

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

---

Scuola di Scienze  
Dipartimento di Fisica e Astronomia  
Corso di Laurea Magistrale in Fisica

**Segmentazione automatica mediante atlanti  
anatomici in radioterapia: automazione e  
performance**

**Relatore:**  
Prof.ssa Maria Pia Morigi

**Presentata da:**  
Valeria Trojani

**Correlatori:**  
Dott. Gabriele Guidi  
Dott. Nicola Maffei

Anno Accademico 2016/2017



# Abstract

Nella pianificazione di trattamenti in ambito radioterapico, la delineazione degli organi a rischio (OAR) è essenziale per la minimizzazione della dose erogata ai tessuti sani. Solitamente, il medico radioterapista esegue la segmentazione delle strutture degli OAR manualmente, con alti costi a livello di tempistiche. Perciò, la ricerca si è focalizzata sullo sviluppo e l'integrazione di algoritmi di segmentazione automatica nella pratica clinica; questi richiedono una valutazione dei loro risultati rispetto al *gold standard* definito dal medico, eseguita attraverso indici di qualità oggettivi.

In questo lavoro di tesi, svolto presso l'Azienda Ospedaliero-Universitaria del Policlinico di Modena, in collaborazione con l'Università degli Studi di Bologna, sono stati sviluppati degli script in grado di automatizzare il processo di segmentazione e di estrazione degli indici (Dice e distanza di Hausdorff) per la suddetta valutazione. In seguito al confronto tra l'algoritmo di segmentazione con atlante anatomico e quello mediante modello, è risultato che il primo presentava delle performance migliori. Gli atlanti considerati, formati da pazienti sottoposti a trattamento PBI (Partial Breast Irradiation), risultano quindi omogenei per tipologia di pazienti e organi in esame. Inoltre, dal confronto di due diversi modi di utilizzo dell'algoritmo di segmentazione automatica mediante atlante su cuore, mammella controlaterale, polmone destro, polmone sinistro, midollo e tiroide, è emerso che, pur non essendoci differenze significative nei loro indici di qualità, uno presentava tempistiche di esecuzione migliori. L'uso di tale metodo in pratica clinica permette una rilevante riduzione del lavoro del medico.

In seguito, sono state gettate le basi per la costruzione di un classificatore in grado di riconoscere un'eventuale sovra-segmentazione di una struttura, al fine di espandere il database di pazienti presente negli atlanti costruiti. Il classificatore ha fornito dei buoni risultati e può quindi essere utilizzato per diminuire l'apporto del medico nel processo di espansione dell'atlante. Inoltre, al fine di permettere un livello di automazione e un'indipendenza dall'operatore sempre maggiore, sono stati proposti diversi miglioramenti per il classificatore costruito, che possono delineare le basi per un successivo lavoro in ambito radioterapico.

# Indice

|                          |          |
|--------------------------|----------|
| <b>Introduzione.....</b> | <b>6</b> |
|--------------------------|----------|

## **Capitolo 1**

|   |    |
|---|----|
| 1.1 Introduzione alla radioterapia.....                                     | 8  |
| 1.2 Introduzione alla segmentazione in ambito medico.....                   | 10 |
| 1.3 Vantaggi della segmentazione automatica mediante atlanti anatomici..... | 15 |

## **Capitolo 2**

|   |    |
|---|----|
| 2.1 Valutazione della bontà di una segmentazione automatica.....                        | 18 |
| 2.2 Metodologia usata nella segmentazione automatica.....                               | 20 |
| 2.2.1. Rappresentazione delle strutture e registrazione delle immagini.....             | 20 |
| 2.2.2. La registrazione rigida di RayStation: Mutual Information Algorithm.....         | 21 |
| 2.2.3. La registrazione ibrida di RayStation: l'algoritmo ANACONDA.....                 | 23 |
| 2.2.4. Segmentazione mediante atlanti anatomici in RayStation.....                      | 25 |
| 2.2.5. Segmentazione mediante modello biomeccanico (Model Based Segmentation, MBS)..... | 27 |
| 2.3 Metodologia e materiali utilizzati nell'analisi.....                                | 29 |

## **Capitolo 3**

|  |    |
|--|----|
| 3.1 Confronto tra MBS e segmentazione con atlante rispetto al <i>gold standard</i> ... | 34 |
|--|----|

3.2 Variazione dei pazienti e delle strutture incluse nell'algoritmo.....38

3.3 Sviluppi futuri.....48

## **Capitolo 4**

4.1 Costruzione di un classificatore per l'individuazione delle segmentazioni corrette al fine di espandere un atlante.....49

4.2 Struttura del classificatore.....52

4.3 Fase di training.....65

4.4 Fase di validazione.....67

4.5 Fase di test.....70

4.6 Discussione dei risultati e limiti di validità del classificatore.....72

4.7 Sviluppi futuri.....74

**Conclusioni.....75**

**Bibliografia.....79**

**Ringraziamenti.....82**

**Appendice.....83**

# Introduzione

In ambito radioterapico, nello sviluppo del piano di trattamento, è richiesto un grande livello di accuratezza nella delineazione degli organi a rischio (OAR), al fine di minimizzare la dose erogata ai tessuti sani.

La figura che esegue la segmentazione di tali organi è quella del medico radioterapista: questo compito, però, risulta avere alti costi a livello di tempistiche e impegno. Per questo motivo, lo sviluppo e l'integrazione di algoritmi di segmentazione automatica sono importanti temi di ricerca in questo settore. Tali algoritmi, infatti, diminuirebbero il carico di lavoro del medico, che potrebbe quindi portare la sua attenzione sulla delineazione dei target o su altri compiti necessari durante il trattamento dei pazienti. Inoltre, mediante l'uso di algoritmi automatici, si ridurrebbero altre problematiche legate alla segmentazione manuale, come ad esempio la variabilità di contornazione tra diversi utenti.

Al fine di poter integrare tali algoritmi di segmentazione nella pratica clinica è necessaria una valutazione diretta da parte del medico sulla segmentazione generata automaticamente o attraverso indici di qualità, aventi come riferimento sempre il *gold standard* del medico, il quale, al giorno d'oggi, risulta un elemento fondamentale nella catena di *image processing* che conduce alla segmentazione degli OAR.

In questo lavoro di tesi, svolto presso la struttura di Fisica Medica e Radioterapia Oncologica dell'Azienda Ospedaliero-Universitaria del Policlinico di Modena, in collaborazione con l'Università degli Studi di Bologna, sono stati condotti diversi studi.

Il primo riguarda un confronto tra la segmentazione automatica generata mediante modello, propria del TPS RayStation® (RaySearch Laboratories AB, Stockholm, Sweden), e la segmentazione automatica generata mediante atlante anatomico. L'atlante utilizzato risulta omogeneo per tipologia di pazienti e organi in esame, in quanto è stato costruito usando come elementi di training 42 pazienti sottoposti al trattamento Partial Breast Irradiation (PBI). Il confronto è stato eseguito per gli organi di midollo, polmone destro e polmone sinistro.

Successivamente è stato condotto un confronto tra due diversi metodi di utilizzo della segmentazione automatica mediante atlante per gli organi di midollo, polmone destro, polmone sinistro, mammella controlaterale, tiroide e cuore. Il metodo di valutazione usato ha previsto il calcolo dell'indice Dice (*Dice Similarity Index, DSC*) e della distanza di Hausdorff media (*Average Hausdorff Distance, AHD*).

A tal proposito, sono stati sviluppati degli script che automatizzassero la costruzione dell'atlante anatomico, la generazione delle segmentazioni automatiche mediante i diversi algoritmi e il calcolo degli indici con cui sono stati condotti i confronti.

In seguito sono state gettate le basi per la costruzione di un classificatore in grado di identificare, mediante l'analisi di parametri significativi, la bontà di una segmentazione senza ricorrere all'uso di indici che prevedano il confronto con il *gold standard* definito dal medico. Il classificatore si inserisce in un *workflow* finalizzato all'eventuale espansione dell'atlante anatomico, identificando, quindi, le segmentazioni adatte o meno a farne parte. Inoltre, esso si pone come obiettivo quello di alleggerire il lavoro del medico, in quanto, in sua assenza, l'identificazione delle ROI da inserire nell'atlante andrebbe condotta comunque da un esperto.

# Capitolo 1

## 1.1 Introduzione alla radioterapia

La radioterapia è una tecnica comunemente utilizzata in Medicina nel campo della cura dei tumori. Questa tecnica prevede l'erogazione di radiazioni ionizzanti al paziente al fine di distruggere le cellule tumorali o almeno limitare la loro proliferazione e, nel frattempo, preservare i tessuti e organi sani dai danni da radiazione.

Nella fase iniziale di un trattamento di radioterapia, è acquisita una CT (Computed Tomography), al fine di avere una precisa localizzazione della zona interessata dal trattamento. In questa immagine sono quindi tracciati i contorni dell'area tumorale (target) e quelli degli organi sani dell'area circostante (chiamati organi a rischio, OAR). Queste operazioni sono fondamentali per determinare la dose totale di radiazione irradiata dal trattamento, tenendo in considerazione le caratteristiche degli organi sani e di quanta dose possono ricevere senza riportare danni.

Durante la delimitazione dei target, devono essere seguite delle regole precise nella determinazione del volume da irradiare. L'ICRU (International Commission on

Radiation Units and Measurements) ha proposto delle linee guida universalmente accettate per la prescrizione della dose irradiata nel trattamento. [1]

Per questi motivi, è necessario introdurre la definizione di diversi volumi relativi sia ai tessuti tumorali sia a quelli sani. La contornazione di questi volumi è obbligatoria durante la pianificazione del trattamento ed è portata a termine manualmente da un medico radioterapista.

Il **GTV (Gross Tumor Volume)**: è il volume macroscopico di malattia, cioè quello che è visibile attraverso le varie modalità di imaging. Può essere delineato sia prima che durante il trattamento per registrare potenziali variazioni nel volume del target.

Il **CTV (Clinical Target Volume)** comprende il GTV insieme ad aree limitrofe, che possono essere probabile sede di interessamento microscopico della malattia. Nel corso del trattamento è spesso necessario ridefinire questo volume, poiché questo viene spesso modificato dalla terapia.

Il **PTV (Planning Target Volume)** è un'espansione del CTV che tiene conto delle incertezze di posizionamento. Viene delineato al fine di assicurarsi che la dose prescritta sia effettivamente erogata al CTV con un margine di confidenza accettabile.

Come già detto in precedenza, anche gli organi a rischio (OAR) sono delineati, al fine di evitare di erogare dose ad aree particolarmente sensibili ai danni da radiazione.

Lo strumento utilizzato al fine di contornare tali strutture d'interesse è il Treatment Planning System (TPS). Il TPS è un insieme di strumenti hardware e software, utilizzati per l'elaborazione del piano di trattamento radiante, che permettono di eseguire diverse operazioni, tra le quali è appunto contemplata la

possibilità di delineare i volumi di interesse per il trattamento. Ciò è possibile sia attraverso un sistema di pennelli di diverse dimensioni, implementato all'interno del software, che permettono la contornazione manuale, sia – nelle zone anatomiche ove disponibile – attraverso diversi algoritmi finalizzati alla contornazione automatica.

## **1.2 Introduzione alla segmentazione in ambito medico**

Nell'ambito dell'immagine processing, il processo di segmentazione ha la finalità di suddividere un'immagine in diverse regioni aventi particolari caratteristiche, il cui livello di dettaglio dipende dalle finalità del problema in esame. [2] Questo significa che, idealmente, il processo di segmentazione si ferma una volta che la zona di interesse è stata identificata.

Se si identifica la regione spaziale da segmentare con  $R$ , la segmentazione è quel processo che partiziona l'insieme  $R$  in diversi sottoinsiemi  $R_j$  tali che:

- 1) ogni pixel di  $R$  appartiene a un determinato  $R_j$ ;
- 2) le sottoregioni  $R_j$  non devono sovrapporsi;
- 3) ogni sottoregione  $R_j$  deve soddisfare una o più condizioni dettate dal tipo di problema in esame;
- 4) due sottoregioni differenti non possono soddisfare la stessa condizione.

Nel contesto di questa tesi, le sottoregioni  $R_j$  citate nella 1) si riferiscono agli OAR e ai target; infatti, nella segmentazione i pixel vengono divisi in appartenenti a una certa regione o non appartenenti a essa. Per questo motivo, la condizione 2) non è applicabile nel caso dei target, in quanto essi per definizione da parte dell'ICRU si sovrappongono tra essi (il PTV è infatti un'espansione del CTV); anche nel caso

degli OAR, nell'eventualità di una segmentazione errata, si possono trovare regioni sovrapposte. Le condizioni che ogni sottoregione deve soddisfare per la 3) sono dettate volta per volta dai diversi algoritmi di segmentazione nel caso di segmentazione automatica o dall'esperienza e dalla conoscenza dell'operatore in caso di segmentazione manuale.

In ambito medico, negli ultimi anni, le innovazioni introdotte nel campo della radioterapia hanno richiesto una sempre più elevata precisione nella definizione dei volumi, in linea con le guide fornite dall'ICRU. Per molti radioterapisti la segmentazione manuale è ancora l'unica scelta possibile, che però richiede alti costi a livello di tempistiche e di manualità. Infatti, delineare da immagini CT le strutture compatibili con gli standard ICRU non è un compito facile, considerando che esse non hanno necessariamente un 'boundary' identificabile in maniera continua. Quello che si fa nella pratica clinica è delineare una regione di interesse (ROI) entro la quale viene accettato sia presente la struttura ricercata. Il risultato è chiaramente dipendente dall'operatore che esegue la contornazione ed è suscettibile a variabilità ed errori tra diversi operatori. Nello studio condotto da Walker G. et al., nel 2014 [3], è stato evidenziato come strutture contornate tramite segmentazione automatica (in alto nella figura 1.1) siano soggette a meno variabilità rispetto a quelle contornate manualmente da 8 esperti (in basso). Lo stesso studio ha anche quantificato il tempo necessario per effettuare una contornazione manuale da parte di un medico come di circa 30 minuti.

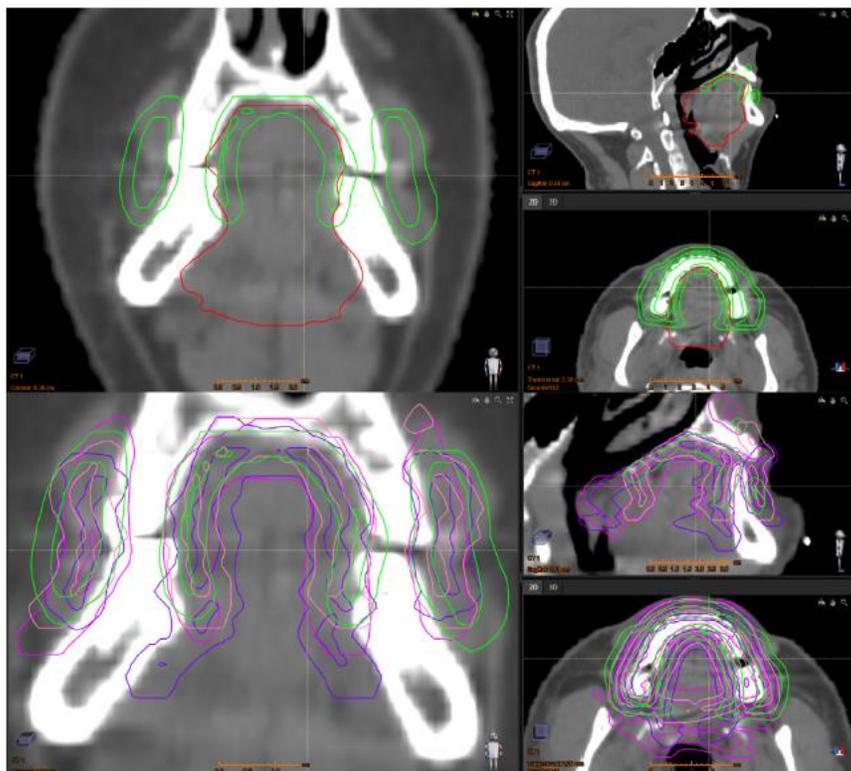


Figura 1.1. In alto ROI contornate tramite atlante, in basso ROI contornate da diversi esperti. (Walker G. et al. 2014)

L'averne una buona contornazione (sia manuale che automatica) dei volumi di interesse è determinante in quanto gli stessi volumi sono poi usati dal TPS nella pianificazione del trattamento da somministrare al paziente: una sovra- o sotto-segmentazione di strutture di interesse può portare a dosi di radiazione pianificata errate e quindi a complicazioni sul paziente sul medio-lungo periodo, cioè anche una volta terminato il trattamento. Inoltre, la visibilità delle regioni di interesse da delineare è frequentemente compromessa dal fatto che esse sono spesso tessuti molli, la cui conformazione strutturale è già intrinsecamente poco visibile in immagini di tipo CT, a causa delle limitazioni fisiche dovute all'interazione tra la radiazione e la materia, oltre che alla loro potenziale struttura irregolare.

Un'altra criticità evidenziata nell'uso di atlanti anatomici, come descritto dallo studio condotto da Commowick O. et al. nel 2008 [5], è la potenziale ampia differenza esistente tra l'anatomia dei pazienti che compongono il database (evidenziata in figura 1.2).

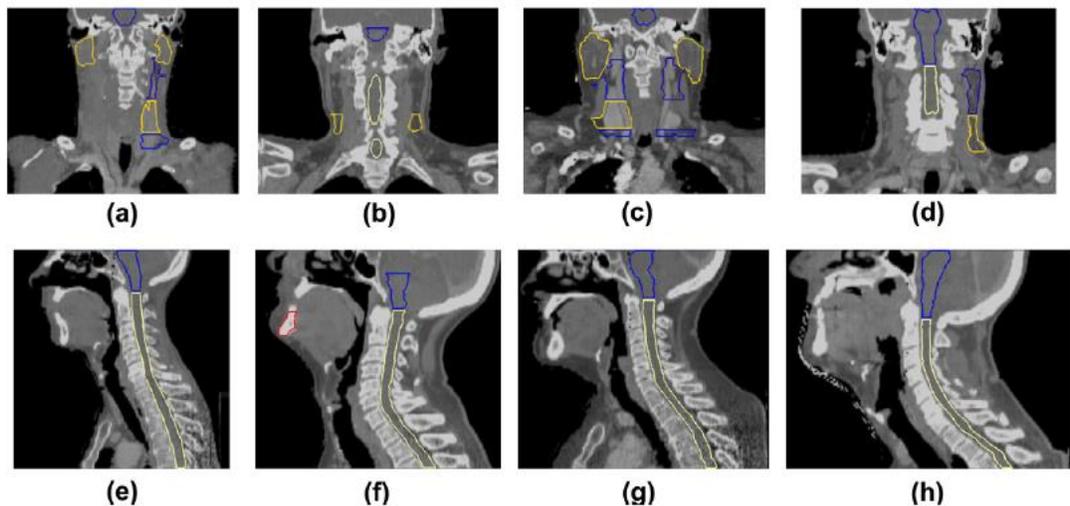


Figura 1.2. Variabilità delle strutture anatomiche all'interno di pazienti in uno stesso atlante.

Queste differenze anatomiche possono introdurre un errore sistematico nella segmentazione automatica, dovuto a un'errata registrazione dall'immagine di riferimento all'immagine target. Una possibile soluzione proposta dagli autori e usata anche nello studio condotto da Ciardo D. et al. nel 2016 [6] è la costruzione di sotto-atlanti riferiti a pazienti con particolari caratteristiche (età, costituzione, provenienza, ...).

Nell'ambito di questa tesi, il processo di segmentazione viene utilizzato al fine di identificare i contorni dei target e degli OARs di una particolare zona anatomica: alla luce dei dati presentati, risulta necessaria una valutazione più approfondita degli algoritmi di segmentazione, in modo da rendere effettivamente possibile l'automazione di questo processo. In ambito commerciale è possibile trovare

svariati TPS, ma non molti prevedono la possibilità di integrare script prodotti dall'utente finale al software commerciale, al fine di automatizzare processi ripetitivi o di eseguire particolari comandi non impartibili usando solamente l'interfaccia grafica.

In ogni caso, nella pratica clinica, il *gold standard*, cioè le strutture a cui si fa riferimento, rimangono sempre quelle delineate dal medico, che comunque è un componente essenziale nella catena di *image processing* che porta alla contornazione delle strutture ICRU.

È possibile adottare diversi approcci a questo tipo di segmentazione:

- Il primo è una segmentazione automatica eseguita senza fornire all'algoritmo una conoscenza aprioristica delle strutture di interesse. Un esempio può essere trovato nel caso dei polmoni o delle ossa, dove una segmentazione basata solo sul livello di grigio dell'immagine può essere condotta, poiché si tratta di aree ad alto livello di contrasto e i contorni della regione di interesse possono essere individuati con facilità. Alcuni di questi algoritmi hanno bisogno di punti iniziali (chiamati '*seeds*') dai quali far partire la contornazione delle strutture, come descritto nello studio di Whitfield et al. nel 2012 [4]. Un algoritmo che si basa su questa metodologia è "*random walker*", sviluppato da Grady L. al Siemens' Princeton Research Laboratory [4]. Sfortunatamente, quello che agli occhi del medico è identificato chiaramente come target o OAR deve essere interpretato dall'algoritmo di segmentazione automatica solamente in termini di dati ricavabili dall'immagine, che, nel caso di una CT, è la mappa di attenuazione dei raggi X, stimata lungo l'area di interesse del paziente. Queste differenze nell'attenuazione dei raggi X sono molto piccole nel caso dei tessuti molli, rendendo gli algoritmi che operano senza conoscenza pregressa sulle strutture poco performanti.

- Un secondo approccio è quello della segmentazione avente alla base una conoscenza a priori sulle strutture di interesse. Questa conoscenza può essere fornita, ad esempio, mediante l'uso di atlanti anatomici (*Atlas Based Segmentation*, ABS). Un atlante è un insieme di immagini di riferimento dove sono state già contornate le strutture di interesse da un esperto. La precisione e la correttezza delle strutture presenti nell'atlante influenza notevolmente i risultati dell'algoritmo. Un altro modo utilizzato per fornire conoscenze sulle strutture all'algoritmo di segmentazione è attraverso modelli della regione da segmentare, nei quali vengono incorporate informazioni sulla variabilità della stessa da paziente a paziente attraverso atlanti probabilistici, che hanno il fine di assegnare a ogni voxel dell'immagine target la relativa probabilità di appartenere a una determinata struttura. Algoritmi di questo tipo sono implementati in diversi TPS di tipo commerciale, come ad esempio Varian® (Varian Medical System Inc., Palo Alto, CA) o RayStation® (RaySearch Laboratories AB, Stockholm, Sweden). Nell'algoritmo implementato da RayStation, inoltre, il successo di questo procedimento dipende in ampia misura dalla scelta del paziente all'interno dell'*atlas*, condotta mediante la valutazione di misure di somiglianza con l'immagine da contornare.

### **1.3 Vantaggi della segmentazione automatica mediante atlanti anatomici**

Sono stati condotti numerosi studi al fine di valutare l'uso di atlanti anatomici nella pratica clinica. I risultati indicano che gli atlanti permettono una notevole riduzione del tempo impiegato nella segmentazione degli organi e allo stesso tempo

riducono la variabilità tra i diversi utenti tipica delle contornazioni manuali. Data la grande rilevanza che ha la segmentazione degli organi nella pratica clinica è comunque necessario fornire dei metodi di valutazione adeguati delle segmentazioni mediante atlante.

Nello studio di Walker G. et al. del 2014 [3], condotto su pazienti affetti da tumori nella zona di testa e collo (Head&Neck - H&N), sono state messe a confronto, rispetto a un *gold standard*, una segmentazione completamente automatica e una segmentazione automatica modificata e corretta da un esperto. Dallo studio è emerso come, in organi quali midollo, mandibola e nervo ottico, la differenza tra questi due tipi di segmentazione non sia significativa, validando quindi anche la segmentazione completamente automatica. In altri organi invece, come ad esempio il chiasma ottico, la cavità ottica e la laringe, la correzione di un esperto migliora significativamente la qualità della segmentazione, rendendo quindi un controllo sull'output dell'*atlas* un elemento fondamentale del *workflow* di contornazione.

Lo studio di Sims R. et al. del 2009 [5], condotto sempre su pazienti H&N, ha evidenziato come la segmentazione automatica fornisse output soddisfacenti in tempi brevi rispetto alla contornazione manuale (circa 7 minuti su un pc standard) ad eccezione delle parotidi e del tronco encefalico, i quali sono stati rispettivamente sovra- e sotto- stimati. Una possibile causa di tale errore è stata ricercata nella elevata variabilità delle caratteristiche dei pazienti e delle strutture contornate presenti nell'atlante, che avrebbe portato a errori nella deformazione delle strutture sull'immagine di riferimento.

L'obiettivo dello studio condotto da Ciardo D. et al. nel 2016 [6] è stato quello di capire se fosse possibile individuare una categoria, qualitativa o quantitativa, di pazienti per i quali una certa segmentazione fosse efficiente o meno. In particolare

Ciaro et al. hanno focalizzato l'attenzione sul caso della mammella, per la quale hanno proposto delle possibili variabili di stratificazione dei diversi pazienti relative non al volume ma piuttosto alla forma della stessa. Lo studio di Eldesoky et al. del 2016 [8] propone invece stratificazioni su categorie relative al tipo di trattamenti chirurgici subiti.

Lo studio di Maffei N. et al. nel 2016 [9] è stato condotto su pazienti con patologia testa e collo presso il Policlinico di Modena. Esso aveva l'obiettivo di valutare sia la segmentazione mediante l'atlante costruito usando contorni effettuati su pazienti della sede di Modena sia quella basata su modelli, propria del TPS RayStation, mediante diversi parametri di valutazione. Dallo studio è risultato che la segmentazione automatica mediante modelli non raggiungeva il livello di accettabilità, definito a priori, per organi come il midollo e le parotidi. Per quanto riguarda la segmentazione tramite atlanti, è stato mostrato come un numero minimo di 20-30 pazienti nell'atlante sia necessario per ottenere una segmentazione accettabile secondo i limiti di cui sopra.

Uno degli obiettivi di questa tesi è quindi quello di condurre uno studio di valutazione di diversi algoritmi di segmentazione automatica implementati mediante il TPS RayStation, utilizzando le metodologie proposte negli studi sopra citati.

# Capitolo 2

## 2.1 Valutazione della bontà di una segmentazione automatica

Al fine di fornire dei parametri fisici per valutare l'effettiva corrispondenza della segmentazione automatica con il *gold standard* alla quale si riferisce, sono stati utilizzati due indici di confronto tra volumi.

Il primo è l'indice di Dice-Sorensen (*Dice Similarity Coefficient*, DSC), che fornisce una stima geometrica della sovrapposizione tra due volumi ed è dato dalla:

$$DSC = 2 \frac{|A \cap B|}{|A| + |B|}$$

dove  $A$  e  $B$  sono i due volumi da confrontare (nel caso di questo lavoro di tesi si tratterà del volume della struttura contornata manualmente dal medico e di quello della struttura generata in maniera automatica dall' algoritmo di segmentazione). L'intervallo in cui può variare questo indice va da 0 (nessuna sovrapposizione) a 1 (completa sovrapposizione). Nella figura 2.1 è mostrata la relazione tra il valore dell'indice Dice e la sovrapposizione dei due volumi da confrontare.

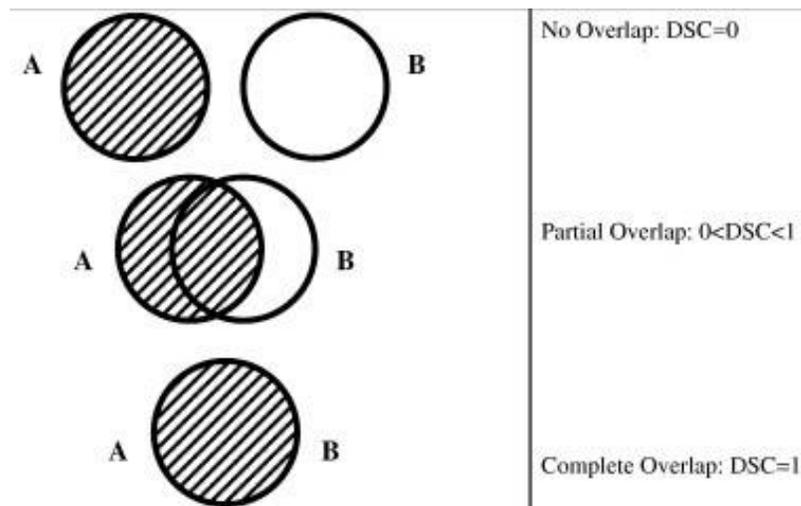


Figura 2.1. Relazione tra indice Dice e sovrapposizione dei volumi considerati.

Il secondo indice è la distanza di Hausdorff [10], usata per avere una misura della distanza tra due superfici. Essa è calcolata tramite:

$$H_{max} = \max(D(S, S'), D(S', S)) = \max(\max_{p \in S} d(p, S'), \max_{p \in S'} d(p, S))$$

dove  $S$  e  $S'$  sono le due superfici in esame,  $d(a, B)$  indica il minimo della distanza euclidea tra il punto  $a$  (tipicamente appartenente a una superficie) e la superficie  $B$ . La distanza viene così definita in modo da essere invariante per simmetria; nel caso di due superfici perfettamente sovrapposte, questo parametro assume un valore nullo. È utile prendere una media della distanza di Hausdorff su tutto il volume in esame (Average Hausdorff Distance, AHD), in quanto il valore massimo della distanza di Hausdorff è molto sensibile a casi di sovra- o sotto- segmentazione solamente nelle zone marginali della ROI in esame, ma non a casi di sovra- o sotto- segmentazione sistematica lungo tutte le slice di un determinato asse, cosa che, invece, viene messa in evidenza dalla AHD. Nella figura 2.2 è mostrato un confronto tra le caratteristiche evidenziate dai due indici usati.

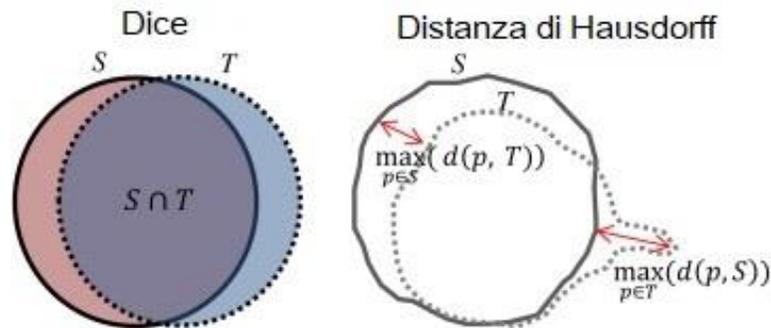


Figura 2.2. Confronto tra indice Dice e distanza di Hausdorff.

## 2.2 Metodologia usata nella segmentazione automatica

Il software usato per effettuare le operazioni di segmentazione sulle strutture di interesse è il TPS commerciale RayStation 6.0, il quale utilizza precisi algoritmi per eseguire il suddetto compito. Sono stati inoltre prodotti degli script (riportati in appendice) nel linguaggio IronPython 2.7.1, integrati poi con il TPS, con i quali è stato possibile calcolare gli indici descritti nel paragrafo precedente finalizzati a quantificare la bontà delle segmentazioni automatiche.

### 2.2.1 Rappresentazione delle strutture e registrazione delle immagini

Nell'ambito dei trattamenti radioterapici, un set di immagini 2D, unite ai contorni definiti dal medico, bastano per definire la forma dell'organo di interesse. In aggiunta a ciò, proprio per riuscire a indentificare maggiormente eventuali variazioni locali di struttura, è possibile utilizzare una rappresentazione mediante modelli 3D formati da triangoli *mesh*. Un *mesh*  $M$  è definito come un set di punti appartenenti a  $\mathbb{R}^3$ , chiamati vertici, le linee che uniscono tali punti, chiamate *edges* e insiemi chiusi di *edges*, chiamati facce. Nel *mesh* è definita anche la forma dell'elemento base della struttura 3D; a tal fine, possono essere usati tutti i tipi di poligoni, ma la scelta ricade solitamente sui triangoli perché sono i più efficaci da utilizzare.

In ambito medico, una registrazione di immagini è un problema di ottimizzazione volto a trovare la trasformazione geometrica che massimizza la somiglianza tra la struttura di riferimento e quella da registrare. In RayStation c'è la possibilità di registrare le immagini secondo due metodologie:

- Registrazione rigida: le trasformazioni cercate sono di tipo lineare, cioè trasformazioni affini (rotazione, traslazione, scala...). La registrazione rigida, a causa della sua linearità, non si adatta alle eventuali differenze locali (i.e. fenomeni di *organ warping/motion*, *tumor shrinkage*) tra le due immagini da registrare.
- Registrazione ibrida: le trasformazioni cercate possono non essere lineari, permettendo una deformazione dell'immagine di riferimento al fine di adattarla all'immagine target.

### **2.1.2 La registrazione rigida di RayStation: Mutual Information algorithm**

L'algoritmo di registrazione rigida usato da RayStation si basa sulle intensità dell'immagine e sfrutta il concetto di informazione mutua (Mutual Information). L'algoritmo di registrazione rigida ha come finalità la massimizzazione di una funzione obiettivo basata sull'entropia di Shannon  $H$ , che è definita come:

$$H = - \sum_i p_i \log(p_i)$$

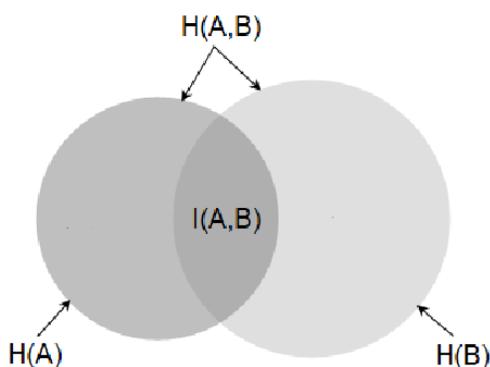
dove  $p_i$  rappresenta la probabilità dell'intensità del segnale. La distribuzione di probabilità del segnale può essere stimata come il numero di occorrenze di un certo valore di intensità nell'immagine, normalizzato sul totale delle occorrenze di tutti i valori di intensità presenti. Un'immagine che presenta pochi valori di intensità al suo interno avrà una bassa entropia: conterrà cioè minore informazione. Avendo due immagini, è possibile costruire un grafico 2D, chiamato

istogramma congiunto, nel quale sono rappresentate le combinazioni delle intensità delle due immagini per tutti i punti corrispondenti. Quando due immagini sono allineate, le strutture anatomiche corrispondenti saranno sovrapposte: l'istogramma congiunto mostrerà quindi valori diversi da zero in corrispondenza degli intervalli di intensità propri delle strutture sovrapposte. È possibile quindi definire l'informazione mutua come:

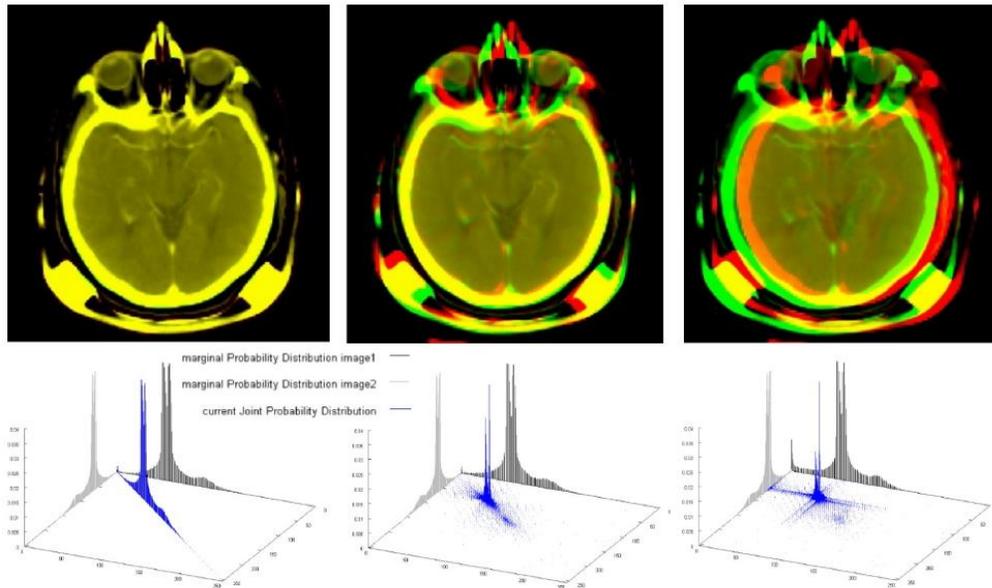
$$I(A,B) = H(A) + H(B) - H(A,B)$$

dove  $H(A)$  e  $H(B)$  sono le entropie marginali di Shannon rispettivamente delle immagini  $A$  e  $B$ , mentre  $H(A,B)$  rappresenta l'entropia congiunta.

Il valore  $I(A,B)$  corrisponde quindi a una misura di somiglianza tra le due immagini da registrare: rappresenta quindi la funzione obiettivo che l'algoritmo di registrazione rigida andrà a massimizzare. Le immagini devono quindi essere allineate in modo che l'informazione da esse contenuta sia massima, ossia che l'entropia congiunta sia al suo valore minimo.



a.



b.

Figura 2.3a. Diagramma di Venn che illustra i concetti di entropie marginali  $H(A)$  e  $H(B)$ , informazione mutua  $I(A,B)$  e entropia congiunta  $H(A,B)$ .

Figura 2.3b. Registrazione rigida effettuata su una CT-MRI di buona (sinistra), media (centro) e bassa (destra) qualità. È possibile vedere come, nel primo caso, l'istogramma congiunto presenti solo una zona ristretta di valori non nulli, corrispondente all'intervallo di intensità delle strutture sovrapposte.

### 2.2.3 La registrazione ibrida di RayStation: l'algoritmo ANACONDA

L'algoritmo per compiere una registrazione ibrida in RayStation è chiamato ANACONDA (*"ANatomically CONstrained Deformation Algorithm"*) e combina informazioni provenienti dall'immagine da contornare, come ad esempio le sue intensità, con le informazioni anatomiche fornite dalle strutture contornate. Il problema della registrazione dell'immagine è formulato come un problema di ottimizzazione non lineare ed è usato un risolutore sviluppato all'interno del software stesso.

Siano definite l'immagine di riferimento  $R$ , quella sulla quale deve essere compiuta la registrazione  $T$  e la trasformazione rigida che allinea le due immagini  $M: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ . La griglia di deformazione iniziale è un set di punti disposti in maniera regolare su un *lattice*, che divide l'immagine in elementi a forma di box (i voxel). L'algoritmo di deformazione trova un campo vettoriale di deformazione  $\mathbf{v}$  che viene definito sulla suddetta griglia: il campo di deformazione nel punto  $x_i \in \mathbb{R}^3$  è definito con  $v_i \in \mathbb{R}^3$ .

Il problema della registrazione è formulato quindi come problema di ottimizzazione non lineare della seguente funzione obiettivo  $f: \mathbb{R}^n \rightarrow \mathbb{R}$

$$f(v) = \alpha C(v) + (\beta H(v) + \gamma S(v)) + \delta D(v)$$

Dove:

$\alpha, \gamma, \delta \in \mathbb{R}$  sono dei pesi non negativi,  $n$  è il numero di variabili (che equivale a tre volte il numero di punti nella griglia di deformazione) e  $\beta: \mathbb{R}^3 \rightarrow \mathbb{R}$  è una funzione peso non negativa.

$C(v)$  misura la somiglianza tra  $T$  e  $R$  e viene calcolato su tutto  $R$  o solo su un subset di essa (ad esempio una singola ROI).

$H(v)$  e  $S(v)$  sono i termini di regolarizzazione del campo di deformazione.  $H(v)$  è ottenuta minimizzando la funzione di Dirichlet<sup>1</sup> del campo di deformazione.

$S(v)$  misura invece l'entità della deformazione e la sua funzione è quella di assicurare che le strutture vengano deformate in modo ragionevole, cosa che non sempre accade, soprattutto nei casi in cui la misura di somiglianza delle immagini

---

<sup>1</sup> La funzione di Dirichlet è una misura di quanto variabile è una funzione. Dato un insieme aperto  $\Omega \in \mathbb{R}^n$  e una funzione  $u: \Omega \rightarrow \mathbb{R}$  l'energia di Dirichlet  $E(u) \in \mathbb{R}$  associata è data dalla:

$$E(u) = \frac{1}{2} \int_{\Omega} \|\nabla u(x)\|^2 dV$$

Dove  $\nabla u(x): \Omega \rightarrow \mathbb{R}^n$  ed è il gradiente della funzione  $u$ .

che guida l'algoritmo non è affidabile (ad esempio, nel caso di questa tesi, se il paziente scelto, nonostante sia quello più simile tra quelli disponibili nell'atlante, si discosta comunque molto dal paziente target).  $S(v)$  è invariante per traslazione e rotazione e dipende solo dalle differenze tra i vettori del campo di deformazione.

Il termine  $D(v)$  ha la funzione di modificare il campo di deformazione nel caso l'utente decida di voler dare un diverso peso alle ROI in esame e impostarne una come più importante rispetto alle altre.

L'algoritmo è inoltre accelerato per GPU ed è implementato all'interno del software commerciale RayStation; i valori dei singoli pesi e delle funzioni peso sono stati definiti in seguito a valutazioni dell'algoritmo eseguiti dalla casa produttrice. Come utenti finali del prodotto non è possibile modificare direttamente i dettagli dell'algoritmo o la forma della funzione obiettivo.

#### **2.2.4 Segmentazione mediante atlanti anatomici in RayStation**

L'algoritmo di segmentazione mediante atlanti proposto da RayStation unisce registrazione rigida e ibrida al fine di ottenere uno strumento di segmentazione automatica. Il punto principale è quello di selezionare all'interno dell'atlante il paziente che ha la maggiore somiglianza con quello in esame attraverso registrazione rigida, scegliendo quello per il quale il valore di informazione mutua tra l'immagine di riferimento e quella target risulta maggiore. A questo punto viene usata la registrazione ibrida per deformare le strutture presenti all'interno dell'immagine di riferimento in relazione all'immagine target. Le strutture presenti nei pazienti che formano l'atlante sono state contornate manualmente da medici e la regione anatomica coperta dall'atlante dovrà essere la stessa dell'immagine

target. In figura 2.4 è mostrato schematicamente il processo di segmentazione mediante atlante utilizzato da RayStation.

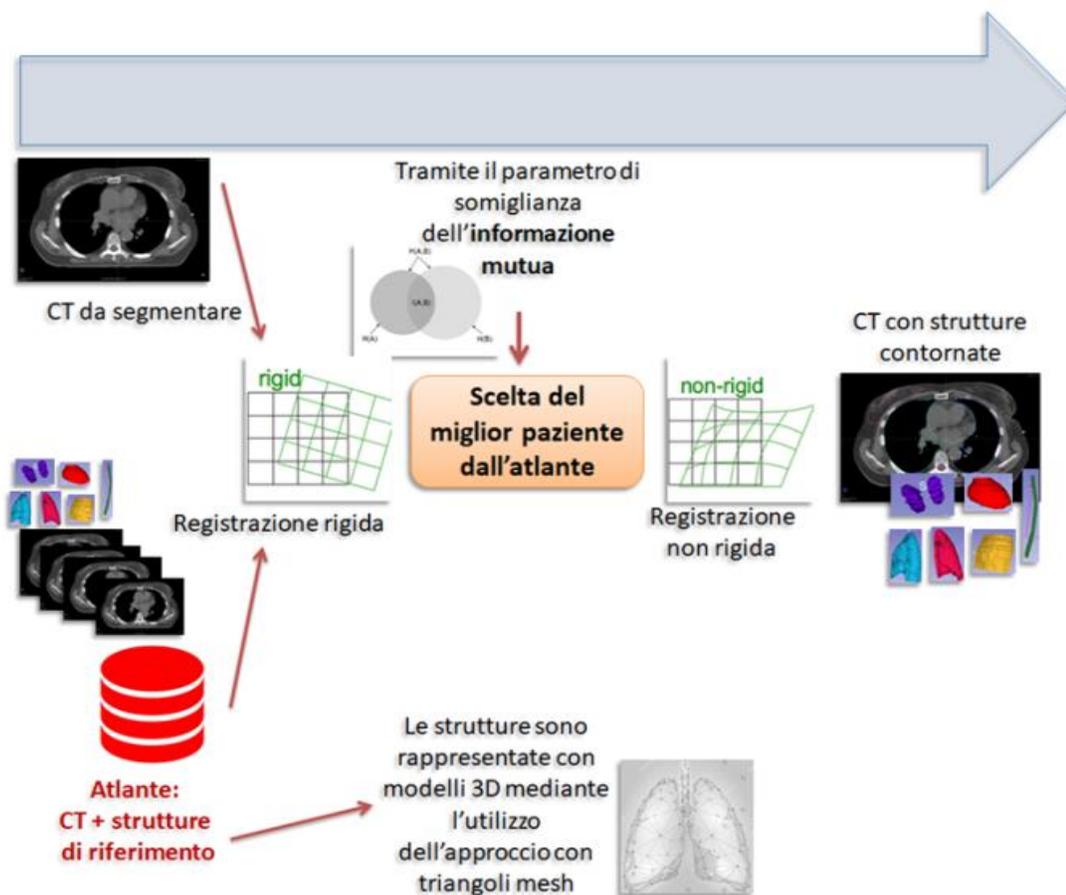


Figura 2.4. Workflow dell'algoritmo di segmentazione tramite atlante usato da RayStation.

Il software RayStation mette a disposizione un'interfaccia grafica (figura 2.5) per la creazione di strutture mediante questa metodologia. È possibile selezionare l'atlante (cioè l'insieme di pazienti) di riferimento (1), i singoli pazienti da includere nell'atlante (2) e infine le strutture che si vogliono creare nella nuova immagine (3). La possibilità di selezionare il numero di strutture da segmentare nasce dal fatto che in alcuni casi potrebbe essere presente una lesione o un artefatto (i.e. protesi

dentaria) nella zona dove è richiesto il contornamento e quindi il risultato che si otterrebbe da una segmentazione automatica non sarebbe corretto perché l'elemento di disturbo in questione interferirebbe con l'algoritmo.

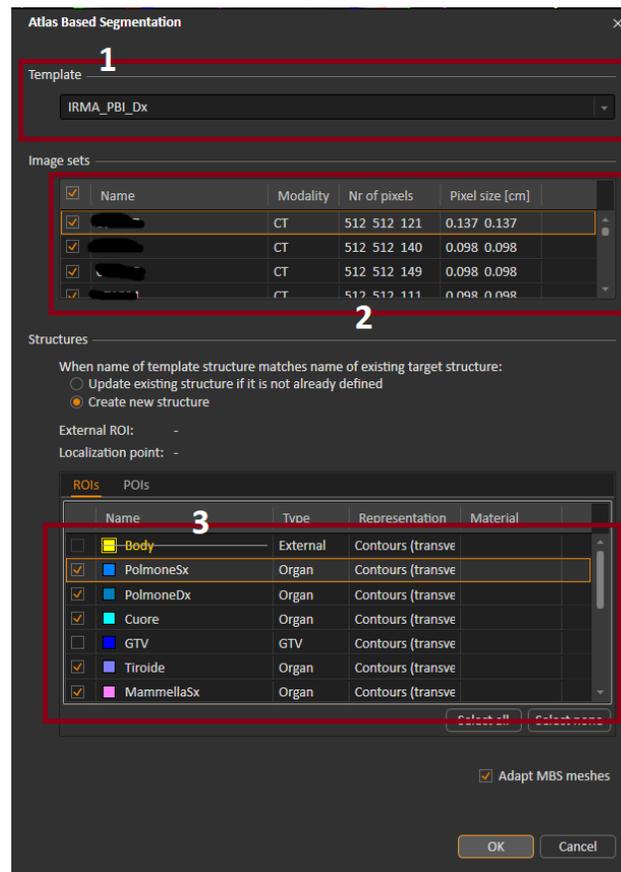


Figura 2.5. Interfaccia grafica della segmentazione mediante atlante.

## 2.2.5 Segmentazione mediante modello biomeccanico (Model Based Segmentation, MBS)

L'idea che sta alla base della MBS è quella di adattare un modello 3D delle strutture da segmentare a una nuova immagine target. Ogni modello di struttura è formato dalla forma media della stessa, unita a dei vincoli che contengono informazioni sulle variazioni di forma più ricorrenti che si possono trovare nella

struttura di interesse. Nella costruzione di un modello, vengono impiegate delle immagini dove la struttura in esame è stata già contornata da un esperto, che vanno a costituire il set di training. Da questo set, viene calcolato il modello medio e, attraverso analisi statistiche, vengono trovate le componenti principali del modello che sono soggette a variazione. Il software di RayStation comprende diversi modelli già definiti e pronti per essere applicati nella segmentazione, sebbene non venga dichiarato il numero di immagini utilizzate per il training e le caratteristiche (numero di voxel, risoluzione,...) delle stesse.

Per segmentare una struttura in una nuova immagine, il suo modello viene automaticamente inizializzato tramite registrazione rigida e ibrida. Al fine di permettere l'adattamento del modello all'immagine in esame, sono necessarie alcune informazioni sui livelli di grigio propri della struttura da contornare: nel caso di modelli già definiti all'interno di RayStation queste informazioni sono inserite di default. In seguito, sono estratti dei punti caratteristici dall'immagine target per ogni faccia del modello, cercandoli nella direzione perpendicolare alle suddette facce. In seguito, l'algoritmo prova ad adattare il modello a questi punti scelti senza, nel frattempo, violare i limiti imposti dalla forma dello stesso. Il processo è, anche in questo caso, un problema di ottimizzazione non lineare; inoltre, è di tipo iterativo e a ogni iterazione vengono estratti nuovi punti caratteristici dall'immagine target in modo da avere ogni volta un migliore adattamento.

Questo tipo di algoritmo prevede eventualmente anche una successiva interazione da parte dell'utente; per questo motivo, le ROI create attraverso questa metodologia dispongono di un *tool* particolare che consente la loro modifica guidata da un parametro di elasticità ((1) nella figura 2.6), che permette un ulteriore adattamento della ROI alla struttura che si sta cercando di contornare.

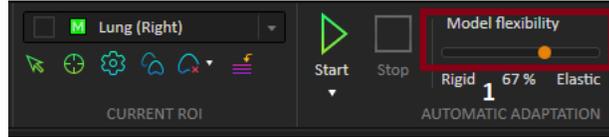


Figura 2.6. Caratteristiche di una ROI MBS.

## 2.3 Metodologia e materiali utilizzati nell'analisi

La prima parte di questo lavoro di tesi è stata la costruzione di un atlante per la regione del torace. I pazienti usati come training e test sono pazienti sottoposti al trattamento Partial Breast Irradiation (PBI). Le ROI sulle quali si è concentrato lo studio sono state: polmone destro, polmone sinistro, midollo, cuore, mammella controlaterale e tiroide. A questo fine, sono stati importati 42 pazienti per implementare l'atlante dal TPS Pinnacle® (Philips Radiation Oncology Systems, Fitchburg, WI), software usato dai medici per eseguire le contornazioni manuali delle strutture di interesse, al TPS RayStation. Il motivo di questo trasferimento è dovuto al fatto che RayStation permette l'integrazione di script prodotti dall'utente per effettuare azioni non eseguibili da GUI e si è rivelato quindi più adatto per un'analisi quantitativa della qualità delle ROI generate dai due algoritmi di segmentazione automatica. Durante la fase di valutazione dell'algoritmo, è stato ogni volta escluso dall'atlante costruito il paziente su cui veniva effettuato il test, in modo da non avere segmentazioni falsate (*leave-one-out cross validation*).

Una volta effettuato il trasferimento di pazienti e strutture in RayStation, è stato necessario impostare un nome unico (in quanto i medici che avevano eseguito la contornazione avevano chiamato le ROI con nominativi diversi) per ogni struttura di interesse e definire le proprietà della ROI, come ad esempio il tipo di tessuto della regione contornata (figura 2.7). A questo fine, è stato elaborato uno script, riportato in appendice 1, in grado di automatizzare questo processo.

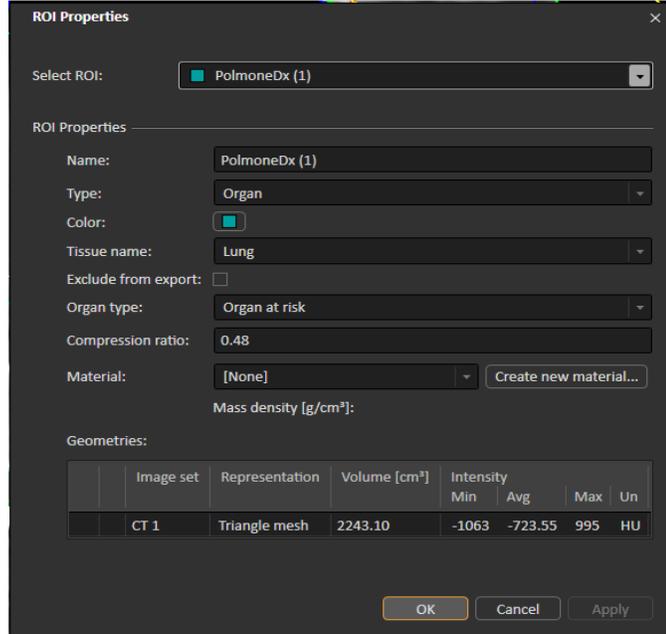


Figura 2.7 Riassunto delle proprietà di una ROI in RayStation.

A questo punto, è stata condotta una valutazione tra le ROI prodotte tramite l'atlante appena costruito e le ROI prodotte dall'algoritmo MBS. Come già descritto in precedenza, i parametri utilizzati per questa valutazione sono stati l'indice Dice e la distanza di Hausdorff. Per la determinazione di tali parametri è stato elaborato un altro script, riportato in appendice 2, che ne ha permesso il calcolo utilizzando direttamente la piattaforma di RayStation, senza il bisogno di esportare le strutture su supporti e software esterni.

In seguito a questa valutazione è stato variato il numero di strutture generate dall'algoritmo. Sono state condotte due diverse segmentazioni mediante atlante: una, nella quale erano state incluse tutte le strutture in una volta sola (usata anche nella parte precedente) e una dove le singole strutture venivano contornate volta per volta, lanciando singolarmente per ognuna di esse l'algoritmo di

segmentazione mediante atlante. I *workflow* relativi a ciascuno dei due metodi di segmentazione sono riportati nelle figure 2.8 e 2.9.

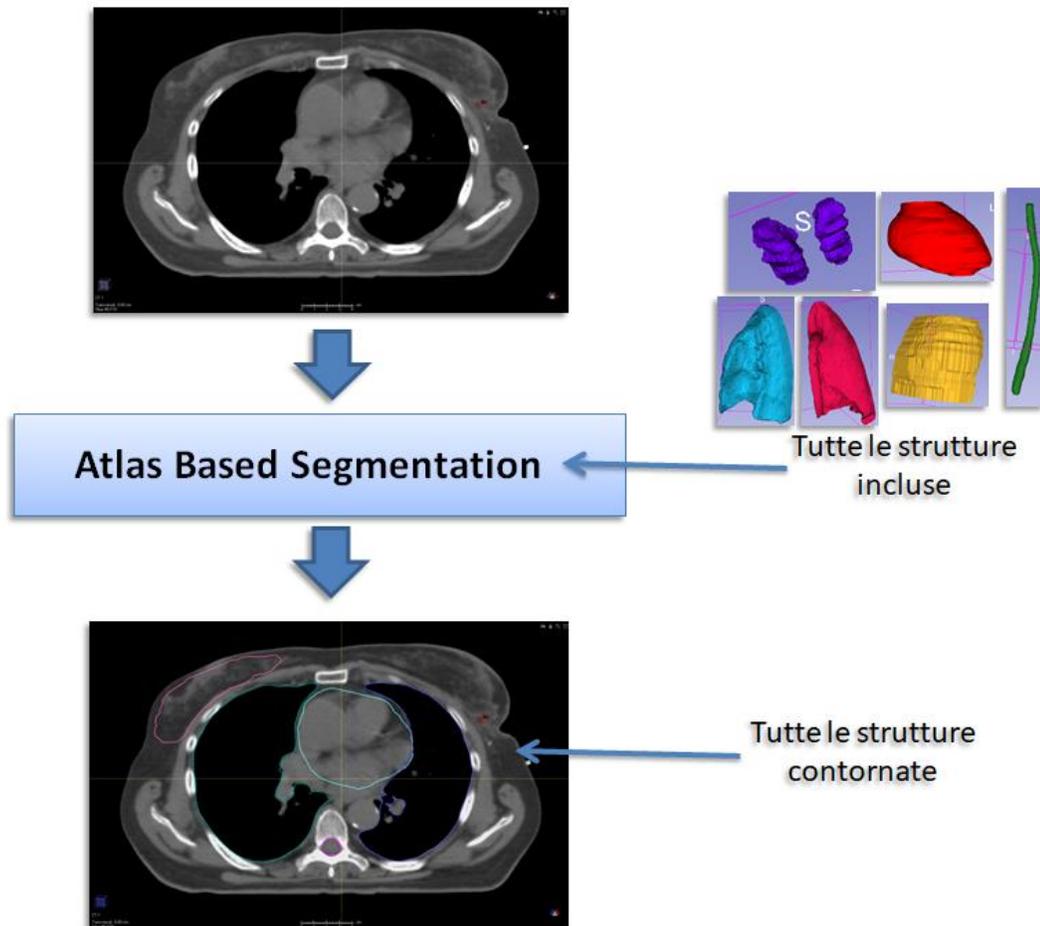


Figura 2.8. *Workflow* usato per il metodo di segmentazione 1.

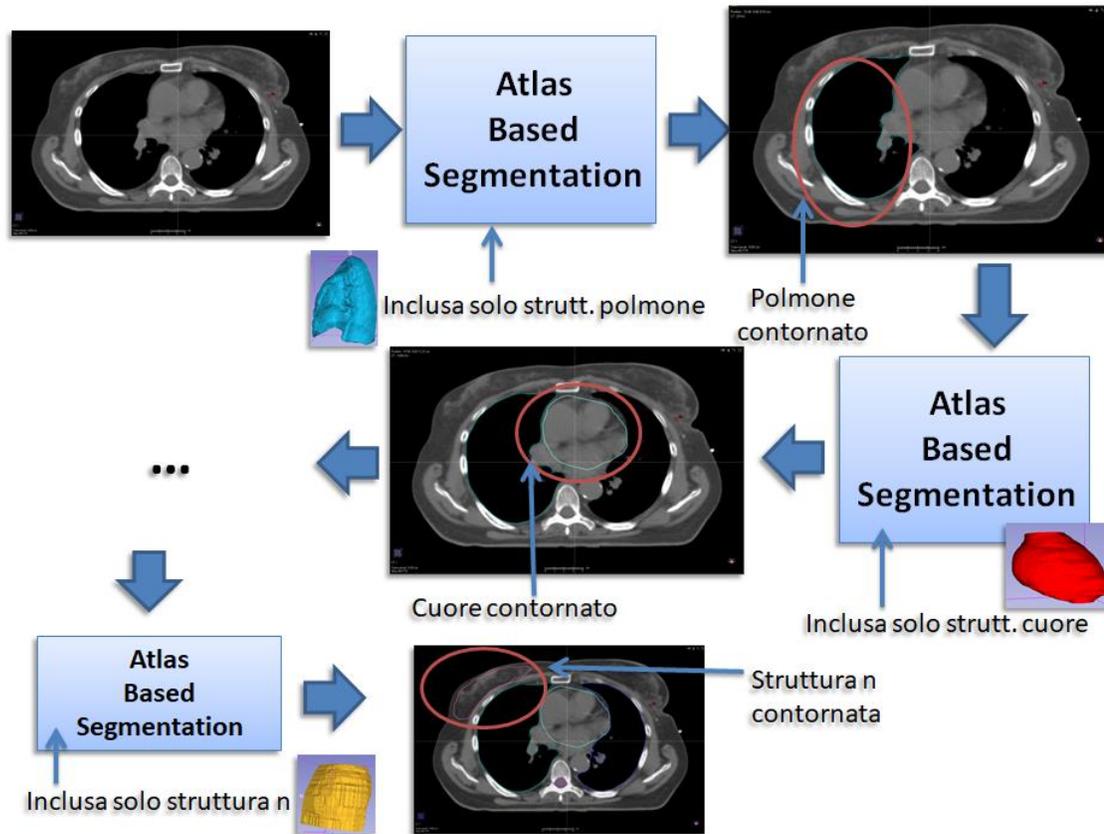


Figura 2.9. *Workflow* del metodo di segmentazione 2.

E' stato deciso di valutare le performance di questa seconda metodologia in rapporto alla prima, in quanto da una prima analisi qualitativa da parte di medici radioterapisti era emerso che contornando le ROI attraverso questo secondo metodo si otteneva un risultato migliore. Come si evincerà dall'analisi dei risultati, discussa nei capitoli successivi, la variazione nel numero di strutture incluse nell'algoritmo influenza sia la scelta del paziente di riferimento dall'atlante, sia l'andamento dell'algoritmo di registrazione ibrida.

Inoltre, sono state registrate le tempistiche di esecuzione di questi due metodi come parametro ulteriore di valutazione dei due algoritmi.

Al fine di automatizzare questi processi e l'acquisizione dei dati in generale, è stato elaborato uno script (appendice 3) che eseguisse le segmentazioni delle singole strutture automaticamente.

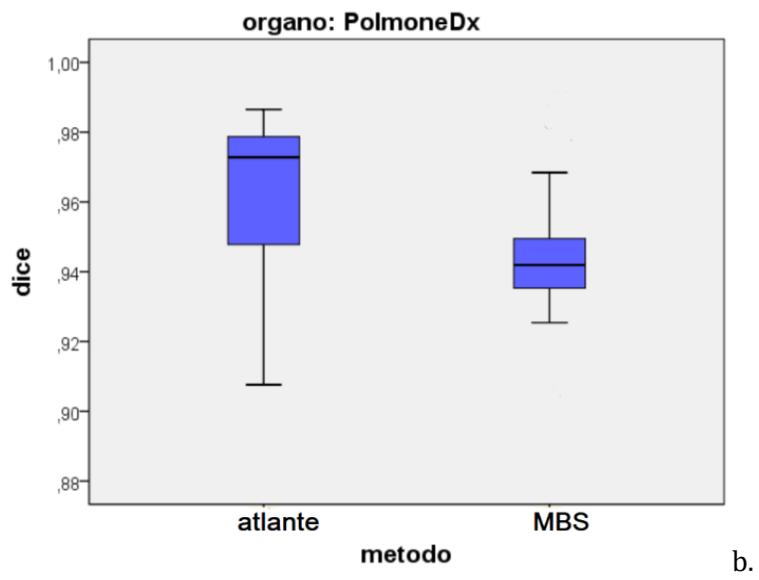
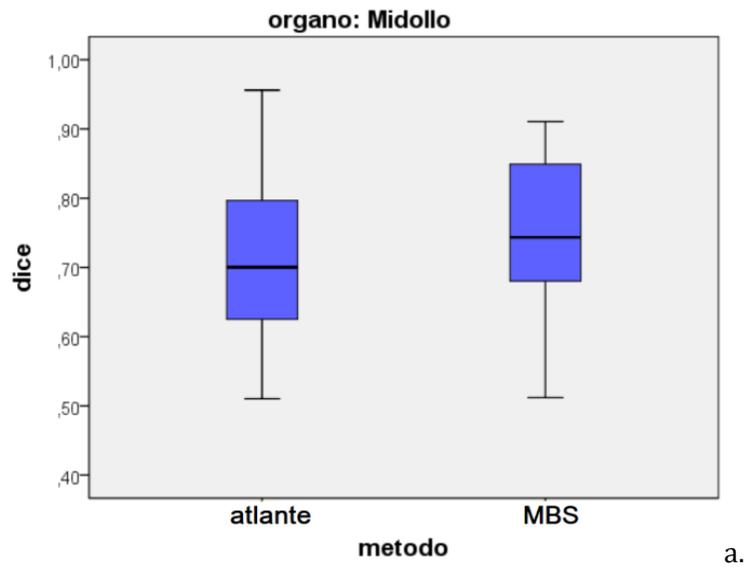
# Capitolo 3

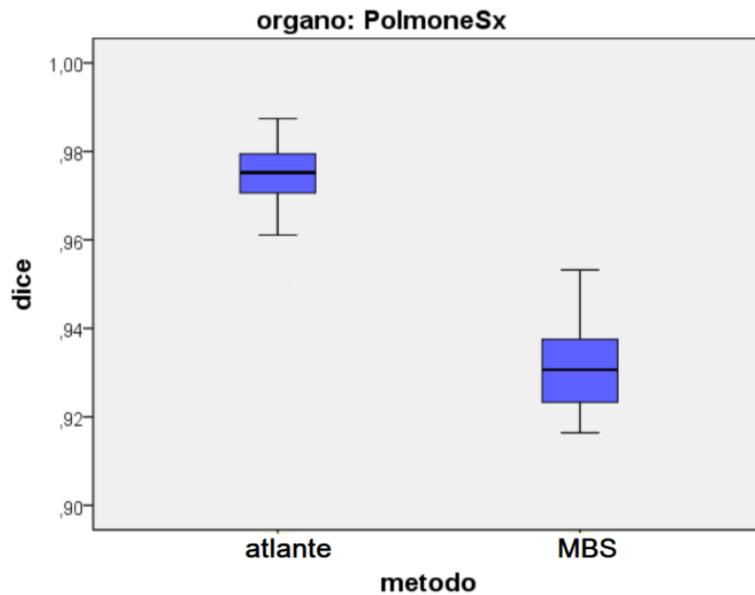
## 3.1 Confronto tra MBS e segmentazione con atlante rispetto al *gold standard*

Di seguito sono riportati i valori di DSC calcolati per i contorni eseguiti tramite la segmentazione con modello (MBS) di RayStation e la segmentazione tramite atlante (usando tutti i 42 pazienti disponibili) rispetto ai contorni precedentemente disegnati dal medico, che sono stati presi come riferimento. Il confronto è stato fatto solo per tre ROI corrispondenti a midollo, polmone destro e polmone sinistro, in quanto i modelli di RayStation erano presenti solo per queste strutture. Per ogni misura vengono forniti la mediana e l'intervallo interquartile (percentile 75, indicato con P\_75 nelle tabelle seguenti, e 25, indicato con P\_25 nelle tabelle seguenti) entro il quale essa varia all'interno di una determinata ROI. L'intervallo di confidenza al 95% delle mediane è, in questo caso, pari a un quarto dell'intervallo interquartile.

| organo    | DSC atlante | P_75 | P_25 | DSC MBS | P_75 | P_25 |
|-----------|-------------|------|------|---------|------|------|
| Midollo   | 0.70        | 0.81 | 0.62 | 0.74    | 0.85 | 0.66 |
| PolmoneDx | 0.97        | 0.98 | 0.95 | 0.94    | 0.95 | 0.93 |
| PolmoneSx | 0.97        | 0.98 | 0.97 | 0.93    | 0.92 | 0.94 |

Tabella 3.1. Indici Dice per le segmentazioni mediante atlante e MBS per le ROI di midollo, polmone destro e polmone sinistro.





c.

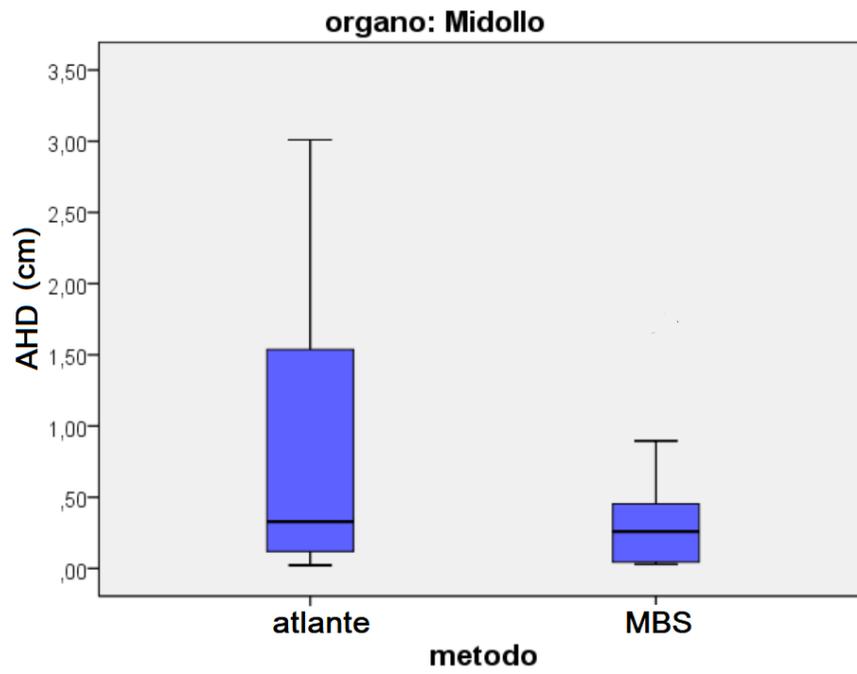
Figura 3.1. *Box plot* degli indici DSC per le ROI di midollo (a), polmone destro (b) e polmone sinistro (c).

Per quanto riguarda il midollo, si nota che i due algoritmi sono equivalenti in termini di indice Dice in quanto i due intervalli si sovrappongono. Per quanto riguarda invece il polmone destro, gli indici Dice relativi all'algoritmo MBS hanno meno dispersione, e un valore mediano inferiore rispetto agli indici Dice relativi all'atlante. Infine, per quanto riguarda la ROI del polmone sinistro, si può notare una minore dispersione degli indici per entrambi gli algoritmi, ma un valore mediano degli indici Dice relativi all'atlante decisamente maggiore rispetto alla MBS.

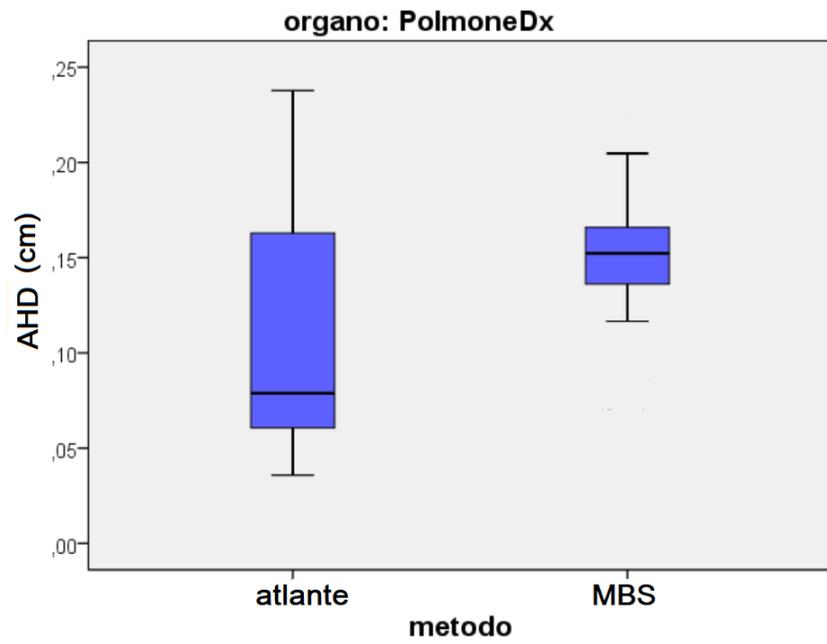
Di seguito vengono riportate le misure relative alla distanza di Hausdorff, in centimetri, delle quali vengono indicati il valore mediano e l'intervallo interquartile, analogamente a quanto fatto per l'indice Dice.

| organo    | AHD atlas (cm) | P_75 | P_25 | AHD MBS (cm) | P_75 | P_25 |
|-----------|----------------|------|------|--------------|------|------|
| Midollo   | 0.33           | 1.04 | 0.14 | 0.33         | 0.64 | 0.05 |
| PolmoneDx | 0.09           | 0.16 | 0.06 | 0.15         | 0.16 | 0.13 |
| PolmoneSx | 0.06           | 0.05 | 0.11 | 0.16         | 0.15 | 0.18 |

Tabella 3.2. Distanze di Hausdorff per le segmentazioni mediante atlante e MBS per le ROI di midollo, polmone destro e polmone sinistro.



a.



b.

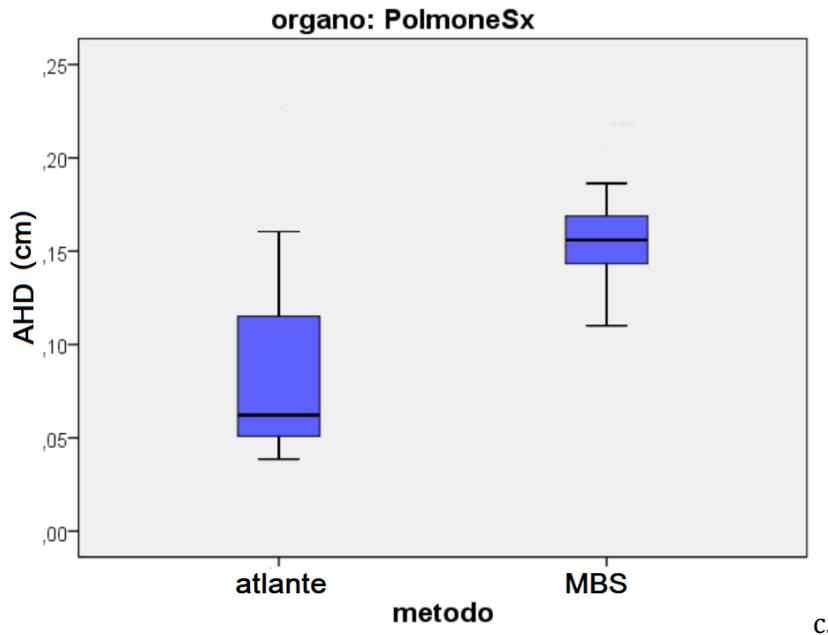


Figura 3.2. *Box plot* delle AHD per le ROI di midollo (a), polmone destro (b) e polmone sinistro (c).

Anche in questo caso, in accordo con i risultati ottenuti per gli indici Dice, si possono trarre conclusioni simili per le ROI prese in esame, in quanto le AHD per la MBS sono significativamente maggiori per i casi dei due polmoni, mentre nel caso del midollo si ha di nuovo sovrapposizione degli intervalli di confidenza delle due grandezze. In questo caso, la variabilità associata alla MBS è sempre minore di quella ottenuta attraverso l'atlante.

### 3.2 Variazione dei pazienti e delle strutture incluse nell'algoritmo

Nelle tabelle e figure seguenti sono riportati gli andamenti dell'indice Dice al variare dei pazienti di training inclusi nell'atlante sia per il metodo di segmentazione che prevede l'inclusione di tutte le strutture (metodo 1) sia per quello in cui le strutture vengono incluse volta per volta (metodo 2). Questa analisi

si è posta come obiettivi sia quello di verificare se uno dei due metodi fosse più efficace dell'altro, sia quello di evidenziare un eventuale miglioramento dei due algoritmi di segmentazione all'aumentare del numero di pazienti introdotti nell'atlante.

### Cuore

| n_pazienti | metodo 1 (DSC) | P_25 | P_75 | metodo 2 (DSC) | P_25 | P_75 |
|------------|----------------|------|------|----------------|------|------|
| 10         | 0.85           | 0.80 | 0.87 | 0.84           | 0.83 | 0.87 |
| 20         | 0.85           | 0.81 | 0.88 | 0.86           | 0.84 | 0.89 |
| 30         | 0.86           | 0.80 | 0.87 | 0.87           | 0.84 | 0.89 |
| 40         | 0.86           | 0.83 | 0.87 | 0.88           | 0.84 | 0.90 |

Tabella 3.3. Indici Dice per le segmentazioni ottenute mediante metodo 1 e metodo 2 per la ROI del cuore.

| n_pazienti | metodo 1 (AHD) (cm) | P_25 | P_75 | metodo 2 (AHD) (cm) | P_25 | P_75 |
|------------|---------------------|------|------|---------------------|------|------|
| 10         | 0.46                | 0.36 | 0.53 | 0.38                | 0.34 | 0.48 |
| 20         | 0.39                | 0.33 | 0.51 | 0.34                | 0.28 | 0.44 |
| 30         | 0.43                | 0.34 | 0.52 | 0.34                | 0.30 | 0.44 |
| 40         | 0.44                | 0.33 | 0.50 | 0.34                | 0.27 | 0.43 |

Tabella 3.4. Distanze di Hausdorff (cm) per le segmentazioni ottenute mediante metodo 1 e metodo 2 per la ROI del cuore.

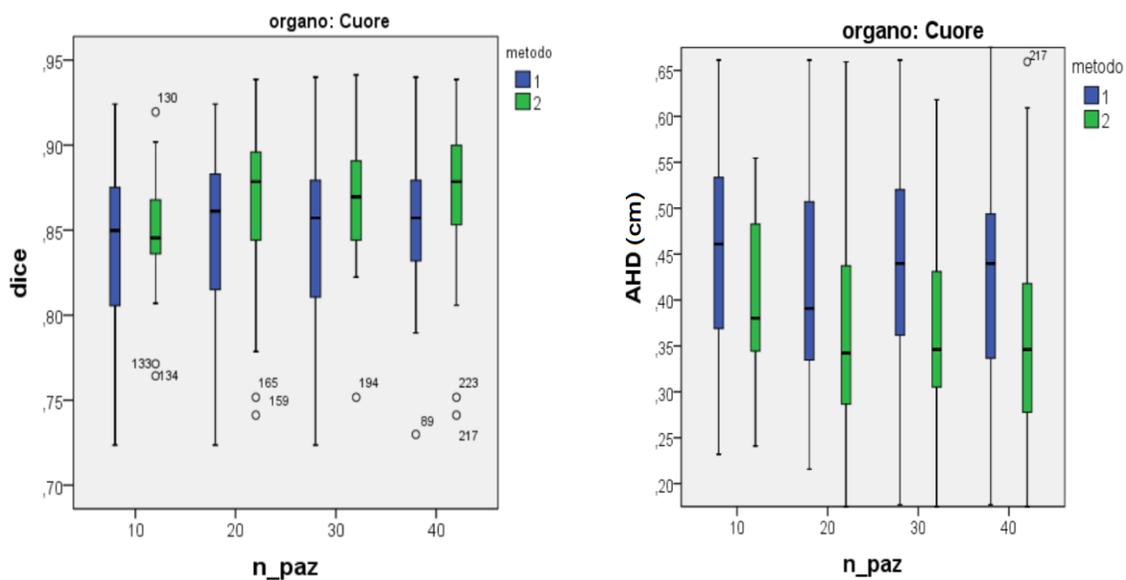


Figura 3.3. Box plot per l'indice Dice (sinistra) e per la AHD (destra) per la ROI del cuore

### Mammella destra

| n_pazienti | metodo 1 (DSC) | P_25 | P_75 | metodo 2 (DSC) | P_25 | P_75 |
|------------|----------------|------|------|----------------|------|------|
| 10         | 0,72           | 0,68 | 0,83 | 0,71           | 0,57 | 0,78 |
| 20         | 0,75           | 0,64 | 0,81 | 0,66           | 0,55 | 0,78 |
| 30         | 0,75           | 0,62 | 0,79 | 0,66           | 0,52 | 0,74 |
| 40         | 0,75           | 0,62 | 0,80 | 0,66           | 0,52 | 0,74 |

Tabella 3.5. Indici Dice per le segmentazioni ottenute mediante metodo 1 e metodo 2 per la ROI della mammella destra.

| n_pazienti | metodo 1 (AHD) (cm) | P_25 | P_75 | metodo 2 (AHD) (cm) | P_25 | P_75 |
|------------|---------------------|------|------|---------------------|------|------|
| 10         | 0,44                | 0,31 | 0,53 | 0,42                | 0,33 | 0,59 |
| 20         | 0,47                | 0,35 | 0,53 | 0,42                | 0,34 | 0,52 |
| 30         | 0,47                | 0,37 | 0,56 | 0,44                | 0,35 | 0,59 |
| 40         | 0,47                | 0,36 | 0,53 | 0,44                | 0,35 | 0,59 |

Tabella 3.6. Distanze di Hausdorff (cm) per le segmentazioni ottenute mediante metodo 1 e metodo 2 per la ROI della mammella destra.

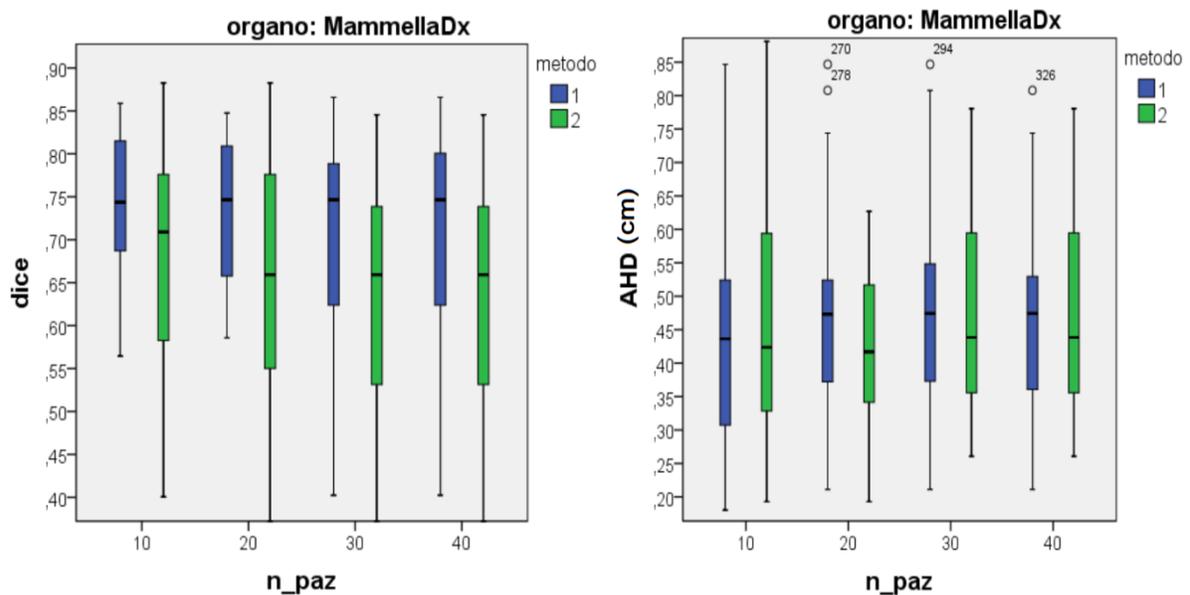


Figura 3.4. Box plot per l'indice Dice (sinistra) e per la AHD (destra) per la ROI della mammella destra.

## Midollo

| n_pazienti | metodo 1 (DSC) | P_25 | P_75 | metodo 2 (DSC) | P_25 | P_75 |
|------------|----------------|------|------|----------------|------|------|
| 10         | 0.72           | 0.68 | 0.79 | 0.78           | 0.69 | 0.85 |
| 20         | 0.71           | 0.66 | 0.76 | 0.74           | 0.69 | 0.83 |
| 30         | 0.62           | 0.74 | 0.79 | 0.76           | 0.69 | 0.84 |
| 40         | 0.70           | 0.65 | 0.82 | 0.78           | 0.65 | 0.86 |

Tabella 3.7. Indici Dice per le segmentazioni ottenute mediante metodo 1 e metodo 2 per la ROI del midollo

| n_pazienti | metodo 1 (AHD) (cm) | P_25 | P_75 | metodo 2 (AHD) (cm) | P_25 | P_75 |
|------------|---------------------|------|------|---------------------|------|------|
| 10         | 0.71                | 0.13 | 1.50 | 0.25                | 0.10 | 1.32 |
| 20         | 0.71                | 0.11 | 1.32 | 0.26                | 0.10 | 1.77 |
| 30         | 0.53                | 0.12 | 1.32 | 0.21                | 0.10 | 1.64 |
| 40         | 0.29                | 0.11 | 1.04 | 0.14                | 0.09 | 0.93 |

Tabella 3.8. Distanze di Hausdorff (cm) per le segmentazioni ottenute mediante metodo 1 e metodo 2 per la ROI del midollo.

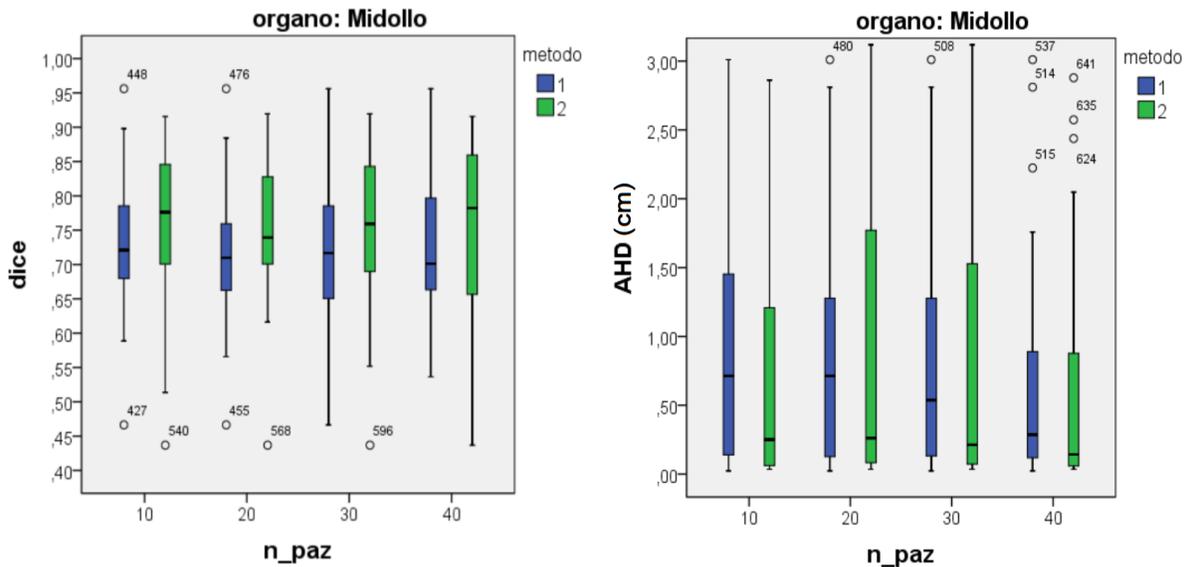


Figura 3.5. Box plot per l'indice Dice (sinistra) e per la AHD (destra) per la ROI del midollo

## Polmone destro

| n_pazienti | metodo 1 (DSC) | P_25 | P_75 | metodo 2 (DSC) | P_25 | P_75 |
|------------|----------------|------|------|----------------|------|------|
| 10         | 0,97           | 0,95 | 0,98 | 0,97           | 0,95 | 0,98 |
| 20         | 0,97           | 0,95 | 0,98 | 0,97           | 0,96 | 0,98 |
| 30         | 0,97           | 0,95 | 0,98 | 0,97           | 0,96 | 0,98 |
| 40         | 0,97           | 0,94 | 0,98 | 0,97           | 0,96 | 0,98 |

Tabella 3.9. Indici Dice per le segmentazioni ottenute mediante metodo 1 e metodo 2 per la ROI del polmone destro.

| n_pazienti | metodo 1 (AHD) (cm) | P_25 | P_75 | metodo 2 (AHD) (cm) | P_25 | P_75 |
|------------|---------------------|------|------|---------------------|------|------|
| 10         | 0,08                | 0,05 | 0,13 | 0,08                | 0,05 | 0,13 |
| 20         | 0,08                | 0,06 | 0,13 | 0,08                | 0,06 | 0,13 |
| 30         | 0,08                | 0,06 | 0,16 | 0,08                | 0,06 | 0,15 |
| 40         | 0,09                | 0,06 | 0,17 | 0,08                | 0,06 | 0,14 |

Tabella 3.10. Distanze di Hausdorff (cm) per le segmentazioni ottenute mediante metodo 1 e metodo 2 per la ROI del polmone destro.

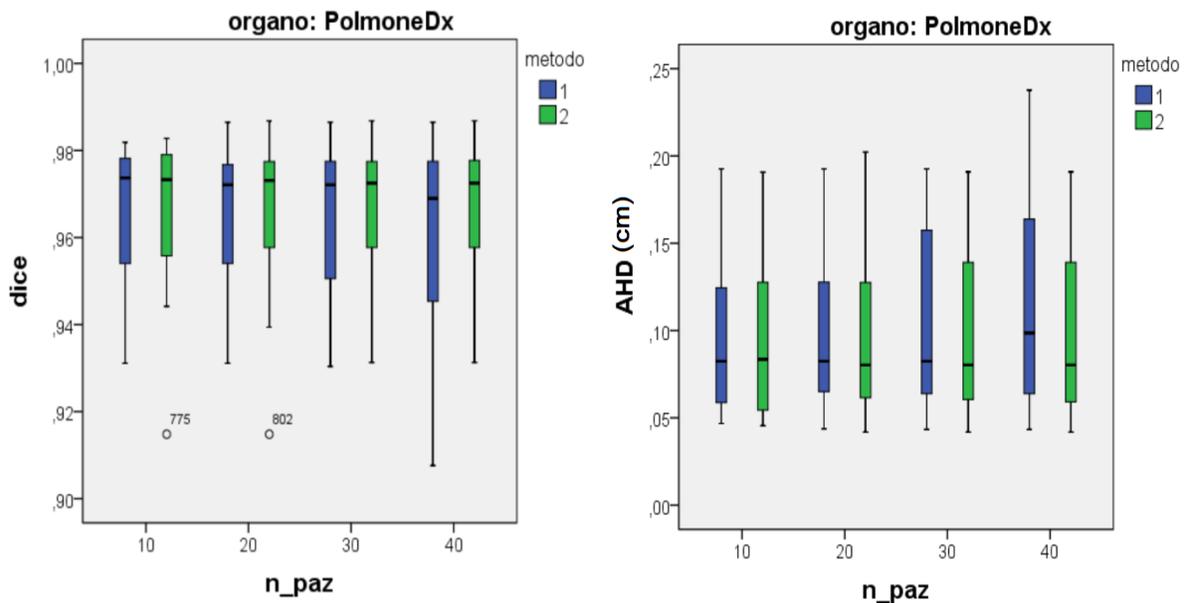


Figura 3.6. Box plot per l'indice Dice (sinistra) e per la AHD (destra) per la ROI del polmone destro.

## Polmone sinistro

| n_pazienti | metodo 1 (DSC) | P_25 | P_75 | metodo 2 (DSC) | P_25 | P_75 |
|------------|----------------|------|------|----------------|------|------|
| 10         | 0.97           | 0.97 | 0.98 | 0.97           | 0.97 | 0.98 |
| 20         | 0.97           | 0.97 | 0.98 | 0.97           | 0.97 | 0.98 |
| 30         | 0.97           | 0.97 | 0.98 | 0.97           | 0.97 | 0.98 |
| 40         | 0.97           | 0.97 | 0.98 | 0.97           | 0.97 | 0.98 |

Tabella 3.11. Indici Dice per le segmentazioni ottenute mediante metodo 1 e metodo 2 per la ROI del polmone sinistro

| n_pazienti | metodo 1 (AHD) (cm) | P_25 | P_75 | metodo 2 (AHD) (cm) | P_25 | P_75 |
|------------|---------------------|------|------|---------------------|------|------|
| 10         | 0.06                | 0.05 | 0.11 | 0.06                | 0.05 | 0.10 |
| 20         | 0.06                | 0.05 | 0.11 | 0.06                | 0.05 | 0.08 |
| 30         | 0.06                | 0.05 | 0.12 | 0.06                | 0.05 | 0.09 |
| 40         | 0.06                | 0.05 | 0.12 | 0.06                | 0.05 | 0.09 |

Tabella 3.12. Distanze di Hausdorff (cm) per le segmentazioni ottenute mediante metodo 1 e metodo 2 per la ROI del polmone sinistro.

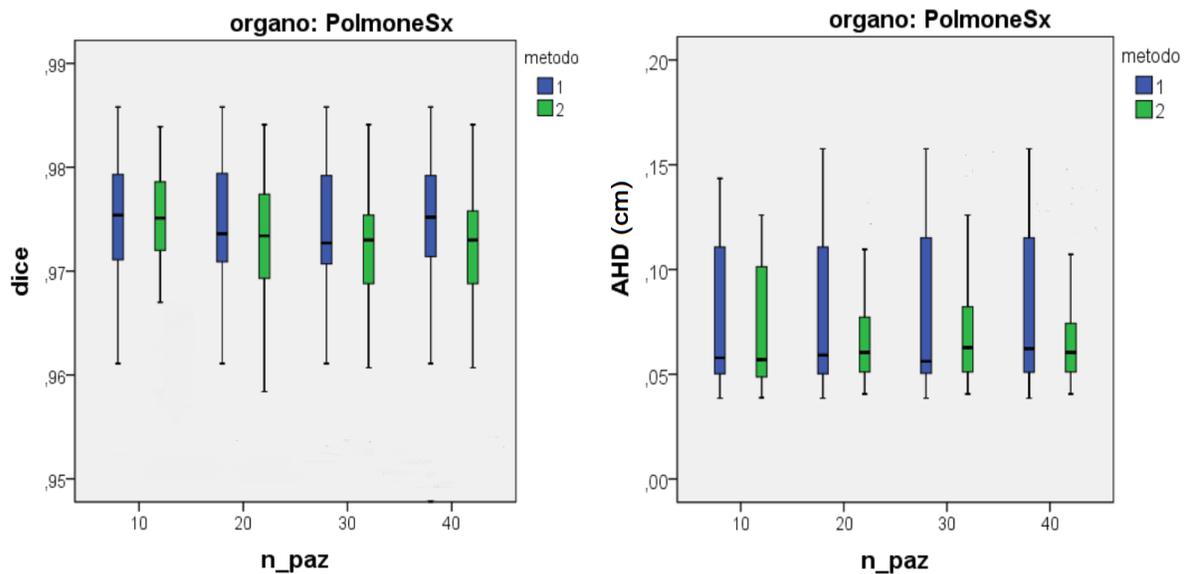


Figura 3.7. Box plot per l'indice Dice (sinistra) e per la AHD (destra) per la ROI del polmone sinistro.

## Tiroide

| n_pazienti | metodo 1 (DSC) | P_25 | P_75 | metodo 2 (DSC) | P_25 | P_75 |
|------------|----------------|------|------|----------------|------|------|
| 10         | 0.33           | 0.22 | 0.46 | 0.45           | 0.27 | 0.60 |
| 20         | 0.30           | 0.20 | 0.43 | 0.34           | 0.21 | 0.48 |
| 30         | 0.30           | 0.18 | 0.42 | 0.33           | 0.14 | 0.55 |
| 40         | 0.33           | 0.18 | 0.44 | 0.24           | 0.13 | 0.54 |

Tabella 3.13. Indici Dice per le segmentazioni ottenute mediante metodo 1 e metodo 2 per la ROI della tiroide.

| n_pazienti | metodo 1 (AHD) (cm) | P_25 | P_75 | metodo 2 (AHD) (cm) | P_25 | P_75 |
|------------|---------------------|------|------|---------------------|------|------|
| 10         | 0.47                | 0.33 | 0.70 | 0.26                | 0.19 | 0.65 |
| 20         | 0.40                | 0.29 | 0.74 | 0.33                | 0.18 | 0.56 |
| 30         | 0.45                | 0.34 | 0.77 | 0.29                | 0.19 | 0.56 |
| 40         | 0.40                | 0.26 | 0.74 | 0.29                | 0.18 | 0.56 |

Tabella 3.14. Distanze di Hausdorff (cm) per le segmentazioni ottenute mediante metodo 1 e metodo 2 per la ROI della tiroide.

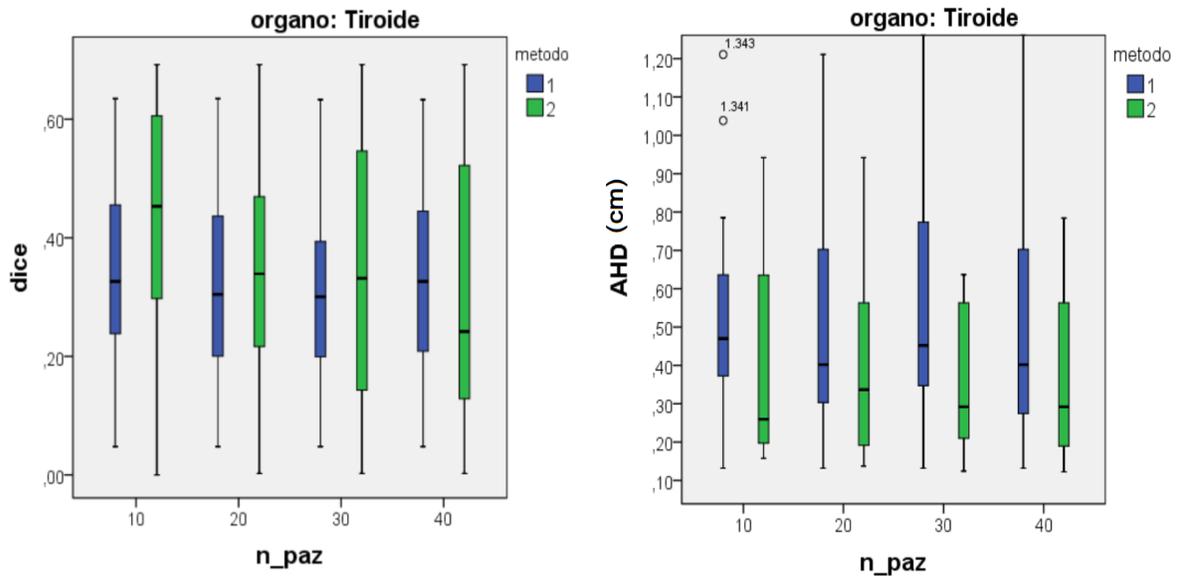


Figura 3.8. Box plot per l'indice Dice (sinistra) e per la AHD (destra) per la ROI della tiroide.

Dall'analisi dei risultati sopra presentati, per quanto riguarda tutte le ROI, si ha che i due metodi, non presentano intervalli di confidenza significativamente differenti per quanto riguarda gli indici AHD e Dice. Si possono comunque fare alcune osservazioni. Ad esempio, per quanto riguarda la ROI del cuore, l'indice Dice del metodo 1 varia su un intervallo più ampio rispetto al 2, mentre la AHD è molto variabile in entrambi i casi. Non c'è molta variabilità tra i risultati relativi ai due polmoni ed entrambi gli algoritmi di segmentazione forniscono risultati aventi indici Dice e AHD molto buoni. Per quanto riguarda la tiroide, si ha che nessuno dei due metodi raggiunge un valore di Dice accettabile, in quanto tutti minori di 0.6. Un tale valore di indice Dice sta a significare che il volume dell'intersezione tra le ROI generate dall'algoritmo e quelle di riferimento è minore del 30% della somma dei due volumi individuali: non è stato quindi ottenuto un *overlap* sufficiente. Nonostante ciò, i valori di AHD relativi alla tiroide risultano comunque bassi in quanto si tratta di un organo piccolo e poco esteso spazialmente. Per quanto riguarda il midollo, invece, dato che ha un'estensione significativa lungo l'asse longitudinale del corpo, si ha che una segmentazione sbagliata può portare anche a una AHD di 2.00 cm.

Per quanto riguarda l'andamento degli indici Dice e AHD, in generale, per tutte le ROI considerate, è stato evidenziato si trova un significativo andamento di proporzionalità inversa (eseguendo una correlazione di Pearson si trova un p-value < 0,01), in accordo con quanto atteso.

Sulla base di questa analisi non è possibile quindi identificare uno dei due metodi come più performante rispetto all'altro.

Si è quindi deciso di valutare il tempo di esecuzione degli algoritmi associati ai metodi 1 e 2.

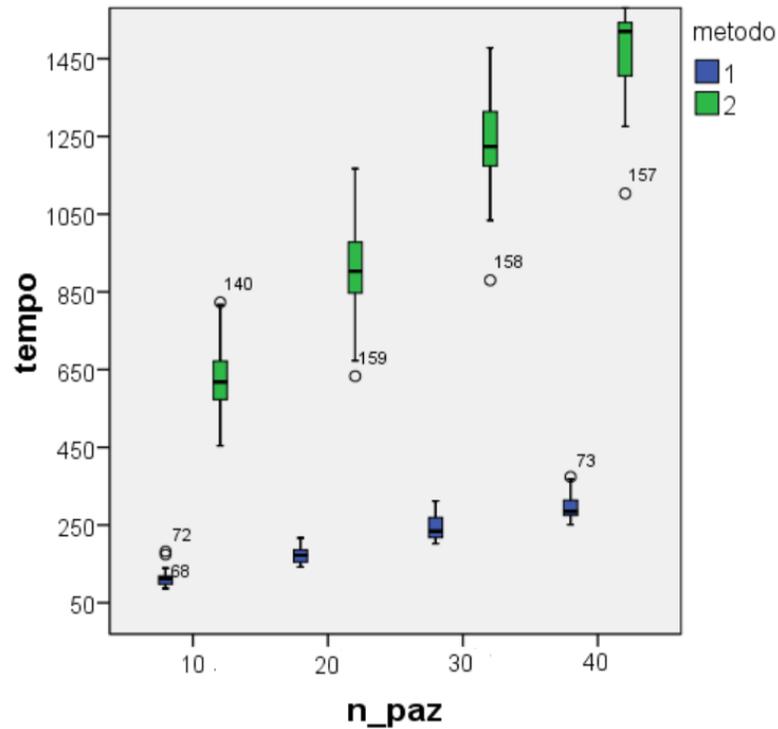


Figura 3.9. Andamento del tempo di esecuzione degli algoritmi associati ai due metodi di segmentazione.

Dal grafico 3.9 si nota che, per quanto riguarda il metodo 1, all'aumentare del numero di pazienti, non aumenta di molto il tempo di esecuzione dell'algoritmo, a differenza del metodo 2 che, oltre ad avere un tempo di esecuzione nettamente maggiore dell'altro, è anche soggetto a un maggior aumento di tempo computazionale all'aumentare del numero di pazienti inseriti nell'atlante.

Per avere una stima dell'aumento delle tempistiche di esecuzione all'aumentare dei pazienti è stato fatto un fit lineare sui valori delle tempi di esecuzione in relazione al numero di pazienti.

| Metodo | Aumento lineare | $R^2$ |
|--------|-----------------|-------|
| 1      | 6 s/paziente    | 0,9   |
| 2      | 30 s/paziente   | 0,9   |

Tabella 3.15 Aumento lineare delle tempistiche di esecuzione degli algoritmi all'aumentare dei pazienti

Dato che dall'analisi degli indici Dice e della distanza di Hausdorff non emergono differenze significative nella qualità della segmentazione generata dai due metodi, si è scelto di mantenere nella pratica clinica il metodo 1, a causa sia del minore tempo di esecuzione che esso richiede, sia del minore aumento lineare del tempo di esecuzione all'aumentare del numero di pazienti.

Come suggerito dallo studio [6], si è evidenziata una correlazione tra indice Dice e volume della ROI di interesse per il metodo scelto (metodo 1). Nell'asse delle ordinate è riportato il coefficiente Dice associato alla ROI in esame con la relativa barra di errore, mentre nelle ascisse è riportata la mediana dei volumi di tutte le ROI dei pazienti in esame.

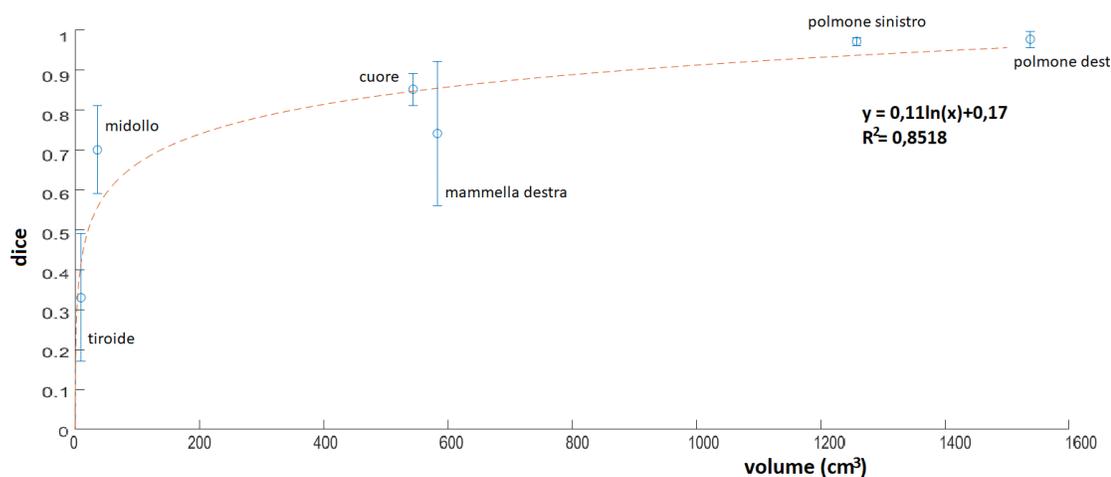


Figura 3.10. Correlazione tra indice Dice e mediana dei volumi delle ROI in esame.

Dal grafico 3.10 si nota una relazione di tipo logaritmico tra l'andamento dell'indice Dice e la mediana del volume della ROI considerata: questo significa che ROI di volume maggiore hanno una segmentazione automatica migliore. Infatti, come evidenziato anche nei risultati precedenti, i casi dei due polmoni presentavano indici Dice maggiori rispetto al caso della tiroide. L'individuazione di tale andamento permette anche la definizione di un volume limite al di sotto del quale l'algoritmo darà valori di Dice minori di una determinata *threshold* fissata a priori.

### **3.3 Sviluppi futuri**

Il metodo di segmentazione mediante l'uso di atlanti anatomici che si è scelto di mantenere nella pratica clinica e del quale è stata automatizzata l'esecuzione ha presentato dei risultati soddisfacenti per tutte le ROI analizzate nella regione del torace tranne che nel caso della tiroide. Per le ROI che hanno presentato buoni risultati, l'intervento che il medico deve apportare ai contorni generati automaticamente è minimo, il che permette un guadagno rilevante nelle tempistiche del lavoro di segmentazione.

L'espansione del database dell'atlante con nuovi pazienti, in futuro, porterà quindi alla riduzione maggiore dell'intervento da parte del medico. Inoltre, lo sviluppo e l'implementazione di sotto-atlanti contenenti pazienti omogenei per caratteristiche (costituzione fisica, età, trattamenti chirurgici subiti, volume delle ROI di interesse...) renderebbe la performance di tale algoritmo di segmentazione ancora migliore.

Inoltre, la costruzione di un atlante dedicato all'individuazione di sotto-strutture di particolari organi (i.e. cuore) potrebbe essere utile al fine di individuare cause relative a tossicità o complicazioni nella fase successiva al trattamento radioterapico.

# Capitolo 4

## 4.1 Costruzione di un classificatore per l'individuazione delle segmentazioni corrette al fine di espandere un atlante

Come riportato nel capitolo 2, la costruzione di un atlante avente al suo interno strutture contornate correttamente è fondamentale per la buona riuscita dell'algoritmo di segmentazione. I metodi di valutazione usati nei capitoli precedenti hanno come prerequisito principale il confronto con un *gold standard* definito, in questo caso, dal medico: non risultano quindi applicabili nel valutare una segmentazione automatica di una struttura per la quale non ne esista una di riferimento del medico. In questi casi quindi è necessaria la ricerca di una metodologia differente. L'indice di valutazione, basato sul giudizio del medico, usato nello studio di Maffei et al., 2016 [9] potrebbe essere adottato in questo caso (attraverso l'utilizzo di un valore di soglia oltre il quale permettere l'inclusione nell'atlante), ma ha lo svantaggio di richiedere la presenza di un medico che potrebbe trovarsi a dover valutare una vasta quantità di pazienti. L'obiettivo che ci si pone in questa sezione del lavoro è quindi la ricerca di parametri che gettino le basi per la costruzione di un classificatore in grado di riconoscere una corretta segmentazione. Come già evidenziato nel capitolo 1, il feedback del medico sulla

catena di *image processing* che porta alla contornazione delle strutture di interesse per il trattamento è essenziale e al giorno d'oggi non è possibile sostituirlo con nessun algoritmo automatico. La funzione del classificatore in oggetto in questa catena è quindi quella di fornire un supporto al medico nell'individuare i pazienti da includere nell'atlante.

Il classificatore ricercato è quindi di tipo binario, dove l'output positivo (P) corrisponde all'aggiunta del paziente all'atlante, mentre quello negativo (N) corrisponde all'esclusione del paziente dall'atlante. Come in ogni problema di predizione a due classi, si può rappresentare l'insieme delle situazioni possibili usando una tabella di contingenza 2x2:

|                  |    | Valore vero:      |                   |
|------------------|----|-------------------|-------------------|
|                  |    | P                 | N                 |
| Valore predetto: | P' | Vero<br>Positivo  | Falso<br>Positivo |
|                  | N' | Falso<br>Negativo | Vero<br>Negativo  |

Figura 4.1. Tabella di contingenza nel caso di un problema di predizione a due classi.

L'output P o N fornito dal classificatore si basa sulla risoluzione dell'espressione logica:

$$C_1 \wedge C_2 \wedge \dots \wedge C_n \quad (4.1)$$

dove  $C_1, \dots, C_n$  sono le diverse condizioni che la segmentazione da valutare dovrà rispettare. In caso di veridicità della suddetta espressione logica, il risultato sarà positivo (1), cioè la ROI sarà inserita direttamente nell'atlante, mentre in caso di espressione falsa il risultato sarà negativo (0), cioè la ROI dovrà essere valutata dal medico.

I pazienti esclusi possono poi essere esaminati da un medico che, tramite il già citato metodo di valutazione con soglia, decide se escluderli completamente dall'atlante in quanto le strutture sono state effettivamente contornate in maniera errata, oppure includerli comunque nell'atlante in caso di falso negativo (FN).

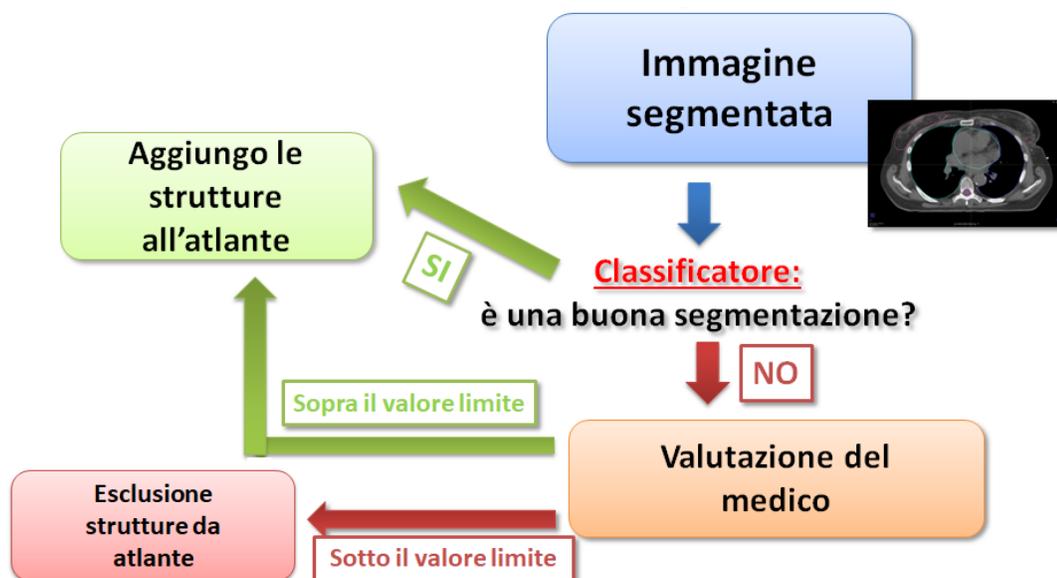


Figura 4.2. *Workflow* del processo di espansione dell'atlante.

Dallo schema mostrato in figura 4.2 è evidente come sia necessario che i parametri del classificatore siano impostati in modo da evitare quanto più possibile i falsi positivi (cioè si vuole evitare di inserire nell'atlante delle strutture mal contornate), anche a costo di ottenere dei falsi negativi, che verranno comunque individuati dal medico nella fase successiva; per questo motivo, nell'espressione 4.1, è stato utilizzato l'operatore AND.

## 4.2 Struttura del classificatore

Nella figura 4.3 è mostrato il *workflow* che si è impostato per il funzionamento del classificatore (in alto) e il processo che si è seguito per la sua costruzione (in basso).

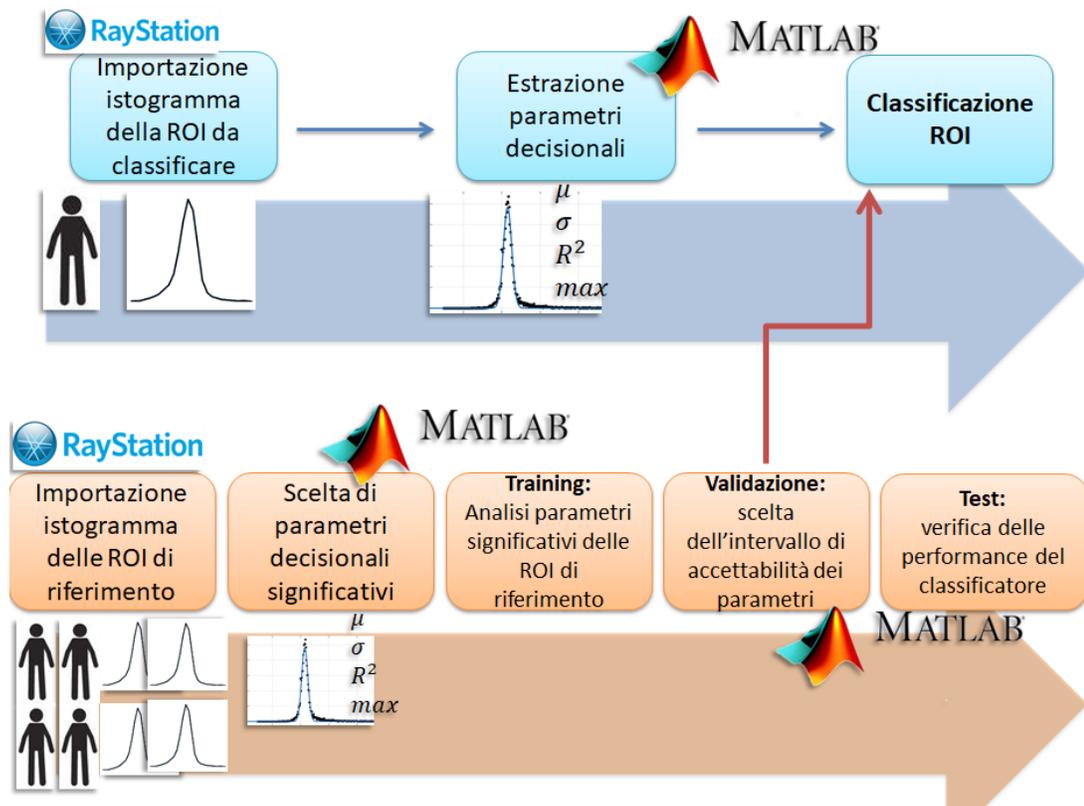


Figura 4.3. *Workflow* di esecuzione (in alto) e costruzione (in basso) del classificatore proposto.

Per la determinazione dei parametri che saranno usati dal classificatore nella decisione, sono stati analizzati gli istogrammi dei voxel contenuti all'interno del volume delle strutture di interesse. I voxel analizzati hanno una base quadrata con lato di 0.977 mm e spessore delle slice di 3 mm; le immagini fornite sono di 512x512 pixel per ognuna delle 121 slice che rappresentano la regione anatomica del torace, sulla quale si è focalizzato lo studio. I pazienti e le strutture utilizzati

sono gli stessi della sezione precedente (42 pazienti provenienti dal database dei trattamenti PBI alla mammella sinistra), a esclusione della ROI della tiroide, a causa dei risultati non soddisfacenti ottenuti per la sua segmentazione. Tali istogrammi sono stati estratti da RayStation mediante uno script (riportato in appendice 4). Nella figura 4.4 sono riportati, a titolo di esempio, gli istogrammi per le ROI di polmone sinistro, mammella destra, cuore e midollo. Tali istogrammi si riferiscono alle ROI di un singolo paziente. Sull'asse delle ascisse sono riportati i valori di Hounsfield Units (HU) e sulle ordinate la loro frequenza normalizzata in base al numero totale di voxel contenuti nel volume della ROI in esame. I bin di tale istogramma sono larghi 3 HU; la larghezza dei bin è impostabile sempre mediante lo script di estrazione dell'istogramma (appendice 4).

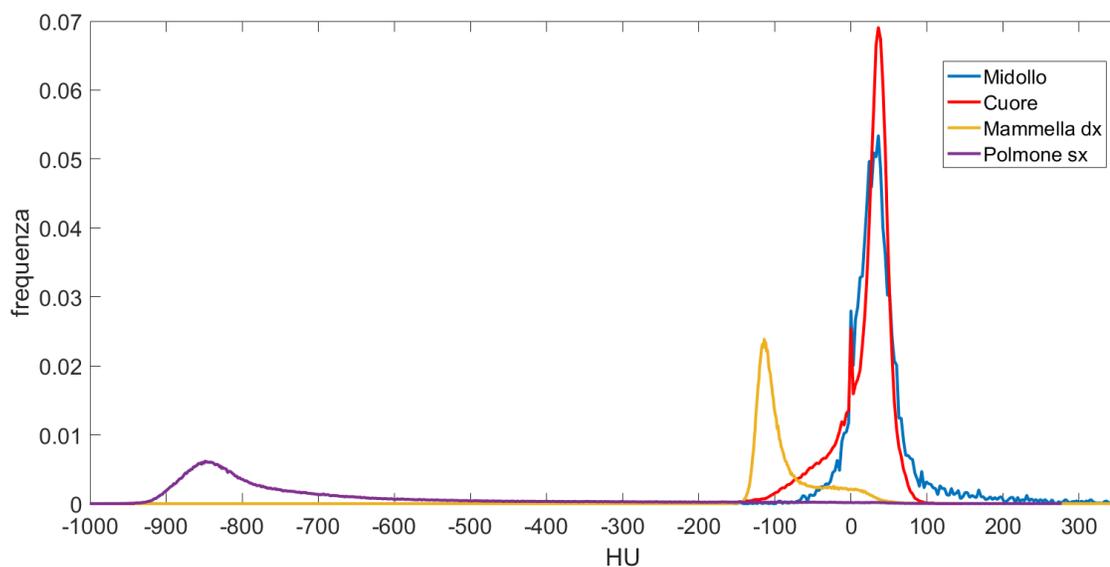
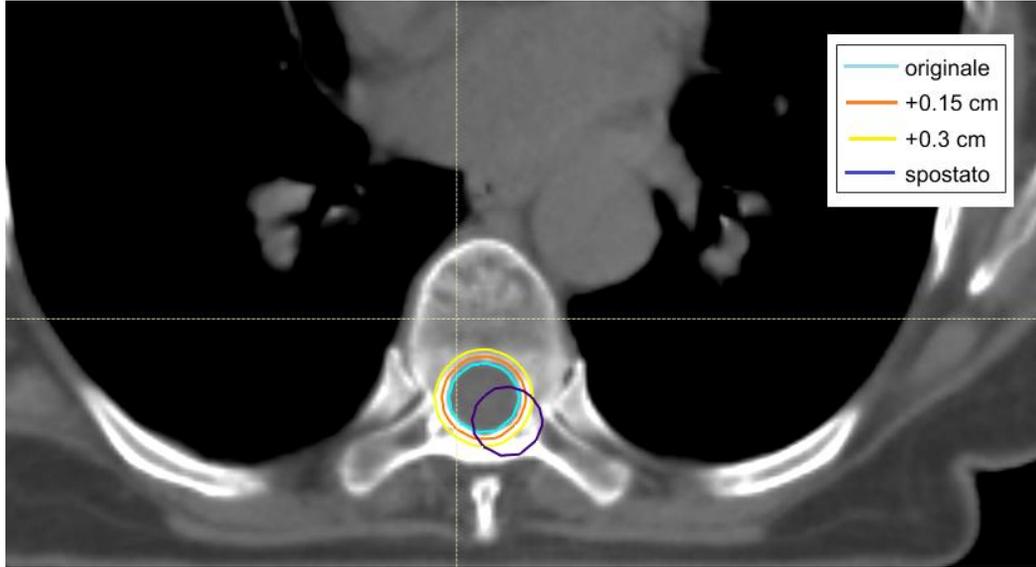


Figura 4.4. Iistogrammi relativi alle ROI di polmone sinistro, mammella destra, cuore e midollo.

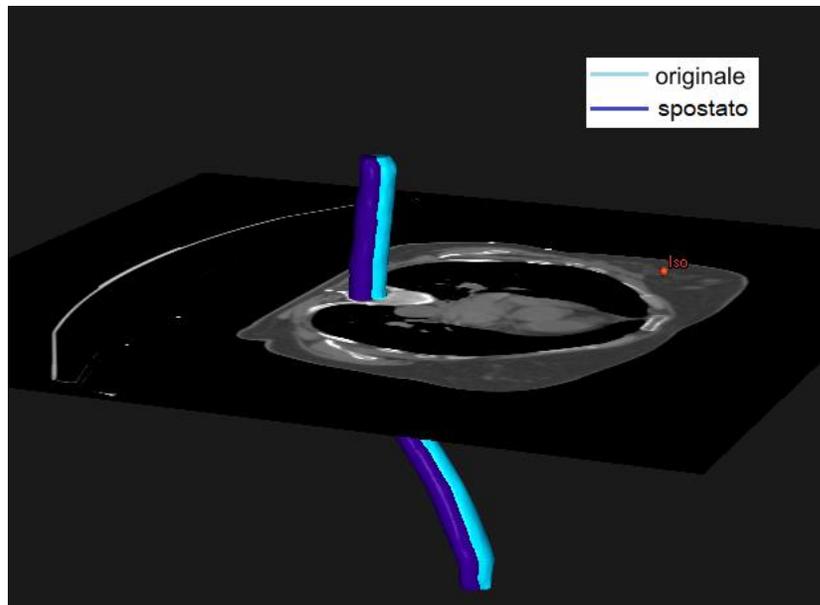
E' stato deciso di applicare il classificatore, in questa fase iniziale di sviluppo, solo alle ROI di cuore e midollo, in quanto modellizzabili come gaussiane. Per la ROI del polmone, a causa del suo volume elevato e della grande variabilità di HU al suo

interno, sarebbe molto più difficile individuare cambiamenti nelle caratteristiche della gaussiana dovuti a un errore di segmentazione. Inoltre, nel caso del polmone, le ottime performance dell'algoritmo di segmentazione riscontrate nella sezione precedente, unite al suo elevato volume, rendono l'individuazione di eventuali voxel non appartenenti a tale ROI molto difficile. Per quanto riguarda la mammella controlaterale, modellizzando la forma del suo istogramma con una gaussiana singola si ottiene un valore di  $R^2$  pari a 0.86: un'analisi più corretta di tale ROI sarebbe quindi ottenibile mediante un modello di curva più complesso.

Dopo la scelta delle ROI per le quali costruire il classificatore, si è passati alla costruzione di ROI modificate, in modo da evidenziare le eventuali differenze con le ROI originali. Per ottenere ciò, sono state espanse le ROI originali, tramite l'apposita funzione di RayStation, di un margine uniforme in tutte le direzioni. Per la regione del cuore, l'espansione è stata di 0.3 cm, 0.5 cm e 0.8 cm. Per la regione del midollo sono state scelte espansioni di 0.15 cm e 0.3 cm. Inoltre, dato che un errore molto frequente nella segmentazione del midollo è quello della traslazione sistematica dello stesso, si è deciso di aggiungere alle ROI modificate una del midollo traslata di 0.5 cm nella direzione della bisettrice lungo il piano perpendicolare all'asse longitudinale. Queste ROI modificate sono state costruite, tramite lo script riportato in appendice 5, per tutti i 42 pazienti analizzati nella sezione precedente. Nella figura 4.5a è riportata una visione sul piano trasversale delle ROI modificate, mentre nella figura 4.5b è riportata una visione lungo l'asse longitudinale dei modelli 3D associati alle ROI del midollo originale e traslato.



a.



b.

Figura 4.5a. Visione trasversale della ROI originale del midollo e di quelle modificate.

Figura 4.5b. Visione longitudinale della ROI originale e della ROI traslata del midollo.

Nella figura 4.6 è invece riportato l'istogramma della ROI originale del midollo insieme a quelli delle ROI modificate.

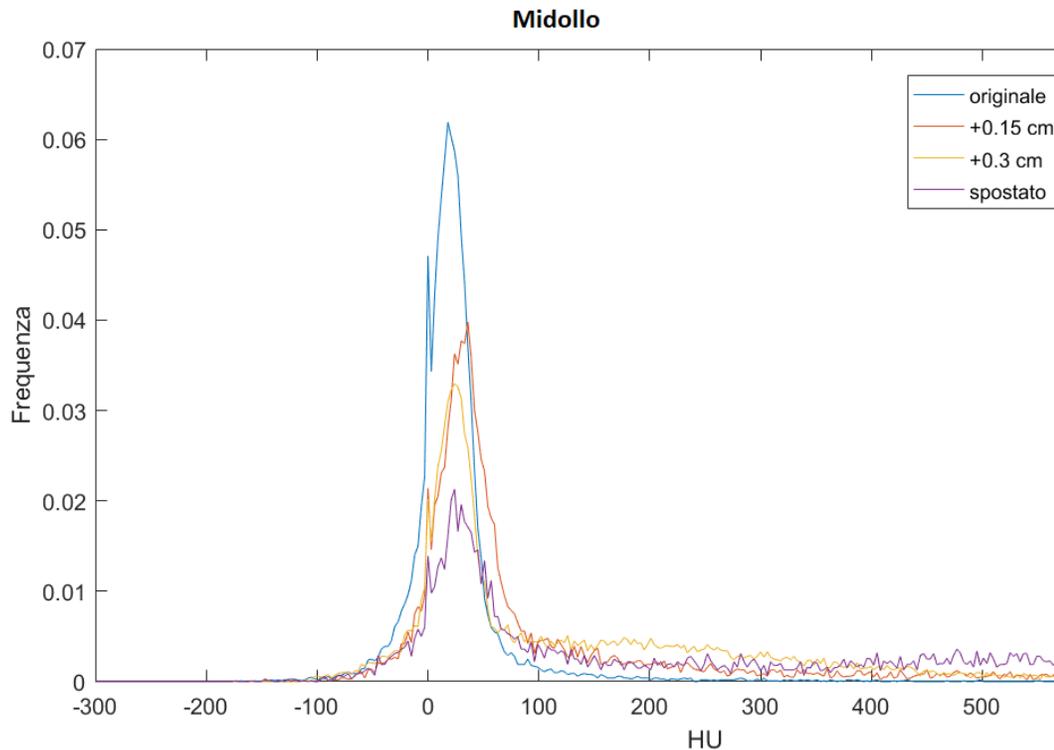


Figura 4.6. Confronto tra andamento della frequenza delle HU nella ROI originale del midollo e nelle ROI modificate.

Anche in questo caso e in quello della figura 4.7 gli istogrammi sono a titolo di esempio e si riferiscono quindi alle ROI appartenenti a un singolo paziente. Indipendentemente dal paziente selezionato, tuttavia, è stato possibile notare come le ROI modificate abbiano un istogramma di altezza minore rispetto a quello della ROI originale. Questo è un risultato atteso, in quanto conseguenza della normalizzazione di tale grafico. Inoltre, sono presenti dei conteggi anche per valori di HU maggiori, dovuti alla presenza di osso attorno al midollo (dato che l'osso presenta un valore di HU maggiore rispetto al tessuto del midollo, come si può notare dalla figura 4.5a).

Nella figura 4.7 è invece riportata la visione trasversale della ROI originale del cuore e delle sue versioni modificate. Anche in questo caso è possibile notare come le ROI espanse comprendano anche voxel esterni all'organo di interesse (nello specifico appartenenti ai polmoni).

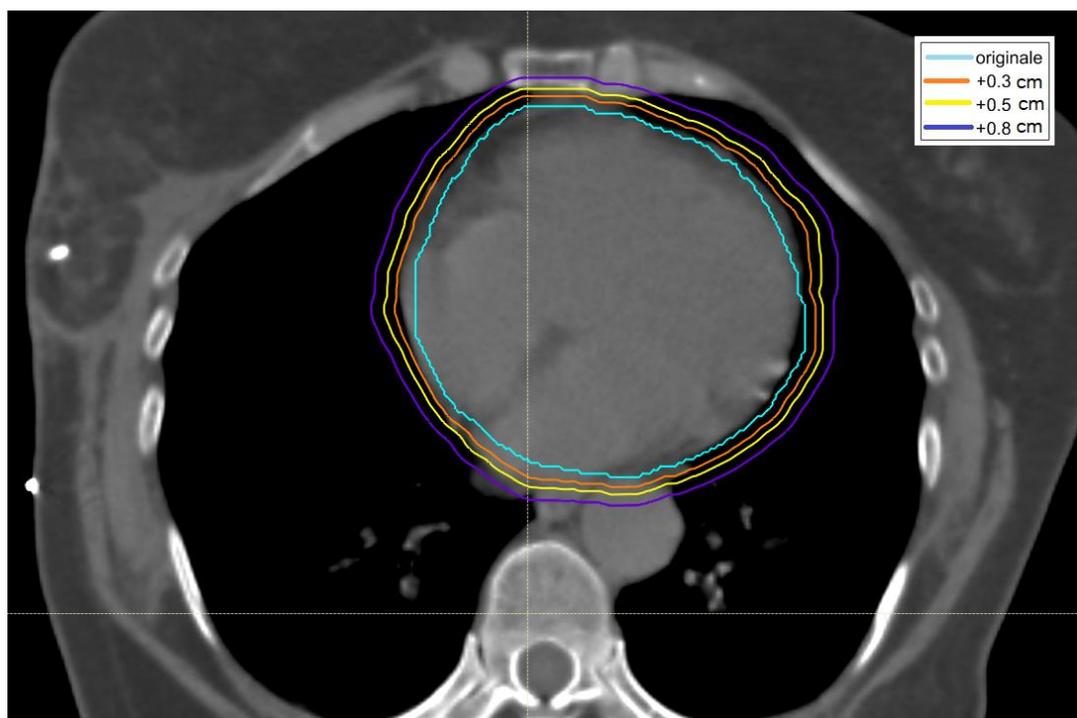


Figura 4.7. Visione trasversale della ROI originale del cuore e delle ROI modificate.

Nella figura 4.8 è invece riportato il confronto tra gli istogrammi delle ROI modificate del cuore e di quella originale. Nel caso di questo istogramma si è deciso di rappresentare una frazione del grafico in scala logaritmica per evidenziare la presenza di conteggi nelle zone con valori di HU minori, a testimonianza del fatto che le ROI espanse comprendevano anche dei voxel appartenenti ai polmoni.

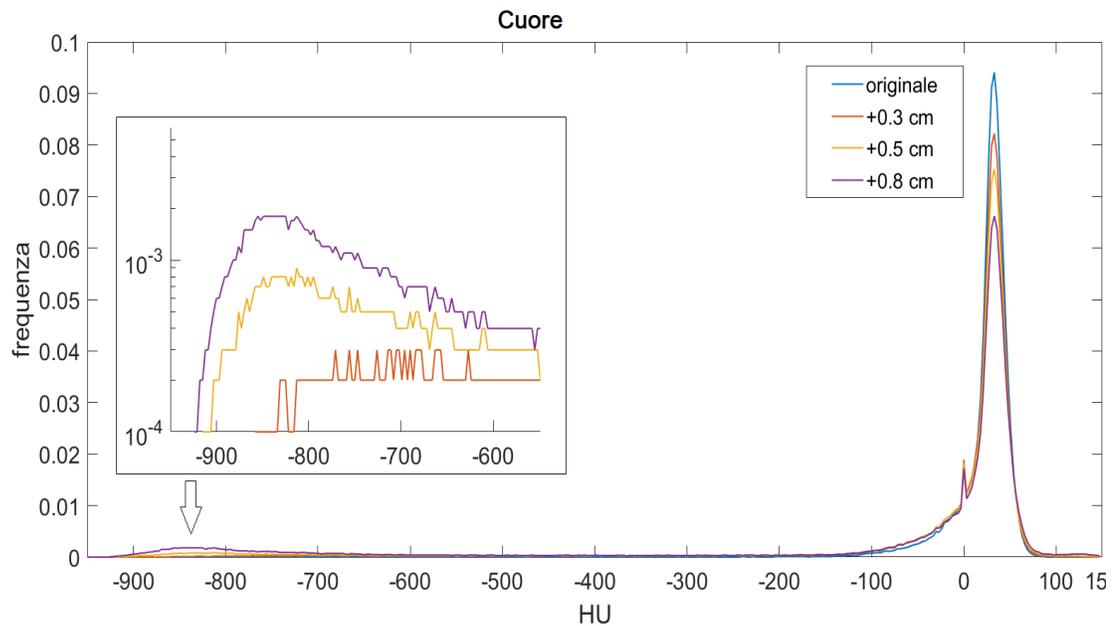


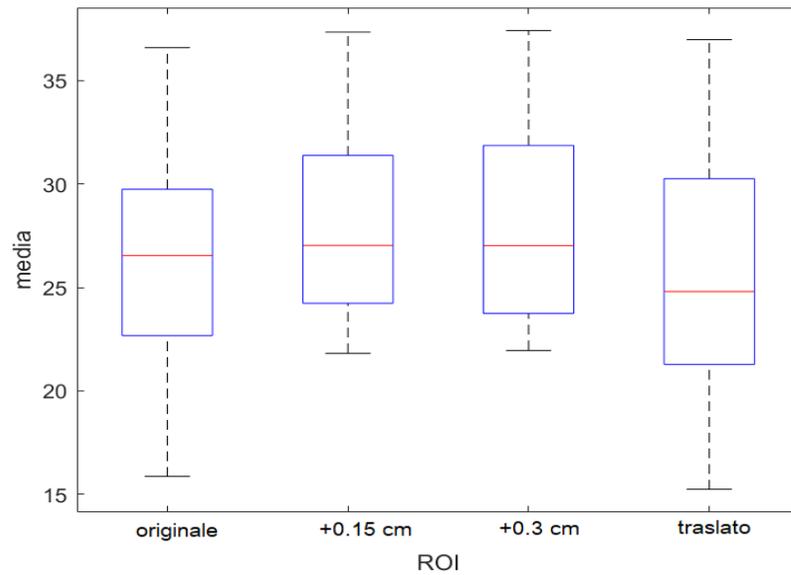
Figura 4.8. Confronto tra andamento della frequenza delle HU nella ROI originale del cuore e nelle ROI modificate. Si è scelto di rappresentare zona del grafico a HU minore in scala logaritmica al fine di mettere in evidenza la presenza di voxel esterni all'organo in esame nelle ROI modificate.

Nel caso della ROI del cuore, gli istogrammi risultano avere una sovrapposizione maggiore tra loro e, come atteso, i conteggi relativi ai voxel appartenenti ai polmoni inclusi nell'espansione sono più evidenti nelle ROI che sono state maggiormente espanse.

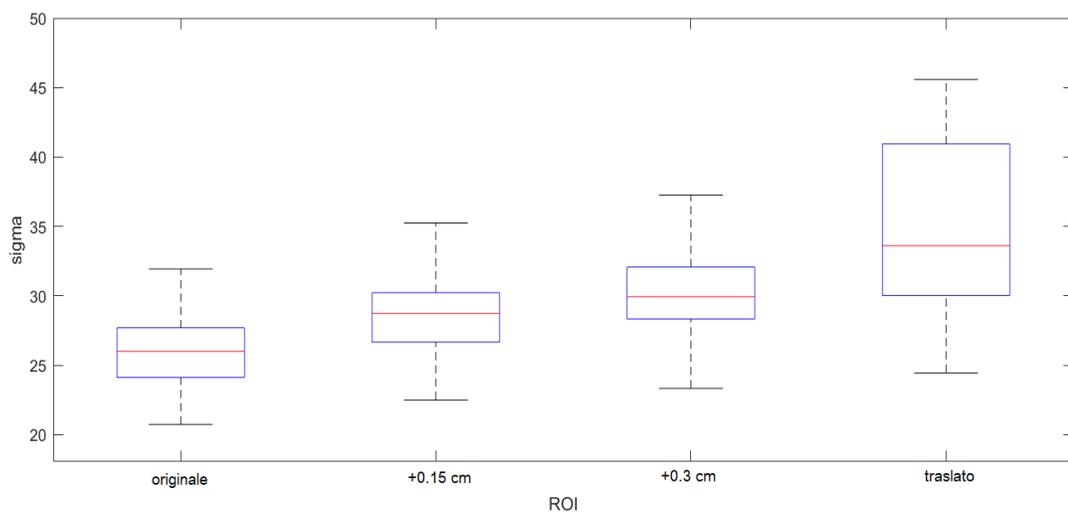
Per quantificare le caratteristiche di questi istogrammi si è deciso di modellarli mediante una curva gaussiana ( $f(x) = A \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ ), usando la funzione di fit implementata in Matlab. Per ogni istogramma è stato quindi estratto il valore massimo di frequenza e, per quanto riguarda il modello di gaussiana a esso associato, sono stati estratti  $\mu$ ,  $\sigma$  e il valore  $R^2$  associato alla bontà del fit. In seguito sono stati costruiti dei *box plot* per ognuna di queste 4 grandezze (riportati

in figura 4.9-4.10), al fine di valutare la loro utilità come criterio decisionale per il classificatore. Questi *box plot* sono stati costruiti analizzando l'andamento delle 4 grandezze per tutti i 42 pazienti disponibili.

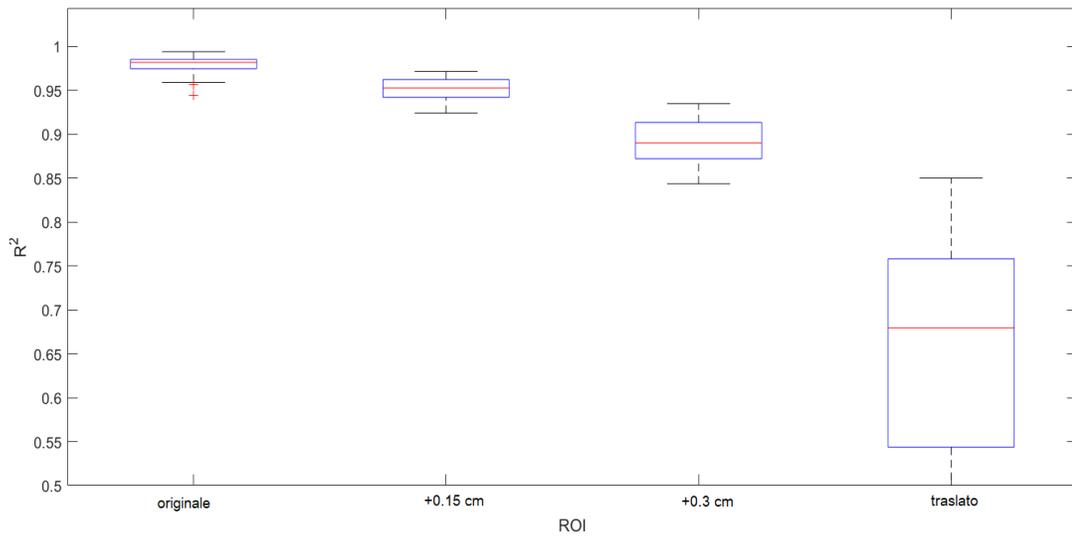
### Midollo



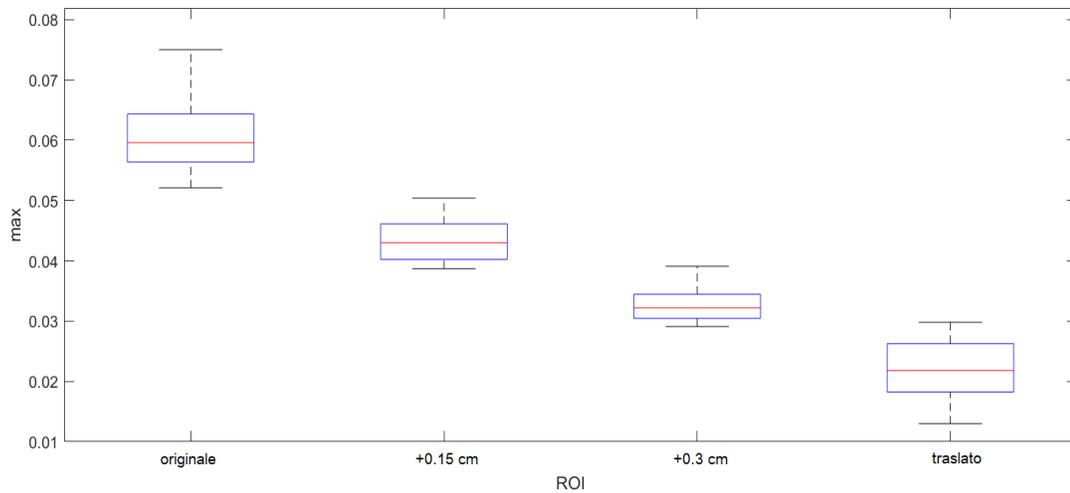
a.



b.



C.

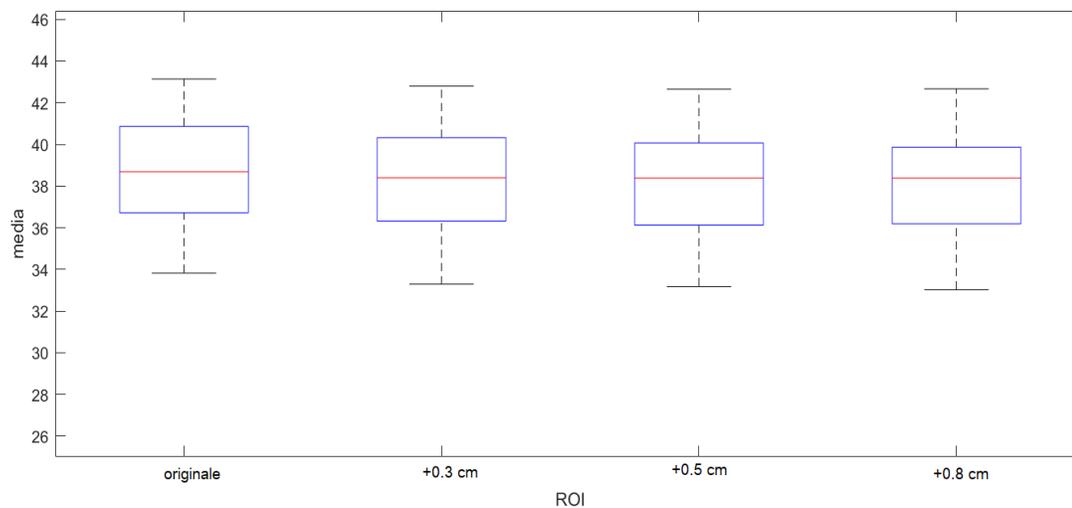


d.

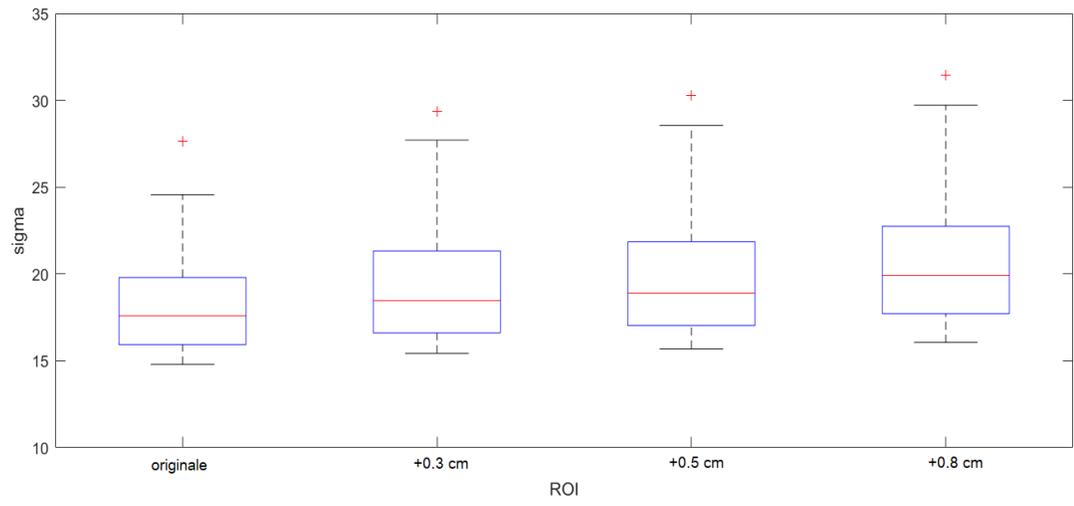
Figura 4.9. *Box plot* riferiti alla distribuzione dei vari parametri scelti nella ROI originale e in quelle modificate per quanto riguarda il midollo. I parametri di media,  $R^2$  e  $\sigma$  si riferiscono alla curva ottenuta eseguendo un fit gaussiano sull'istogramma della ROI, mentre il valore max corrisponde al valore di frequenza massima registrato nell'istogramma. I valori riportati sono riferiti ai 42 pazienti disponibili provenienti dal database PBI della mammella sinistra.

Per quanto riguarda il parametro della media  $\mu$  si ha che gli intervalli individuati dai *box plot* si sovrappongono per tutte le ROI prese in esame e non è quindi un buon criterio da usare nel classificatore. Per quanto riguarda i restanti parametri, in particolare quelli relativi al valore massimo di frequenza e a  $R^2$ , gli intervalli non sono sovrapposti ed è quindi ragionevole usarli come parametri nella decisione del classificatore. Il parametro  $\sigma$  presenta degli intervalli sovrapposti, ma nel caso delle ROI traslate del midollo risulta comunque un parametro utile alla loro individuazione: per questo motivo si è scelto di inserirlo ugualmente nel classificatore.

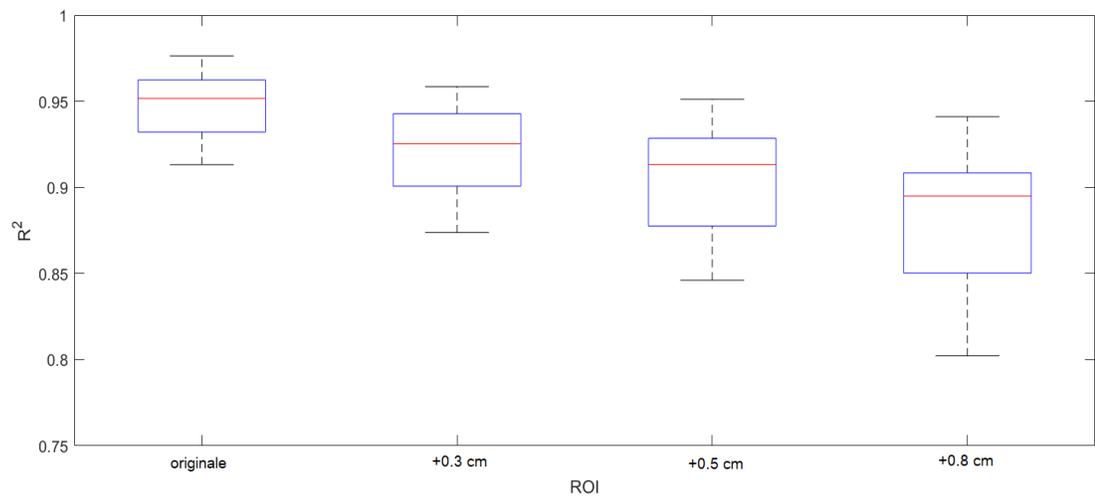
### Cuore



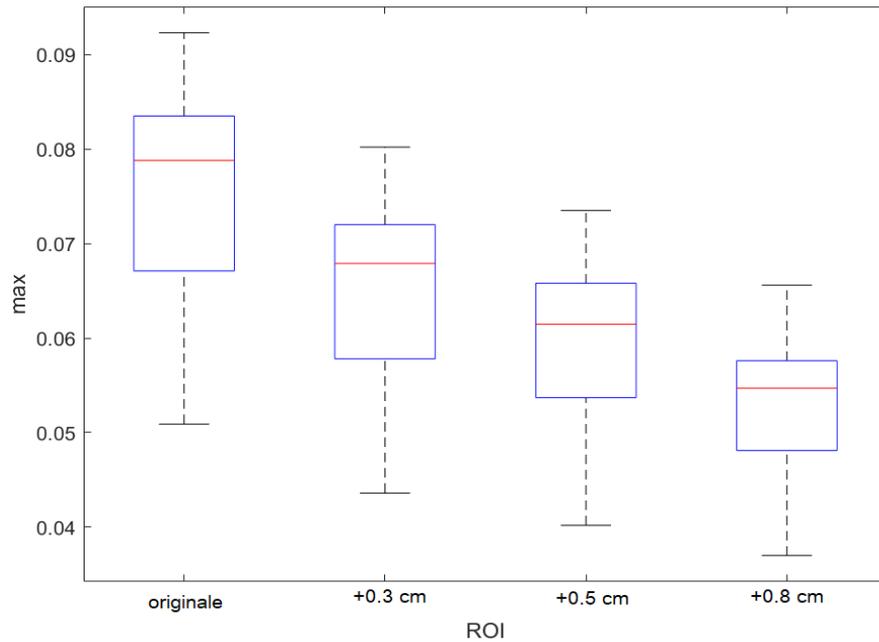
a.



b.



c.



d.

Figura 4.10. Distribuzione dei vari parametri scelti nella ROI originale e in quelle modificate per quanto riguarda il cuore. I parametri di media,  $R^2$  e  $\sigma$  si riferiscono alla curva ottenuta eseguendo un fit gaussiano sull'istogramma della ROI, mentre il valore max corrisponde al valore di frequenza massima registrato nell'istogramma. I valori riportati sono riferiti ai 42 pazienti disponibili provenienti dal database PBI relativo alla mammella sinistra.

Come atteso, dall'analisi dell'istogramma, anche in questo caso, la media  $\mu$  non si rivela un buon parametro di decisione perché gli intervalli dei *box plot* sono sovrapposti. La stessa cosa si può dire per il parametro  $\sigma$ , per il quale gli intervalli delle ROI si sovrappongono in maniera tale da non renderlo un parametro rilevante per la decisione. I parametri  $R^2$  e valore massimo della frequenza nell'istogramma sono invece buoni parametri decisionali, a causa degli intervalli poco sovrapposti delle grandezze. È possibile, inoltre, notare come la mediana di  $R^2$  del cuore sia minore di quella del midollo, evidenziando una maggiore somiglianza dell'istogramma di quest'ultimo al modello di curva gaussiana.

Sulla base dei risultati ottenuti sono stati quindi scelti i parametri di massima frequenza,  $R^2$  e  $\sigma$  per quanto riguarda il classificatore utilizzato per il midollo, mentre sono stati scelti i parametri di massima frequenza e  $R^2$  per il classificatore adibito al riconoscimento delle ROI del cuore.

Il classificatore che si andrà a costruire riceverà quindi in input i valori dei parametri scelti per quella determinata ROI, estratti dall'istogramma relativo alla ROI da classificare:

- se tali valori soddisfano la condizione  $p_{min} < p < p_{max}$  per tutti i parametri decisionali usati per quella ROI, questa viene classificata come corretta e inserita nell'atlante;
- se tali valori non soddisfano la suddetta condizione anche solo per uno dei parametri di decisione per la ROI in esame, questa viene classificata come errata e sottoposta al giudizio del medico.

Da queste considerazioni, risulta evidente come la definizione non solo di quali parametri usare nella decisione, ma anche dell'intervallo di accettabilità (cioè i valori di  $p_{min}$  e  $p_{max}$ ) degli stessi sia una fase essenziale nella costruzione del classificatore. Al fine di stabilire tali intervalli, il set dei 42 pazienti sottoposti a trattamento PBI alla mammella sinistra è stato diviso in due:

- 37 pazienti di addestramento, dai quali sono stati estratti i vettori contenenti i parametri di decisione adottati per la ROI in esame. Dalla distribuzione di tali parametri sono stati estratti poi i valori di percentile 25 e 75, usati, in prima analisi, come minimo e massimo dell'intervallo di accettabilità.
- 5 pazienti di validazione usati per modificare eventualmente i valori di percentile associati agli intervalli di accettabilità, al fine di ottenere una risposta del

classificatore volta più alla minimizzazione dei falsi positivi rispetto ai falsi negativi. La motivazione di questa scelta risiede nel *workflow* nel quale il classificatore è inserito, mostrato in figura 4.2.

Infine, per valutare le effettive performance del classificatore, è stato effettuato un test finale su 13 nuovi pazienti provenienti dal database del trattamento PBI relativo alla mammella destra.

### 4.3 Fase di training

Come già anticipato, in questa fase sono stati presi come input gli istogrammi relativi ai 37 pazienti del database PBI della mammella sinistra. Nella figura 4.11 è illustrato il procedimento seguito durante questa fase.

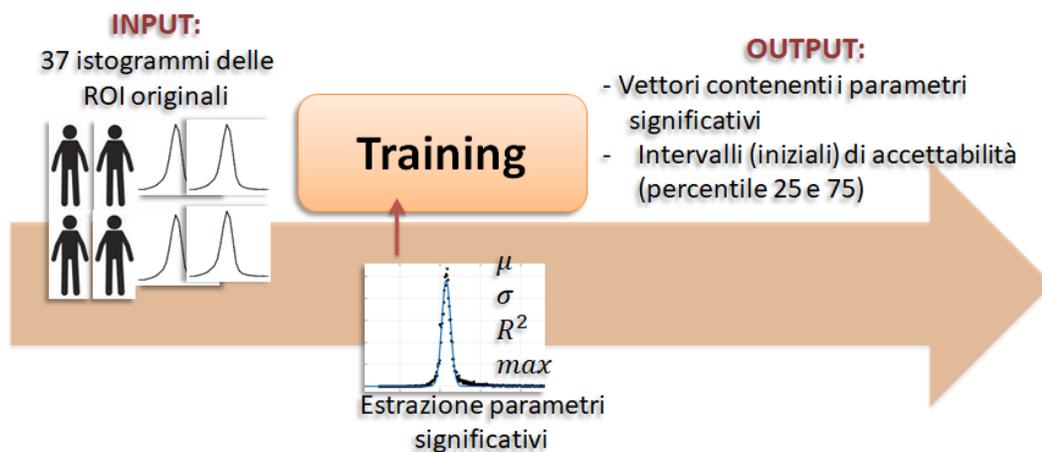


Figura 4.11. Procedimento seguito durante la fase di training del classificatore.

Per quanto riguarda la ROI del midollo sono stati estratti i seguenti parametri per ognuno dei 37 istogrammi riferiti alle ROI originali:

- massimo valore di frequenza relativo alle HU;
- valore di  $R^2$  relativo alla curva proveniente dal fit gaussiano eseguito sull'istogramma;

- valore di  $\sigma$  associato alla suddetta curva gaussiana.

Sono stati quindi ottenuti 3 vettori relativi a ogni parametro composti da 37 elementi provenienti dai pazienti presi in considerazione. Lo script che esegue questa prima fase di training e fornisce in output i suddetti vettori è riportato in appendice 6. Dalle distribuzioni di questi 3 parametri sono stati ricavati i percentili 25 e 75, che sono stati inizialmente usati come valori di  $p_{\min}$  e  $p_{\max}$  nella condizione di decisione del classificatore.

Per quanto riguarda la ROI del cuore sono stati estratti i seguenti parametri per ognuno dei 37 istogrammi riferiti alle ROI originali:

- massimo valore di frequenza relativo alle HU;
- valore di  $R^2$  relativo alla curva proveniente dal fit gaussiano eseguito sull'istogramma.

Anche in questo caso sono stati quindi ottenuti 3 vettori relativi a ogni parametro, composti da 37 elementi provenienti dai pazienti considerati. I valori di  $p_{\min}$  e  $p_{\max}$  sono stati impostati in maniera analoga al caso del midollo.

## 4.4 Fase di validazione



Figura 4.12. Procedimento seguito durante la fase di validazione del classificatore.

Come mostrato nella figura 4.12, durante la fase di validazione sono stati utilizzati i 5 pazienti rimanenti dal database PBI relativo alla mammella sinistra al fine di definire gli intervalli di accettabilità definitivi per i parametri utilizzati per ogni ROI.

Per questi pazienti sono stati estratti 4 diversi istogrammi:

- 1 istogramma relativo alla ROI originale;
- 3 istogrammi relativi alle ROI modificate.

In totale sono stati quindi analizzati e classificati, usando gli intervalli di accettabilità definiti durante la fase di training, 20 istogrammi. Di questi, utilizzando un approccio supervisionato, 5 erano relativi a ROI che dovevano essere classificate come corrette e 15 erano istogrammi di ROI che dovevano essere classificate come errate. Il classificatore prende quindi in input i parametri associati a una collezione di istogrammi, i vettori contenenti i parametri di riferimento provenienti dalla fase di training e i valori di percentile da usare nella classificazione. Inoltre, fornisce in output un vettore avente un numero di righe pari al numero di istogrammi di input e un numero di colonne corrispondente ai

parametri utilizzati, più una finale contenente il risultato della decisione del classificatore. Tale valore proviene dalla risoluzione dell'espressione logica 4.1, nella quale sono presenti tutte le condizioni relative ai parametri significativi per la ROI in questione. Questo vettore di output presenta un valore 1 nel suo elemento  $v_{ij}$  quando l'istogramma  $i$  ha soddisfatto la sua corrispondente condizione  $j$ , in caso contrario presenta un valore 0. Lo script che esegue la classificazione è riportato in appendice 7.

Con riferimento alla tabella di contingenza mostrata in figura 4.1, le situazioni che si sono verificate sono le seguenti:

- Iistogramma relativo a una ROI modificata → output positivo: la ROI è stata classificata in maniera errata ed è quindi necessaria una riduzione degli intervalli di accettabilità (caso di Falso Positivo);
- Iistogramma relativo a una ROI modificata → output negativo: la ROI è stata classificata correttamente e non vengono quindi modificati gli estremi degli intervalli di accettabilità (caso di Vero Negativo);
- Iistogramma relativo a una ROI originale → output positivo: la ROI è stata classificata correttamente e non vengono quindi modificati gli estremi degli intervalli di accettabilità (caso di Vero Positivo);
- Iistogramma relativo a una ROI originale → output negativo: la ROI è stata classificata in maniera errata ed è quindi necessaria una espansione degli intervalli di accettabilità (caso di Falso Negativo).

Le modifiche agli intervalli di accettabilità sono state apportate variando manualmente i valori di percentile forniti al classificatore e avendo sempre come priorità quella di ottenere un numero nullo di falsi positivi. Come detto in precedenza, l'output fornito dal classificatore contiene le informazioni su quale

parametro ha dato una risposta diversa da quella attesa: grazie a ciò, sono stati modificati ogni volta solo gli intervalli relativi ai parametri che avevano causato una classificazione errata.

### Midollo

Le condizioni per la decisione nel caso del midollo sono quindi:

- $C_1: \sigma_{min} < \sigma_{test} < \sigma_{max}$
- $C_2: R^2_{min} < R^2_{test}$
- $C_3: Max_{min} < Max_{test} < Max_{max}$

I valori minimi e massimi sono definiti per ogni ROI dal percentile impostato nella fase di valutazione. Nel caso del midollo i valori trovati sono quelli riportati in Tabella 4.1.

| P        | P <sub>min</sub> (percentile) | P <sub>max</sub> (percentile) |
|----------|-------------------------------|-------------------------------|
| $\sigma$ | 23 HU (10)                    | 30 HU (85)                    |
| $R^2$    | 0,97 (25)                     | -                             |
| max      | 0,050 (10)                    | 0,075 (95)                    |

Tabella 4.1. Valori degli intervalli di accettabilità per la ROI del midollo stabiliti in seguito alla fase di validazione.

Con questi valori, tutte le 15 ROI di validazione modificate vengono individuate come errate, mentre 4 su 5 ROI originali vengono individuate come aggiungibili immediatamente all'atlante.

### Cuore

Alla luce dei risultati presentati nella sezione precedente, le condizioni per la decisione nel caso del cuore sono quindi:

- $C_1: R^2_{min} < R^2_{test}$
- $C_2: Max_{min} < Max_{test} < Max_{max}$

I valori minimi e massimi sono definiti per ogni ROI dal percentile impostato nella fase di validazione. Nel caso del midollo i valori trovati sono riportati in Tabella 4.2.

| P              | P <sub>min</sub> (percentile) | P <sub>max</sub> (percentile) |
|----------------|-------------------------------|-------------------------------|
| R <sup>2</sup> | 0,95 (35)                     | -                             |
| max            | 0,075 (35)                    | 0,090 (95)                    |

Tabella 4.2. Valori degli intervalli di accettabilità per la ROI del cuore stabiliti in seguito alla fase di validazione.

Con questi valori, tutte le 15 ROI di validazione modificate vengono individuate come errate, mentre 3 su 5 ROI originali vengono individuate come aggiungibili immediatamente all'atlante.

## 4.5 Fase di test

In questa ultima fase, sono state valutate le performance del classificatore sui 13 nuovi pazienti provenienti dal database PBI riferito alla mammella destra, come illustrato nella figura 4.13.

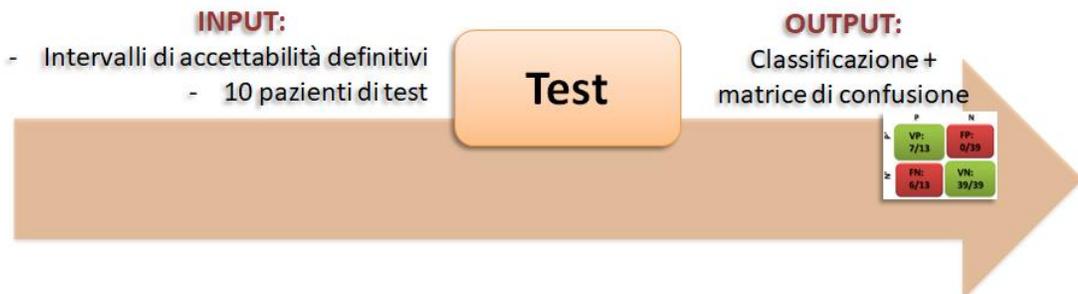


Figura 4.13. Procedimento seguito durante la fase di test del classificatore.

Anche per questi 13 pazienti sono stati estratti l'istogramma originale e i 3 relativi alle ROI modificate come illustrato nella fase di validazione, per un totale di 52 istogrammi da classificare. Gli intervalli di accettabilità sono, questa volta, quelli riportati nelle tabelle 4.1 e 4.2.

## Midollo

Per quanto riguarda il midollo, 45 ROI sono state classificate come errate e 7 sono state classificate come aggiungibili immediatamente all'atlante. E' stata costruita una matrice di confusione per valutare i risultati del classificatore. Il rate dei Veri Positivi (VP) è stato di 7/13, quello dei Falsi Positivi (FP) di 0/39, quello dei Falsi Negativi (FN) di 6/13 e quello dei Veri Negativi (VN) di 39/39.

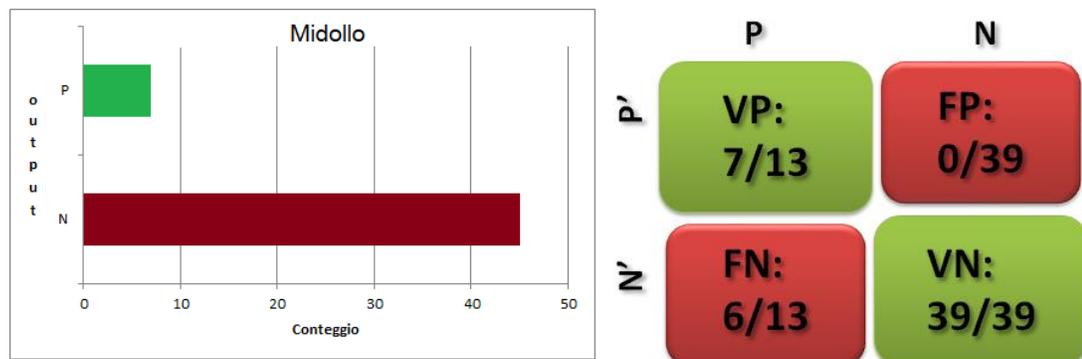


Figura 4.14. Risultati e performance del classificatore per quanto riguarda la ROI del midollo.

## Cuore

Per quanto riguarda il cuore, 44 ROI sono state classificate come errate e 8 come aggiungibili immediatamente all'atlante. E' stata costruita una matrice di confusione per valutare i risultati del classificatore. Il rate dei Veri Positivi (VP) è stato di 6/13, quello dei Falsi Positivi (FP) di 2/39, quello dei Falsi Negativi (FN) di 7/13 e quello dei Veri Negativi (VN) di 37/39.

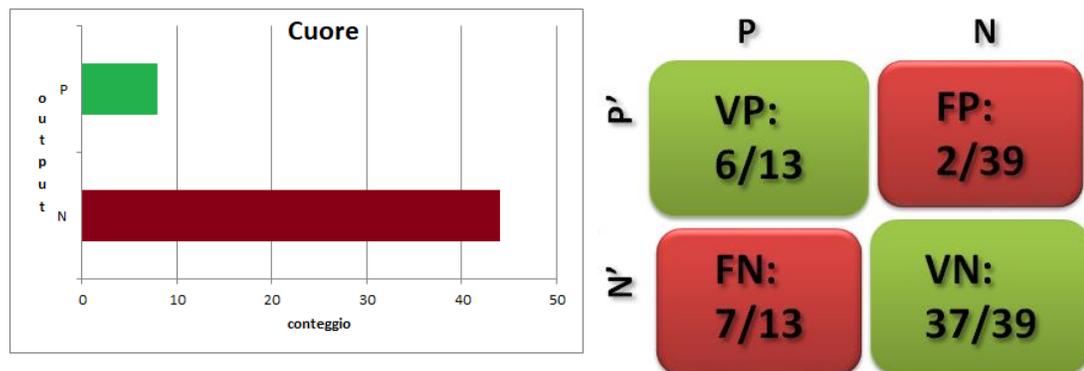


Figura 4.15. Risultati e performance del classificatore per quanto riguarda la ROI del cuore.

A seguito della fase di test per la ROI del cuore, sono stati analizzati gli istogrammi appartenenti alle due ROI classificate come FP. I due campioni in questione sono ROI modificate provenienti dalla stessa ROI originale, aventi come espansione 0.3 cm e 0.5 cm. Dall'analisi degli istogrammi delle ROI coinvolte è risultato che la ROI originale è stata anch'essa mal classificata come errata, in quanto avente un valore di massimo della frequenza maggiore dell'estremo superiore dell'intervallo di accettabilità. Questa condizione nell'istogramma di partenza ha portato a uno spostamento dei valori di massimo anche delle ROI modificate, portando tali valori a rientrare nell'intervallo di accettabilità stabilito per quanto riguarda il parametro del massimo.

#### 4.6 Discussione dei risultati e limiti di validità del classificatore

Il classificatore sopra descritto, alla luce dei risultati della sezione precedente, risulta in grado di individuare solo dei casi di sovra-segmentazione nel caso delle ROI del cuore e del midollo. Questo è dovuto al fatto che l'istogramma, per sua definizione, è in grado di individuare molto bene la presenza di voxel estranei alla struttura in esame, mentre non sono date informazioni sulla loro collocazione

spaziale. Una eventuale sotto-segmentazione di cuore e midollo potrebbe comunque essere individuata tramite istogramma, ma in questo caso l'unica grandezza utile sarebbe l'altezza massima di tale curva, che, come unico parametro di decisione, renderebbe il classificatore molto debole. Questo è dovuto al fatto che all'interno di queste due ROI sono presenti voxel aventi un limitato intervallo di HU.

Inoltre, il classificatore risulta efficace solo se la sovra-segmentazione produce ROI modificate aventi un alto gradiente di HU: ciò è possibile nel caso di cuore-polmone e nel caso di midollo-osso. Per la struttura della mammella, invece, un approccio del genere non risulterebbe performante in quanto non sono presenti attorno a essa zone aventi valori di HU significativamente differenti.

Per quanto riguarda la ROI del cuore, la valutazione corretta dei VP diminuirebbe il numero di ROI da valutare per il medico di circa il 46%, in base ai risultati ottenuti nella fase di test. D'altra parte, però, è risultato che le due condizioni di decisione non sono sufficienti a garantire l'assenza di Falsi Positivi nelle ROI accettate dal classificatore ed è quindi necessaria l'introduzione di un eventuale parametro ulteriore nella decisione. Dall'analisi delle segmentazioni automatiche generate da RayStation, si è notato che un altro errore molto frequente è quello di una sotto-segmentazione nella zona superiore del cuore lungo l'asse longitudinale. La regione interessata è però caratterizzata da un basso gradiente: un'analisi solo attraverso istogrammi che non tiene conto della forma della struttura in esame non è performante proprio per questo motivo.

Per quanto riguarda la ROI del midollo, invece, non è stata riscontrata la presenza di FP, mentre la valutazione corretta dei VP diminuirebbe le ROI da esaminare da parte del medico di circa il 54%, in base ai risultati ottenuti in fase di test.

## 4.7 Sviluppi futuri

Il classificatore descritto nella sezione precedente, in grado di individuare segmentazioni errate solamente in occasione di una eventuale sovra-segmentazione della struttura di interesse, alla luce di quanto detto sopra, potrebbe essere migliorato nei seguenti modi:

- 1) Aumento del campione di addestramento dal quale vengono estratte le soglie dei parametri di decisione;
- 2) Utilizzo di un modello di curva più complesso della gaussiana singola, in modo da riuscire a modellizzare più correttamente altri istogrammi, come ad esempio quello della mammella;
- 3) Automazione maggiore del classificatore, ad esempio nella fase di validazione e nella fase di test delle performance;
- 4) Integrazione sia dello script di controllo delle ROI, sia dello script di addestramento del classificatore nella piattaforma di RayStation, con eventuale possibilità di modifica dei parametri attraverso GUI;
- 5) Aggiunta di maggiori valori di espansione alle ROI modificate, anche non uniformi;
- 6) Utilizzo di algoritmi di machine learning nella classificazione mediante istogrammi;
- 7) Utilizzo di modelli biomeccanici integrati al classificatore proposto per effettuare un controllo non solo sulle intensità delle ROI prese in esame, ma anche sulla forma delle strutture contornate per mettere in evidenza eventuali anomalie di forma rispetto al campione medio descritto dal modello.

# Conclusioni

Un tema molto importante nella ricerca in ambito radioterapico è lo sviluppo e l'integrazione di algoritmi di segmentazione automatica nella pratica clinica. Ciò diminuirebbe il carico di lavoro del medico, che potrebbe dedicare più tempo alla delineazione dei target o ad altri compiti necessari durante il trattamento dei pazienti. Inoltre, mediante l'uso di algoritmi automatici, si ridurrebbero altre problematiche legate alla segmentazione manuale, come ad esempio la variabilità di contornazione tra diversi utenti.

Al fine di poter integrare tali algoritmi nella pratica clinica è necessaria una loro valutazione attraverso indici di qualità aventi come *gold standard* i contorni effettuati dal medico. Gli indici di qualità utilizzati in questo lavoro di tesi sono stati l'indice Dice (DSC) e la distanza di Hausdorff media (AHD).

Sono stati quindi sviluppati degli script che automatizzassero la costruzione dell'atlante, la generazione delle segmentazioni automatiche mediante i diversi algoritmi di segmentazione e il calcolo degli indici mediante i quali sono stati condotti i confronti tra i diversi algoritmi. L'utilizzo di tali script nella pratica clinica permette quindi un risparmio di tempo, in quanto la segmentazione delle strutture di interesse è eseguita in maniera automatica su tutti i pazienti dove è necessaria, senza dover ricorrere all'uso di GUI che permette l'esecuzione dell'algoritmo solo sul paziente corrente.

In primo luogo sono state confrontate segmentazioni provenienti da algoritmi aventi alla base una diversa applicazione della conoscenza a priori sulle strutture da segmentare: l'algoritmo MBS, proprio del TPS RayStation, e l'algoritmo di segmentazione tramite atlante anatomico, costruito usando 42 pazienti sottoposti a trattamento PBI alla mammella sinistra del Policlinico di Modena. Sono state

valutate le segmentazioni di midollo, polmone destro e polmone sinistro. Dai risultati ottenuti, è emerso che, per quanto riguarda il midollo, non è possibile individuare un algoritmo di segmentazione migliore dell'altro, in quanto le differenze tra i due set di indici ottenuti non sono significative. Per quanto riguarda le ROI di polmone destro e sinistro si è evidenziato che la segmentazione mediante atlante forniva risultati significativamente migliori di quella mediante MBS.

In seguito, sono stati confrontati due metodi di utilizzo dell'algoritmo di segmentazione mediante atlante. Al primo metodo sono fornite in input tutte le strutture da segmentare in una sola sessione (metodo 1). Il secondo metodo prevedeva l'esecuzione dell'algoritmo in modo sequenziale, considerando una struttura per volta (metodo 2). Sono state inoltre valutate le performance all'aumentare del numero di pazienti inseriti nell'atlante. Dai risultati ottenuti è emerso che non è possibile individuare una differenza significativa tra i due metodi basandosi solo sugli indici DSC e AHD. Per questo si è deciso di valutare il tempo di esecuzione dei due metodi ed è risultato che il metodo 1 richiedeva minor tempo di esecuzione (valore mediano per 42 pazienti per il metodo 1: 280 s, valore mediano per 42 pazienti per il metodo 2: 1580 s). Per questo motivo, gli studi sono stati condotti utilizzando il metodo 1. Inoltre, le segmentazioni prodotte da questo metodo sono soddisfacenti per le ROI di cuore, midollo, mammella controlaterale, polmone destro e polmone sinistro, mentre non producono risultati soddisfacenti (DSC < 0.6) per la ROI della tiroide. Per gli organi che hanno presentato risultati soddisfacenti, le correzioni che il medico deve apportare alle segmentazioni automatiche sono minime, il che permette un guadagno rilevante nelle tempistiche del processo di segmentazione. Le prospettive future per questo settore di ricerca sono aumentare il database di pazienti nell'atlante e lo sviluppo e l'implementazione di sotto-atlanti contenenti pazienti omogenei per caratteristiche (costituzione fisica, età, trattamenti chirurgici subiti, volume delle

ROI di interesse...) che renderebbero la performance di tale algoritmo di segmentazione ancora migliore.

In linea con la volontà di espandere il numero di pazienti e quindi di strutture contenute nell'atlante anatomico, è stato ricercato un metodo di valutazione delle segmentazioni che richiedesse il minor supporto possibile da parte del medico. Sono state quindi gettate le basi per la costruzione di un classificatore in grado di individuare una segmentazione errata da una corretta e ne sono state valutate le performance. I parametri di decisione del classificatore sono stati impostati avendo come priorità l'assenza di Falsi Positivi nell'output del classificatore, anche a costo di avere dei Falsi Negativi in più. Tali parametri sono stati estratti facendo uso degli istogrammi relativi ai voxel contenuti nei volumi delle ROI di interesse. Il classificatore è stato testato sulle ROI di midollo e cuore. I pazienti utilizzati sono i 42 del database PBI riferito alla mammella sinistra, già considerati nelle sezioni precedenti, dei quali 37 sono stati usati per il training e 5 per la validazione. Per il test sulle performance del classificatore sono stati presi in considerazione 13 pazienti provenienti dal database relativo ai pazienti sottoposti a PBI sulla mammella destra. Per i pazienti di validazione e test, al fine di simulare il caso di ROI non corretta, sono state modificate le ROI originali espandendole uniformemente e, nel caso del midollo, traslando la ROI lungo il piano trasversale. Sono stati quindi esaminati 20 istogrammi (5 corretti e 15 errati) nella fase di validazione e 52 istogrammi nella fase di test. In questa fase, per il caso del midollo, il classificatore non ha riportato nessuna istanza di Falso Positivo e, con un rate di 7/13 Veri Positivi, ha diminuito di circa il 54% la quantità di ROI da analizzare da parte del medico. Per il caso del cuore, nonostante un rate di Veri Positivi di 6/13, corrispondente a una diminuzione di circa il 46% della quantità di ROI da analizzare, è stata riscontrata la presenza di 2/39 Falsi Positivi. Questo risultato ha indicato quindi la necessità dell'aggiunta di un eventuale parametro ulteriore per il

classificatore riferito al cuore. Il classificatore proposto in questo lavoro di tesi può essere migliorato in futuro aumentando la statistica di addestramento e validazione. Inoltre, si potrebbe supportare la decisione mediante meccanismi di classificazione di *machine learning*.

# Bibliografia

[1] V.Grégoire, T.R.Mackie. *State of the art on dose prescription, reporting and recording in Intensity-Modulated Radiation Therapy (ICRU report No. 83)*. Cancer/Radiothérapie, 15(6): 555-559, 2011.

[2] Rafael C. Gonzalez, Richard E. Woods. *Digital Image Processing*, Pearson, 2007.

[3] Gary V. Walker, Musaddiq Awan, Randa Tao, Eugene J. Koay, Nicholas S. Boehling, Jonathan D. Grant, Dean F. Sittig, Gary Brandon Gunn, Adam S. Garden, Jack Phan, William H. Morrison, David I. Rosenthal, Abdallah Sherif Radwan Mohamed, Clifton David Fuller. *Prospective randomized double-blind study of atlas-based organ-at-risk autosegmentation-assisted radiation planning in head and neck cancer*, Radiotherapy and Oncology 112 (321-325), 2014.

[4] Whitfield G.A., Price G.J., Moore C.J., *Automated delineation of radiotherapy volumes: are we going in the right direction?*. Br J Radiol, 2013.

[5] Commowick O., Grégoire V., Malandain G., *Atlas-based delineation of lymph node levels in head and neck computed tomography images*, Radiotherapy and Oncology 87 (281-289), 2008.

[6] Ciardo D, Gerardi MA, Vigorito S, Morra A, Dell'acqua V, Diaz FJ, Cattani F, Zaffino P, Ricotti R, Spadea MF, Riboldi M, Orecchia R, Baroni G, Leonardi MC, Jereczek-Fossa BA<sup>8</sup>. *Atlas-based segmentation in breast cancer radiotherapy: Evaluation of specific and generic-purpose atlases*, The Breast 32 (44-52), 2017.

[7] Sims R, Isambert A, Grégoire V, Bidault F, Fresco L, Sage J, Mills J, Bourhis J, Lefkopoulos D, Commowick O, Benkebil M, Malandain G. *A pre-clinical assessment of an atlas-based automatic segmentation tool for the head and neck.*, Radiotherapy and Oncology 93 (474-478), 2009.

[8] Eldesoky AR, Yates ES, Nyeng TB, Thomsen MS, Nielsen HM, Poortmans P, Kirkove C, Krause M, Kamby C, Mjaaland I, Blix ES, Jensen I, Berg M, Lorenzen EL, Taheri-Kadkhoda Z, Offersen BV. *Internal and external validation of an ESTRO delineation guideline - dependent automated segmentation tool for loco-regional radiation therapy of early breast cancer*. Radiotherapy and Oncology 121 (424-430), 2016.

- [9] Maffei N, Guidi G, D'Angelo E, Meduri B, Lohr F, Costi T. *Can atlas-based automatic segmentation contour H&N ROIs like a physician?*, RayStation User Meeting italiano, 2016.
- [10] Nicolas Aspert, Diego Santa-Cruz, Touradj Ebrahimi. *MESH : MEASURING ERRORS BETWEEN SURFACES USING THE HAUSDORFF DISTANCE*, IEEE International Conference in Multimedia and Expo, 2002.
- [11] RaySearch Laboratories, *RayStation 6 User Manual*, 2016.
- [12] RaySearch Laboratories, *RayStation 6 Reference Manual*, 2016.
- [13] RaySearch Laboratories, *RayStation 6 Scripting Guidelines*, 2016.
- [14] RaySearch Laboratories, *Deformable registration in RayStation*, 2017
- [15] Weistrand O, Svensson S. *The ANACONDA algorithm for deformable image registration in radiotherapy*, Medical Physics 42, 2015.
- [16] Dean J, Welsh L, McQuaid D, Wong K et al., *Assessment of fully-automated atlas-based segmentation of novel oral mucosal surface organ-at-risk*, (in press), Radiotherapy and Oncology.
- [17] Zhang T, Chi Y, Meldolesi E, *Automatic delineation of on-line Head-and-Neck computed tomography images: toward on-line adaptive radiotherapy*, Radiotherapy and Oncology 68 (522-530), 2007.
- [18] Iglesias J, Sabuncu M, *Multi-atlas segmentation of biomedical images: a survey*, Medical Image Analysis 24 (205-219), 2015.
- [19] Zhou R, Liao Z, Pan T, Milgrom SA, Pinnix CC, Shi A, Tang L, Yang J, Liu Y, Gomez D, Nguyen QN, Dabaja BS, Court L, Yang J. *Cardiac atlas development and validation for automatic segmentation of cardiac substructures*. Radiotherapy and Oncology 122 (66-71), 2017.
- [20] Asman AJ, Huo Y, Plassard AJ, Landman BA. *Multi-atlas learner fusion: An efficient segmentation approach for large-scale data*. Medical Image Analysis 26 (82-91), 2015.
- [21] Ahmed R. Eldesoky, Giulio Francolini, Mette S. Thomsen, Esben S. Yates, Tine B. Nyeng, Carine Kirkove, Claus Kamby, Egil S. Blix, Mette H. Nielsen, Zahra Taheri-Kadkhoda, Martin Berg, Birgitte V. Offersen, *Dosimetric assessment of an Atlas based automated segmentation for loco-regional radiation therapy of early breast cancer in the Skagen Trial 1: A multi-institutional study*, Clinical and Transational Radiation Oncology 2 (36-40), 2017.
- [22] Lachner R. (2008) *Rigid Registration of Medical Images by Maximization of Mutual Information*. In: Breitner M.H., Denk G., Rentrop P. (eds) From Nano to Space. Springer, Berlin, Heidelberg.

[23] Karasawa K, Oda M, Kitasaka T, Misawa K, Fujiwara M, Chu C, Zheng G, Rueckert D, Mori K, *Multi-atlas pancreas segmentation: Atlas selection based on vessel structure*, Medical Image Analysis 29 (18-28), 2017.

[24] [http://www.dkfz.de/en/medphys/Therapy\\_planning\\_development/Projects/Registration/Rigid\\_Registration.html](http://www.dkfz.de/en/medphys/Therapy_planning_development/Projects/Registration/Rigid_Registration.html)

# Ringraziamenti

A seguito della conclusione di questo lavoro di tesi, vorrei spendere delle parole di ringraziamento per tutte le persone che mi hanno sostenuto e aiutato durante questo periodo.

Ringrazio la professoressa Maria Pia Morigi, la quale, anche durante il mio percorso di tesi triennale, oltre ad avermi dato l'opportunità di svolgere questo lavoro di tesi, è stata sempre disponibile a darmi suggerimenti e consigli.

Ringrazio il direttore del dipartimento di Fisica Medica Tiziana Costi e il direttore del dipartimento di Radioterapia Oncologica Frank Lohr per avermi dato la disponibilità a svolgere questa tesi.

Ringrazio il dott. Gabriele Guidi e il dott. Nicola Maffei per l'aiuto, l'incoraggiamento, le correzioni e i suggerimenti durante il mio periodo di tesi. Vorrei ringraziare inoltre il personale della Fisica Medica per avermi accolto durante questi mesi e a tutti i medici professionisti del reparto di Radioterapia Oncologica dell'Azienda Ospedaliero-Universitaria di Modena che hanno indirettamente contribuito allo svolgimento di questo lavoro.

Ringrazio i miei genitori per il loro supporto economico, ma soprattutto morale, durante tutto il mio percorso di studi e per la loro disponibilità ad ascoltare, consigliarmi e supportarmi in tutte le mie scelte, accademiche e non, anche quelle più difficili.

Infine, ma chiaramente non per importanza, i miei ringraziamenti vanno a tutti i miei amici, sia coloro con cui ho avuto l'occasione di condividere i miei studi triennali e magistrali, sia quelli con cui ho condiviso interessi ed esperienze. Attraverso i vostri incoraggiamenti costanti e la vostra vicinanza emotiva sempre presente, siete in ogni occasione la mia colonna insostituibile di supporto.

# Appendice

## Appendice 1. Script di preparazione pre-inserimento nell'atlante

```
from connect import *

case=get_current("Case") #Case è l'oggetto dalla quale si possono modificare le cose tipo le ROI
patient=get_current("Patient") #patient è un ogetto che ha come attributi patientID, patientName, ecc

possibili_nomi_PolmoneDx=['PolmoneDx','Polmone dx','polmone dx','polmonedx','polm dx','polmdx']
possibili_nomi_PolmoneSx=['PolmoneSx','Polmone sx','polmone sx','polmonesx','polmone sn','polm sn','polmsn','polm sx','polmsx','polmone sin','polmonesn','polmonesin']
possibili_nomi_MammellaSx=['MammellaSx','Mammella sin','Mammella sx','mammella sx','mammella sin','mammellasx','mamm sx','mammsn','mamm sn','mammsx']
possibili_nomi_MammellaDx=['MammellaDx','mammella dx','mammelladx','mammella d','mamm dx','mammdx']
possibili_nomi_body=['Body','body','BODY']
possibili_nomi_PTV_PBI=['PTV_PBI','PTV-PBI','PTVPBI','PTV PBI','PTV']
possibili_nomi_PTV_WBI=['PTV_WBI','PTV-WBI','PTV1WBI','PTV WBI','PTV1 WBI','PTVWBI','PTV1']
possibili_nomi_CTV_WBI=['CTV_WBI','CTV-WBI','CTV1WBI','CTV WBI','CTV1 WBI','CTVWBI']
possibili_nomi_CTV_PBI=['CTV_PBI','CTV-PBI','CTV PBI','CTVPBI','CTV1 PBI','CTV1PBI','CTV1 PBI']
possibili_nomi_PTV_EVAL=['PTV_EVAL','PTV-EVAL','PTV-eval','PTV EVAL','PTVEVAL','PTVeval','PTV eval']

for r in case.PatientModel.RegionsOfInterest:
    for nome in possibili_nomi_PolmoneDx:
        if r.Name==nome:
            r.Name='PolmoneDx'
            break
        else: continue
for r in case.PatientModel.RegionsOfInterest:
    for nome in possibili_nomi_PolmoneSx:
        if r.Name==nome:
            r.Name='PolmoneSx'
            break
        else: continue
for r in case.PatientModel.RegionsOfInterest:
    for nome in possibili_nomi_MammellaSx:
        if r.Name==nome:
            r.Name='MammellaSx'
            break
        else: continue
```

```

for r in case.PatientModel.RegionsOfInterest:
    for nome in possibili_nomi_MammellaDx:
        if r.Name==nome:
            r.Name='MammellaDx'
            break
        else: continue
for r in case.PatientModel.RegionsOfInterest:
    for nome in possibili_nomi_body:
        if r.Name==nome:
            r.Name='Body'
            break
        else: continue
for r in case.PatientModel.RegionsOfInterest:
    if r.Name=='cuore':
        r.Name='Cuore'
        break
    else: continue
for r in case.PatientModel.RegionsOfInterest:
    if r.Name=='tiroide':
        r.Name='Tiroide'
        break
    else: continue
for r in case.PatientModel.RegionsOfInterest:
    if r.Name=='midollo':
        r.Name='Midollo'
        break
    else: continue
for r in case.PatientModel.RegionsOfInterest:
    for nome in possibili_nomi_PTV_PBI:
        if r.Name==nome:
            r.Name='PTV_PBI'
            break
        else: continue
for r in case.PatientModel.RegionsOfInterest:
    for nome in possibili_nomi_PTV_EVAL:
        if r.Name==nome:
            r.Name='PTV_EVAL'
            break
        else: continue
for r in case.PatientModel.RegionsOfInterest:
    for nome in possibili_nomi_PTV_WBI:
        if r.Name==nome:
            r.Name='PTV_WBI'
            break
        else: continue
for r in case.PatientModel.RegionsOfInterest:
    for nome in possibili_nomi_CTV_WBI:
        if r.Name==nome:
            r.Name='CTV_WBI'
            break
        else: continue
for r in case.PatientModel.RegionsOfInterest:
    for nome in possibili_nomi_CTV_PBI:
        if r.Name==nome:
            r.Name='CTV_PBI'
            break
        else: continue
for r in case.PatientModel.RegionsOfInterest:
    if r.Name=="PolmoneDx":
        case.PatientModel.RegionsOfInterest['PolmoneDx'].OrganData.R
            esponseFunctionTissueName="Lung"
    if r.Name=="PolmoneSx":
        case.PatientModel.RegionsOfInterest['PolmoneSx'].OrganData.R
            esponseFunctionTissueName="Lung"
    if r.Name=="MammellaSx":
        case.PatientModel.RegionsOfInterest['MammellaSx'].OrganData.R
            esponseFunctionTissueName="Unspecified"

```

```

if r.Name=="MammellaDx":
    case.PatientModel.RegionsOfInterest['MammellaDx'].OrganData.ResponseFunctionTissueName="Unspecified"
if r.Name=="Cuore":
    case.PatientModel.RegionsOfInterest['Cuore'].OrganData.ResponseFunctionTissueName="Heart"
if r.Name=="Midollo":
    case.PatientModel.RegionsOfInterest['Midollo'].OrganData.ResponseFunctionTissueName="Spinal chord"
if r.Name=="Tiroide":
    case.PatientModel.RegionsOfInterest['Tiroide'].OrganData.ResponseFunctionTissueName="Unspecified"
if r.Name=="Body":
case.PatientModel.RegionsOfInterest['Body'].SetAsExternal()
case.PatientModel.RegionsOfInterest['Body'].OrganData.ResponseFunctionTissueName="Skin"

if r.Name=="GTV":
    case.PatientModel.RegionsOfInterest['GTV'].Type="Gtv"
    case.PatientModel.RegionsOfInterest['GTV'].OrganData.ResponseFunctionTissueName="Unspecified"
    case.PatientModel.RegionsOfInterest['GTV'].OrganData.OrganType="Target"

if r.Name=="CTV_PBI":
    case.PatientModel.RegionsOfInterest['CTV_PBI'].Type="Ctv"
    case.PatientModel.RegionsOfInterest['CTV_PBI'].OrganData.ResponseFunctionTissueName="Unspecified"
    case.PatientModel.RegionsOfInterest['CTV_PBI'].OrganData.OrganType="Target"
if r.Name=="CTV_WBI":
    case.PatientModel.RegionsOfInterest['CTV_WBI'].Type="Ctv"
    case.PatientModel.RegionsOfInterest['CTV_WBI'].OrganData.ResponseFunctionTissueName="Unspecified"
    case.PatientModel.RegionsOfInterest['CTV_WBI'].OrganData.OrganType="Target"
if r.Name=="PTV_WBI":
    case.PatientModel.RegionsOfInterest['PTV_WBI'].Type="Ptv"
    case.PatientModel.RegionsOfInterest['PTV_WBI'].OrganData.ResponseFunctionTissueName="Unspecified"
    case.PatientModel.RegionsOfInterest['PTV_WBI'].OrganData.OrganType="Target"
if r.Name=="PTV_EVAL":
    case.PatientModel.RegionsOfInterest['PTV_EVAL'].Type="Ptv"
    case.PatientModel.RegionsOfInterest['PTV_EVAL'].OrganData.ResponseFunctionTissueName="Unspecified"
    case.PatientModel.RegionsOfInterest['PTV_EVAL'].OrganData.OrganType="Target"
if r.Name=="PTV_PBI":
    case.PatientModel.RegionsOfInterest['PTV_PBI'].Type="Ptv"
    case.PatientModel.RegionsOfInterest['PTV_PBI'].OrganData.ResponseFunctionTissueName="Unspecified"
    case.PatientModel.RegionsOfInterest['PTV_PBI'].OrganData.OrganType="Target"

```

```

patient.Save() #salva il paziente

```

## Appendice 2. Script di calcolo dei valori di Dice e Hausdorff automatico e con GUI (agisce su paziente singolo)

### 2.1 Automatico

```
# Calcolo degli indici Dice e distanze di Hausdorff per i pazienti indicati (tramite il loro ID) in id_list

from connect import *

import os, sys

def write_indici (Dice,HausAver,HausMax,volume,progressivo,file_name):
    with open(file_name, 'r+') as outfile:
        outfile.seek(0,2)
        outfile.write(progressivo + ";" + Dice + ";" + HausAver + ";" + HausMax + ";"
            + volume + "\n")

id_list=[] #scrivere ID pazienti di interesse

for id in id_list:
    patient_db = get_current("PatientDB")
    id_p = id
    info = patient_db.QueryPatientInfo(Filter={'PatientID':id_p})
    if len(info) == 1: # Check that info contains exactly one item.
        patient = patient_db.LoadPatient(PatientInfo=info[0])
        patient = get_current('Patient')
        # Get a handle to the first case of the patient
        case_to_load = patient.Cases[0]
        # Load the case
        case_to_load.SetCurrent()

        case=get_current("Case")
        roi_attuali=[r.OfRoi.Name for r in
            case.PatientModel.StructureSets[case.Examinations[0].Name].RoiGeometries if
            r.PrimaryShape != None]

        Roi_daCalcSx=["PolmoneSx","PolmoneDx","Cuore","Midollo","TestaOmeroSx","Tiroide","MammellaDx"]
        Roi_daCalcDx=["PolmoneSx","PolmoneDx","Cuore","Midollo","TestaOmeroDx","Tiroide","MammellaSx"]

        if "MammellaSx" in roi_attuali:
            Roi_daCalc=Roi_daCalcDx #vedo a quale irma appartiene il paziente
        else:
            Roi_daCalc=Roi_daCalcSx

        input=8

        case = get_current("Case")
        examination = get_current("Examination")
        patient=get_current("Patient")
        ID_p=patient.PatientID

        percorso = os.path.dirname(os.path.abspath(__file__))

        f = open("{0}\{1}_DiceHaus.csv".format(percorso, ID_p), 'w')
        f.close()
```

```

file_name = "{0}\{1}_DiceHaus.csv".format(percorso, ID_p)
with open(file_name, 'r+') as outfile:
    outfile.seek(0,2) #va all'ultimo carattere del file di testo
    outfile.write(" ;Dice;HausAver;HausMax;Volume" + "\n")

for i in range(0,len(Roi_daCalc)):
    RoiCorrente=Roi_daCalc[i]
    for n in range(1,input+1):

        if RoiCorrente in roi_attuali:
            n1=str(n)
            NomeRoiA=RoiCorrente
            NomeRoiB=RoiCorrente + " (" + n1 + ")"
            #calcolo indice hausdorff
            struttura=case.PatientModel.StructureSets[0]
            HausDiz1=dict(struttura.RoiSurfaceToSurfaceDistanceBasedOnDT(ReferenceRoiName=NomeRoiA,TargetRoiName=NomeRoiB)) # Haus AB
            HausDiz2=dict(struttura.RoiSurfaceToSurfaceDistanceBasedOnDT(ReferenceRoiName=NomeRoiB,TargetRoiName=NomeRoiA)) # Haus BA
            HausAver1=HausDiz1["Average"]
            HausMax1=HausDiz1["Max"]
            HausAver2=HausDiz2["Average"]
            HausMax2=HausDiz2["Max"]
            HausAver=max(HausAver1,HausAver2)
            HausMax=max(HausMax1,HausMax2)

            #calcolo indice di dice
            DiceDiz=dict(struttura.ComparisonOfRoiGeometries(RoiA=NomeRoiA,RoiB=NomeRoiB))

            Dice=DiceDiz["DiceSimilarityCoefficient"]
            #calcolo volume
            volume=struttura.RoiGeometries[RoiCorrente].GetRoiVolume()

            write_indici(str(round(Dice,4)),str(round(HausAver,4)),str(round(HausMax,4)),str(round(volume,4)),NomeRoiB,file_name)
        else: continue

MBS_models=["Lung (Left)","Lung (Right)", "SpinalCord (Thorax)"]
ref_models=["PolmoneSx","PolmoneDx","Midollo"]
for j in range(0,len(MBS_models)):
    model=MBS_models[j]
    if model in roi_attuali:
        #calcolo indice hausdorff
        struttura=case.PatientModel.StructureSets[0]
        HausDiz1=dict(struttura.RoiSurfaceToSurfaceDistanceBasedOnDT(ReferenceRoiName=model,TargetRoiName=ref_models[j])) # Haus AB
        HausDiz2=dict(struttura.RoiSurfaceToSurfaceDistanceBasedOnDT(ReferenceRoiName=ref_models[j],TargetRoiName=model)) # Haus BA
        HausAver1=HausDiz1["Average"]
        HausMax1=HausDiz1["Max"]
        HausAver2=HausDiz2["Average"]
        HausMax2=HausDiz2["Max"]
        HausAver=max(HausAver1,HausAver2)
        HausMax=max(HausMax1,HausMax2)
        #calcolo indice di dice
        DiceDiz=dict(struttura.ComparisonOfRoiGeometries(RoiA=ref_models[j],RoiB=model))
        Dice=DiceDiz["DiceSimilarityCoefficient"]
        #calcolo volume

```

```

        volume=struttura.RoiGeometries[model].GetRoiVolume()
        write_indici(str(round(Dice,4)),str(round(HausAver,4))
        ,str(round(HausMax,4)),str(round(volume,4)),model,file
        _name)
    else: continue
else: continue

```

## 2.2 Con GUI

```

#calcolo degli indici Dice e distanza di Hausdorff del numero di ROI selezionate in input
from connect import *

import clr
clr.AddReference("System.Windows.Forms")
clr.AddReference("System.Drawing")

import sys

from System.Windows.Forms import Application, Form, Label, Button, TextBox, ComboBox
from System.Drawing import Point, Size

class NumRoi(Form):
    def __init__(self):
        # grandezza della finestra
        self.Size=Size(300,200)
        # titolo della finestra
        self.Text='Numero Roi'

        #aggiunta testo/label
        self.label=Label()
        self.label.Text="Inserisci numero di Roi delle quali calcolare gli indici"
        self.label.Location=Point(15,15)
        self.label.AutoSize=True
        self.Controls.Add(self.label)

        # aggiunta textbox
        self.textbox=TextBox()
        self.textbox.Text="0"
        self.textbox.Location=Point(15,60)
        self.textbox.AutoSize=True
        self.Controls.Add(self.textbox)

        #aggiunta bottone
        self.button1=Button()
        self.button1.Text="Enter"
        self.button1.Location=Point(15,100)
        self.button1.Click+=self.button_clicked
        self.Controls.Add(self.button1)

    def button_clicked(self, sender, event):
        self.numinput=int(self.textbox.Text)
        self.Close()

def write_indici (Dice,HausAver,HausMax,volume,progressivo,file_name):
    with open(file_name, 'r+') as outfile:
        outfile.seek(0,2)
        outfile.write(progressivo + ";" + Dice + ";" + HausAver + ";" + HausMax + ";"
        + volume + "\n")

case=get_current("Case")

```

```

roi_attuali=[r.OfRoi.Name for r in
case.PatientModel.StructureSets[case.Examinations[0].Name].RoiGeometries if
r.PrimaryShape != None]

Roi_daCalcSx=["PolmoneSx","PolmoneDx","Cuore","Midollo","TestaOmeroSx","Tiroide","MammellaDx
"]
Roi_daCalcDx=["PolmoneSx","PolmoneDx","Cuore","Midollo","TestaOmeroDx","Tiroide","MammellaSx
"]

if "MammellaSx" in roi_attuali:
    Roi_daCalc=Roi_daCalcDx #vedo a quale irma appartiene il paziente
else:
    Roi_daCalc=Roi_daCalcSx

form=NumRoi()
Application.Run(form)
input=form.numinput

case = get_current("Case")
examination = get_current("Examination")
patient=get_current("Patient")
ID_p=patient.PatientID

percorso = os.path.dirname(os.path.abspath(__file__))

f = open("{0}\{1}_DiceHaus.csv".format(percorso, ID_p), 'w')
f.close()

file_name = "{0}\{1}_DiceHaus.csv".format(percorso, ID_p)
with open(file_name, 'r+') as outfile:
    outfile.seek(0,2) #va all'ultimo carattere del file di testo
    outfile.write(" ;Dice;HausAver;HausMax;Volume" + "\n")

for i in range(0,len(Roi_daCalc)):
    RoiCorrente=Roi_daCalc[i]
    for n in range(1,input+1):

        if RoiCorrente in roi_attuali:
            n1=str(n)
            NomeRoiA=RoiCorrente
            NomeRoiB=RoiCorrente + " (" + n1 + ")"
            #calcolo indice hausdorff
            struttura=case.PatientModel.StructureSets[0]
            HausDiz1=dict(struttura.RoiSurfaceToSurfaceDistanceBasedOnDT(Referenc
eRoiName=NomeRoiA,TargetRoiName=NomeRoiB)) # Haus AB
            HausDiz2=dict(struttura.RoiSurfaceToSurfaceDistanceBasedOnDT(Referenc
eRoiName=NomeRoiB,TargetRoiName=NomeRoiA)) # Haus BA
            HausAver1=HausDiz1["Average"]
            HausMax1=HausDiz1["Max"]
            HausAver2=HausDiz2["Average"]
            HausMax2=HausDiz2["Max"]
            HausAver=max(HausAver1,HausAver2)
            HausMax=max(HausMax1,HausMax2)

            #calcolo indice di dice
            DiceDiz=dict(struttura.ComparisonOfRoiGeometries (RoiA=NomeRoiA,RoiB=N
omeRoiB))
            Dice=DiceDiz["DiceSimilarityCoefficient"]
            #calcolo volume
            volume=struttura.RoiGeometries[RoiCorrente].GetRoiVolume()
            write_indici(str(round(Dice,4)),str(round(HausAver,4)),str(round(Haus
Max,4)),str(round(volume,4)),NomeRoiB,file_name)
        else: continue

MBS_models=["Lung (Left)","Lung (Right)", "SpinalCord (Thorax)"]
ref_models=["PolmoneSx","PolmoneDx","Midollo"]
for j in range(0,len(MBS_models)):

```

```

model=MBS_models[j]
if model in roi_attuali:
    #calcolo indice hausdorff
    struttura=case.PatientModel.StructureSets[0]
    HausDiz1=dict(struttura.RoiSurfaceToSurfaceDistanceBasedOnDT(Referenc
eRoiName=model,TargetRoiName=ref_models[j])) # Haus AB
    HausDiz2=dict(struttura.RoiSurfaceToSurfaceDistanceBasedOnDT(Referenc
eRoiName=ref_models[j],TargetRoiName=model)) # Haus BA
    HausAver1=HausDiz1["Average"]
    HausMax1=HausDiz1["Max"]
    HausAver2=HausDiz2["Average"]
    HausMax2=HausDiz2["Max"]
    HausAver=max(HausAver1,HausAver2)
    HausMax=max(HausMax1,HausMax2)

    #calcolo indice di dice
    DiceDiz=dict(struttura.ComparisonOfRoiGeometries(RoiA=ref_models[j],R
oiB=model))
    Dice=DiceDiz["DiceSimilarityCoefficient"]
    #calcolo volume
    volume=struttura.RoiGeometries[model].GetRoiVolume()
    write_indici(str(round(Dice,4)),str(round(HausAver,4)),str(round(Haus
Max,4)),str(round(volume,4)),model,file_name)
else: continue

```

## Appendice 3. Script di generazione delle segmentazioni automatiche

### 3.1 Automatico, valido per database PBI

```
# Script che permette la generazione delle segmentazioni secondo
# i due metodi per il paziente corrente automaticamente - valido
# per i database PBI

from connect import *

import os, sys
import datetime

id_list=[] #scrivere qui gli id dei pazienti
nAtlas_list=[40,30,20,10]
for id in id_list:
    patient_db = get_current("PatientDB")
    id_p = id
    info = patient_db.QueryPatientInfo(Filter=('PatientID':id_p))
    if len(info) == 1: # Check that info contains exactly one item.
        patient = patient_db.LoadPatient(PatientInfo=info[0])
        patient = get_current('Patient')
        # Get a handle to the first case of the patient
        case_to_load = patient.Cases[0]
        # Load the case
        case_to_load.SetCurrent()

        case=get_current("Case")
        db=get_current("PatientDB")
        roi_attuali=[r.OfRoi.Name for r in
        case.PatientModel.StructureSets[case.Examinations[0].Name].RoiGeometries if
        r.PrimaryShape != None]
        Roi_daCalcSx=["PolmoneSx","PolmoneDx","Cuore","Midollo","TestaOmeroSx","Tiroide","MammellaDx"]
        Roi_daCalcDx=["PolmoneSx","PolmoneDx","Cuore","Midollo","TestaOmeroDx","Tiroide","MammellaSx"]

        if "MammellaSx" in roi_attuali:
            roi_names=Roi_daCalcDx
            TemplName='IRMA_PBI_Dx'
            db_IRMA=db.TemplatePatientModels['IRMA_PBI_Dx']
        else:
            roi_names=Roi_daCalcSx
            TemplName='IRMA_PBI_Sx'
            db_IRMA=db.TemplatePatientModels['IRMA_PBI_Sx']

        ExmNames=[e.Name for e in db_IRMA.StructureSetExaminations] #lista dei pazienti atlas
        n_paz=len(ExmNames)

        patient=get_current("Patient")
        examination=get_current("Examination")

        ID_p=patient.PatientID
        if ID_p in ExmNames:
            ExmNames.remove(ID_p) #rimuovo il paziente stesso dall'atlas se è presente
        print roi_names
        diz_exam=[]
        for e in case.Examinations:
            diz_exam.append(e)

        #creazione roi mbs

        case.PatientModel.MBSAutoInitializer(MbsRois=[{ 'CaseType': "Thorax", 'ModelName': "Lung
        (Left)", 'RoiName': "Lung (Left)", 'RoiColor': "255, 51, 204" }, { 'CaseType': "Thorax",
        'ModelName': "Lung (Right)", 'RoiName': "Lung (Right)", 'RoiColor': "0, 255, 0" },
        { 'CaseType': "Thorax", 'ModelName': "SpinalCord (Thorax)", 'RoiName': "SpinalCord
```

```

(Thorax)", 'RoiColor': "0, 0, 255" ]], CreateNewRois=True, Examination=examination,
UseAtlasBasedInitialization=True)

case.PatientModel.AdaptMbsMeshes (Examination=examination, RoiNames=["Lung (Left)", "Lung
(Right)", "SpinalCord (Thorax)"], CustomStatistics=None, CustomSettings=None)

case.PatientModel.StructureSets[0].RoiGeometries['Lung
(Left)'].SetRepresentation(Representation='Contours')

case.PatientModel.StructureSets[0].RoiGeometries['Lung
(Right)'].SetRepresentation(Representation='Contours')

case.PatientModel.StructureSets[0].RoiGeometries['SpinalCord
(Thorax)'].SetRepresentation(Representation='Contours')

patient.Save ()

for n in nAtlas_list:

    nAtlas=n
    if nAtlas > len(ExmNames)+1:
        print "Il numero di pazienti selezionato è maggiore dei pazienti
        attualmente presenti nell'atlante"
    else:
        ExmNames=ExmNames[0:nAtlas]
        print ExmNames
        for exam in diz_exam:
            tstart = datetime.datetime.now() #parte timer

            case.PatientModel.CreateStructuresFromAtlas (SourceTemplateName
            =TemplName, SourceExaminationsNames=ExmNames,
            SourceRoiNames=roi_names, SourcePoiNames=[],
            AssociateStructuresByName=False, TargetExamination=exam)

            tend = datetime.datetime.now() #Ferma timer
            timediff = tend - tstart #Tempo di esecuzione
            timediff = str(timediff)[:4]
            print timediff
            for roi in roi_names:
                if roi in roi_attuali: continue
                else:

case.PatientModel.RegionsOfInterest[roi].DeleteRoi ()
patient.Save ()

percorso = os.path.dirname(os.path.abspath(__file__))
f = open("{0}\{1}_tempi_one_{2}paz.csv".format(percorso,ID_p,n),'w')
f.close()
file_name = "{0}\{1}_tempi_one_{2}paz.csv".format(percorso,ID_p,n)
for exam in diz_exam:
    for roi in roi_names:

        if roi in roi_attuali:
            RoiName=[roi]

            tstart = datetime.datetime.now()

            case.PatientModel.CreateStructuresFromAtlas (S
            ourceTemplateName=TemplName,
            SourceExaminationsNames=ExmNames,
            SourceRoiNames=RoiName, SourcePoiNames=[],
            AssociateStructuresByName=False,
            TargetExamination=exam)

            tend=datetime.datetime.now()
            timediff = tend - tstart
            timediff = str(timediff)[:4]
            with open(file_name, 'r+') as outfile:
                outfile.seek(0,2)
                outfile.write(roi + ";" + timediff +
                "\n")

        else: continue

patient.Save ()

else: continue

```

## 3.2 Con GUI, valido per database PBI

```
# Script che permette la generazione delle segmentazioni secondo
# i due metodi per il paziente corrente mediante l'utilizzo di
# GUI - valido per i database PBI

from connect import *

import clr
clr.AddReference("System.Windows.Forms")
clr.AddReference("System.Drawing")
import os, sys
import datetime
import random as rn

from System.Windows.Forms import Application, Form, Label, Button, TextBox, ComboBox
from System.Drawing import Point, Size

case=get_current("Case")
db=get_current("PatientDB")
roi_attuali=[r.OfRoi.Name for r in
case.PatientModel.StructureSets[case.Examinations[0].Name].RoiGeometries if r.PrimaryShape != None]
Roi_daCalcSx=["PolmoneSx", "PolmoneDx", "Cuore", "Midollo", "TestaOmeroSx", "Tiroide", "MammellaDx"]
Roi_daCalcDx=["PolmoneSx", "PolmoneDx", "Cuore", "Midollo", "TestaOmeroDx", "Tiroide", "MammellaSx"]

if "MammellaSx" in roi_attuali: #vedo a quale irma appartiene il paziente: se ho mammella sx tra le roi
allora sar  irma dx perch  mammella sx si riferisce al contorno della mammella sana
    roi_names=Roi_daCalcDx
    TemplName='IRMA_PBI_Dx'
    db_IRMA=db.TemplatePatientModels['IRMA_PBI_Dx']
else:
    roi_names=Roi_daCalcSx
    TemplName='IRMA_PBI_Sx'
    db_IRMA=db.TemplatePatientModels['IRMA_PBI_Sx']

ExmNames=[e.Name for e in db_IRMA.StructureSetExaminations] #lista dei pazienti presenti nell'atlas
n_paz=len(ExmNames)
class metodo(Form):
    def __init__(self):
        # grandezza della finestra
        self.Size=Size(300,300)
        # titolo della finestra
        self.Text='Selezione metodo'

        #aggiunta testo/label
        self.label=Label()
        self.label.Text="Inserisci numero atlanti da usare"
        self.label.Location=Point(15,15)
        self.label.AutoSize=True
        self.Controls.Add(self.label)

        # aggiunta textbox
        self.textbox=TextBox()
        self.textbox.Text=str(n_paz)
        self.textbox.Location=Point(15,60)
        self.textbox.AutoSize=True
        self.Controls.Add(self.textbox)

        #aggiunta testo/label
        self.label=Label()
        self.label.Text="Seleziona metodo"
        self.label.Location=Point(15,115)
        self.label.AutoSize=True
```

```

self.Controls.Add(self.label)

# aggiunta combobox
self.combobox=ComboBox()
self.combobox.DataSource=["1 struttura","tutte le strutture"]
self.combobox.Location=Point(15,160)
self.combobox.AutoSize=True
self.Controls.Add(self.combobox)

#aggiunta bottone
self.button1=Button()
self.button1.Text="Enter"
self.button1.Location=Point(15,200)
self.button1.Click+=self.button_clicked
self.Controls.Add(self.button1)

def button_clicked(self, sender, event):
    self.scelta=self.combobox.SelectedValue
    self.numinput=int(self.textbox.Text)
    self.Close()

patient=get_current("Patient")

ID_p=patient.PatientID
if ID_p in ExmNames:
    ExmNames.remove(ID_p) #rimuovo il paziente stesso dall'atlas se è presente
print roi_names
diz_exam=[]
for e in case.Examinations:
    diz_exam.append(e)

form=metodo()
Application.Run(form)
scelta=form.scelta
nAtlas=form.numinput
if nAtlas > len(ExmNames)+1:
    print "Il numero di pazienti selezionato è maggiore dei pazienti attualmente presenti
nell'atlante"
else:
    ExmNames=ExmNames[0:nAtlas]
    print ExmNames
    if scelta=="tutte le strutture":
        for exam in diz_exam:
            tstart = datetime.datetime.now() #parte timer
            case.PatientModel.CreateStructuresFromAtlas(SourceTemplateName=TemplName,
            SourceExaminationsNames=ExmNames, SourceRoiNames=roi_names, SourcePoiNames=[],
            AssociateStructuresByName=False, TargetExamination=exam)
            tend = datetime.datetime.now() #Ferma timer
            timediff = tend - tstart #Tempo di esecuzione
            timediff = str(timediff)[-4] #Tolgo le ultime 4 cifre convertendo in stringa
            print timediff
            for roi in roi_names:
                if roi in roi_attuali: continue
                else:
                    case.PatientModel.RegionsOfInterest[roi].DeleteRoi()
            patient.Save()

    else:
        percorso = os.path.dirname(os.path.abspath(__file__))
        f = open("{0}\{1}_tempi_one.csv".format(percorso,ID_p), 'w')
        f.close()
        file_name = "{0}\{1}_tempi_one.csv".format(percorso, ID_p)
        for exam in diz_exam:
            for roi in roi_names:

                if roi in roi_attuali:
                    RoiName=[roi]

```

```

tstart = datetime.datetime.now()
case.PatientModel.CreateStructuresFromAtlas(SourceTemplateName
=TemplName, SourceExaminationsNames=ExmNames,
SourceRoiNames=RoiName, SourcePoiNames=[],
AssociateStructuresByName=False, TargetExamination=exam)
tend=datetime.datetime.now()
timediff = tend - tstart #Tempo di esecuzione
timediff = str(timediff)[:4]
with open(file_name, 'r+') as outfile:
    outfile.seek(0,2)
    outfile.write(roi + ";" + timediff + "\n")
else: continue

patient.Save()

```

### 3.3 Con GUI + selezione atlante per il quale applicarlo

```

# Script che permette la generazione delle segmentazioni secondo
# i due metodi per il paziente corrente automaticamente - valido per ogni database

from connect import *

import clr
clr.AddReference("System.Windows.Forms")
clr.AddReference("System.Drawing")
import os, sys
import datetime
import random as rn

from System.Windows.Forms import Application, Form, Label, Button, TextBox, ComboBox
from System.Drawing import Point, Size

db=get_current("PatientDB")
case=get_current("Case")
patient=get_current("Patient")

class metodo(Form):
    def __init__(self):
        # grandezza della finestra
        self.Size=Size(350,400)
        # titolo della finestra
        self.Text='Selezione metodo e atlanti'

        #aggiunta testo/label
        self.label=Label()
        self.label.Text="Seleziona atlante"
        self.label.Location=Point(15,15)
        self.label.AutoSize=True
        self.Controls.Add(self.label)

        # aggiunta combobox
        self.combobox1=ComboBox()
        self.combobox1.DataSource=[a.Name for a in db.TemplatePatientModels] #lista degli
atlanti disponibili
        self.combobox1.Location=Point(15,60)
        self.combobox1.Size=Size(190,150)
        self.Controls.Add(self.combobox1)

        #aggiunta testo/label
        self.labell=Label()
        self.labell.Text="Inserisci numero pazienti da usare nell'atlante scelto"
        self.labell.Location=Point(15,115)
        self.labell.AutoSize=True
        self.Controls.Add(self.labell)

        # aggiunta textbox
        self.textbox=TextBox()

```

```

        self.textbox.Text="0"
        self.textbox.Location=Point(15,160)
        self.textbox.AutoSize=True
        self.Controls.Add(self.textbox)

        #aggiunta testo/label
        self.label2=Label()
        self.label2.Text="Seleziona metodo"
        self.label2.Location=Point(15,215)
        self.label2.AutoSize=True
        self.Controls.Add(self.label2)

        # aggiunta combobox
        self.combobox=ComboBox()
        self.combobox.DataSource=["1 struttura","tutte le strutture"]
        self.combobox.Location=Point(15,260)
        self.combobox.AutoSize=True
        self.Controls.Add(self.combobox)

        #aggiunta bottone
        self.button1=Button()
        self.button1.Text="Enter"
        self.button1.Location=Point(15,300)
        self.button1.Click+=self.button_clicked
        self.Controls.Add(self.button1)

    def button_clicked(self, sender, event):
        self.scelta=self.combobox.SelectedValue
        self.numinput=int(self.textbox.Text)
        self.AtlasName=self.combobox1.SelectedValue
        self.Close()

roi_attuali=[r.OfRoi.Name for r in
case.PatientModel.StructureSets[case.Examinations[0].Name].RoiGeometries if r.PrimaryShape != None]

roi_names=roi_attuali

form=metodo()
Application.Run(form)
scelta=form.scelta
nAtlas=form.numinput+1
AtlasName=form.AtlasName
db_corrente=db.TemplatePatientModels[AtlasName]
print AtlasName

ExmNames_all=[e.Name for e in db_corrente.StructureSetExaminations] #lista con tutti i pazienti
nell'atlante scelto
ID_p=patient.PatientID
if ID_p in ExmNames_all:
    ExmNames_all.remove(ID_p) #rimuovo il paziente stesso dall'atlas se è presente
print roi_names
diz_exam=[]
for e in case.Examinations:
    diz_exam.append(e)

if nAtlas > len(ExmNames_all):
    print "Il numero di pazienti selezionato è maggiore dei pazienti attualmente presenti
nell'atlante"
else:
    ExmNames=[] #lista con gli atlanti che verranno usati nell' algoritmo di segmentazione
    for j in range(1,nAtlas): # con questo mi costruisco un vettore con il numero di atlanti scelto
dall'utente scegliendo a caso atlanti dal database totale
        n=rn.randrange(0,len(ExmNames_all),1)
        ExmNames.append(ExmNames_all[n])

print ExmNames
if scelta=="tutte le strutture":
    for exam in diz_exam:
        tstart = datetime.datetime.now() #parte timer

```

```

        case.PatientModel.CreateStructuresFromAtlas(SourceTemplateName=AtlasName,
        SourceExaminationsNames=ExmNames, SourceRoiNames=roi_names, SourcePoiNames=[],
        AssociateStructuresByName=False, TargetExamination=exam)

        tend = datetime.datetime.now() #Ferma timer
        timediff = tend - tstart      #Tempo di esecuzione
        timediff = str(timediff)[-4]  #Tolgo le ultime 4 cifre convertendo in stringa
        print timediff

        for roi in roi_names:
            if roi in roi_attuali: continue
            else:
                case.PatientModel.RegionsOfInterest[roi].DeleteRoi()
        patient.Save()

else:
    percorso = os.path.dirname(os.path.abspath(__file__))
    f = open("{0}\{1}_tempi_one.csv".format(percorso, ID_p), 'w')
    f.close()
    file_name = "{0}\{1}_tempi_one.csv".format(percorso, ID_p)
    for exam in diz_exam:
        for roi in roi_names:

            if roi in roi_attuali:
                RoiName=[roi]

                tstart = datetime.datetime.now()

                case.PatientModel.CreateStructuresFromAtlas(SourceTemplateName
                =AtlasName, SourceExaminationsNames=ExmNames,
                SourceRoiNames=RoiName, SourcePoiNames=[],
                AssociateStructuresByName=False, TargetExamination=exam)

                tend=datetime.datetime.now()
                timediff = tend - tstart      #Tempo di esecuzione
                timediff = str(timediff)[-4]
                with open(file_name, 'r+') as outfile:
                    outfile.seek(0,2)
                    outfile.write(roi + ";" + timediff + "\n")
            else: continue

    patient.Save()

```

## Appendice 4. Script di estrazione degli istogrammi

```
# Esportazione degli istogrammi in un file output in fomato .csv
# delle roi indicate in roi_names per tutti i pazienti indicati in id_list
# valido per la ROI del cuore

from connect import *

from System import Array
import math
import sys
import clr
clr.AddReference('System.Windows.Forms')
clr.AddReference('System.Drawing')
import System.Windows.Forms as WinForms
from System.Windows.Forms import DialogResult, FolderBrowserDialog
import re

# selezione cartella output
dialog = FolderBrowserDialog()
dialog.SelectedPath = "C:\\\"

if dialog.ShowDialog() == DialogResult.OK:
    try:
        path = dialog.SelectedPath
    except IOError, e:
        print 'An error occurred:'

id_list=[] #lista ID pazienti interessati

for id in id_list:
    patient_db = get_current("PatientDB")
    id_p = id
    info = patient_db.QueryPatientInfo(Filter={'PatientID':id_p})
    if len(info) == 1: # info deve avere un solo elemento
        patient = patient_db.LoadPatient(PatientInfo=info[0])
        patient = get_current('Patient')
        # case da caricare
        case_to_load = patient.Cases[0]
        # caricamento caso
        case_to_load.SetCurrent()

        patient = get_current("Patient")
        case = get_current("Case")
        exam=get_current("Examination")

        IDp=patient.PatientID

        roi_attuali=[r.OfRoi.Name for r in
        case.PatientModel.StructureSets[case.Examinations[0].Name].RoiGeometries if
        r.PrimaryShape != None]
        roi_names = ["Cuore","cuore03","cuore05","cuore08"] #nomi delle ROI delle quali si vuole
        esportare l'istogramma
        roi_attuali=[r.OfRoi.Name for r in
        case.PatientModel.StructureSets[case.Examinations[0].Name].RoiGeometries if
        r.PrimaryShape != None]

        if "Cuore" in roi_attuali:

            for roi in roi_names:
                f = open("{0}\{1}_{2}_HU.csv".format(path, IDp, roi), 'w')
                f.close()
                file = "{0}\{1}_{2}_HU.csv".format(path, IDp, roi)

                with open(file, 'r+') as outfile:
                    outfile.seek(0,2)
```

```

        outfile.write("ID_paziente;HU;norm_voxel;volume;nome_roi"+
"\n")

struttura=case.PatientModel.StructureSets[0]
volume=struttura.RoiGeometries[roi].GetRoiVolume()
BinSize=3
hist =
case.Examinations[exam.Name].Series[0].ImageStack.GetImageStackHistogram(BinSize = BinSize, ExcludePixelPadding = False, RoiName = roi,
ConvertCbctValueToHu = False)
lenArray=((1005+150)/BinSize)+1
array=Array.CreateInstance(float,lenArray,2)
i=0
n_voxel=0.0
n_rim=0.0
for key in hist:
    n_voxel=n_voxel+key.Value
    if key.Key>-1005:
        if key.Key<150:
            array[i,0]=key.Key
            array[i,1]=key.Value
            i=i+1
        else: continue
    else: continue

for j in range(0,i):
    array[j,1]=array[j,1]/n_voxel
with open(file, 'r+') as outfile:
    outfile.seek(0,2)
    for k in range(0,i):
        outfile.write(str(IDp)+" "+str(round(array[k,0],4))+" "+str(round(array[k,1],4))+" "+str(round(volume,4))+" "+roi+ "\n")

        else: continue
    else: continue

# Esportazione degli istogrammi in un file output in formato .csv
# delle roi indicate in roi_names per tutti i pazienti indicati in id_list
# valido per la ROI del midollo

from System import Array
import math
from connect import *

import sys
import clr
clr.AddReference('System.Windows.Forms')
clr.AddReference('System.Drawing')
import System.Windows.Forms as WinForms
from System.Windows.Forms import DialogResult, FolderBrowserDialog
import re

#selezione cartella output
dialog = FolderBrowserDialog()
dialog.SelectedPath = "C:\\\"

if dialog.ShowDialog() == DialogResult.OK:
    try:
        path = dialog.SelectedPath
    except IOError, e:
        print 'An error occurred:'

id_list=[] #scrivere ID pazienti interessati
for id in id_list:
    patient_db = get_current("PatientDB")
    id_p = id
    info = patient_db.QueryPatientInfo(Filter='PatientID:id_p')
    if len(info) == 1: # info deve avere un solo elemento
        patient = patient_db.LoadPatient(PatientInfo=info[0])

```

```

patient = get_current('Patient')
# case da caricare
case_to_load = patient.Cases[0]
# caricamento caso
case_to_load.SetCurrent()

patient = get_current("Patient")
case = get_current("Case")
exam=get_current("Examination")

IDp=patient.PatientID

roi_names = [] #scrivere nomi ROI delle quale si vuole estrarre l'istogramma
for roi in roi_names:
    f = open("{0}\{1}_{2}_HU.csv".format(path,IDp,roi),'w')
    f.close()
    file = "{0}\{1}_{2}_HU.csv".format(path,IDp,roi)

    with open(file, 'r+') as outfile:
        outfile.seek(0,2)
        outfile.write("ID_paziente;HU;norm_voxel;volume;nome_roi"+ "\n")
    # estrazione istogramma
    struttura=case.PatientModel.StructureSets[0]
    volume=struttura.RoiGeometries[roi].GetRoiVolume()
    BinSize=3
    hist =
    case.Examinations[exam.Name].Series[0].ImageStack.GetImageStackHistogram(BinSize,
    e = BinSize, ExcludePixelPadding = False, RoiName = roi, ConvertCbctValueToHu =
    False)
    lenArray=((1206+306)/BinSize)+1
    array=Array.CreateInstance(float,lenArray,2)
    i=0
    n_voxel=0.0
    n_rim=0.0
    for key in hist:
        n_voxel=n_voxel+key.Value
        if key.Key>-306:
            if key.Key<1206:
                array[i,0]=key.Key
                array[i,1]=key.Value
                i=i+1
            else: continue
        else: continue

    for j in range(0,i):
        array[j,1]=array[j,1]/n_voxel
    with open(file, 'r+') as outfile:
        outfile.seek(0,2)
        for k in range(0,i):
            outfile.write(str(IDp)+" "+str(round(array[k,0],4))+" "+str(round(array[k,1],4))+" "+str(round(volume,4))+" "+roi+ "\n")

else: continue

```

## Appendice 5. Script di generazione delle ROI modificate

```
# Generazione delle ROI modificate per tutti i pazienti indicati in id_list. Valido per la ROI del cuore

from connect import *

import sys

id_list=[] #inserire ID pazienti di interesse
for id in id_list:
    patient_db = get_current("PatientDB")
    id_p = id
    info = patient_db.QueryPatientInfo(Filter='PatientID':id_p)
    if len(info) == 1:
        patient = patient_db.LoadPatient(PatientInfo=info[0])
        patient = get_current('Patient')
        case_to_load = patient.Cases[0]
        case_to_load.SetCurrent()

        patient = get_current("Patient")
        case = get_current("Case")
        exam=get_current("Examination")

        IDp=patient.PatientID
        roi_attuali=[r.OfRoi.Name for r in
        case.PatientModel.StructureSets[case.Examinations[0].Name].RoiGeometries if
        r.PrimaryShape != None]

        if "Cuore" in roi_attuali:

            #creazione roi sbagliate:

            #cuore03

            retval_0 = case.PatientModel.CreateRoi(Name="cuore03", Color="Cyan",
            Type="Organ", TissueName=None, RbeCellTypeName=None, RoiMaterial=None)

            retval_0.SetMarginExpression(SourceRoiName="Cuore", MarginSettings={ 'Type':
            "Expand", 'Superior': 0.3, 'Inferior': 0.3, 'Anterior': 0.3, 'Posterior': 0.3,
            'Right': 0.3, 'Left': 0.3 })

            retval_0.UpdateDerivedGeometry(Examination=exam, Algorithm="Auto")

            #cuore015

            retval_1 = case.PatientModel.CreateRoi(Name="cuore05", Color="Cyan",
            Type="Organ", TissueName=None, RbeCellTypeName=None, RoiMaterial=None)

            retval_1.SetMarginExpression(SourceRoiName="Cuore", MarginSettings={ 'Type':
            "Expand", 'Superior': 0.5, 'Inferior': 0.5, 'Anterior': 0.5, 'Posterior': 0.5,
            'Right': 0.5, 'Left': 0.5 })

            retval_1.UpdateDerivedGeometry(Examination=exam, Algorithm="Auto")

            #cuore08

            retval_2 = case.PatientModel.CreateRoi(Name="cuore08", Color="Cyan",
            Type="Organ", TissueName=None, RbeCellTypeName=None, RoiMaterial=None)

            retval_2.SetMarginExpression(SourceRoiName="Cuore", MarginSettings={ 'Type':
            "Expand", 'Superior': 0.8, 'Inferior': 0.8, 'Anterior': 0.8, 'Posterior': 0.8,
            'Right': 0.8, 'Left': 0.8 })

            retval_2.UpdateDerivedGeometry(Examination=exam, Algorithm="Auto")

            exam=case.Examinations[0].Name
            patient.Save() #salva il paziente
```

```

# Generazione delle ROI modificate per tutti i pazienti indicati in id_list. Valido per la ROI del
midollo

from connect import *
import sys

id_list=[] # scrivere id pazienti interessati

for id in id_list:
    patient_db = get_current("PatientDB")
    id_p = id
    info = patient_db.QueryPatientInfo(Filter=('PatientID':id_p))
    if len(info) == 1:
        patient = patient_db.LoadPatient(PatientInfo=info[0])
        patient = get_current('Patient')
        case_to_load = patient.Cases[0]
        case_to_load.SetCurrent()

        patient = get_current("Patient")
        case = get_current("Case")
        exam=get_current("Examination")

        IDp=patient.PatientID

        #creazione roi sbagliate:

        #midollo03

        retval_0 = case.PatientModel.CreateRoi(Name="midollo03", Color="Cyan", Type="Organ",
        TissueName=None, RbeCellTypeName=None, RoiMaterial=None)

        retval_0.SetMarginExpression(SourceRoiName="Midollo", MarginSettings={ 'Type': "Expand",
        'Superior': 0.3, 'Inferior': 0.3, 'Anterior': 0.3, 'Posterior': 0.3, 'Right': 0.3,
        'Left': 0.3 })

        retval_0.UpdateDerivedGeometry(Examination=exam, Algorithm="Auto")

        #midollo015

        retval_1 = case.PatientModel.CreateRoi(Name="midollo015", Color="Cyan", Type="Organ",
        TissueName=None, RbeCellTypeName=None, RoiMaterial=None)

        retval_1.SetMarginExpression(SourceRoiName="Midollo", MarginSettings={ 'Type': "Expand",
        'Superior': 0.15, 'Inferior': 0.15, 'Anterior': 0.15, 'Posterior': 0.15, 'Right': 0.15,
        'Left': 0.15 })

        retval_1.UpdateDerivedGeometry(Examination=exam, Algorithm="Auto")

        #midollo_spost

        retval_2 = case.PatientModel.CreateRoi(Name="midollo_spost", Color="Cyan", Type="Organ",
        TissueName=None, RbeCellTypeName=None, RoiMaterial=None)

        retval_2.SetMarginExpression(SourceRoiName="Midollo", MarginSettings={ 'Type': "Expand",
        'Superior': 0.0, 'Inferior': 0.0, 'Anterior': 0.0, 'Posterior': 0.0, 'Right': 0.0,
        'Left': 0.0 })

        retval_2.UpdateDerivedGeometry(Examination=exam, Algorithm="Auto")

        midollo_sp=case.PatientModel.RegionsOfInterest['midollo_spost']
        #la colonna Mx4 viene usato per la traslazione, tutte le altre scalano!
        midollo_sp.TransformROI3D(Examination=exam,TransformationMatrix= {'M11':1, 'M12':0,
        'M13':0, 'M14':0.5,
        'M21':0, 'M22':1, 'M23':0, 'M24':0.5,
        'M31':0, 'M32':0, 'M33':1, 'M34':0,
        'M41':0, 'M42':0, 'M43':0, 'M44':1})

        exam=case.Examinations[0].Name
        patient.Save() #salva il paziente

```

## Appendice 6. Script che esegue il training del classificatore

```
%% estrazione media, sigma, massimo e R^2 da tutti i pazienti presenti
%midollo
HU=(-303:3:1203)';
info= dir;
vett_wl=zeros(size(info,1)-3,2);
vett_rsq=zeros(size(info,1)-3,1);
vett_max=zeros(size(info,1)-3,1);
for j=1:(size(info,1)-3)
    filename=info(j+2).name;
    n_v = importSPHU(filename); % importa vettore con le frequenze dai file csv
    [modello,prop]=fit(HU,n_v,'gauss1');
    vett_wl(j,1)=modello.b1; %media sulla prima colonna e sigma sulla seconda
    vett_wl(j,2)=modello.c1;
    vett_rsq(j,1)=prop.rsquare;
    vett_max(j,1)=max(n_v);
end

%% estrazione media, sigma, massimo e R^2 da tutti i pazienti presenti
% cuore
HU=(-1002:3:147)';
info= dir;
vett_wl=zeros(size(info,1),2);
vett_rsq=zeros(size(info,1)-3,1);
vett_max=zeros(size(info,1)-3,1);
for j=1:(size(info,1)-3)
    filename=info(j+2).name;
    n_v = importfileHU_cuore(filename); % importa vettore con le frequenze dai file csv
    [modello,prop]=fit(HU,n_v,'gauss1');
    vett_wl(j,1)=modello.b1; %media sulla prima colonna e sigma sulla seconda
    vett_wl(j,2)=modello.c1;
    vett_rsq(j,1)=prop.rsquare;
    vett_max(j,1)=max(n_v);
end
```

## Appendice 7. Script che esegue la classificazione

```
%classificatore midollo
vett_max_g=vett_max;
vett_rsq_g=vett_rsq;
vett_g=vett_wl;
HU=(-303:3:1203)';
info= dir;
vett_res=zeros(size(info,1)-3,4);
for j=1:(size(info,1)-3)
    filename=info(j+2).name;
    n_v = importSPHU(filename); % importa vettore con le frequenze dai file csv
    [modello,prop]=fit(HU,n_v,'gauss1');
    c1=modello.c1;
    rsq=prop.rsquare;
    [mass,ind]=max(n_v);
    if (mass > prctile(vett_max_g,10)) && (mass < prctile(vett_max_g,85))
        vett_res(j,1)=1;
    else
        vett_res(j,1)=0;
    end
    if rsq > prctile(vett_rsq_g,25)
        vett_res(j,2)=1;
    else
        vett_res(j,2)=0;
    end
    if (c1 > prctile(vett_g(:,2),10)) && (c1 < prctile(vett_g(:,2),95))
        vett_res(j,3)=1;
    else
        vett_res(j,3)=0;
    end
    if vett_res(j,1) && vett_res(j,2) && vett_res(j,3)
        vett_res(j,4)=1;
    else
        vett_res(j,4)=0;
    end
end
end
```

```

%classificatore cuore
vett_max_g=vett_max;
vett_rsq_g=vett_rsq;
vett_g=vett_w1;
HU=(-1002:3:147)';
info= dir;
vett_res=zeros(size(info,1)-3,3);
for j=1:(size(info,1)-2)
    filename=info(j+2).name;
    n_v = importSPHU(filename); % importa vettore con le frequenze dai file csv
    [modello,prop]=fit(HU,n_v,'gauss1');
    cl=modello.cl;
    rsq=prop.rsquare;
    [mass,ind]=max(n_v);
    if (mass > prctile(vett_max_g,35)) && (mass < prctile(vett_max_g,95))
        vett_res(j,1)=1;
    else
        vett_res(j,1)=0;
    end
    if rsq > prctile(vett_rsq_g,35)
        vett_res(j,2)=1;
    else
        vett_res(j,2)=0;
    end

    if vett_res(j,1) && vett_res(j,2)
        vett_res(j,3)=1;
    else
        vett_res(j,3)=0;
    end
end
end

```