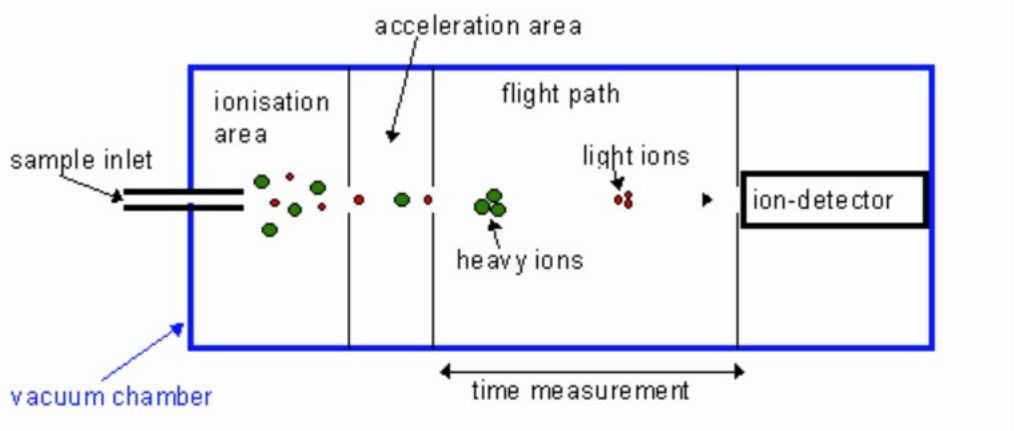


Figura 2.14: Schema analizzatore TOF
Fonte: https://alevelnotes.com/notes/chemistry/elements-of-life/mass-spectrometry

Analizzatore a tempo di volo (TOF)

Separa gli ioni in virtù del tempo che impiegano a percorrere una certa distanza nota. Si basa sul fatto che ioni con diverso valore m/z e uguale energia cinetica impiegano tempi differenti a percorre una certa distanza: ioni più pesanti impiegano più tempo. Gli ioni prodotti dalla sorgente vengono accelerati da un potenziale di accelerazione che conferisce a tutti la stessa energia cinetica (vedi figura 2.14). Dall'equazione 2.2 ricaviamo che

$$v = \sqrt{\frac{2zV}{m}} \tag{2.5}$$

dunque minore è la massa dello ione, maggiore è la sua velocità. Inoltre se lo strumento possiede un tubo di lunghezza L, detta $v = L/t$ e sostituendo a v l'equazione 2.5 si ottiene:

$$t = L/v = \frac{L}{\sqrt{\frac{2V}{m/z}}} \tag{2.6}$$

L'analizzatore TOF deve lavorare ad impulsi perché tutti gli ioni devono essere espulsi dalla sorgente nello stesso momento. Se così non fosse, gli ioni verrebbero prodotti continuativamente, e al rivelatore arriverebbe un flusso continuo di ioni che non sarebbero più separabili in funzione del tempo di arrivo. L'analizzatore TOF deve perciò utilizzare tecniche di ionizzazione pulsata come la MALDI di-

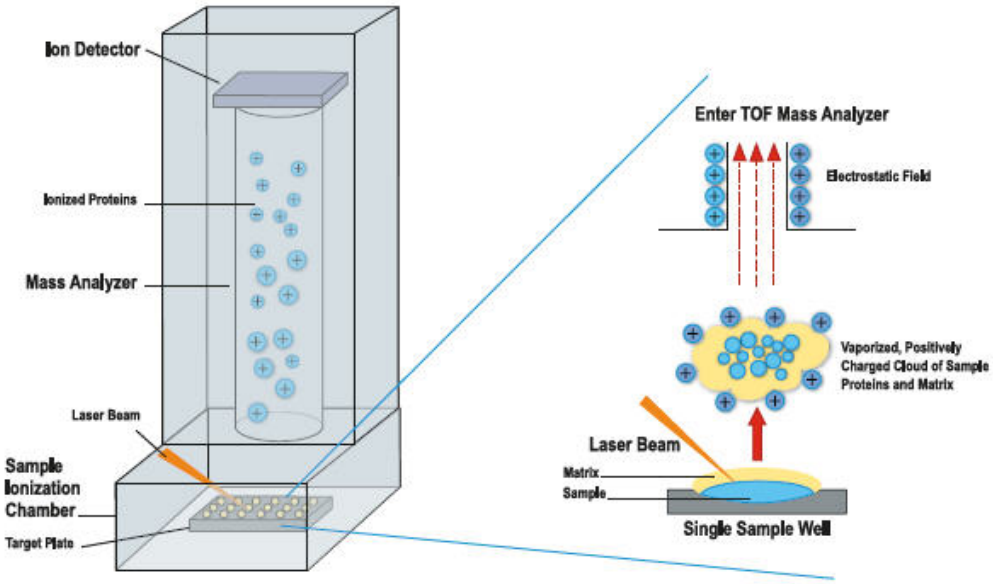


Figura 2.15: Schema MALDI-TOF

Fonte: http://clinchem.aaccjnls.org/content/61/1/100

scussa precedentemente. Nella figura 2.15 è riportato lo schema di MALDI-TOF.

2.1.3 Rilevazione degli ioni

Per poter produrre lo spettro di massa è necessario un rilevatore di ioni, dispositivo che "trasforma" un fascio di ioni in ingresso in un segnale elettrico proporzionale alla quantità degli ioni.

Questo segnale viene successivamente amplificato e registrato. Un tipico rilevatore (riportato in figura 2.16) è costituito da:

- un dinodo di conversione,
- un moltiplicatore di elettroni a dinodo continuo o "Channeltron"

Il dinodo di conversione è una superficie metallica concava posta ad angolo retto rispetto al fascio di ioni che giungono dall'analizzatore, alla quale è applicato ad un potenziale negativo. Quando uno ione positivo colpisce un dinodo di conversione si osserva l'emissione di particelle secondarie (ioni negativi ed elettro-

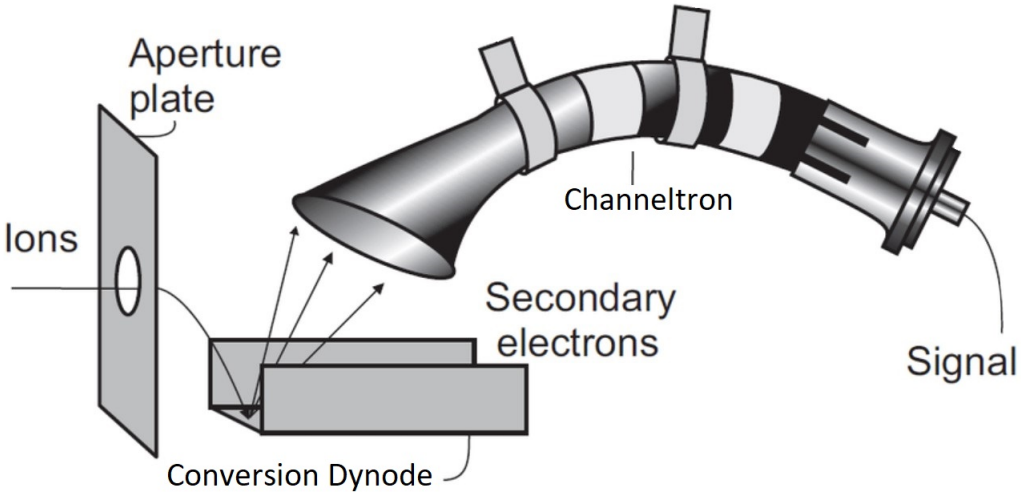


Figura 2.16: Schema rivelatore

Immagine modificata da https://www.researchgate.net/figure/Figure-A9-Principle-of-the-conversion-dynode-and-connection-scheme-of-the-Channeltron_fig46_48908229

ni) che vengono focalizzati dalla superficie curva del dinodo e accelerati verso il moltiplicatore di elettroni.

Gli ioni positivi (ma vale anche un concetto analogo per quelli negativi) trasformati in elettroni dal dinodo di conversione vengono amplificati attraverso un effetto a cascata nel *channeltron* di elettroni a forma di corno ricurvo per produrre un segnale elettrico (vedi figura 2.17).

Questo dispositivo è largamente impiegato in strumenti a quadrupolo e ad ion trap.

Grazie alla particolare forma gli elettroni emessi non percorrono molto spazio prima di colpire nuovamente la superficie interna del moltiplicatore, causando così l'emissione di nuovi elettroni. Alla fine si forma una cascata di elettroni che si traduce in una corrente misurabile.

Il segnale che esce dal rivelatore va elaborato tramite il calcolatore dello spettrometro per la presentazione dello spettro di massa. Oltre a questo il calcolatore:

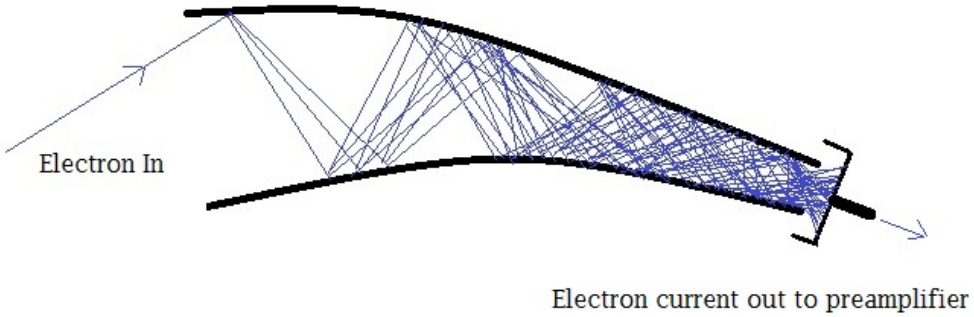


Figura 2.17: Effetto moltiplicativo del channeltron
Fonte: https://www.rbdinstruments.com/blog/how-an-electron-multiplier-works/

- Acquisisce e memorizza lo spettro
- Esegue analisi quantitative e qualitative sullo spettro ottenuto

Peak detection

Lo spettro in uscita dallo spettrometro deve essere successivamente elaborato al fine di ottenere una sequenza di picchi ben definiti. Questa procedura, detta peak detection, incide molto sul risultato finale dell'analisi, specialmente se si usano spettrometri ad altissima risoluzione come quello usato in questa tesi. Esistono vari algoritmi, basati principalmente su direct peak location, derivata prima e seconda, trasformate wavelet continue e fitting [12]

2.1.4 Risoluzione

Nelle sezioni precedenti è stata menzionata spesso la risoluzione. In realtà esistono due definizioni di risoluzione, di seguito riportate.

Definizione IUPAC

La risoluzione necessaria a separare due picchi A e B è:

$$R = \frac{M}{\Delta M} \tag{2.7}$$

dove R è la risoluzione, M è il valore di m/z del picco A e ΔM (detto anche potere risolvente) è la differenza tra i valori di m/z di due picchi contigui picco B e picco A. Questa è la definizione sottintesa precedentemente.

Definizione alternativa

Alcuni danno significati a R e ΔM nell'equazione 2.7, ovvero indicano con R il potere risolvente e la risoluzione come la più piccola ΔM che permette di distinguere tra due ioni differenti. In questo caso metodi diversi danno luogo a diverse misure di ΔM, a seconda di dove si misura tale valore nello spettro (i più utilizzati sono larghezza di un picco o valle tra due picchi prese ad altezze diverse).

2.2 Lo strumento utilizzato per la raccolta dei dati

Per l'esperimento che ha permesso la raccolta dei dati dai campioni di branzini di questo esperimento si è utilizzato uno spettrometro all'avanguardia. Rispetto ad altre soluzioni descritte precedentemente, questo spettrometro permette un'analisi del campione veloce e molto accurata con costi di acquisto e analisi contenuti. Si tratta di uno spettrometro Orbitrap con sorgente di ionizzazione DART (vedi figura 2.18).

Presenta diverse modifiche vantaggiose rispetto ad uno spettrometro "classico". Di seguito sono riportate sinteticamente le sue caratteristiche fondamentali. Per una descrizione più accurata si rimanda a [7] e [8].

Figura 2.18: Modello dello spettrometro utilizzato in questa tesi. Lo ionizzatore è sostituito con il DART

Fonte: https://assets.thermofisher.com/TFS-Assets/CMD/product-images/Exactive-Plus-Right.eps-650.jpg

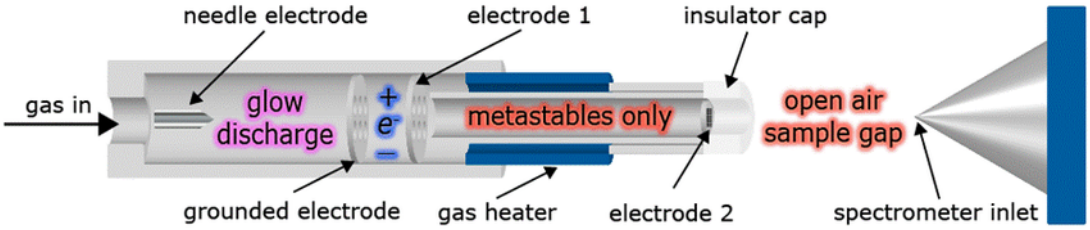


Figura 2.19: Schema di funzionamento DART

Fonte: https://en.wikipedia.org/wiki/Direct_analysis_in_real_time

2.2.1 Sorgente di ionizzazione DART

Nello spettrometro utilizzato la sorgente di ionizzazione è detta DART, acronimo inglese che sta per "Analisi diretta in tempo reale". E' stata sviluppata per la prima volta attorno al 2003 e ha diversi vantaggi rispetto, per esempio, a una ionizzazione chimica che richiede l'utilizzo di reagenti. Infatti la DART permette l'analisi del campione senza che questo venga precedentemente trattato (tramite pulizia e/o estrazione mediante solventi). Si ottengono ottimi risultati analizzando diversi tipi di sostanze su molteplici superfici [8].

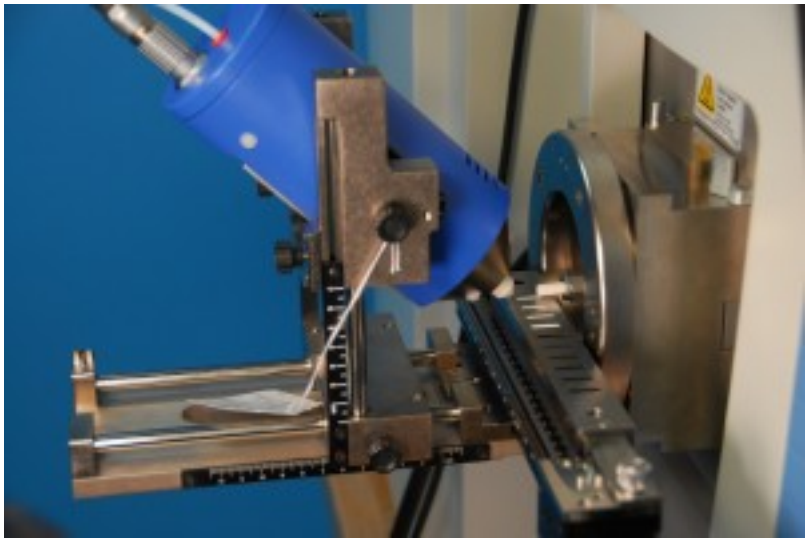


Figura 2.20: Ionizzatore DART
Fonte: http://www.msconsult.dk/en/instrument-sales/ionsense-dart/

Inoltre permette di ottenere i risultati in tempi rapidi: un intero esperimento, comprese le fasi iniziali e l'elaborazione dei dati, dura meno di due ore. Il principio di funzionamento è semplice: un flusso di gas di ioni contenente particelle metastabili (di solito elio o azoto) viene indirizzato sul campione ed è responsabile del desorbimento e della ionizzazione delle molecole di analita presenti sulla superficie del campione.

2.2.2 Analizzatore Orbitrap

E' un particolare tipo di analizzatore IT (vedi precedentemente) costituito da un elettrodo cavo all'interno che circonda un elettrodo coassiale interno simile a un fuso che intrappola gli ioni in un movimento orbitale attorno a sè stesso (vedi figura 2.21). La *corrente immagine* generata dagli ioni intrappolati viene rilevata e convertita in uno spettro di massa usando la trasformata di Fourier del segnale di frequenza. Le fasi principali che compongono l'analisi Orbitrap sono riportate di seguito.

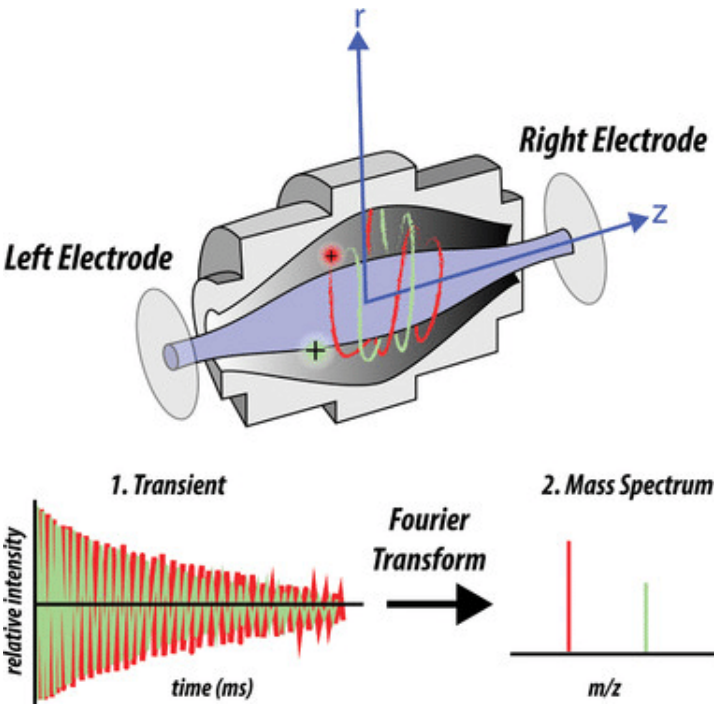


Figura 2.21: Schema Orbitrap

Fonte: https://www.researchgate.net/figure/The-Orbitrap-FT-mass-analyzer-In-the-Orbitrap-ions-oscillate-around-a-central_fig9_306542804

Intrappolamento

Gli ioni sono intrappolati a causa della loro attrazione elettrostatica sull'elettrodo bilanciata dalla loro inerzia. Si trovano quindi a ruotare attorno all'elettrodo interno su traiettorie assomigliano a delle eliche. A causa delle proprietà del potenziale quadro-logaritmico[7] il loro movimento assiale è armonico, cioè è completamente indipendente non solo dal movimento attorno all'elettrodo interno ma anche da tutti i parametri iniziali degli ioni (tranne i loro rapporti m/z).

Iniezione

Per iniettare gli ioni nella trappola, il campo tra gli elettrodi viene prima ridotto. Quando i pacchetti di ioni vengono iniettati tangenzialmente al campo, il campo elettrico viene aumentato con una rampa di tensione. Gli ioni vengono schiacciati verso l'elettrodo interno fino a raggiungere l'orbita desiderata all'interno della trappola. Quindi la rampa viene fermata, il campo diventa statico e il rilevamento può iniziare. Ogni pacchetto di ioni contiene una moltitudine di ioni a velocità diverse distribuite su un determinato volume. Questi ioni si muovono con frequenze di rotazione diverse ma con la stessa frequenza assiale. Ciò significa che gli ioni con un rapporto massa-carica specifico diffondono in anelli che oscillano lungo il barilotto interno. Questo metodo di iniezione funziona bene con sorgenti pulsate tipo MALDI, mentre non funziona con sorgenti continue come l'elettrospray. In questo caso, prima di procedere con l'iniezione, viene usata una "C-trap" (vedi figura 2.22). Permette di raggruppare gli ioni in pacchetti prima di spedirli verso la trappola tramite rapido abbassamento del voltaggio RF e applicazione di un gradiente DC lungo la trappola. Si creano pacchetti simili a quelli generati delle sorgenti laser.

Eccitazione

Per eccitare le oscillazioni assiali coerenti degli anelli ionici basta iniettare i pacchetti di ioni lontano dal minimo del potenziale assiale [7] (che corrisponde alla

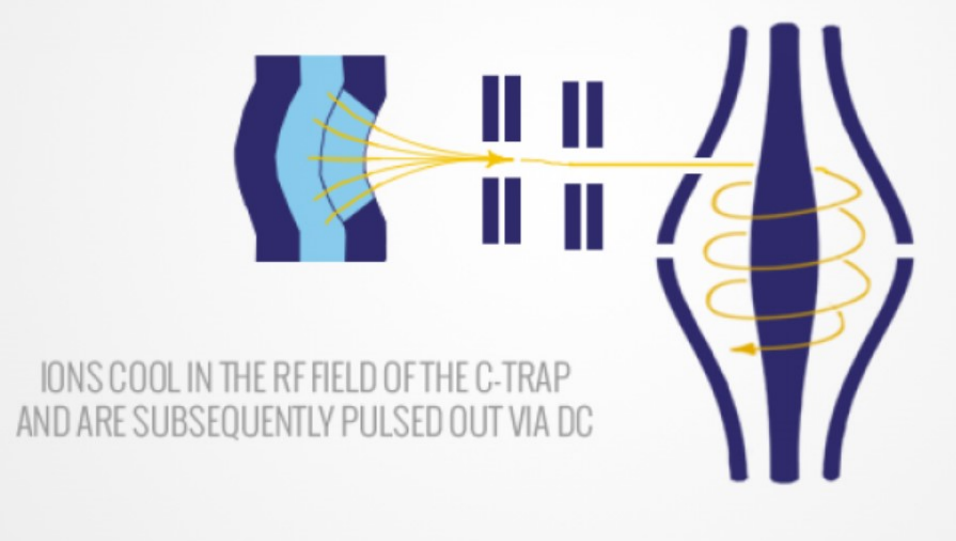


Figura 2.22: Schema trappola C
Fonte: http://www.massspecpro.com/mass-analyzers/orbitrap

parte più spessa di entrambi gli elettrodi): questo avvia automaticamente le loro oscillazioni assiali, eliminando la necessità di ulteriori eccitazioni. Questo consente anche l'avvio del processo di rilevamento non appena l'elettronica di rilevamento si ripristina dalla rampa di tensione necessaria per l'iniezione.

Rivelazione

Le oscillazioni assiali degli anelli ionici sono rilevate dalla loro "corrente immagine" indotta sull'elettrodo esterno che è diviso in due sensori simmetrici di raccolta collegati ad un amplificatore differenziale. La trappola stessa, di fatto, costituisce il rilevatore di massa. Tutti gli ioni vengono rilevati simultaneamente in un determinato periodo di tempo e la risoluzione può essere migliorata aumentando la forza del campo o aumentando il periodo di rilevamento.

Capitolo 3

Descrizione dei dati da analizzare

3.1 Metodo di raccolta dei campioni

Sono stati valutati 60 esemplari di branzino (Dicentrarchus labrax) di origine greca, allevati intensivamente in gabbia a mare. Dal medesimo allevamento sono stati acquistati due lotti di 30 esemplari ciascuno, differenti per taglia (200-300g e 300-400g), trasportati in casse di polistirene con ghiaccio in scaglie e conferiti in 36 ore ad una ditta di prodotti ittici all'ingrosso. A 48 ore dalla raccolta gli esemplari presentavano ottime caratteristiche di freschezza (rigor, occhio brillante, branchie rosso vivo) e da ciascun lotto sono stati sottoposti a sfilettatura 30 soggetti, effettuata da operatore della ditta, con successiva eviscerazione. La temperatura dei filetti è stata misurata all'inizio della lavorazione (+0.5 °C) e al termine della lavorazione (+4.5°C). Da ciascun esemplare sono stati ottenuti due filetti, distinti in sacchetti identificati con numero progressivo, di cui uno è stato congelato e uno mantenuto alla temperatura di ghiaccio fondente fino al momento dell'analisi. I campioni sottoposti a congelamento sono stati posti in abbattitore con temperatura di esercizio tra -20°C e -30°C per 2 ore fino a raffreddare i filetti al cuore a -18°C. Successivamente i campioni sono stati raggruppati in una cassa di polistirene stoccata presso la ditta in cella freezer a -20°C per 6 settimane. I filetti freschi sono stati trasportati, immediatamente dopo la sfilettatura, in laboratorio in contenitore isotermico con ghiaccio in scaglie. I filetti congelati sono

stati decongelati nell'arco di 24 ore in cella a +4°C presso la medesima ditta. Successivamente sono stati trasportati in laboratorio in contenitore isotermico con ghiaccio in scaglie e sottoposti alla medesima modalità di prelievo.

3.2 Procedimento di acquisizione degli spettri

3.2.1 Preparazione dei campioni

Tutti i campioni sono stati preparati per l'analisi allo spettrometro aggiungendo a 1g di campione 9ml di EtAc (etil-acetato). E' una soluzione lipofila, permette cioè di analizzare solo le sostanze apolari, ovvero quelle di interesse in questo ambito. Successivamente i preparati sono agitati al vortex, lasciati 10' a sonicare e infine analizzati. 5 ml delle soluzioni così ottenuti vengono depositati su DIP Stick (in triplicato) lasciati asciugare e poi processati in modalità di ionizzazione positiva.

3.2.2 Settaggi dello strumento

Settaggi DART	
Temperatura	360°
Grid Voltage	250V
Gas	Elio
Polarità	Positiva
Velocità scorrimento	0.3 mm/s
Dopante	NH_3

Settaggi Orbitrap	
Polarità	ioni negativi
Range	da 75 a 1250 m/z
Risoluzione	70.000 FWHM
AGC target	3e6
Tempo di analisi	0.66 min
Capillarity	250 °C
S-lens RF	55
CID	0 eV

3.2.3 Trascrizione e decodifica dei dati dello spettrometro

Al termine dell'acquisizione strumentale vengono eseguite alcune operazioni di trascrizione decodifica e identificazione (realizzate dall'IZSVe) di seguito riportate:

- Acquisizione spettro DART mediante Thermo Xcalibur
- Creazione spettro medio e sottrazione del background
- Memorizzazione file RAW dello spettro medio
- Conversione RAW to mzML mediante Proteowitzard/MS convert
- Creazione dei file .csv mediante una specifica procedura R
- Organizzazione dei file in directory e sottodirectories
- Creazione di un file .zip
- Analsi statistica mediante sito on-line MetaboAnayst con cui vengono eseguite le fasi preparatorie dei dati come Peak Matching and Alignment
- Creazione del database di spettri ripuliti ed allineati da analizzare

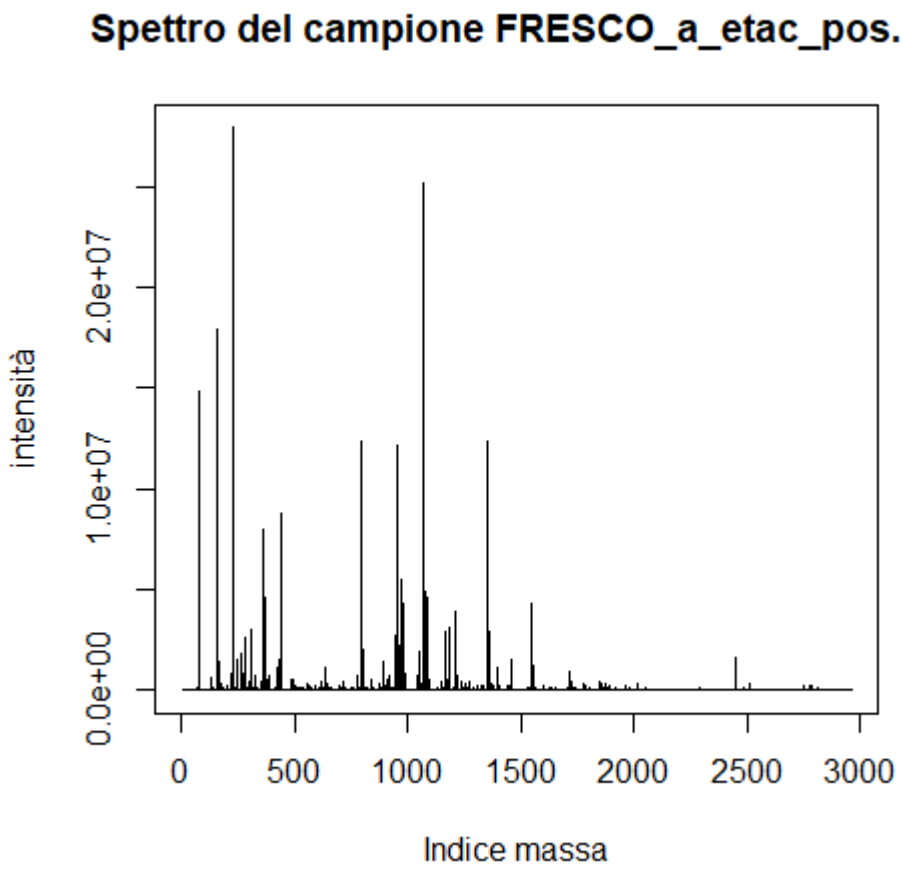


Figura 3.1: Spettro di massa

3.3 Analisi preliminare dei dati

3.3.1 Spettri ottenuti

In figura 3.1 è riportato un esempio di spettro "ripulito" (corrisponde a una colonna del database di spettri). Per ogni campione si ottiene una lista di 2959 intensità, una per ogni massa rilevata compresa tra 75 e 1069 Da (poichè il collision index dissociation o CID è settato a 0 eV, m/z coincide con m, visto che z=1).

3.3.2 Descrizione dei dati

Prima di procedere con la classificazione è utile procedere ad una analisi dei dati per poter delineare una strategia da seguire per la normalizzazione. Seguono i grafici "diagnostici" realizzati a tale scopo con una breve descrizione.

MvA

Il grafico MvA è un particolare grafico, usato soprattutto in ambito genomico, che permette di evidenziare, se presenti, eventuali bias nella raccolta di due campioni. Dati due campioni $s1$ e $s2$, si definiscono $M = log(s1) - log(s2)$ e $A = 0.5 * (log(s1) + log(s2))$. In questo dataset non risultano bias significativi: i grafici sono ben allineati attorno a media nulla con deviazione dalla media trascurabile rispetto al valore molto elevato delle intensità (dell'ordine di 10^7 in alcuni casi). Ne vengono riportati alcuni, se si desidera vederli tutti (o qualcuno in particolare) nell'appendice è riportato il codice per realizzarli (in tutto sono 62128). In figura 3.2 sono riportati tre MvA plot tra campioni freschi, in figura 3.3 sono riportati tre MvA plot tra campioni decongelati e in figura 3.4 sono riportati due MvA plot tra un campione fresco e un decongelato.

Cross-Correlazione tra campioni inter-classe e intra-classe

Permettono di capire quanto i campioni di una stessa classe sono "simili" e quanto campioni di classi diverse siano "diversi". Vengono calcolati i coefficienti di correlazione di Pearson. Date due variabili statistiche, il coefficiente di correlazione di Pearson è un indice che esprime un'eventuale relazione di linearità tra esse. Varia tra -1 e 1: 1 corrisponde alla perfetta correlazione lineare positiva, 0 corrisponde a un'assenza di correlazione lineare e -1 corrisponde alla perfetta correlazione lineare negativa.

Nel grafico di figura 3.5 è riportato l'istogramma dei valori di cross-correlazione tra campioni freschi, campioni decongelati e tra freschi e decongelati. Sono esclusi i valori di cross-correlazione tra gli stessi campioni (quelli sulla diagonale della

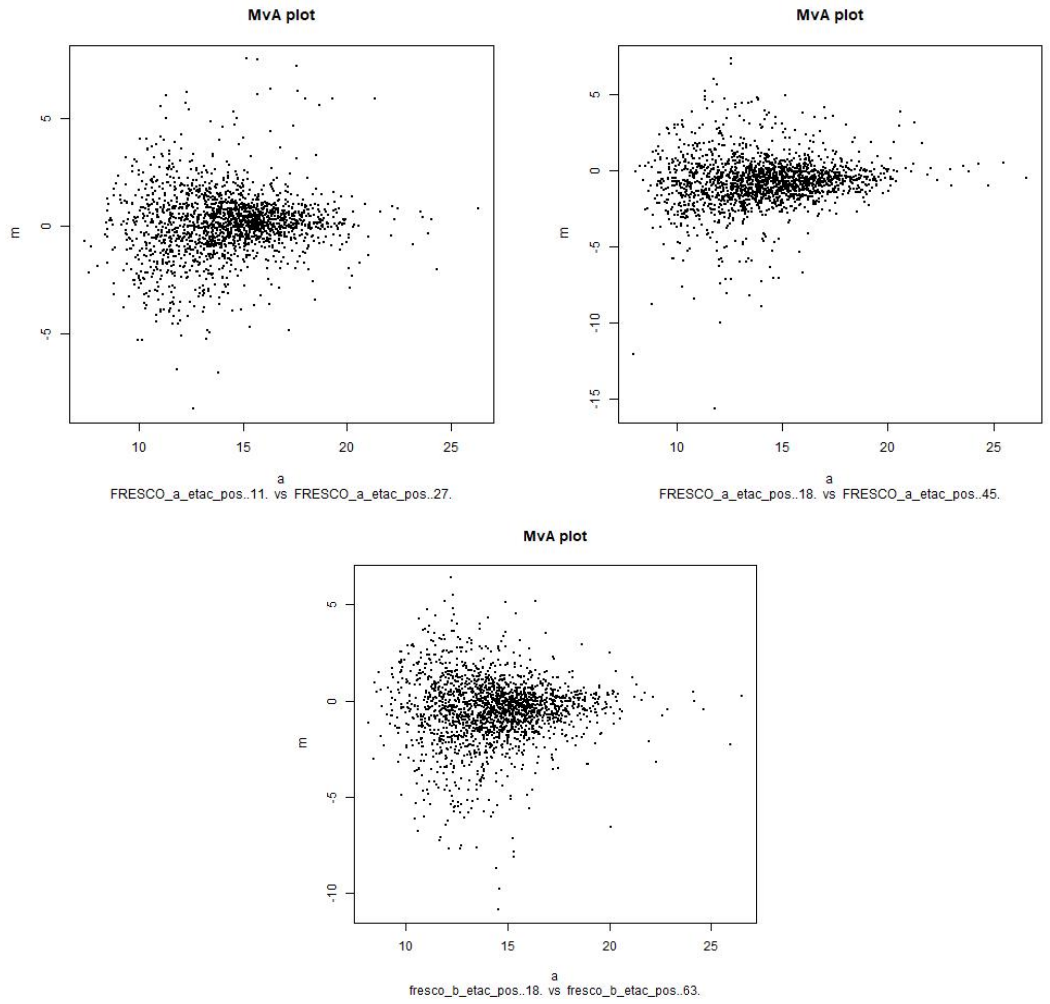


Figura 3.2: MvA plot tra freschi

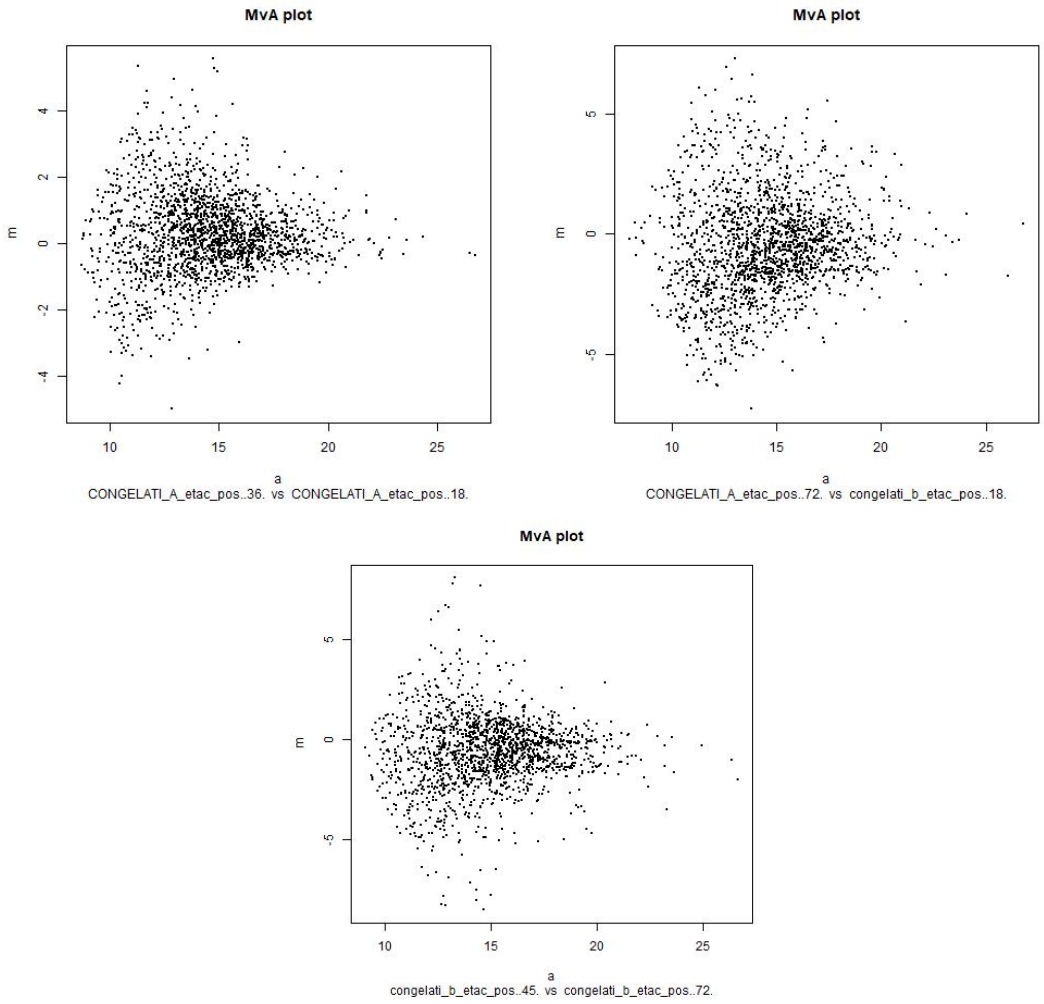


Figura 3.3: MvA plot tra congelati

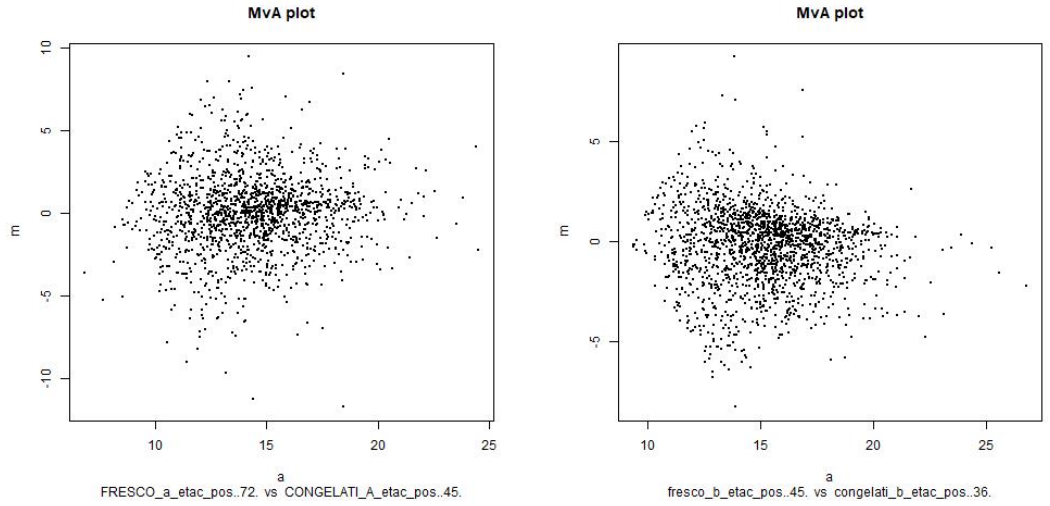


Figura 3.4: MvA plot tra un fresco e un decongelato

matrice di correlazione). Per rendere confrontabili i campioni, questi sono stati standardizzati (a media nulla e varianza unitaria) e negli istogrammi è rappresentata la densità (l'area è normalizzata a 1 per tutti e tre). Risulta che in generale i campioni sono molto correlati tra loro, in particolare quelli decongelati, il 50% infatti ha cross-correlazione maggiore di 0.9.

Cross-correlazione tra features inter-classe e intra-classe

Evidenziano eventuali similitudini tra features (masse) diverse. Nel grafico in figura 3.6 (anche qui, prima di calcolare la cross-correlazione, le features sono state standardizzate e l'istogramma riporta le densità) si nota chiaramente che, mentre buona parte delle features sono scorrelate (come si auspica), alcune presentano una correlazione piuttosto elevata. Questo indica che sarà necessario eliminare le features che risultano "ridondanti". Questo step è spiegato con maggior chiarezza nel capitolo 6.

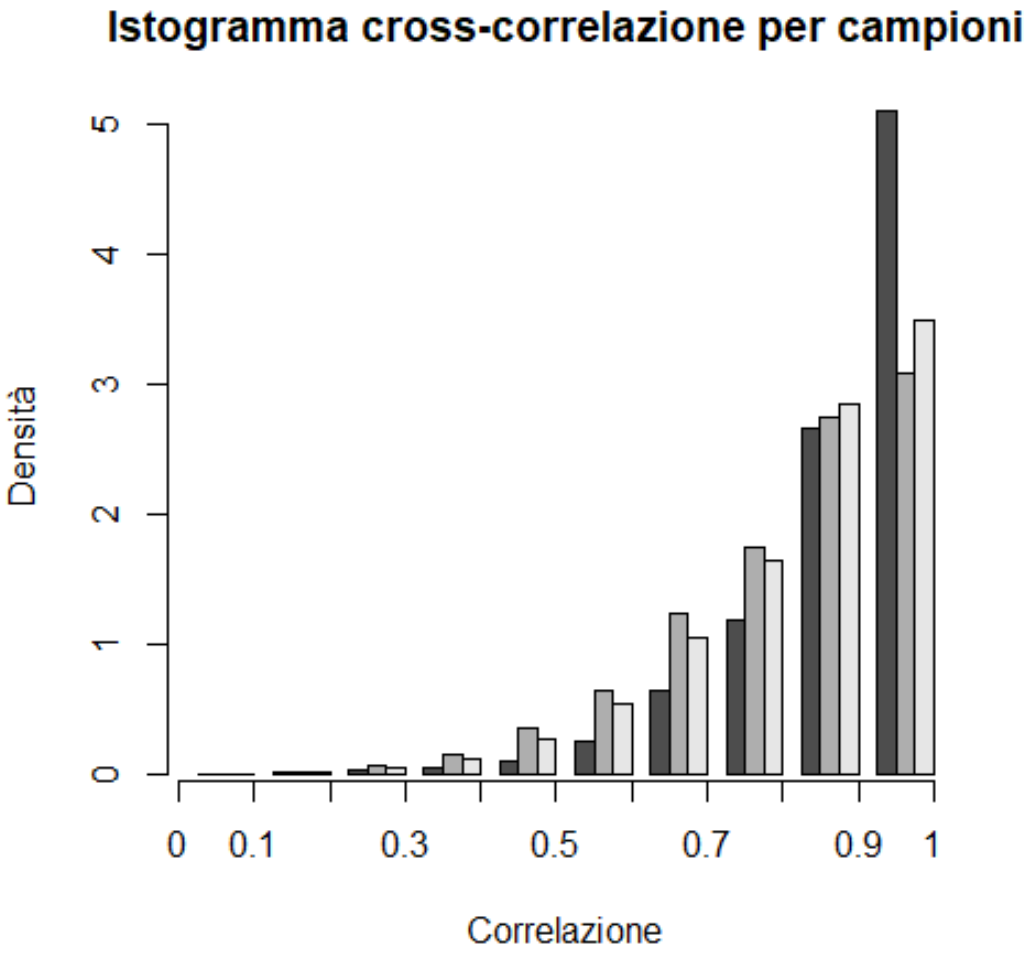


Figura 3.5: Cross correlazione tra soli freschi (grigio), soli decongelati (nero) e tra freschi e decongelati (bianco)

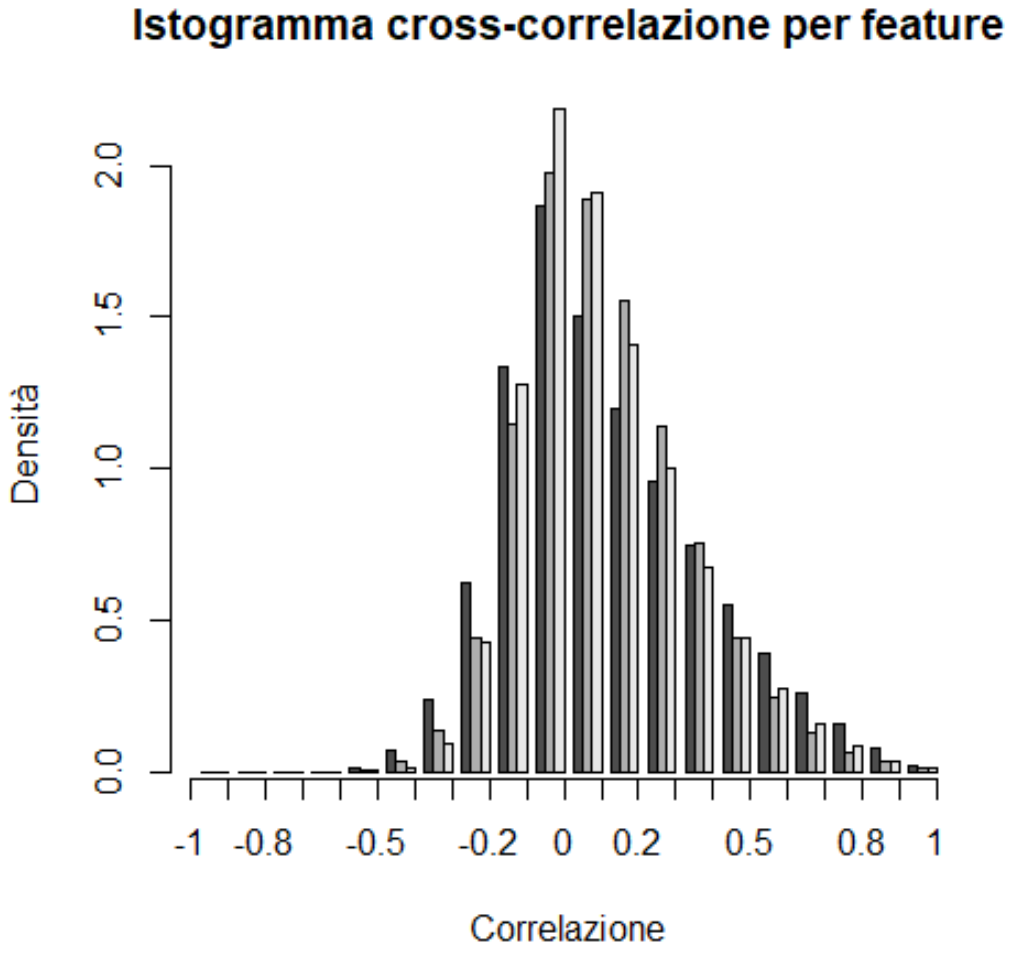


Figura 3.6: Cross-correlazione tra features, considerando soli freschi (grigio), soli decongelati (nero) e freschi e decongelati (bianco)

Capitolo 4

Normalizzazione dei dati

Sebbene dai grafici del capitolo precedente non risulti necessario procedere con una normalizzazione per campioni, il tool MetaboAnalyst consiglia di effettuarla comunque. In questo capitolo verranno confrontati tra loro i diversi metodi che propone MetaboAnalyst, con l'obiettivo di trovarne uno, se esiste, in grado di migliorare effettivamente i dati. Per rendere il confronto più robusto, l'unico pre-processing dei dati è stato sostituire gli NA (circa il 25% dei dati) con l'half minimum positive value delle intensità. Si ipotizza ragionevolmente che se un valore di intensità è NA allora la corrispondente massa non è presente nel campione. In questi casi MetaboAnalyst suggerisce la sostituzione con l'half minimum positive value. Nel classificatore tuttavia tali NA sono sostituiti con 0: questo non comporta nessuna differenza sostanziale nel risultato finale, anzi il risultato è più accurato.

4.0.1 Calcolo delle normalizzazioni con MetaboAnalyst

Per ciascuna normalizzazione viene caricato il dataset originale (con i campioni in colonna). Come discusso precedentemente si seleziona *Replace by a small value (half of the minimum positive value in the original data)* nella sezione **Missing value estimation**, togliendo la spunta su **Step 1. Remove features with too many missing values**. Non si vuole infatti rimuovere nessuna feature per

il momento. Anche nella sezione **Data Filtering** viene selezionato *None* per non effettuare nessun filtraggio dei dati. Infine nella sezione **Normalization overview** viene selezionata una normalizzazione tra quelle presenti in **Sample normalization**, lasciando gli altri campi a *None*. In seguito sono descritte le normalizzazioni che propone MetaboAnalyst.

4.1 Quantile Normalization

Sfrutta la funzione *preprocessCore::quantileNormalize()* di R ed elimina quelle features (masse) che sono allo stesso rank in tutti i campioni (e quindi le loro intensità sono sostituite dalla media delle intensità dalla funzione). Di fatto linearizza i qqplot per campioni, come si può vedere nell'esempio di figura 4.1. Fa l'assunzione che ci sia una distribuzione comune di intensità tra campioni. Questa assunzione è verosimile: si stanno analizzando campioni molto simili tra loro, è ragionevole supporre che le intensità delle masse seguano una distribuzione comune [2].

4.2 Normalization by sum

Come suggerisce il nome questa normalizzazione divide (per campione) le intensità di ciascuna massa per la somma delle intensità, moltiplicandola successivamente per 1000. In questo modo la somma delle intensità di ciascun campione risulta pari a 1000.

4.3 Normalization by median

Simile a **Sum Normalization** questa normalizzazione divide (per campione) ciascuna intensità per la mediana delle intensità. In questo modo la mediana delle intensità di ciascun campione risulta unitaria, come si può notare in figura 4.2.

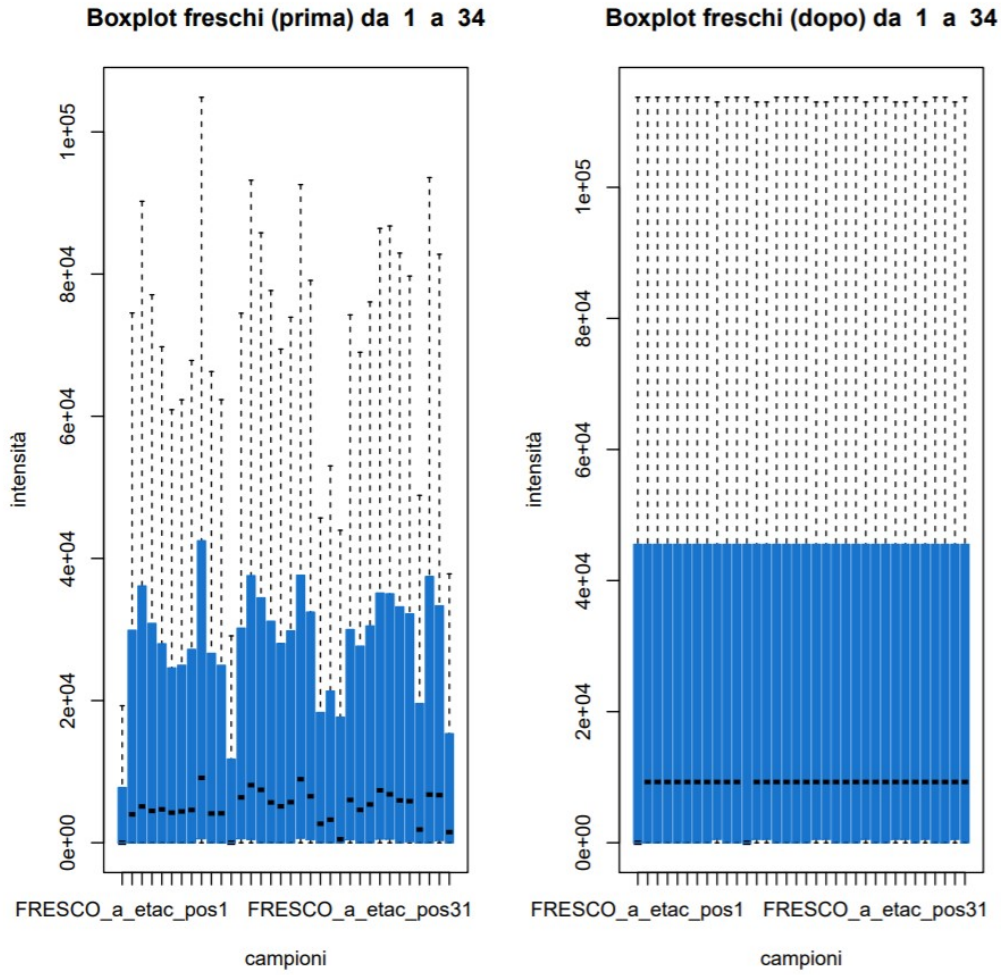


Figura 4.1: Esempio di come si modificano i boxplot per campioni con la quantile normalization

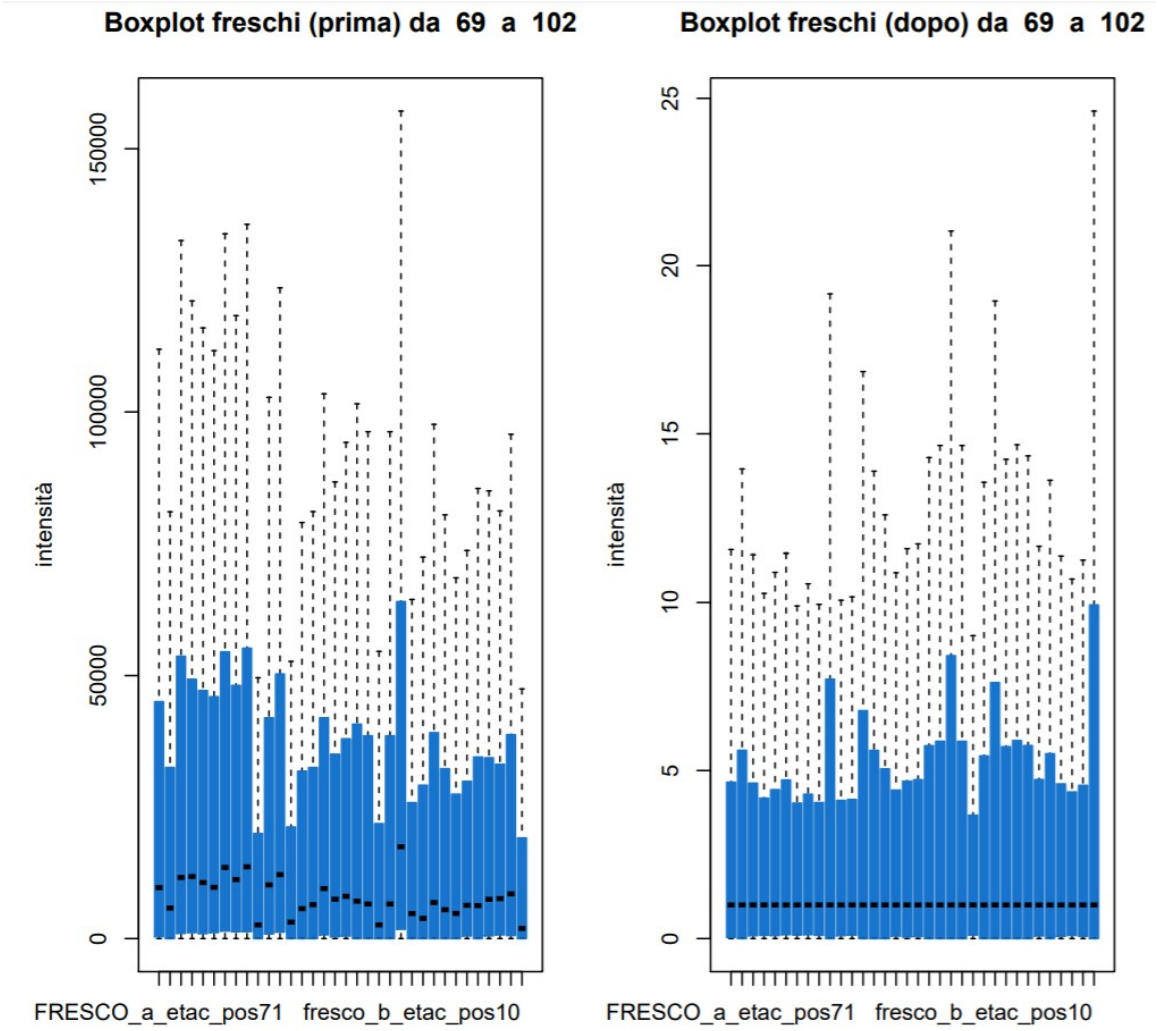


Figura 4.2: Esempio di come la normalizzazione by median modifica i boxplot

4.4 Normalization by a pooled sample from group

Sfrutta la Probabilistic Quotient Normalization nel seguente modo:

1. Crea un campione `ref` di riferimento facendo la media delle intensità (per masse) all'interno del gruppo specificato (nel nostro caso o freschi o congelati)

2. Ad ogni campione (sia fresco che congelato) applica la seguente normalizzazione:

```
campione_norm <- campione/median(campione/ref)
```

4.5 Sample-specific normalization (i.e. weight, volume)

Questa normalizzazione moltiplica le intensità di ciascun campione per una costante in modo da correggere eventuali differenze nell'analisi del campione. In questo esperimento non è necessario: tutti i campioni sono stati preparati allo stesso modo e nelle stesse quantità.

4.6 Normalization by reference sample (PQN)

Il funzionamento è lo stesso della **Normalization by a pooled sample from group** ma `ref` è uno specifico campione del dataset. In questo tipo di database non ha senso preferire un campione in particolare da usare come campione di riferimento, quindi anche questa normalizzazione, come la precedente, non è stata utilizzata.

4.7 Confronto normalizzazioni

Tutte queste normalizzazioni (per campione) sono state confrontate tra loro con l'obiettivo di selezionare quella più adatta allo specifico dataset in esame. Nella valutazione sono considerati i seguenti grafici:

- MvA plot: come spiegato precedentemente, evidenzia se sono presenti eventuali bias nella raccolta dei campioni. Una buona normalizzazione non deve peggiorare questi grafici, al più migliorarli.

- Boxplot: è una rappresentazione grafica utilizzata per descrivere la distribuzione di un campione tramite semplici indici di dispersione e di posizione. In seguito alla normalizzazione ci si aspetta che i boxplot di campioni diversi diventino più simili tra loro.

- Corrgram: tramite il grafico delle cross-correlazioni si valuta quanto campioni di classi diverse sono diversi e quanto campioni della stessa classe sono simili. Con una normalizzazione ottimale si dovrebbe ottenere un aumento dell'indice di correlazione per campioni della stessa classe e una diminuzione dell'indice per campioni di classi diverse: questo faciliterebbe il compito di classificare correttamente i campioni.

- Distanza euclidea: con un ragionamento simile al precedente grafico ci si aspetta che una buona normalizzazione diminuisca la distanza tra campioni della stessa classe e aumenti la distanza tra campioni di classi diverse.

Una volta calcolati i grafici questi sono stati confrontati nella versione "prima" e "dopo" la normalizzazione. Data la dimensione notevole del database non è possibile riportare i grafici in questa tesi senza perdere qualità. Se lo si desidera sono disponibili nel CD-ROM allegato alla tesi (vedi elenco dei file in appendice). Segue una descrizione a parole dei risultati per ciascuna tipologia di grafico:

- MvA plot: il miglioramento non sembra significativo per nessuna normalizzazione

- Boxplot: nella quantile normalization viene "appiattito", presenta varianza ridotta per normalizzazioni somma e media e varianza aumentata per le normalizzazioni pooled
- Corrgram: peggiora leggermente per la quantile normalization, mentre per le altre rimane uguale ai dati originali
- Distanza Euclidea: il cambiamento non sembra significativo per nessuna normalizzazione

In conclusione è stato scelto di non effettuare nessuna normalizzazione per campioni, in quanto nessuna è risultata significativamente utile.

Capitolo 5

Rimozione di features chiaramente discriminanti

Per avere un feedback iniziale sulla difficoltà nel classificare i campioni si è scelto di utilizzare un classificatore NaiveBayes, allenato sul 70% del database e testato sul rimanente 30%. I risultati sono fin troppo ottimistici: tutti i campioni sono classificati correttamente. Dopo una attenta analisi delle masse ci si è accorti che in questo database sono presenti alcune feature che sono chiaramente discriminanti, ovvero la loro intensità varia moltissimo tra campioni freschi e campioni decongelati, permettendo di classificare i campioni senza nemmeno usare un classificatore. Questa però è una situazione "fortunata", molto spesso infatti non si riesce a trovare nessuna feature in grado di classificare correttamente i campioni presa singolarmente, come confermato dal dott. Roberto Piro. L'obiettivo di questa tesi è trovare un classificatore che sia il più generale possibile, quindi si è scelto di creare artificialmente una situazione più complicata (ma più verosimile) togliendo le feature chiaramente discriminanti. Segue una descrizione del procedimento adottato.

5.1 Primo tentativo: Naive Bayes

Come primo approccio si è scelto di allenare con il 70% dei campioni un classificatore NaiveBayes su ciascuna delle 2959 features. Sono state scartate poi quelle features che riuscivano a classificare correttamente più dell'80% dei rimanenti campioni e si è proceduto con la classificazione come riportato nel capitolo 6.

Questo approccio, sebbene di rapida esecuzione, non si è rivelato quello più appropriato perchè selezionava come buone features delle features che anche visivamente non erano per nulla in grado di classificare (in pratica, basandosi solo sulla probabilità condizionata, capitava spesso che selezionasse features come buone casualmente).

5.2 Secondo tentativo: SVM lineare

Come secondo approccio, rivelatosi vincente, è stata implementata la seguente pipeline:

1. Per ciascuna features è stato allenato un classificatore SVM lineare mediante 5-fold cross validation sull'intero dataset, sfruttando la funzione *train* di R (vedi appendice);

2. Sono state selezionate e rimosse quelle features che presentavano accuracy (mediata sulle cinque cross validations) maggiore dell'80%

Questo metodo ha permesso di selezionare molto bene quelle features che, anche visivamente, erano in grado di classificare i campioni. Due esempi di tali features sono riportati nelle figure 5.1 e 5.2.

La soglia dell'80% in entrambi i tentativi è stata scelta come compromesso tra l'eliminare abbastanza features discriminanti (che avrebbero polarizzato successivamente il classificatore) ma non troppe per non rischiare di restare senza features buone per la classificazione.

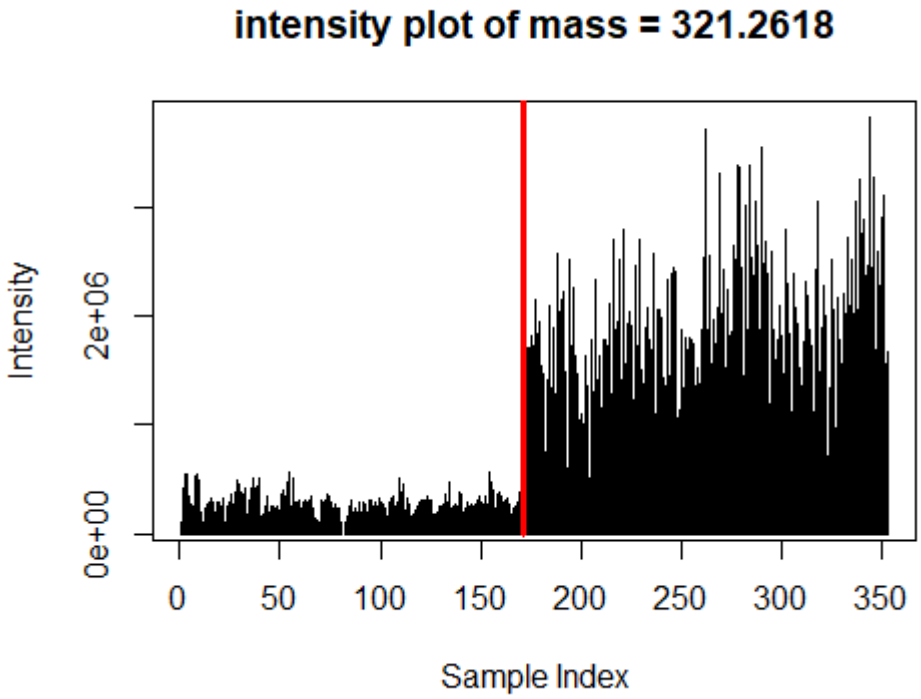


Figura 5.1: Come si può vedere questa massa è presente in quantità nettamente superiore nei campioni decongelati rispetto a quelli freschi

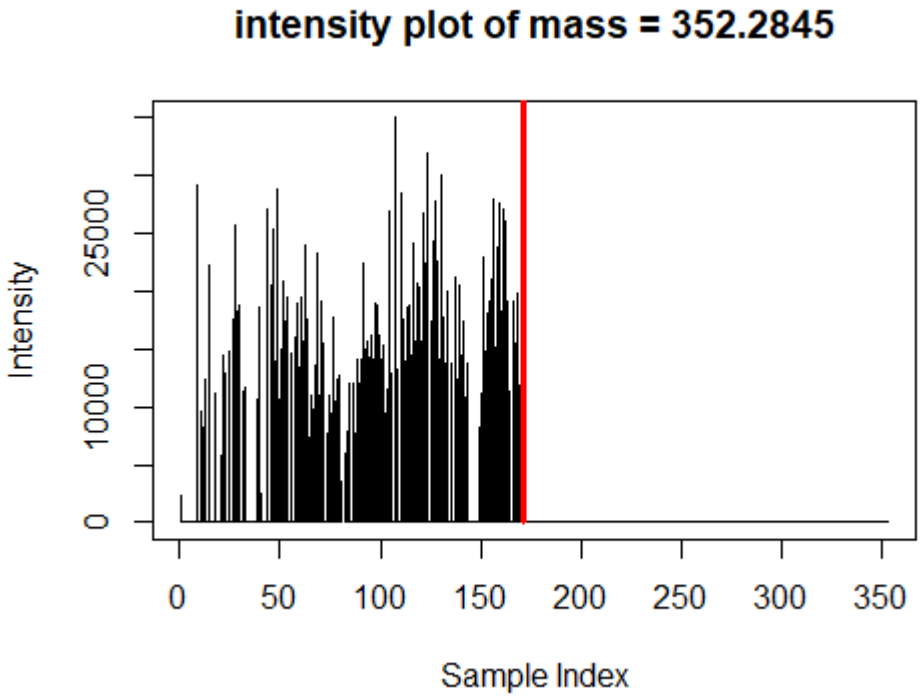


Figura 5.2: Questa massa non è mai presente nei campioni decongelati

Capitolo 6

Classificazione

Per la classificazione è stato realizzato un protocollo di analisi ispirato a quello proposto in un paper del 2015 per il riconoscimento di marcatori tossicogenomici in bovini da carne in seguito a trattamenti con corticosteroidi[3], opportunamente modificato ed adattato al problema in esame.

Segue una descrizione di tutti gli step della classificazione.

6.1 Preprocessing

6.1.1 Eliminazione features discriminanti

Come discusso precedentemente vengono rimosse le feature discriminanti, per creare una situazione più verosimile.

6.1.2 Clustering per isotopi

Le features rimanenti vengono poi clusterizzate in modo che features simili (isotopi) cadano all'interno di uno stesso gruppo. Si sa infatti che, a causa del carbonio (presente nelle molecole organiche) nello spettro saranno presenti sia le molecole con carbonio-12 (il 98.89% del carbonio presente in natura[1]) sia quelle con

[1]https://www.wolframalpha.com/input/?i=carbon-12+-+carbon-13

carbonio-13 (l'1.11% del totale[2]), ad una distanza tra loro di 1.00287 a meno della tolleranza dello spettrometro. Gli isotopi portano la stessa informazione (a meno di un fattore di scala i grafici delle intensità di due isotopi nei vari campioni sono identici), quindi vanno rimossi per ridurre la ridondanza nel classificatore. Per farlo viene calcolata la matrice di cross-correlazione tra le features rimanenti dal punto 6.1.1 e vengono definite isotopi "putativi" quelle features che presentano una correlazione $= 0.9958$. Questa soglia è stata scelta dopo aver verificato che venissero selezionati correttamente solo isotopi: abbassandola, infatti, succede che vengono selezionati come isotopi features che chiaramente non lo sono, essendo troppo distanti nello spettro.

Una volta individuate le coppie di isotopi si procede tramite grafo a mettere assieme le coppie: se A e B sono isotopi e B e C sono isotopi allora A,B e C vengono raggruppati assieme. Infine di ciascun gruppo si tiene solo la prima feature (corrisponde alla quella con peso inferiore nel gruppo) e vengono rimosse le altre.

6.2 Suddivisione dei dati

Dal database così ottenuto, composto da 353 campioni di branzini, ciascuno con il suo spettro di 2723 m/z (tolte le 180 discriminanti e gli isotopi), vengono creati:

- un database di test finale composto da 35 campioni freschi e 35 campioni congelati (in tutto sono il 20% dei campioni totali) che verrà utilizzato per valutare le performance del classificatore ottimizzato (quello che verrà utilizzato in futuro per classificare campioni di cui non si conosce la label)
- un database di train composto dai rimanenti 283 campioni.

6.3 Bootstrap

Per 100 volte viene ripetuta la seguente pipeline di bootstrap:

[2]https://www.wolframalpha.com/input/?i=carbon-12+-+carbon-13

6.3.1 Suddivisione dati di bootstrap e normalizzazione

Viene creato un database di train interno btrain ottenuto campionando con ripetizione il database di train ottenuto precedentemente e un database di test interno btest composto da campioni non selezionati per il btrain (con l'accortezza di non prendere gli stessi campioni sia in btrain che in btest). Successivamente viene normalizzato il btrain per features (media nulla e varianza unitaria) e la stessa normalizzazione (usando media e varianza del btrain) viene applicata anche al btest (per simulare quello che avverrebbe nella realtà).

6.3.2 5-fold cross validation per i parametri del classificatore

Tramite 5-Fold Cross Validation si cerca il costo ottimo C per la linear SVM utilizzando il database btrain.

6.3.3 Entropian Recursive Feature Elimination

Una volta trovato il C ottimo vengono ordinate le features dalla più importante alla meno importante tramite Entropian Recursive Feature Elimination (le 100 features più importanti sono ordinate tramite Recursive Feature Elimination) utilizzando il database btrain. L'algoritmo viene riportato in appendice.

6.3.4 Train e test delle SVM

Sempre utilizzando il C ottimo si allenano 100 SVM lineari sul btrain dove la i-esima SVM usa solo le i features più importanti e si usa il database btest per valutarne le performance calcolando il parametro MCC (Matthews correlation coefficient).

Il coefficiente di correlazione di Matthews viene utilizzato nell'apprendimento automatico come misura della qualità delle classificazioni binarie introdotta dal

biochimico Brian W. Matthews nel 1975[3]. Il MCC è in sostanza un coefficiente di correlazione tra le classificazioni binarie osservate e previste; restituisce un valore compreso tra -1 e $+1$. Un coefficiente di $+1$ rappresenta una previsione perfetta, 0 non migliore della previsione casuale e -1 indica un totale disaccordo tra previsione e osservazione.

L'MCC è generalmente considerato come una delle migliori misure in grado di riassumere la tabella di verità in un unico numero [15].

E' definito come:

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

In questo modo si ottiene un grafico di MCC al variare del numero di features "ranked" usate.

6.4 Selezione numero ottimo di features

Una volta ottenuti i 100 grafici di MCC in funzione del numero di ranked features utilizzate, uno per ogni bootstrap, questi sono mediati (vedi 6.1) per scegliere il numero ottimo di features (il più piccolo numero di features che dà le più alte performance) da utilizzare per il classificatore finale, quello che verrà effettivamente utilizzato quando si dovrà classificare un nuovo campione.

6.5 Allenamento classificatore finale e valutazione delle performance

Si allena infine il classificatore finale con il C ottimo e il numero ottimo di features sul database di train completo ricavato al punto 1 e se ne valutano le performance sul database di test ricavato sempre al punto 1.

Il procedimento è ben rappresentato nella figura 6.2

[3]https://en.wikipedia.org/wiki/Matthews_correlation_coefficient

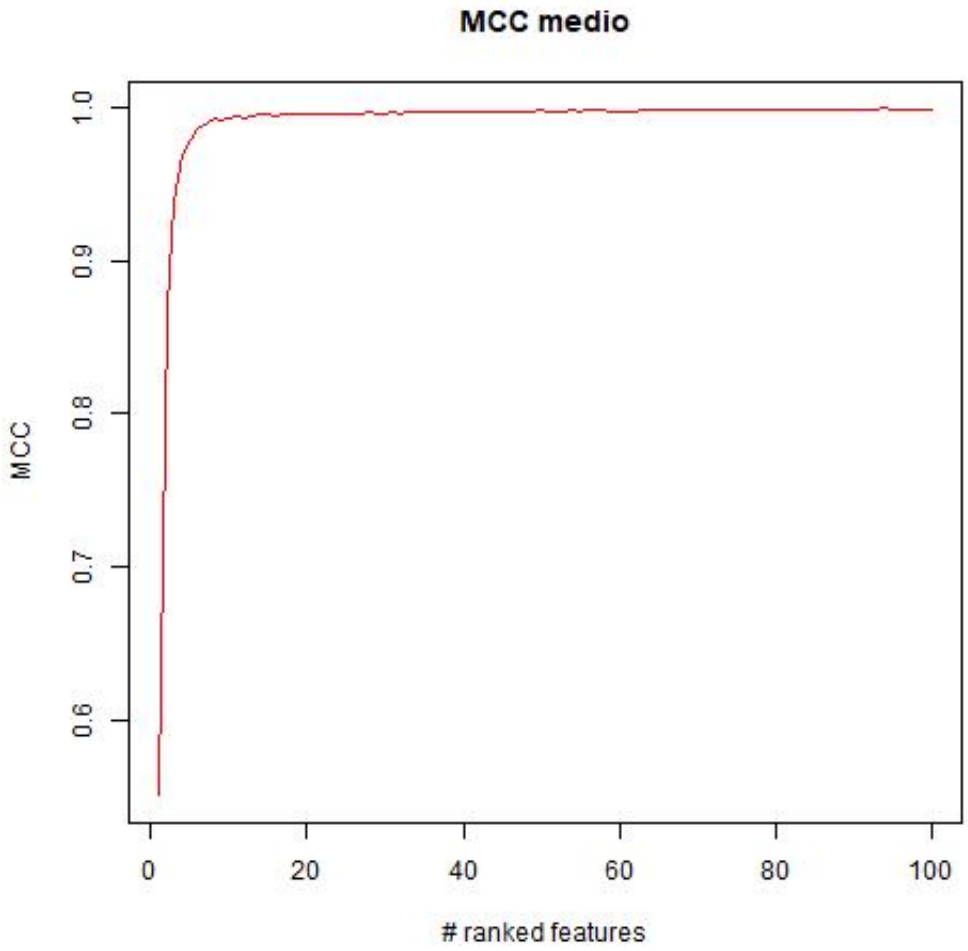


Figura 6.1: grafico MCC medio

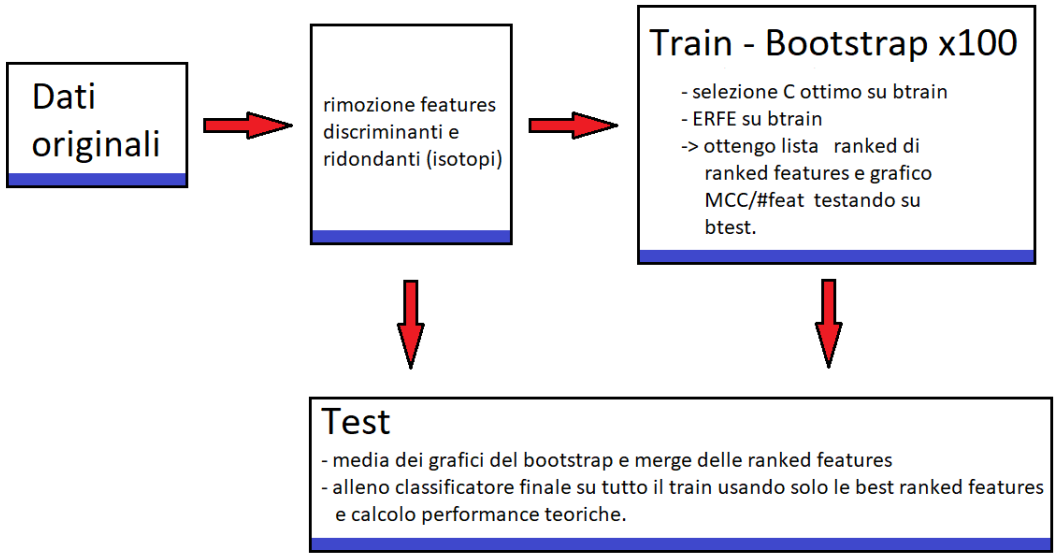


Figura 6.2: Diagramma di funzionamento dell'algoritmo

Capitolo 7

Discussione dei risultati ottenuti

Il classificatore finale, allenato su sole 14 *best features* opportunamente selezionate (in seguito a rimozioni e ordinamento globale) e su 283 dei 353 campioni, ha ottenuto uno score MCC del 97% sui restanti 70 campioni mai utilizzati nella procedura di messa a punto del classificatore. Questo fa ben sperare per successive applicazioni di questo algoritmo ad altri dataset.

Quando si dovrà utilizzare il classificatore per decidere la classe di nuovi campioni si potranno comunque usare le features discriminanti trovate in precedenza (dopo aver verificato che sono effettivamente discriminanti e non legate a questo specifico dataset).

7.1 I campioni più *difficili*

E' interessante vedere quali sono i campioni che, durante i vari bootstrap, sono stati classificati erroneamente più volte. Ce ne sono due che sono nettamente più *difficili* da classificare rispetto agli altri, e sono il *congelati_b_etac_pos..59.* e il *FRESCO_a_etac_pos..72..* Entrambi sono caratterizzati da pochi valori di intensità molto elevati, probabilmente dovuti a problemi con lo spettrometro. In figura 7.1 sono riportati due campioni di esempio confrontati con i due più difficili da classificare.

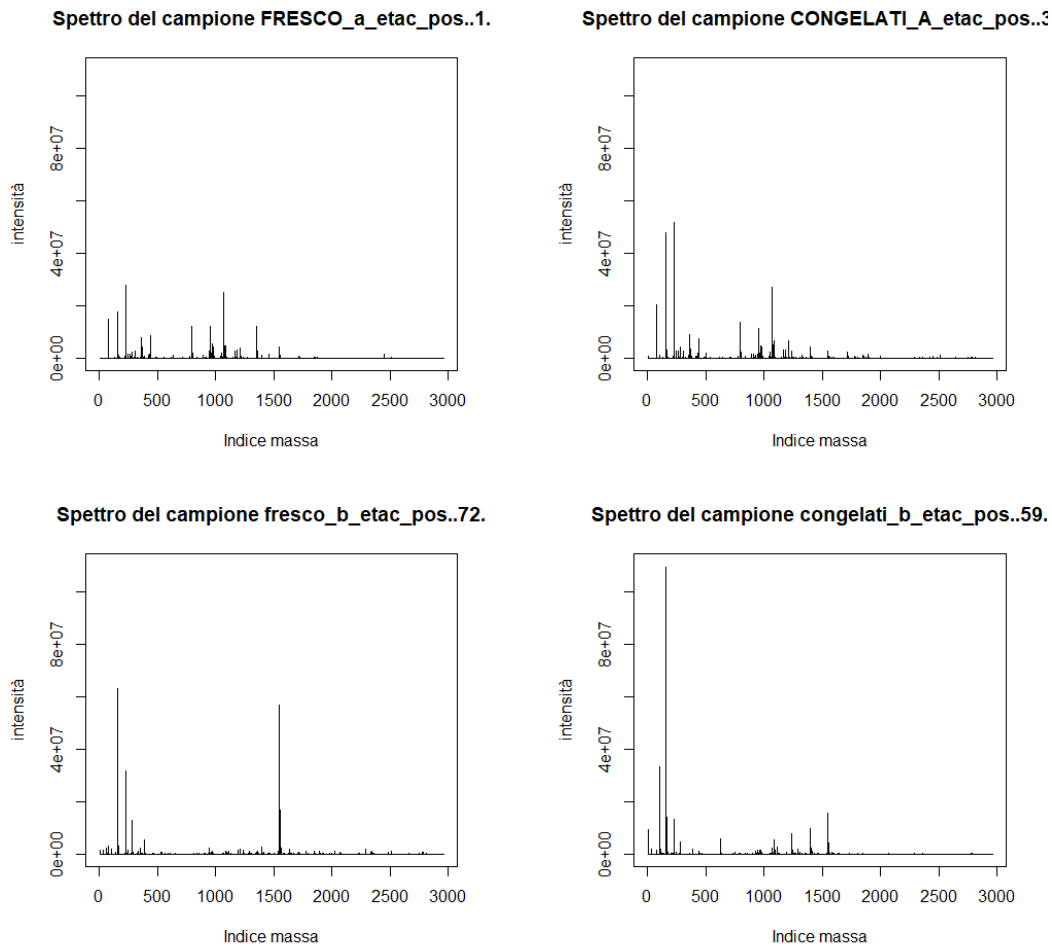


Figura 7.1: I due campioni più difficili da classificare (in basso) confrontati con due campioni di esempio (in alto)

Appendice A

Codice utilizzato nella tesi

Segue una presentazione del codice utilizzato per l'analisi e la visualizzazione dei dati. Delle funzioni più semplici si riporta solamente una breve descrizione, per maggiori informazioni si rimanda al CD-ROM allegato alla tesi.

A.1 Elenco completo di tutti i file della tesi

Segue un elenco di tutti i file e le cartelle creati per svolgere la tesi. Alcune precisazioni:

- I file in *corsivo* non sono utilizzati nella versione finale, ma sono stati utili per analizzare inizialmente i dati.
- Nella cartella *confronto*, i file in formato *pdf* sono grafici diagnostici creati con il file *plotconfronto.R* e i file in formato *csv* sono il risultato delle normalizzazioni di MetaboAnalyst.
- il file *main.R* crea alcuni grafici diagnostici. Utili per avere una visione iniziale dei dati, di questi sono stati conservati solo *Masse presenti solo nei freschi.pdf* e *Masse presenti solo nei congelati.pdf*. Se si desidera vedere tutti i plot si rimanda al CD-ROM.

```
/main
└─ /classificazione
```

```
    ├── erfe.R
    ├── k_fold_cv.R
    ├── MCC.R
    ├── remove_features.R
    └── svm_rfe.R
└── /confronto
    ├── /mediannorm
    │   ├── boxplot.pdf
    │   ├── corrgram.pdf
    │   ├── corrgram_diff.pdf
    │   ├── corrgram_diff_Cong.pdf
    │   ├── corrgram_diff_Freschi.pdf
    │   ├── corrgram_full.pdf
    │   ├── corrgram_full_Freschi.pdf
    │   ├── corrgramfull_Cong.pdf
    │   ├── data_normalized.csv
    │   ├── data_original.csv
    │   ├── data_processed.csv
    │   ├── euclideandist.pdf
    │   └── MvA.pdf
    ├── /pooledcongelatinorm
    │   ├── boxplot.pdf
    │   ├── corrgram.pdf
    │   ├── corrgram_diff.pdf
    │   ├── corrgram_diff_Cong.pdf
    │   ├── corrgram_diff_Freschi.pdf
    │   ├── corrgram_full.pdf
    │   ├── corrgram_full_Freschi.pdf
    │   ├── corrgramfull_Cong.pdf
    │   ├── data_normalized.csv
    │   ├── data_original.csv
    │   ├── data_processed.csv
    │   ├── euclideandist.pdf
    │   └── MvA.pdf
    └── /pooledfreschinorm
        ├── boxplot.pdf
        ├── corrgram.pdf
        ├── corrgram_diff.pdf
        ├── corrgram_diff_Cong.pdf
        ├── corrgram_diff_Freschi.pdf
        ├── corrgram_full.pdf
        ├── corrgram_full_Freschi.pdf
        ├── corrgramfull_Cong.pdf
        ├── data_normalized.csv
        └── data_original.csv
```

```
    ├── data_processed.csv
    ├── euclideandist.pdf
    └── MvA.pdf
  ├── /quantilenorm
    ├── boxplot.pdf
    ├── corrgram.pdf
    ├── corrgram_diff.pdf
    ├── corrgram_diff_Cong.pdf
    ├── corrgram_diff_Freschi.pdf
    ├── corrgram_full.pdf
    ├── corrgram_full_Freschi.pdf
    ├── corrgramfull_Cong.pdf
    ├── data_normalized.csv
    ├── data_original.csv
    ├── data_processed.csv
    ├── euclideandist.pdf
    └── MvA.pdf
  └── /sumnorm
    ├── boxplot.pdf
    ├── corrgram.pdf
    ├── corrgram_diff.pdf
    ├── corrgram_diff_Cong.pdf
    ├── corrgram_diff_Freschi.pdf
    ├── corrgram_full.pdf
    ├── corrgram_full_Freschi.pdf
    ├── corrgramfull_Cong.pdf
    ├── data_normalized.csv
    ├── data_original.csv
    ├── data_processed.csv
    ├── euclideandist.pdf
    └── MvA.pdf
├── /dati_originali
  ├── Analisi descrittiva.pdf
  ├── data_original_etac_pos.csv
  ├── image.scale.R
  ├── Masse presenti solo nei congelati.pdf
  └── Masse presenti solo nei freschi.pdf
└── /my_functions
  ├── boxplotfun.R
  ├── corplot.R
  ├── crosscorrelazioni.R
  ├── crosssv.R
  ├── eucliplot.R
  ├── hardest_samples.R
  └── massplot.R
```

```
    ├── mergeFeatures.R
    ├── MSplot.R
    ├── mvaplot.R
    ├── mvatesiplot.R
    ├── myImagePlot.R
    ├── parfunctions.R
    ├── pcaplot.R
    ├── plotconfronto.R
    ├── plotFC.R
    ├── plotMap.R
    ├── plotMap_tpeak.R
    └── SVMrem.R
  └── /risultati
├── confrontoNorm.R
├── initial_remove.R
├── main.R
├── main.Rproj
├── naiveBayes.R
└── SVMclassifier.R
```

A.2 MetaboAnalyst

Segue il codice realizzato per confrontare le normalizzazioni per campioni di MetaboAnalyst

A.2.1 confrontoNorm.R

Crea i grafici per tutte e 5 le normalizzazioni di MetaboAnalyst. Sfrutta i database che restituisce MetaboAnalyst in seguito a ciascuna normalizzazione e la funzione *plotconfronto.R* (vedi A.2.2).

```
## confronto le normalization by sample


source('my_functions/mvaplot.R')
source('my_functions/boxplotfun.R')
source("my_functions/corplot.R")
source("my_functions/eucliplot.R")
```

```
source("my_functions/parfunctions.R")
source("my_functions/plotconfronto.R")
library(data.table)

ind_freschi <- seq(1,170)
ind_congelati <- seq(171,353)

##########################
# quantile normalization #
##########################
#or_data <- read.csv("data_original.csv",stringsAsFactors = FALSE)
pr_data <- read.csv("confronto/quantilenorm/data_processed.csv",
    stringsAsFactors = FALSE)
nr_data <- read.csv("confronto/quantilenorm/data_normalized.csv",
    stringsAsFactors = FALSE)
plotconfronto("confronto/quantilenorm")

####################
# sum normalization #
####################
#rm(or_data)
rm(pr_data)
rm(nr_data)
#or_data <- read.csv("data_original.csv",stringsAsFactors = FALSE)
pr_data <- read.csv("confronto/sumnorm/data_processed.csv",
    stringsAsFactors = FALSE)
nr_data <- read.csv("confronto/sumnorm/data_normalized.csv",
    stringsAsFactors = FALSE)
plotconfronto("confronto/sumnorm")
```

```
########################
# median normalization #
########################
#rm(or_data)
rm(pr_data)
rm(nr_data)
#or_data <- read.csv("data_original.csv",
    stringsAsFactors = FALSE)
pr_data <- read.csv("confronto/mediannorm/data_processed.csv",
    stringsAsFactors = FALSE)
nr_data <- read.csv("confronto/mediannorm/data_normalized.csv",
    stringsAsFactors = FALSE)
plotconfronto("confronto/mediannorm")

################################
# pooled freschi normalization #
################################
#rm(or_data)
rm(pr_data)
rm(nr_data)
#or_data <- read.csv("data_original.csv",
    stringsAsFactors = FALSE)
pr_data <- read.csv("confronto/pooledfreschinorm/data_processed.csv",
    stringsAsFactors = FALSE)
nr_data <- read.csv("confronto/pooledfreschinorm/data_normalized.csv",
    stringsAsFactors = FALSE)
plotconfronto("confronto/pooledfreschinorm")

##################################
# pooled congelati normalization #
```

```
##################################
#rm(or_data)
rm(pr_data)
rm(nr_data)
#or_data <- read.csv("data_original.csv",
    stringsAsFactors = FALSE)
pr_data <- read.csv("confronto/pooledcongelatinorm/data_processed.csv",
    stringsAsFactors = FALSE)
nr_data <- read.csv("confronto/pooledcongelatinorm/data_normalized.csv",
    stringsAsFactors = FALSE)
plotconfronto("confronto/pooledcongelatinorm")
```

A.2.2 plotconfronto.R

Aggregatore di funzioni plot, vedi A.2.3, A.2.4, A.2.5, A.2.6.

A.2.3 mvaplot.R

```
    mvaplot <- function(sample1,sample2) {
  # m <- log2(as.numeric(sample1)/as.numeric(sample2))
  # a <- log2(sqrt(abs(as.numeric(sample1)-as.numeric(sample2))))
  a <- 0.5*(log2(as.numeric((sample1)))+log2(as.numeric(sample2)))
  m <- log2(sample1)-log2(sample2)
  plot(a,m,pch = 19,cex=0.3,main="MvA plot",
    sub=paste(rownames(sample1)," vs ",rownames(sample2)),)
}
```

A.2.4 boxplotfun.R

Permette di ottenere tutti i grafici boxplot, di cui sono alcuni sono stati mostrati nella tesi.

```
boxplotfun <- function(pr_table,nr_table,ind_freschi,ind_congelati,
```

```
  dir_tosave) {
pdf(file=paste(dir_tosave,"/boxplot.pdf",sep = ""), paper="a4")
layout(matrix(c(1,2,1,2),2,2,byrow=TRUE))# par(omi = rep(.5, 4))

for (i in 1:5) {
    boxplot(t(pr_table[((1+34*(i-1)):(34*i)),]),col="dodgerblue3",
      outpch = 20, outcex=0.5, outcol="dodgerblue3",
      boxcol="dodgerblue3", outline=FALSE)
    title(main=paste("Boxplot freschi (prima) da ",
      as.character(1+34*(i-1))," a ",(34*i)),
          xlab="campioni",ylab="intensità")

    boxplot(t(nr_table[((1+34*(i-1)):(34*i)),]),col="dodgerblue3",
      outpch = 20, outcex=0.5,
      outcol="dodgerblue3",
      boxcol="dodgerblue3",outline=FALSE)
    title(main=paste("Boxplot freschi (dopo) da ",
      as.character(1+34*(i-1))," a ",(34*i)),
        xlab="campioni",ylab="intensità")
    }

for (i in 1:3) {
  boxplot(t(pr_table[((171+61*(i-1)):(170+(61*i))),]),
      col="goldenrod2", outpch = 20, outcex=0.5,
      outcol="goldenrod2", boxcol="goldenrod2",outline=FALSE)
  title(main=paste("Boxplot cong. (prima) da ",
      as.character(1+34*(i-1))," a ",(34*i)),
      xlab="campioni",ylab="intensità")
  boxplot(t(nr_table[((171+61*(i-1)):(170+(61*i))),]),
      col="goldenrod2", outpch = 20, outcex=0.5,
```

```
        outcol="goldenrod2",
        boxcol="goldenrod2",outline=FALSE)
        title(main=paste("Boxplot cong. (dopo) da ",
        as.character(1+34*(i-1))," a ",(34*i)),
        xlab="campioni",ylab="intensità")
  }
  dev.off()
}
```

A.2.5 corplot.R

```
    corplot <- function(pr_table,nr_table,dir_tosave) {
  rnp <- rownames(pr_table)
  cnp <- colnames(nr_table)
  rnn <- rownames(nr_table)
  cnn <- colnames(nr_table)
  pr_table <- transpose(as.data.frame(apply(pr_table,1,normalizzo)))
    #media 0 var 1
  rownames(pr_table) <- rnp
  colnames(pr_table) <- cnp
  nr_table <- transpose(as.data.frame(apply(nr_table,1,normalizzo)))
    #media 0 var 1
  rownames(nr_table) <- rnn
  colnames(nr_table) <- cnn
  rb <- colorRampPalette(c("darkred","red","darkblue","dodgerblue3"))

  curr1f <- pr_table[1:170,]
  curr1c <- pr_table[171:353,]
  curr2f <- nr_table[1:170,]
  curr2c <- nr_table[171:353,]
  x <- rownames(curr1f)
```

```
rownames(curr1f) <- paste(substr(x,1,1),substr(x, nchar(x)-3+1,
  nchar(x)),sep="")
rownames(curr2f) <- rownames(curr1f)
x <- rownames(curr1c)
rownames(curr1c) <- paste(substr(x,1,1),substr(x, nchar(x)-3+1,
  nchar(x)),sep="")
rownames(curr2c) <- rownames(curr1c)
c1f <- cor(t(curr1f),use = "complete.obs")
c1c <- cor(t(curr1c),use = "complete.obs")
c2f <- cor(t(curr2f),use = "complete.obs")
c2c <- cor(t(curr2c),use = "complete.obs")
df <- c2f-c1f
dc <- c2c-c1c
pdf(file=paste(dir_tosave,'/corrgram_diff_Freschi.pdf',sep = ""),
  width = 100, height = 100,title='Corrgram differenza Nr-Pr')
corrplot.mixed(df,upper="color",lower = "number",lower.col = "black",
  col.lab="black",number.cex=0.6, tl.cex=0.7)
dev.off()
pdf(file=paste(dir_tosave,'/corrgram_diff_Cong.pdf',sep = ""),
  width = 100, height = 100,title='Corrgram differenza Nr-Pr')
corrplot.mixed(dc,upper="color",lower = "number",lower.col = "black",
  col.lab="black",number.cex=0.6, tl.cex=0.7)
dev.off()
pdf(file=paste(dir_tosave,'/corrgram_full_Freschi.pdf',sep = ""),
  width = 100, height = 100,title='Corrgram finale')
corrplot.mixed(c2f,upper="color",lower = "number",lower.col = "black",
  col.lab="black",number.cex=0.6, tl.cex=0.7)
dev.off()
pdf(file=paste(dir_tosave,'/corrgramfull_Cong.pdf',sep = ""),
  width = 100, height = 100,title='Corrgram finale')
```

```
  corrplot.mixed(c2c,upper="color",lower = "number",lower.col = "black",
    col.lab="black",number.cex=0.6, tl.cex=0.7)
  dev.off()
}
```

A.2.6 eucliplot.R

Plotta la distanza euclidea tra i campioni dopo averli normalizzati.

```
    eucliplot <- function(pr_table,nr_table,dir_tosave) {
library(fields)
rnp <- rownames(pr_table)
cnp <- colnames(nr_table)
rnn <- rownames(nr_table)
cnn <- colnames(nr_table)
pr_table <- transpose(as.data.frame(apply(pr_table,1,normalizzo)))
    #media 0 var 1
rownames(pr_table) <- rnp
colnames(pr_table) <- cnp
nr_table <- transpose(as.data.frame(apply(nr_table,1,normalizzo)))
    #media 0 var 1
rownames(nr_table) <- rnn
colnames(nr_table) <- cnn

palette <- hcl.colors(20, palette="viridis",alpha=NULL,rev = FALSE,
fixup = TRUE)
pdf(file=paste(dir_tosave,'/euclideandist.pdf',sep = ""),width = 200,
    height = 100,title='Plot distanza euclidea prima e dopo')
dst1 <- dist(pr_table, method = "euclidean", diag = FALSE,
    upper = FALSE, p = 2)
dst1 <- data.matrix(dst1)
```

```
dim <- ncol(dst1)
image(1:dim, 1:dim, dst1, axes = FALSE, xlab="", ylab="",col=palette)
image.plot(1:dim,1:dim,dst1,add=TRUE,legend.only=TRUE,legend.width=5,
    legend.shrink=.5, horizontal=TRUE,col=palette)
pnames <- paste(substr(rnp,1,1),substr(rnp,nchar(rnp)-3+1,
    nchar(rnp)),sep="")
axis(1, 1:dim, pnames, cex.axis = 0.5, las=3)
axis(2, 1:dim, pnames, cex.axis = 0.5, las=1)
#text(expand.grid(1:dim, 1:dim), sprintf("%0.1f", dst), cex=0.5)

dst2 <- dist(nr_table, method = "euclidean", diag = FALSE,
    upper = FALSE, p = 2)
dst2 <- data.matrix(dst2)
dim <- ncol(dst2)

image(1:dim, 1:dim, dst2, axes = FALSE, xlab="", ylab="",col=palette)
image.plot(1:dim,1:dim,dst2,add=TRUE,legend.only=TRUE,
    legend.width=5, legend.shrink=.5, horizontal=TRUE,col=palette)
nnames <- paste(substr(rnn,1,1),substr(rnn,nchar(rnn)-3+1,
    nchar(rnn)),sep="")
axis(1, 1:dim, nnames, cex.axis = 0.5, las=3)
axis(2, 1:dim, nnames, cex.axis = 0.5, las=1)

diff <- dst2-dst1
image(1:dim, 1:dim, diff, axes = FALSE, xlab="", ylab="",col=palette)
image.plot(1:dim,1:dim,diff,add=TRUE,legend.only=TRUE,
    legend.width=5, legend.shrink=.5, horizontal=TRUE,col=palette)
nnames <- paste(substr(rnn,1,1),substr(rnn,nchar(rnn)-3+1,
    nchar(rnn)),sep="")
axis(1, 1:dim, nnames, cex.axis = 0.5, las=3)
```

```
axis(2, 1:dim, nnames, cex.axis = 0.5, las=1)
#text(expand.grid(1:dim, 1:dim), sprintf("%0.1f", dst), cex=0.5)
dev.off()
}
```

A.3 Classificazione

A.3.1 SVMClassifier.R

Il classificatore vero e proprio. A partire dal database iniziale restituisce il classificatore ottimo, le sue performance stimate, tutti i classificatori intermedi e e le loro performance.

```
    source("my_functions/MSplot.R")
source("my_functions/parfunctions.R")
source("my_functions/plotFC.R")
source("my_functions/massplot.R")
source('my_functions/mvaplot.R')
source('my_functions/pcaplot.R')
source('my_functions/boxplotfun.R')
source('my_functions/mergeFeatures.R')
source('my_functions/plotMap.R')
source('my_functions/plotMap_tpeak.R')
library('data.table')
#library("factoextra")#serve per PCAplot
#library('caret')
#library('mlbench')
library('snow')
library('doSNOW')
library('e1071')
library('igraph')
```

```
source("classificazione/erfe.R")
source("classificazione/svm_rfe.R")
source("classificazione/MCC.R")
source("classificazione/k_fold_cv.R")


## Implemento parallelizzazione (dimezza circa il tempo di
    esecuzione)
cl <<- makeCluster(4)
registerDoSNOW(cl)
clusterExport(cl,list("svm","MCC"))
# library(doMC)
# registerDoMC(cores = 4)


################################
##carico i dati e li sistemo ##
################################
peaktable <- read.csv("dati_originali/data_original_etac_pos.csv",
    stringsAsFactors = FALSE)
sizept <- dim(peaktable)
labels <- peaktable[1,seq(2,sizept[2])]
# labels[which(labels=="congelati_etac_pos")] <- -1
# labels[which(labels=="branzini_fresci_etac_pos")] <- 1
mass <- as.numeric(peaktable[2:sizept[1],1])
tpeaktable <- transpose(peaktable[2:sizept[1],2:sizept[2]])
tpeaktable <- transpose(as.data.frame(apply(tpeaktable,1,numerize)))
tpeaktable[is.na(tpeaktable)] <- 0
rownames(tpeaktable) <- colnames(peaktable)[2:sizept[2]]
```

```
## Normalizzazione ##########################################
#tpeaktable <- as.data.frame(apply(tpeaktable,2,normalizzo)) #
############################################################

# ### Shuffled labels #######
# set.seed(1)               #
# labels <- sample(labels)  #
# ###########################

tpeaktable <- cbind(tpeaktable,as.data.frame(t(labels)))
colnames(tpeaktable)[sizept[1]] <- "Label"



# load('NB_acuracy.RData')
# # tpeaktable <- tpeaktable[,-which(disc_feat)]
# # mass <- mass[-which(disc_feat)]
# tpeaktable <- tpeaktable[,-which(acuracy>0.8)]
# mass <- mass[-which(acuracy>0.8)]
load('SVM_5cv_accuracy.RData')
tpeaktable <- tpeaktable[,-which(accuracy>0.8)]
mass <- mass[-which(accuracy>0.8)]
## ########################################################
## Seleziono test_dataset finale (da fare dopo bootsrap) ##
## ########################################################
final_test_indices <- c(1:35,319:353)
#final_test_dataset <- tpeaktable[final_test_indices,]
complete_train_indices <- 1:(sizept[2]-1)
complete_train_indices <- complete_train_indices[-final_test_indices]
complete_train_dataset <- tpeaktable[complete_train_indices,]
```

```
##Creo combinazioni bootstrap
set.seed(1)
B <- 100
b_trains <- rep(NA,B)
ft <- function(x) {
  x <- round(runif(length(complete_train_indices),
    max=length(complete_train_indices),min=1))
}
b_trains <- lapply(b_trains,ft)



classifiers <- list()
#predictions <- list()
unbiased_MCC <- list()
risultato_classificazione <- list()
ranked_features_total <- list()
progression<-winProgressBar(title = paste("Bootstrap"),
    min = 0,max = 3*B , width = 300)

#cluster di features
cc <- cor(complete_train_dataset[,1:(dim(complete_train_dataset)[2]-1)],
    use="complete.obs",method="pearson")
cc[upper.tri(cc,diag = T)]<-NA
couples <- arrayInd(which(abs(cc)>0.9958),dim(cc))
    #valutato con massplot, approccio conservativo
data <- matrix(as.character(couples),ncol=2)
#make a graph
gg <- graph.edgelist(data, directed=F)
gruppi_masse <- split(V(gg)$name, clusters(gg)$membership)
```

```
gm <- as.numeric(unlist(gruppi_masse))

#Visto che ho trovato cluster di isotopi, tengo solo la prima
#massa del cluster ed elimino le altre
to_mantain <- foreach(i=1:length(gruppi_masse)) %dopar%{
    return(gruppi_masse[[i]][1])}
to_mantain <- as.numeric(unlist(to_mantain))
to_del <- setdiff(gm,to_mantain)

#aggiorno i dataset
complete_train_dataset <- complete_train_dataset[,-to_del]
#final_test_dataset <- final_test_dataset[,-to_del]
mass_after_cluster <- mass[-to_del] #usato solo con massplot
tpeaktable_after_cluster <- tpeaktable[,-to_del]



nfeat_MCC <- 100 #Per il plot del MCC
for (b in 1:B) {
  pv <- getWinProgressBar(progression)
  setWinProgressBar(progression,pv+1,title=paste("Bootstrap",b,"di",
    B,"- fase 1/3"))
  indici_train <- b_trains[[b]]
  train_dataset <- complete_train_dataset[indici_train,]
  k <- unique(train_dataset)
  w <- setdiff(rownames(complete_train_dataset),rownames(k))
  test_dataset <- complete_train_dataset[w,]

  #normalizzo train (media 0 var 1) e applico la stessa
  #normalizzazione al test
  medie_train <- apply(train_dataset[,c(1:(length(
```

```
  mass_after_cluster)))], 2, media)
varianze_train <- apply(train_dataset[,c(1:(length(
  mass_after_cluster)))], 2,varianza)
labels_train <- as.data.frame(train_dataset[,
  length(mass_after_cluster)+1])
labels_test <- as.data.frame(test_dataset[,
  length(mass_after_cluster)+1])
train_dataset <- cbind(as.data.frame(t(apply(train_dataset[,
  c(1:(length(mass_after_cluster)))],1,
  kfunction,medie=medie_train,varianze=varianze_train))),
  labels_train)
colnames(train_dataset)[length(mass_after_cluster)+1] <- "Label"
test_dataset <- cbind(as.data.frame(t(apply(test_dataset[,
  c(1:(length(mass_after_cluster)))],1,
  kfunction,medie=medie_train,varianze=varianze_train))),
  labels_test)
colnames(test_dataset)[length(
  mass_after_cluster)+1] <- "Label"

#cluster di features
#cc <- cor(train_dataset[,1:(dim(complete_train_dataset)[2]-1)],
  use="complete.obs",method="pearson")

#5-fold cross validation
# best_parameters <- k_fold_cv(train_dataset)
# best_costo <- best_parameters[1]
# print(paste("il miglior MCC del bootstrap",b,"è",
  best_parameters[2],"con costo",best_parameters[1]))
best_costo <- 1
```

```
setWinProgressBar(progression,pv+2,title=paste("Bootstrap",b,"di",
  B,"- fase 2/3"))
ranked_features <- erfe(train_dataset,best_costo,gruppi_masse)
ranked_features_total[[b]] <- ranked_features
setWinProgressBar(progression,pv+3,title=paste("Bootstrap",b,"di",
  B,"- fase 3/3"))

classifiers_b <- list()
unbiased_MCC_b <- list()
risultato_classificazione_b <- list()
# for (i in 1:round(length(ranked_features)/6)) {
for (i in 1:nfeat_MCC) {
  indici <- c(ranked_features[1:i],ncol(train_dataset))
  dati <- train_dataset[,indici]
  classifiers_b[[i]] <- svm(Label ~ .,data=dati,kernel = 'linear',
      cost=best_costo)
  to_test <- test_dataset[,indici]
  colnames(to_test) <- colnames(train_dataset)[indici]
  #predictions_b[[i]] <- predict(classifiers_b[[i]],to_test[-(i+1)])
  #k <- table(pred = predictions_b[[i]], true = test_dataset[,2960])
  predi <- predict(classifiers_b[[i]],to_test[-(i+1)])
  risultato_classificazione_b[[i]] <- cbind(predi,
      test_dataset[,ncol(train_dataset)])
  k <- table(pred = predi, true = test_dataset[,ncol(train_dataset)])
  unbiased_MCC_b[[i]] <- MCC(k)
}
classifiers[[b]] <- classifiers_b
#predictions[[b]] <- predictions_b
unbiased_MCC[[b]] <- unbiased_MCC_b
risultato_classificazione[[b]] <- risultato_classificazione_b
```

```
}
invisible(close(progression))

rank_list <- mergeFeatures(ranked_features_total)
first_ranked <- as.numeric(names(sort(rank_list))[1:100])

library(viridis)
c_palette <- viridis(B)
for (z in 1:B) {
  MCCplot(unlist(unbiased_MCC[[z]]),viridis_list = c_palette[z],
    folder = "risultati/MCC/",name = paste("Bootstrap",z))
}

mean_MCC <- rep(0,nfeat_MCC)
for (i in 1:B) {
  mean_MCC <- mean_MCC+unlist(unbiased_MCC[i])
}
mean_MCC <- mean_MCC/B
MCCplot(mean_MCC,viridis_list = "red",folder = "risultati/MCC/",
    name="medio")


#optim_feature_number <- which(mean_MCC==max(mean_MCC))[1]
#optim_feature_number <- which(mean_MCC[1:30]==max(
    mean_MCC[1:30]))[1]
optim_feature_number <- 14 #punto di gomito del
    #grafico MCC medio
#final evaluation
tpeaktable_after_cluster_for_evaluation <- cbind(
    tpeaktable_after_cluster[,first_ranked[1
```

```
    :optim_feature_number]],as.data.frame(t(labels)))
colnames(tpeaktable_after_cluster_for_evaluation)[
    optim_feature_number+1] <- "Label"
final_train <- tpeaktable_after_cluster_for_evaluation[
    complete_train_indices,]
final_test <- tpeaktable_after_cluster_for_evaluation[
    final_test_indices,]
colnames(final_test) <- colnames(final_train)

medie_ftrain <- apply(final_train[,c(
    1:optim_feature_number)], 2, media)
varianze_ftrain <- apply(final_train[,c(1:optim_feature_number)],
2,varianza)
labels_ftrain <- as.data.frame(final_train[,optim_feature_number+1])
labels_ftest <- as.data.frame(final_test[,optim_feature_number+1])
final_train <- cbind(as.data.frame(t(apply(final_train[,c(1:
    optim_feature_number)],1,kfunction,medie=medie_ftrain,
    varianze=varianze_ftrain))), labels_ftrain)
colnames(final_train)[optim_feature_number+1] <- "Label"
final_test <- cbind(as.data.frame(t(apply(final_test[,c(1:
    optim_feature_number)],1,kfunction,medie=medie_ftrain,
    varianze=varianze_ftrain))),labels_ftest)
colnames(final_test)[optim_feature_number+1] <- "Label"

final_classifier <- svm(Label ~ .,data=final_train,
kernel = 'linear',cost=1)



#predictions_b[[i]] <- predict(classifiers_b[[i]],to_test[-(i+1)])
#k <- table(pred = predictions_b[[i]], true = test_dataset[,2960])
```

```
predi <- predict(final_classifier,
final_test[-(optim_feature_number + 1)])
final_k <- table(pred = predi, true = final_test[,(
    optim_feature_number+1)])
final_MCC <- MCC(final_k)
```

A.3.2 erfe.R

Implementa l'algoritmo di Entropian Recusive Feature Elimination con alcune modifiche: vista la natura dei nostri dati il secondo caso dell'algoritmo (quello in cui l'entropia dei bin è inferiore alla soglia e/o la media dei pesi rimappati in $[0, 1]$ è minore della soglia) è stato riscritto in modo da non usare il logaritmo come prevede l'algoritmo originale.

```
    erfe <- function(train_dataset,best_costo,gruppi_masse) {
  Rf <- c(1:(ncol(train_dataset)-1)) #all'inizio ho tutte le feat
  Fr <- c()# nessuna nel vettore ranked
  Rt <- 100 #numero di features per cui farò RFE
  pind <- 1 #indice per plot
  while (length(Rf)>Rt) {
    col <- c(Rf,ncol(train_dataset)) #ad ogni ciclo
    #cambia la lunghezza delle feature non ranked
    S <- train_dataset[,col]
    m <- svm(Label ~ .,data=S,kernel = 'linear',cost=best_costo)
    w <- t(m[["coefs"]])%*%m[["SV"]] #calcolo i pesi
    dJ <- w^2

    if (pind==1) {
      jpeg(paste("risultati/whist","_B",b,"_n",pind,".jpg"))
      hist(dJ,50)
      dev.off()
```

```
  pind <- pind+1
}

OldRange = (max(dJ) - min(dJ))
NewRange = (1 - 0)
pw = (((dJ - min(dJ)) * NewRange) / OldRange) # dJ in [0,1]
nint <- round(sqrt(length(Rf)))#come suggerito dal paper

dint <- 1/nint #width of interval
n_pw <- c()
n_pw[1] <- sum(pw<=dint)
for (i in 2:nint) {
  n_pw[i] <- sum(pw>((i-1)*dint)& pw<=(i*dint)) #per ogni
  #intervallo calcolo quanti pw cadono dentro
}

p <- n_pw/length(Rf) #distribuzione dei pw normalizzata

ind <- which(p<=0)
p[ind] <- NaN
newp <- min(p,na.rm=T)/2 #poi devo fare il log, quindi gli
#intervalli vuoti non vanno bene
p[ind] <- newp

Ht <- log2(nint)/2 #suggerito dal paper
Mt <- 0.2

H <- -sum(p[!is.na(p)]*log2(p[!is.na(p)]))
M <- mean(pw)
```

```
if(H > Ht & M>Mt) {
  indici_Rf <- c()
  for (i in 1:length(pw)) {
    if(pw[i]>=0 & pw[i]<=(1/nint)){
      indici_Rf <- c(i,indici_Rf) #seleziono quelle features
      #i cui pw sono in [0,1/nint]
    }
  }
  # temp <- rep(NA,length(pw))
  # temp[indici_Rf] <- pw[indici_Rf]
  # toadd_Fr <- order(temp)
  # Fr <- c(toadd_Fr[1:length(indici_Rf)],Fr)

  #aggiungo indici di cluster

  Fr <- c(Rf[indici_Rf],Fr) #metto le feature selezionate in Fr
  Rf <- Rf[-indici_Rf]# e le rimuovo da Rf
  print(paste("Ho fatto il caso 1 eliminando ",
    length(indici_Rf)," features"))
} else {
  indici_Rf <- c()

  # ind <- which(pw==0)
  # pw[ind] <- NaN
  # newpw <- min(pw,na.rm=T)/2
  # pw[ind] <- newpw
  A <- sum(pw<=M)
  conv=0
  while (conv==0) {
    M <- M/2
```

```
        btest <- sum(pw<=M)
        if(btest<=A/2){
          conv=1 #elimino un numero di features minore della metà
          #di quelle con pw in [0,M/2]
        }
      }
      indici_Rf <- which(pw<=M)
      Fr <- c(Rf[indici_Rf],Fr)
      Rf <- Rf[-indici_Rf]
      print(paste("Ho fatto il caso 2 eliminando ",
        length(indici_Rf)," features"))
    }
  }
  Fr <- svm_rfe(train_dataset,Rf,Fr,best_costo)
  return(Fr)
}
```

A.3.3 svm_rfe.R

Implementa la semplice Recursive Feature Elimination

```
    svm_rfe <- function(train_dataset,Rf,Fr,best_costo) {
  while(length(Rf>0)){
    col <- c(Rf,ncol(train_dataset))
    S <- train_dataset[,col]
    m <- svm(Label ~ .,data=S,kernel = 'linear',cost=best_costo)
    w <- t(m[["coefs"]])%*%m[["SV"]]
    dJ <- w^2
    Fr <- c(Rf[which(dJ==min(dJ))],Fr)
    Rf <- Rf[-which(dJ==min(dJ))]
  }
```

```
  return(Fr)
}
```

A.3.4 SVMrem.R

Calcola l'accuracy teorica di un classificatore SVM lineare per ciascuna feature. Per il training usa un 5-fold cross-validation.

```
source("my_functions/MSplot.R")
source("my_functions/parfunctions.R")
source("my_functions/plotFC.R")
source("my_functions/massplot.R")
source('my_functions/mvaplot.R')
source('my_functions/pcaplot.R')
source('my_functions/boxplotfun.R')
source('my_functions/mergeFeatures.R')
source('my_functions/plotMap.R')
source('my_functions/plotMap_tpeak.R')
library('data.table')
#library("factoextra")#serve per PCAplot
library('caret')
#library('mlbench')
library('snow')
library('doSNOW')
library('e1071')
library('igraph')

source("classificazione/erfe.R")
source("classificazione/svm_rfe.R")
source("classificazione/MCC.R")
source("classificazione/k_fold_cv.R")
```

```
################################
##carico i dati e li sistemo ##
################################
peaktable <- read.csv("dati_originali/data_original_etac_pos.csv",
    stringsAsFactors = FALSE)
sizept <- dim(peaktable)
labels <- peaktable[1,seq(2,sizept[2])]
# labels[which(labels=="congelati_etac_pos")] <- -1
# labels[which(labels=="branzini_fresci_etac_pos")] <- 1
mass <- as.numeric(peaktable[2:sizept[1],1])
tpeaktable <- transpose(peaktable[2:sizept[1],2:sizept[2]])
tpeaktable <- transpose(as.data.frame(apply(tpeaktable,1,numerize)))
tpeaktable[is.na(tpeaktable)] <- 0
rownames(tpeaktable) <- colnames(peaktable)[2:sizept[2]]

## Normalizzazione ##########################################
#tpeaktable <- as.data.frame(apply(tpeaktable,2,normalizzo)) #
############################################################

# ### Shuffled labels #######
# set.seed(1)               #
# labels <- sample(labels)  #
# ############################

tpeaktable <- cbind(tpeaktable,as.data.frame(t(labels)))
colnames(tpeaktable)[sizept[1]] <- "Label"

control <- trainControl(method="cv",number = 5)
```

```
metric <- "Accuracy"


clusterExport(cl,list("train"))
acc <- foreach(i= 1:2959)%dopar% {
fit.svm <- train(Label~., data=tpeaktable[,c(i,2960)],
    method="svmLinear", metric=metric, trControl=control)
return(mean(unlist(fit.svm$resample$Accuracy)))
}
accuracy <- unlist(acc)
save(accuracy,file="SVM_5cv_accuracy.RData")
```

Bibliografia

[1] Julia H. Jungmann, Nadine E. Mascini, Andras Kiss, Donald F. Smith, Ivo Klinkert, Gert B. Eijkel, Marc C. Duursma, Berta Cillero Pastor, Kamila Chughtai, Sanaullah Chughtai, Ron M. A. Heeren, *Mass Spectrometry Basics for Young Students: An Interactive Laboratory Tour*, American Society for Mass Spectrometry, 2013

[2] Ben Bolstad, *Probe Level Quantile Normalization of High Density Oligonucleotide Array Data*, Division of Biostatistics, University of California,Berkeley, 2001 (http://bmbolstad.com/stuff/qnorm.pdf),

[3] Sara Pegolo,Barbara Di Camillo, Clara Montesissa, Francesca Tiziana Cannizzo, Bartolomeo Biolatti, Luca Bargelloni *Toxicogenomic markers for corticosteroid treatment in beef cattle: Integrated analysis of transcriptomic data*, 2015, Food and chemical toxicology 77

[4] *Mass Spectrometry*, chimica.unimi.it/extfiles/unimidire/178601/ attachment/franco-dragoni.pdf

[5] *Spettrometria di massa*, http://www.dbt.univr.it/documenti/ OccorrenzaIns/matdid/matdid380779.pdf

[6] *Spettrometria di massa*, http://www.dmf.unisalento.it/ daqatlas/ minilab/spettrometria_massa.pdf

[7] Qizhi Hu, Robert J. Noll, Hongyan Li, Alexander Makarov, Mark Hardmanc and R. Graham Cooks,*The Orbitrap: a new mass spectrometer*, 2005, JOUR-

NAL OF MASS SPECTROMETRY J. Mass Spectrom. 2005; 40: 430–443 Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/jms.856

[8] Robert B. Cody,James A. Larame and H. Dupont Durst, *Versatile New Ion Source for the Analysis of Materials in Open Air under Ambient Conditions*, JEOL USA, Inc., 11 Dearborn Road, Peabody, Massachusetts 01960, EAI Corporation, 1308 Continential Drive, Suite J, Abingdon, Maryland 21009, and U.S. Army Edgewood Chemical Biological Center, Aberdeen Proving Ground, Maryland 21010, Anal. Chem. 2005, 77, 2297-2302

[9] *Mass spectrometry introduction*, University of Pittsburgh, https://www.chem.pitt.edu/facilities/mass-spectrometry/mass-spectrometry-introduction

[10] Michela Begala, *Tipi di analizzatori*, Università di Cagliari, https://people.unica.it/michelabegala/files/2010/06/MS-Lezione-III-analizzatori.pdf

[11] Michela Begala, *Metodi di ionizzazione*, https://people.unica.it/michelabegala/files/2010/06/MS-Lezione-II-metodi-di-ionizzazione1.pdf

[12] Ying Zhenga, Runlong Fan, Chunling Qiu, Zhen Liu, Di Tian,*An improved algorithm for peak detection in mass spectra based on continuous wavelet transform*, 2016, International Journal of Mass Spectrometry, https://www.sciencedirect.com/science/article/pii/S1387380616301956

[13] *Pesce fresco e pesce congelato: nuova frontiera nei controlli della filiera ittica*, http://www.izsto.it/images/stories/meistro_18_12_15.pdf

[14] Jasmine Chong, Othman Soufan, Carin Li, Iurie Caraus, Shuzhao Li, Guillaume Bourque, David S Wishart, Jianguo Xia, *MetaboAnalyst 4.0: towards more transparent and integrative metabolo-*

mics analysis, 2018, Nucleic Acids Research, Volume 46, Issue W1, https://academic.oup.com/nar/article/46/W1/W486/4995686

[15] David MW Powers, *Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation*, Technical Report SIE-07-001, 2007 http://www.flinders.edu.au/science_engineering/fms/School-CSEM/publications/tech_reps-research_artfcts/TRRA_2007.pdf

Ringraziamenti

Desidero ringraziare la prof.ssa Di Camillo, relatore di questa tesi, per avermi guidato in questi mesi nelle ricerche e nella stesura dell'elaborato,

il dott. Roberto Piro dell'IZSVe per avermi fornito tutte le informazioni necessarie a comprendere le metodologie utilizzate per ottenere i dati analizzati in questa tesi,

la mia famiglia e i miei amici per avermi supportato (e sopportato) in questi anni trascorsi all'Università.