

Dipartimento Informatica – Scienza e Ingegneria (DISI)

Corso di Laurea in Ingegneria e Scienze Informatiche

**Segmentazione di singole cellule all'interno di ROI in
immagini di provini istologici attraverso l'utilizzo del
software open-source *QuPath***

Tesi in: Programmazione

Relatore

Prof.ssa Antonella Carbonaro

Presentata da

Filippo Pilutti

Correlatori

Dott. Filippo Piccinini

Dott.ssa Marcella Tazzari

Prof. Gastone Castellani

*“Anche un viaggio di mille miglia
inizia con un singolo passo.”*

Lao Tzu

Parole Chiave

Histology

Microscopy

Single-cell analysis

Open-source software

QuPath

Abstract (English version)

In the field of histological analysis, one of the typical issues is the analysis of single cells contained in regions of interest (*i.e.*, ROIs). Today, several commercial, freely-available, and open-source software options are accessible for this task and have revolutionized the way researchers scrutinize cellular structures and tissues, providing the opportunity to extract quantitative data. In this work, we reviewed and compared the tools currently available for performing single-cell analysis within a histological image ROI. The primary aim of this investigation is to review the different functionalities offered by these software solutions. In particular, we focused the analysis on three issues: (*a*) possibility of analyzing large SVS, DICOM and TIFF files; (*b*) opportunities for automatically extracting or manually defining squared, circular, polygonal and freehand ROIs; (*c*) opportunities to automatically/semi-automatically/manually classify single-cells within a selected ROI. Thanks to this work, a deeper understanding of the strengths and limitations of different software platforms emerges facilitating informed decision-making for researchers looking for a tool to analyze histological images.

Abstract (versione italiana)

Nel campo dell'analisi istologica, uno dei principali ostacoli è rappresentato dall'analisi delle singole cellule all'interno di specifiche regioni di interesse, note come ROI (Regioni di Interesse). Attualmente, esistono diverse soluzioni software per affrontare questa sfida, che spaziano da prodotti commerciali a software gratuiti e open-source. Questi strumenti hanno rivoluzionato il modo in cui i ricercatori esaminano le strutture cellulari e i tessuti, offrendo la possibilità di ottenere dati quantitativi fondamentali per la ricerca biomedica. Questa Tesi si propone di esaminare e confrontare gli strumenti disponibili per condurre l'analisi delle singole cellule all'interno di una ROI in un'immagine istologica. L'obiettivo primario di questa ricerca è esaminare le diverse funzionalità offerte da tali soluzioni software, concentrandosi su tre principali aspetti: (a) la capacità di analizzare file di grandi dimensioni nei formati SVS, DICOM e TIFF; (b) le modalità per estrarre automaticamente o definire manualmente ROI quadrate, circolari, poligonali e a mano libera; (c) le opzioni per classificare automaticamente, semi-automaticamente o manualmente le singole cellule all'interno di una ROI. Grazie a questa Tesi, emerge una comprensione più approfondita dei pregi e dei limiti delle varie piattaforme software, agevolando la scelta consapevole dei ricercatori che cercano uno strumento per l'analisi di immagini istologiche.

Indice

Introduzione.....	11
1 Analisi a singola cellula in immagini di provini istologici	13
2 Software per analisi a singola cellula	15
2.1 Software open-source.....	15
2.1.1 <i>CellProfiler</i>	15
2.1.2 <i>Cytomine</i>	17
2.1.3 <i>Digital Slide Archive</i>	19
2.1.4 <i>Icy</i>	20
2.1.5 <i>ImageJ/Fiji</i>	21
2.1.6 <i>QuPath</i>	23
2.2 Software freely-available	24
2.2.1 <i>Aperio ImageScope</i>	25
2.2.2 <i>NIS-Elements Viewer</i>	26
2.2.3 <i>Sedeen Viewer</i>	27
2.2.4 <i>SlideViewer</i>	28
2.3 Software commerciali.....	29
2.3.1 <i>Amira</i>	29
2.3.2 <i>Arivis</i>	31
2.3.3 <i>HALO</i>	32
2.3.4 <i>Imaris</i>	33
2.4 Tabelle comparative	35
3 QuPath e modalità di scrittura plugin	40
3.1 Requisiti di sistema.....	40
3.2 Installazione.....	40
3.2.1 Windows.....	40
3.2.2 macOS.....	40
3.2.3 Linux.....	41
3.3 Modalità d'uso	41
3.3.1 Esecuzione di <i>QuPath</i>	41
3.3.2 Finestra principale	42
3.3.3 Caricamento di un'immagine	43
3.3.4 Oggetti in <i>QuPath</i>	45
3.3.5 Disegnare annotazioni	45
3.3.6 Rilevamento cellule.....	47
3.3.7 Salvataggio	48
3.3.8 Gestione dei progetti	48
3.4 Scripting.....	49
3.4.1 Workflow	49
3.4.2 Dai workflow agli script.....	50
3.4.3 <i>Groovy</i>	51
3.4.4 Script personalizzati	52

3.5 Estensioni.....	52
3.5.1 Creazione di un'estensione.....	52
4 Algoritmi in QuPath per la Segmentazione di singole cellule all'interno di ROI.....	56
4.1 Segmentazione automatica.....	56
4.1.1 Segmentazione <i>Watershed</i>	56
4.1.2 Stain separation.....	57
4.1.3 Stima dei vettori di colorazione.....	58
4.1.4 Segmentazione cellulare.....	62
4.1.5 Misurazioni.....	63
4.2 Segmentazione semi-automatica.....	65
4.3 Segmentazione manuale.....	65
5 Analisi quantitativa sperimentale.....	67
5.1 Descrizione caso di studio.....	67
5.2 Descrizione provino istologico.....	68
5.3 Descrizione ground truth.....	69
5.3.1 Ground truth nuclei totali.....	70
5.3.2 Ground truth con classificazione marcatori.....	70
5.4 Conteggio totale: risultati sperimentali.....	71
5.4.1 Tool idonei.....	71
5.4.2 <i>Amira</i>	72
5.4.3 <i>Arivis</i>	73
5.4.4 <i>CellProfiler</i>	73
5.4.5 <i>HALO</i>	74
5.4.6 <i>ImageJ/Fiji</i>	75
5.4.7 <i>NIS-Elements</i>	76
5.4.8 <i>QuPath</i>	77
5.4.9 Discussione dei risultati.....	78
5.5 Conteggio con classificazione dei marcatori: risultati sperimentali.....	78
5.5.1 Tool idonei.....	78
5.5.2 <i>Amira</i>	80
5.5.3 <i>Arivis</i>	81
5.5.4 <i>HALO</i>	82
5.5.5 <i>ImageJ/Fiji</i>	82
5.5.6 <i>NIS-Elements</i>	83
5.5.7 <i>QuPath</i>	84
5.5.8 Discussione dei risultati.....	85
6 Conclusioni e sviluppi futuri.....	87
Bibliografia.....	90
Ringraziamenti.....	92

Introduzione

Questo progetto di Tesi è stato sviluppato all'interno del gruppo di ricerca "*Data Science for Health*" (DS4H), composto da ricercatori e professori dell'Università di Bologna (UniBo) e dell'*Istituto Romagnolo dei Tumori Dino Amadori* (IRST) di Meldola (FC). Il gruppo DS4H si propone di coordinare professionisti e risorse provenienti dai settori dell'informatica, dell'ingegneria dell'informazione, della fisica, nonché dalle discipline biomediche, della biologia, della chimica e della medicina. In particolare, il progetto di Tesi intitolato "*Segmentazione di ROI ed analisi di singole cellule tumorali in immagini di provini istologici attraverso l'utilizzo del software open-source QuPath*" nasce da una specifica necessità sorta nel contesto del gruppo di ricerca di approfondire il funzionamento e l'utilizzo dell'applicativo *QuPath* per l'analisi di immagini di provini istologici. L'obiettivo di questo progetto è fornire una disamina di questo software, con particolare interesse alle funzionalità a disposizione per lo studio di intese scansioni digitali ad alta definizione di provini istologici, andando anche a fare un confronto con gli altri strumenti a disposizione sul mercato.

Per avere chiaro il contesto operativo, è importante spiegare brevemente la metodologia di valutazione dei tumori nei pazienti da parte dei professionisti dell'IRST avendo a disposizione una biopsia del tessuto. Tutto parte dalla creazione di adeguati provini istologici dai campioni di tessuto prelevati dal paziente, a cui vengono applicati specifici marcatori utili a identificare le cellule con caratteristiche tumorali. Dopo uno strutturato processo di preparazione del provino istologico, si procede alla acquisizione di immagini digitali ad alta definizione, andando a scannerizzare per intero il provino attraverso l'utilizzo di microscopi, tipicamente degli slide scanner a campo largo. L'interesse principale di questo progetto ricade proprio nello studio completo ed efficiente di queste immagini di microscopia che arrivano a contenere decine di gigabytes di dati grafici bidimensionali. L'obiettivo finale, quindi, è quello di fornire ai Life Scientists (e.g. medici, biologi e biotecnologi) un quadro generale dei vari software a loro disposizione per l'analisi di queste immagini, per poi andare ad approfondire in maniera più dettagliata il programma *QuPath*.

QuPath è un software open-source progettato per l'analisi e la visualizzazione di immagini biomediche, in particolare immagini di microscopia. È ampiamente utilizzato in ambito scientifico e di ricerca, soprattutto nelle scienze della vita e nella patologia grazie alla sua interfaccia utente intuitiva che consente di esplorare, analizzare e interpretare dati con precisione e affidabilità. *QuPath* consente agli utenti di caricare e visualizzare immagini in diversi formati, compresi file di

grandi dimensioni come file SVS e DICOM. Offre strumenti per segmentare strutture e regioni di interesse (ROI), ad esempio aree tumorali contenenti cellule di uno specifico fenotipo o caratteristiche particolari. Inoltre, fornisce strumenti per l'analisi quantitativa delle immagini, consentendo agli utenti di misurare caratteristiche morfologiche, come dimensioni e forme delle strutture cellulari, nonché altre proprietà quantitative legate ad esempio alla cromaticità. Il software consente infine l'integrazione di algoritmi di machine learning per l'automazione di determinate attività, come la classificazione delle cellule o la rilevazione di determinati pattern all'interno dei campioni.

L'individuazione di questa necessità e la proposta di progetto sono state formulate dalla Prof.ssa Antonella Carbonaro e dal Dott. Filippo Piccinini, utilizzando come supporto e caso di studio immagini ottenute da campioni istologici di pazienti reali. Queste immagini (in versione anonimizzata) sono state fornite dalla Dott.ssa Marcella Tazzari, principal investigator della IRST "*Immunotherapy Cell Therapy and Biobank (ITCB)*" unit, e dal Prof. Gastone Castellani, direttore della Scuola di Specializzazione in Fisica Medica dell'Università di Bologna. Le immagini rappresentano delle scansioni ad alta risoluzione (40x) di interi campioni istologici colorati con marcatori chimici quali DAB e ematossilina e sono state catturate utilizzando uno scanner *Aperio CS2* di *Leica Biosystems*.

1 Analisi a singola cellula in immagini di provini istologici

L'istologia è la disciplina scientifica dedicata all'approfondimento della struttura morfologica e funzionale dei tessuti e delle cellule che li costituiscono. Tale studio si realizza attraverso l'analisi microscopica di sottili campioni tissutali fissati su vetro, definiti provini istologici. In campo medico l'analisi istologica svolge un ruolo fondamentale nella diagnosi e nel monitoraggio di un vasto numero di malattie, in particolare nella ricerca di cellule tumorali [1].

Preparazione di un provino istologico

Affinché un campione possa essere correttamente studiato al microscopio, esso va preparato e trattato adeguatamente. Questa procedura di preparazione prevede cinque fasi fondamentali: fissazione, disidratazione, inclusione, sezionamento, colorazione.

Fissazione

L'obiettivo del processo di fissazione è quello di prevenire o arrestare la degenerazione cellulare che comincia non appena il campione viene prelevato. Questo può essere ottenuto attraverso l'immersione del campione in una soluzione chimica fissativa (come ad esempio la formalina), o attraverso procedure fisiche come la crio-conservazione [2].

Disidratazione

Questo processo è necessario per eliminare l'acqua presente nel tessuto al fine di renderlo compatibile con il mezzo di inclusione. Sono previsti passaggi in serie del preparato in alcoli a gradazione progressivamente maggiore.

Inclusione

Questo procedimento serve a dare al tessuto una consistenza solida, rendendolo adatto al taglio. Ciò si realizza impregnando il campione di sostanze indurenti (ad esempio la paraffina).

Sezionamento

A questo punto può avvenire il sezionamento, ovvero il taglio del campione di tessuto in una sottile sezione. Per l'osservazione mediante microscopio ottico si utilizza un microtomo, che permette di

ottenere spessori della grandezza di pochi micrometri. Queste sezioni vengono poi collocate in particolari vetrini adatti all'analisi istologica.

Colorazione

Il processo di colorazione viene usato per evidenziare caratteristiche di particolare interesse nel tessuto, così come per differenziare tra loro gli elementi strutturali che lo compongono attraverso il loro colore e/o intensità di gradazione. Questo procedimento risulta fondamentale per semplificare l'individuazione e l'analisi di determinate cellule e proprietà che si vogliono studiare. La colorazione avviene impiegando prodotti chimici naturali o sintetici che, legandosi a particolari cellule o componenti del tessuto, li vanno a colorare aumentando il contrasto e consentendo la distinzione delle porzioni di interesse specifico [3]. Il processo di colorazione è un passaggio di particolare interesse nell'ambito di questo progetto di Tesi. Infatti, avendo un'immagine a contrasto elevato tra i vari componenti del campione risulta possibile realizzare algoritmi o procedure informatiche in grado di semplificare di molto il lavoro di analisi da parte del medico o del biologo. Si potrà quindi disporre di strumenti in grado di riconoscere automaticamente certi tipi di cellule, oppure algoritmi in grado di scartare le porzioni del campione di scarso interesse per il caso di studio specifico. In *Figura 1.1* è rappresentata una sezione di un provino istologico prelevato da un melanoma. Si nota che il processo di colorazione rende possibile l'individuazione ad occhio nudo dei diversi componenti del tessuto.

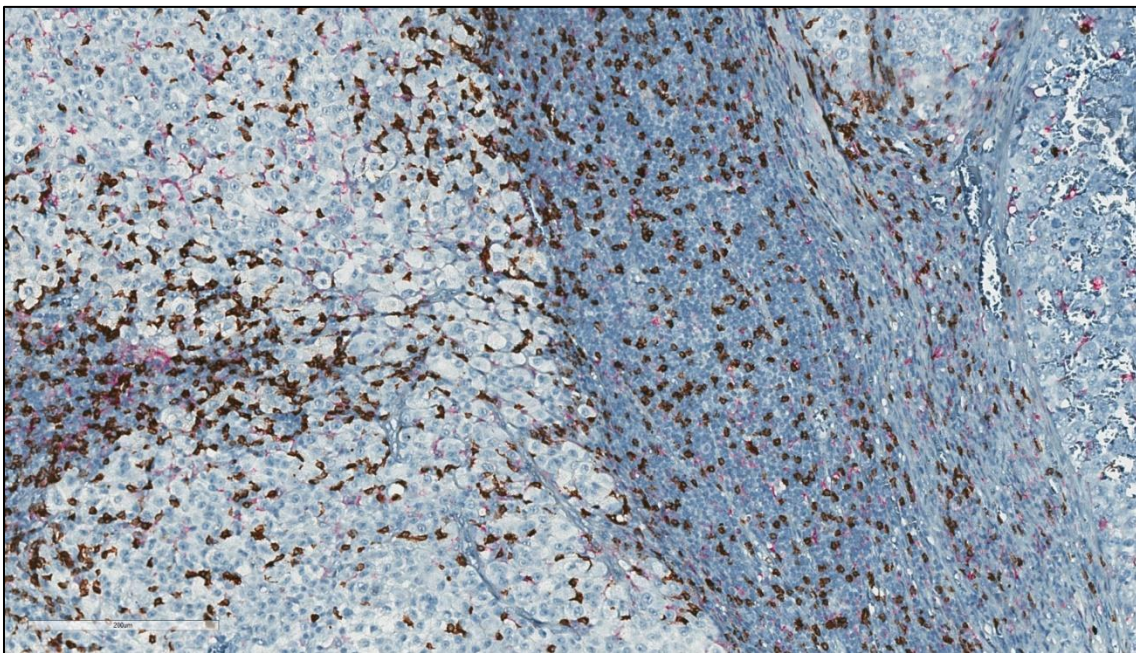


Figura 1.1 - Sezione di provino istologico proveniente da melanoma

2 Software per analisi a singola cellula

L'analisi a singola cellula di provini istologici rappresenta un campo cruciale nella ricerca biomedica. In questo contesto esistono numerosi software che stanno assumendo un ruolo fondamentale sia nella ricerca che nella diagnosi patologica, offrendo soluzioni accessibili e flessibili per l'analisi delle immagini istologiche. Alcune caratteristiche fondamentali di questi software sono la facilità di utilizzo anche da parte di utenti non esperti nel campo informatico, l'espandibilità (attraverso plugin o script) con algoritmi o procedure personalizzate, la gestione di diverse tipologie di immagini provenienti da campioni differenti. Andiamo ad esaminare le principali piattaforme software per l'analisi a singola cellula di provini istologici, che in base alla loro modalità di fruizione si suddividono in tre categorie: software open-source, software freely-available e software commerciali.

2.1 Software open-source

I software open-source sono caratterizzati dalla loro natura collaborativa e trasparente. In questo modello di sviluppo, il codice sorgente è reso pubblico ed accessibile a chiunque, consentendo ad una vasta comunità di sviluppatori di contribuire, migliorare e adattare il software alle proprie specifiche esigenze.

2.1.1 CellProfiler

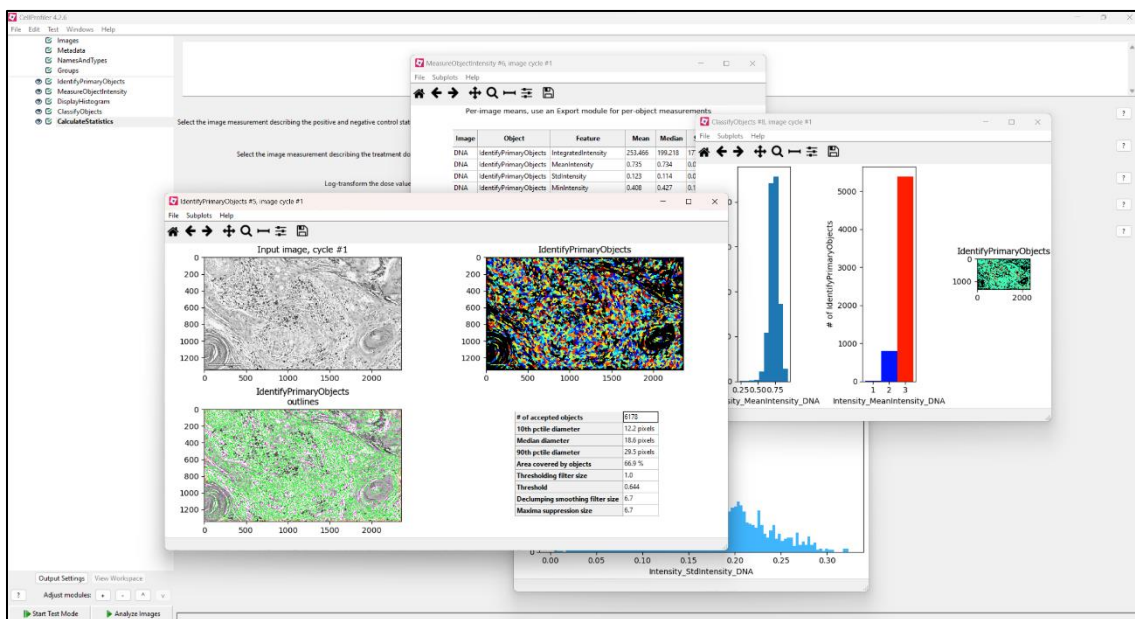


Figura 2.1 – Interfaccia utente di CellProfiler

CellProfiler è un programma open-source progettato dal Broad Institute. E' stato sviluppato originariamente in *MATLAB* e poi in *Python* per l'analisi ad alto rendimento di immagini cellulari. La capacità dei moderni microscopi di ottenere enormi quantità di immagini ad alta risoluzione in un lasso di tempo contenuto va a creare un collo di bottiglia nella fase di analisi. Questo software permette l'analisi automatizzata di migliaia di campioni cellulari in poco tempo, processo che risulterebbe impossibile attraverso il semplice studio ad occhio nudo: si definisce quindi come una analisi quantitativa, al contrario di quella ad occhio nudo che è qualitativa. A differenza degli altri strumenti utilizzati per l'analisi cellulare (come *ImageJ*), che sono più indirizzati verso lo studio dettagliato di immagini singole (come avviene per esempio in *PhotoShop*), *CellProfiler* si specializza nell'analisi automatizzata di grandi quantità di campioni. Tipicamente il flusso di lavoro su *CellProfiler* si svolge attraverso la creazione di pipeline di singoli moduli. Ogni modulo processa l'immagine in qualche modo, e i moduli sono applicati in ordine sequenziale per creare appunto una pipeline, solitamente in questo ordine: elaborazione dell'immagine, identificazione degli oggetti, misurazioni. La maggior parte dei moduli ha un funzionamento automatico, ma ne esistono anche di interattivi, che richiedono un qualche input da parte dell'utente. I vari moduli possono venire combinati e abbinati in base alle necessità dello specifico progetto; all'inizio del processo di analisi ogni immagine attraversa la pipeline e viene processata in ordine sequenziale da ogni modulo. *CellProfiler* fornisce un'ampia suite di moduli integrati che rendono possibile una vasta gamma di analisi differenti. Esiste ovviamente anche la possibilità di estendere il software andando a sviluppare ulteriori algoritmi da aggiungere poi come moduli extra [4].

Limitazioni principali

CellProfiler risulta essere un software molto potente, che presenta però alcune limitazioni, in particolare nel contesto di questo progetto di Tesi.

- **Interfaccia utente e usabilità:** l'interfaccia utente potrebbe non risultare molto user-friendly, in particolare per utenti non esperti in campo informatico. L'utilizzo del software al suo pieno potenziale richiede una certa curva di apprendimento e almeno una conoscenza basilica della programmazione per creare flussi di lavoro personalizzati.
- **Segmentazione di regioni:** la definizione di particolari regioni di interesse all'interno delle immagini può risultare un processo laborioso in quanto avviene in modo manuale, al contrario di *QuPath* che offre funzionalità di segmentazione automatiche.

- **Analisi quantitativa:** *CellProfiler* risulta più adatto per flussi di lavoro che coinvolgono un grande numero di immagini da analizzare; *QuPath* invece offre strumenti specializzati nell'analisi dettagliata di singole immagini di grandi dimensioni.
- **Aggiornamenti e comunità:** la comunità di *CellProfiler* non è così attiva come quella di *QuPath* e i tempi di rilascio di nuovi aggiornamenti non sono sempre regolari.
- **Dimensione dei file:** *CellProfiler* supporta nativamente solamente file di immagini di grandezza non superiore ai 2GB; quindi, per poter analizzare scansioni istologiche di grandi dimensioni queste andranno suddivise in immagini più piccole.

2.1.2 Cytomine

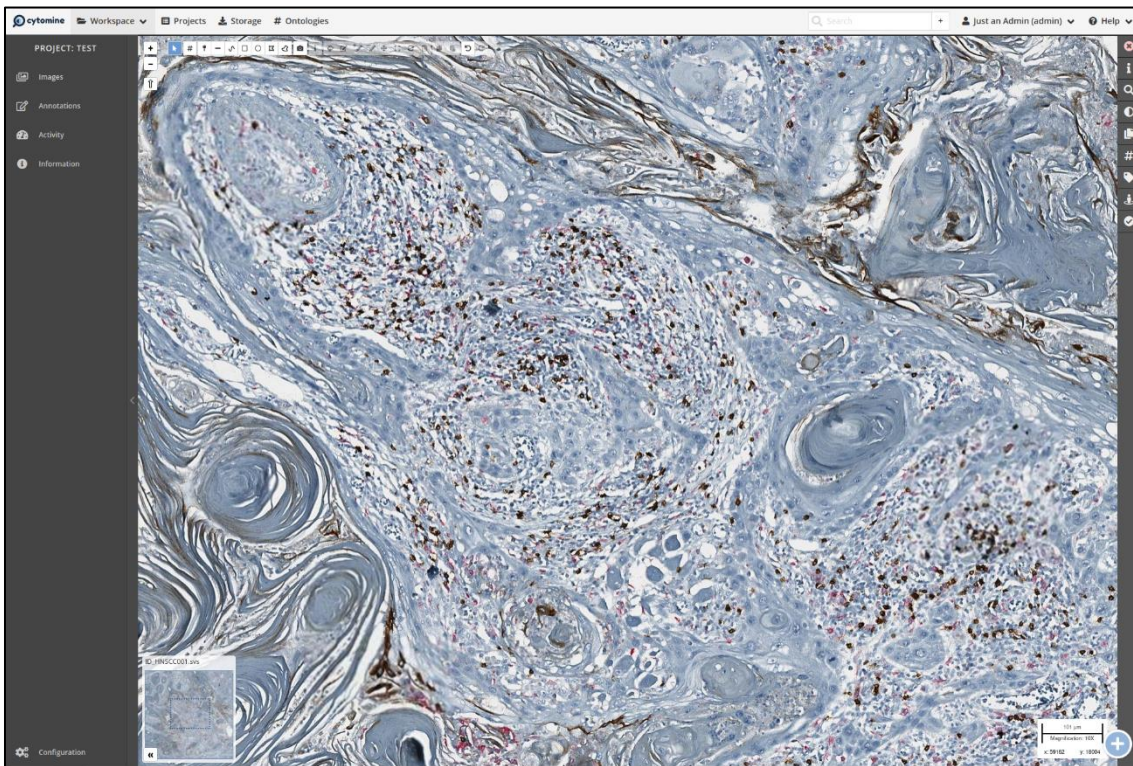


Figura 2.2 - Interfaccia utente di Cytomine

Cytomine è una piattaforma web open source progettata prestando particolare attenzione all'aspetto collaborativo dell'analisi di immagini istologiche che fornisce un ambiente flessibile e scalabile atto a facilitare la collaborazione tra esperti di settori diversi. *Cytomine* è stato sviluppato prevalentemente in Python e la sua prima versione risale al 2012; attualmente viene distribuito come piattaforma web composta da diversi servizi incapsulati in contenitori *Docker*. Un punto di forza di questo programma è il supporto nativo a numerosi formati di file anche di grandi dimensioni (multi-gigapixel), inclusi i principali file di patologia digitale e di microscopia (ad esempio TIFF). Il software supporta inoltre l'annotazione collaborativa delle immagini,

consentendo a più utenti di annotare e marcare sezioni o regioni di interesse e rende semplice la condivisione dei propri progetti attraverso un'interfaccia web che permette anche di seguire le operazioni di altri utenti in tempo reale. Ulteriori features offerte da *Cytomine* sono l'integrabilità di algoritmi di apprendimento automatico e la possibilità di estensione con nuovi algoritmi e plugins. La piattaforma offre anche un servizio cloud a pagamento per il mantenimento dei progetti e delle immagini. *Cytomine* è disponibile come applicazione web (accessibile dal link: <https://open.cytomine.com/#/>) utilizzando un account demo fornito dagli sviluppatori. In alternativa è possibile richiedere delle credenziali personali per accedere a tutte le funzionalità, oppure installare un'istanza locale di *Cytomine* (disponibile solo su sistemi operativi *Linux*) [5].

Limitazioni principali

- **Algoritmi di analisi:** la versione *Community Edition* di *Cytomine* non mette a disposizione algoritmi o funzionalità di analisi avanzate integrati, ma andranno implementati autonomamente o aggiunti con estensioni.
- **Segmentazione automatica:** *Cytomine* non offre funzionalità di segmentazione automatica di ROI o di cellule. Le immagini possono venire annotate solo manualmente.
- **Complessità di configurazione:** l'installazione in locale e la successiva configurazione di *Cytomine* può risultare un processo lungo e complesso per gli utenti meno esperti, in quanto si tratta di un software rilasciato sotto forma di servizi *Docker*.
- **Usabilità:** in particolare per quanto riguarda il caricamento di immagini e la gestione dei progetti, *Cytomine* non risulta essere il più user-friendly tra i software considerati.

2.1.3 Digital Slide Archive

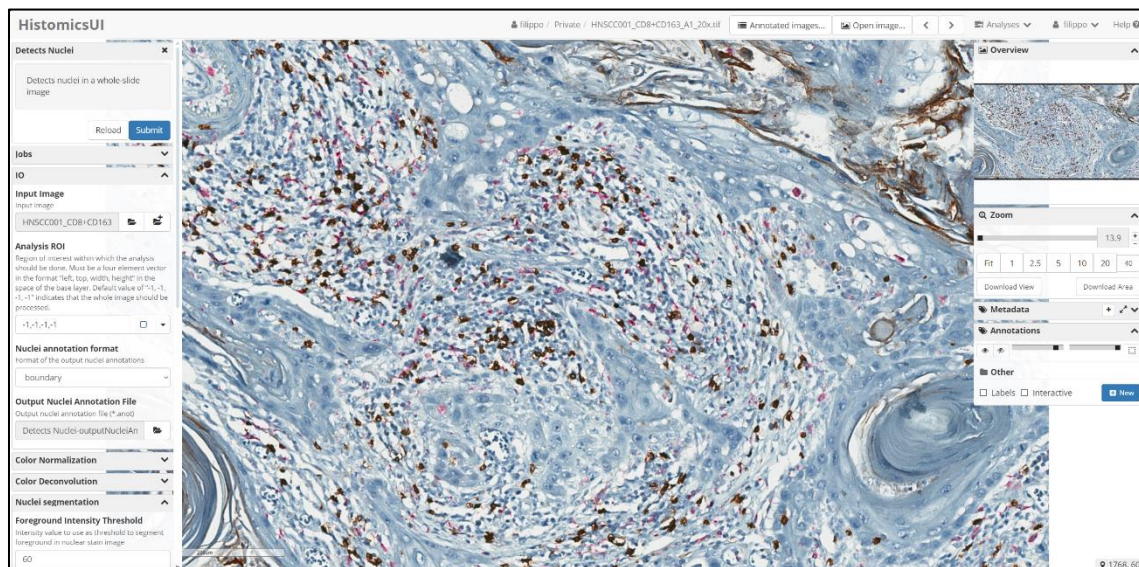


Figura 2.3 – Interfaccia utente di Digital Slide Archive (HistomicsUI)

Digital Slide Archive è una piattaforma progettata per l'archiviazione, la gestione, la visualizzazione e l'analisi di ampie raccolte di dati di immagini. Questa piattaforma si compone di un toolkit di analisi (*HistomicsTK*), un'interfaccia utente per la visualizzazione delle immagini e la loro annotazione (*HistomicsUI*), uno strato Database basato su *MongoDB* e un server web che offre API avanzate e un sistema di gestione dei dati (attraverso l'utilizzo di *Girder*). *Digital Slide Archive* consente di organizzare un gran numero di immagini provenienti da diversi set di risorse, implementare controlli di accesso per utenti diversi, annotare e revisionare le immagini, eseguire algoritmi di analisi sui campioni. *HistomicsTK* è il pacchetto di strumenti di analisi integrato in questa piattaforma: si tratta di un package *Python* utilizzabile anche come libreria stand-alone. Gli algoritmi messi a disposizione da *HistomicsTK* offrono funzionalità di analisi avanzata (come segmentazione di ROI, analisi di texture, misurazioni quantitative), di elaborazione delle immagini (come normalizzazione dell'illuminazione, correzione dei colori) e di apprendimento automatico. L'architettura software di *Digital Slide Archive* è basata su diversi servizi (e.g *Girder*, *MongoDB*) distribuiti attraverso container *Docker* [6].

Limitazioni principali

- **Complessità d'installazione:** l'installazione in locale di *Digital Slide Archive* risulta un processo complicato e laborioso per gli utenti non esperti di sistemi operativi *Linux*.
- **Usabilità:** l'interfaccia utente *HistomicsUI* e l'utilizzo della piattaforma *Digital Slide Archive* potrebbero risultare complessi per gli utenti meno esperti.

- **Requisiti hardware e software:** l'installazione locale di *Digital Slide Archive* è possibile solo su sistemi operativi *Linux* e *MacOS*. L'esecuzione del programma e di algoritmi di analisi particolarmente esigenti potrebbe richiedere un'elevata potenza di calcolo.

2.1.4 Icy

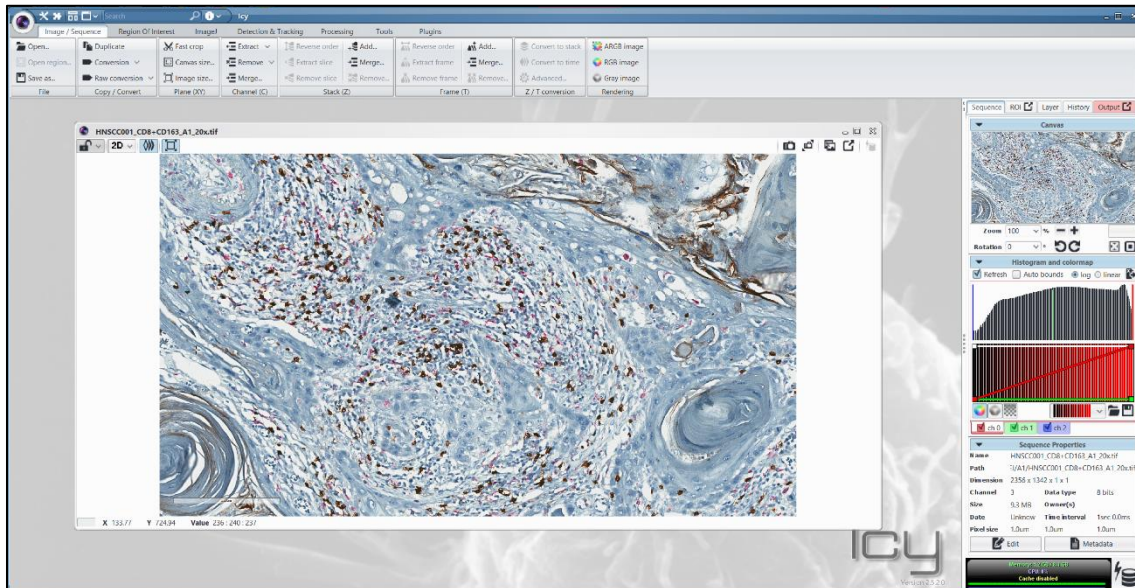


Figura 2.4 - Interfaccia utente di Icy

Icy è un software open source multiplatforma che offre una suite di funzionalità e plugin per visualizzare, annotare e analizzare immagini di microscopia digitale, sviluppato in linguaggio *Java* a partire dal 2011 presso l'*Institut Pasteur* di Parigi. *Icy* risulta essere altamente estensibile, sia attraverso l'implementazione di plugin che attraverso l'editor di script integrato che permette la scrittura di script in *Javascript* o *Python*. La community di utenti di *Icy* è molto attiva, infatti esistono numerosi plugin molto validi per automatizzare e semplificare l'analisi a singola cellula delle immagini, come ad esempio *Active Contours* per la segmentazione automatica dei nuclei cellulari. Sono presenti, inoltre, numerose funzionalità per la visualizzazione e lo studio di dati in tre dimensioni. Uno degli aspetti più interessanti di questo software è il plugin *Protocols*, che fornisce un ambiente grafico per la creazione di flussi di lavoro di analisi delle immagini in modo completamente grafico e senza alcuna conoscenza richiesta di programmazione, e la possibilità di automatizzarli su molte immagini. Attraverso il sito web di *Icy* è anche possibile caricare i propri protocolli e scaricare quelli creati da altri utenti [7].

Limitazioni principali

- **Dimensione dei file:** *Icy* non supporta il caricamento di immagini di grosse dimensioni (>2GB); per essere visualizzate esse andranno scomposte o tagliate.
- **Segmentazione di regioni:** rispetto ad altri programmi, in *Icy* gli strumenti per la segmentazione manuale di ROI sono limitati.
- **Apprendimento automatico:** attualmente *Icy* non supporta ancora l'integrazione con algoritmi o sistemi di apprendimento automatico per la classificazione automatica di regioni e cellule.

2.1.5 ImageJ/Fiji

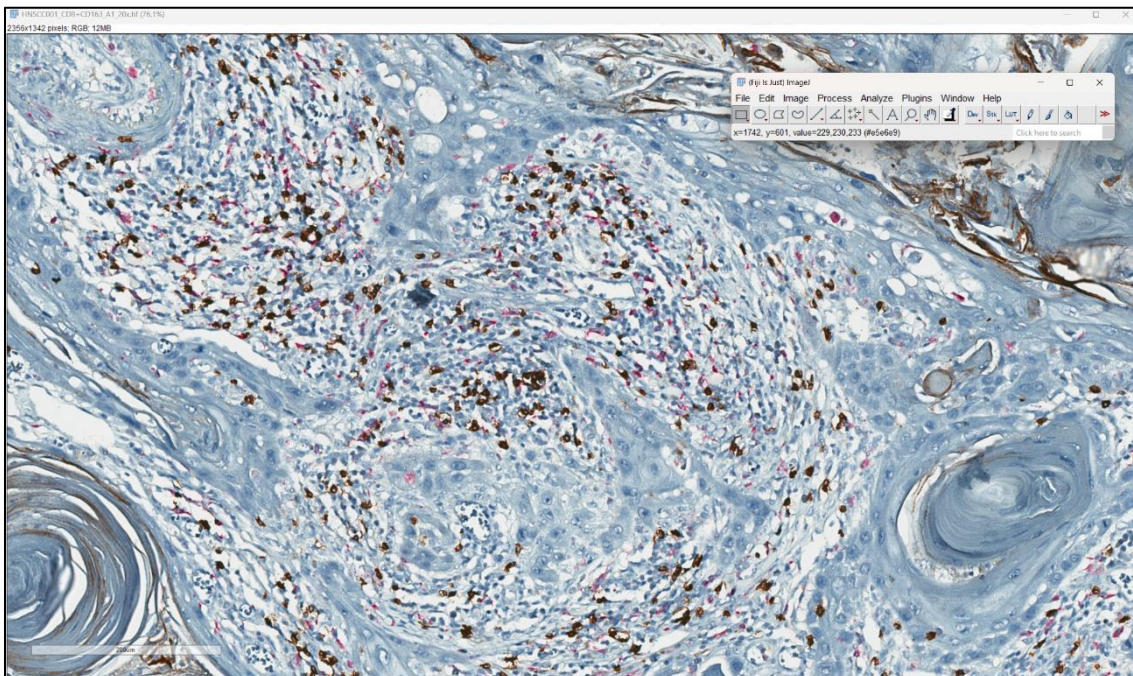


Figura 2.5 - Interfaccia utente di ImageJ/Fiji

ImageJ è un programma per l'analisi di immagini che negli ultimi anni è divenuto uno standard di fatto nell'ambito scientifico. Grazie alla sua facilità di utilizzo e alla sua architettura estendibile a plugin, attorno a questo software si è creata una grande community di utenti e contributori che spaziano dalla biologia alle scienze fisiche. *ImageJ* offre funzionalità per l'annotazione manuale di regioni all'interno dell'immagine, sia attraverso strumenti di disegno con forme prestabilite (e.g. cerchio, ovale, rettangolo) sia a mano libera [5] [6]. *Fiji* (acronimo di: "*Fiji Is Just ImageJ*") è una distribuzione di *ImageJ* specializzata nell'analisi di immagini biologiche, che nasce come una piattaforma per la collaborazione tra le comunità di ricerca in informatica e biologia. A tal fine adotta moderne pratiche di ingegneria del software per unire potenti librerie software a un'ampia

gamma di linguaggi di scripting, fornendo un ambiente per il rapido sviluppo di nuovi algoritmi di analisi di immagini. *Fiji* fornisce inoltre un sistema integrato di aggiornamenti per facilitare la condivisione dei nuovi plugin per *ImageJ* con gli utenti finali. La distribuzione fornisce inoltre una grande quantità di plugin già inclusi e pronti all'utilizzo e questo l'ha resa recentemente molto popolare all'interno della comunità scientifica. In particolare, il plugin *Bio-formats* rende possibile il caricamento di immagini dei principali formati di microscopio digitale. La distribuzione *ImageJ/Fiji*, quindi, consiste in una suite di plugin che possono essere aggiunti per svolgere una grande quantità di operazioni diverse. Il plugin *Particle Analysis* è utile per l'analisi a singola cellula, però richiede diversi passaggi di elaborazione dell'immagine che potrebbero risultare complessi per gli utenti non esperti. Per prima cosa, l'immagine andrà convertita in un'immagine binaria in bianco e nero, impostando un adeguato threshold per la distinzione delle cellule dallo sfondo con una buona precisione, successivamente bisognerà effettuare una segmentazione (e.g. watershed) per individuare tutte le singole cellule e infine sarà possibile eseguire il plugin per la misurazione e l'analisi delle cellule [8] [9].

Limitazioni principali

Nonostante *ImageJ* sia di fatto uno standard per l'analisi di immagini in contesto scientifico, presenta alcune limitazioni nell'ambito della patologia digitale.

- **Interfaccia utente e usabilità:** l'interfaccia utente di *ImageJ* risulta più generica e potrebbe richiedere un periodo di apprendimento medio/lungo per gli utenti meno esperti, mentre quella di *QuPath* è stata progettata specificatamente per l'analisi patologica.
- **Segmentazione di regioni:** attraverso plugin aggiuntivi e personalizzazioni *ImageJ* può ottenere risultati simili a quelli di *QuPath* nella segmentazione automatica di regioni, ma potrebbe non essere altrettanto efficiente nel riconoscimento di strutture istologiche complesse. *QuPath* offre inoltre l'integrazione di algoritmi di apprendimento automatico.
- **Integrazione con database e gestione dei dati:** *ImageJ* richiede soluzioni personalizzate per la connessione a database di patologia e non supporta nativamente immagini di dimensioni molto grandi.
- **Ottimizzazione per l'analisi patologica:** *QuPath* è stato sviluppato con un focus specifico sull'analisi patologica e offre quindi una soluzione pronta all'utilizzo senza bisogno di personalizzazioni o aggiunte. *ImageJ* richiede configurazioni maggiori per adattarsi completamente ai flussi di lavoro tipici della patologia, operazioni che per gli utenti meno esperti potrebbero risultare complesse.

- **Dimensione dei file:** *ImageJ/Fiji* non offre il supporto nativo a file di dimensioni maggiori di 2GB.
- **Analisi a singola cellula:** il flusso di lavoro per la segmentazione e l'analisi di singole cellule richiede diversi passaggi e risulta lungo e complesso.

2.1.6 *QuPath*

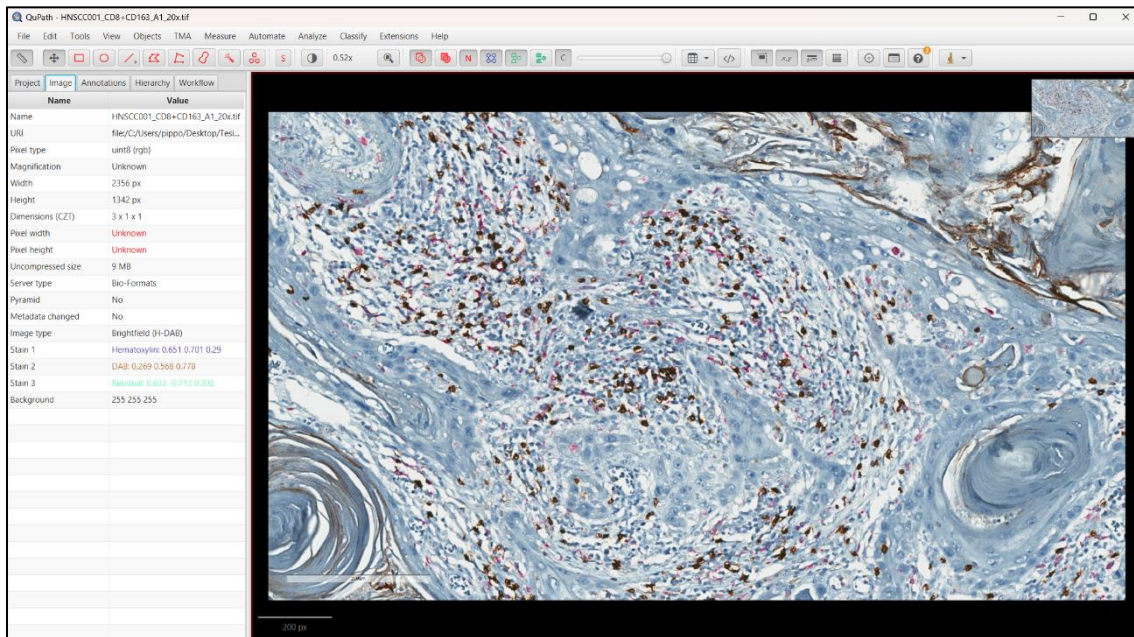


Figura 2.6 - Interfaccia utente di *QuPath*

QuPath offre una soluzione user-friendly, estensibile e open-source per la patologia digitale e l'analisi di immagini intere di sezioni istologiche, con una particolare attenzione al campo biomedico. Il software offre un ampio pannello di strumenti per l'identificazione dei tumori e la valutazione ad alto rendimento dei biomarcatori e fornisce ai ricercatori potenti funzionalità di elaborazione batch e scripting, nonché una piattaforma estensibile con cui sviluppare e condividere nuovi algoritmi per analizzare immagini tissutali complesse. *QuPath* nasce dalla necessità di gestire l'analisi di immagini di intere sezioni istologiche, che grazie alle moderne tecnologie di microscopia digitale possono arrivare a contenere fino a 40 GB di dati 2D. Al suo nucleo, *QuPath* è un visualizzatore di immagini di sezioni istologiche multiplatforma e multi-threaded, che incorpora estese funzionalità di annotazione e visualizzazione. Oltre a questo, offre anche una serie di algoritmi innovativi per fornire non solo soluzioni pronte per i problemi più comuni dell'analisi patologica, ma anche i mattoni di base per creare flussi di lavoro personalizzati e la possibilità di collegarli per l'elaborazione batch attraverso potenti funzionalità di scripting. Infine, *QuPath* permette agli sviluppatori di estendere il software a proprio piacimento per risolvere nuove sfide e

applicazioni, consentendo anche lo scambio di dati efficiente con altri strumenti quali *ImageJ* o *MATLAB*, che altrimenti fornirebbero un supporto limitato per l'analisi di immagini intere di grandi dimensioni. Caratteristica chiave alla base della funzionalità di *QuPath*, che rappresenta anche un importante fattore tecnico distintivo rispetto agli altri software precedentemente citati, è il suo modello dei dati gerarchico e orientato agli oggetti. Qui, un oggetto rappresenta principalmente una struttura o una regione all'interno dell'immagine, che può essere generata e manipolata sia attraverso tool interattivi (ad esempio per annotare una regione di interesse) sia attraverso comandi di segmentazione automatizzati (ad esempio per individuare singoli nuclei o cellule). Esistono inoltre anche altri tipi di oggetti, che supportano l'assegnazione di classificazioni, misurazioni e collegamenti ad altri oggetti 'genitori' e 'figli' con un sistema gerarchico facilmente interrogabile e manipolabile tramite comandi integrati o scripting. Questo modello generico rende possibile la rappresentazione e visualizzazione delle relazioni tra una grande quantità di oggetti in maniera efficiente ed intuitiva e supporta inoltre l'addestramento di classificatori di oggetti mediante tecniche di apprendimento automatico [10].

Limitazioni principali

- **Scripting e flussi di lavoro:** sebbene sia possibile convertire un workflow direttamente in uno script, sono richieste almeno delle conoscenze di base di programmazione per la personalizzazione e modifica degli script.
- **Requisiti hardware:** visto che in *QuPath* è possibile caricare immagini di grandi dimensioni, per l'esecuzione degli algoritmi di calcolo più complessi sono richieste discrete potenza di calcolo e memoria disponibile.
- **Curva di apprendimento:** per sfruttare a pieno anche le potenzialità più avanzate di *QuPath* è richiesta una certa curva di apprendimento.

2.2 Software *freely-available*

I software *freely-available* sono software accessibili ed utilizzabili gratuitamente, ma che non seguono il modello open-source; quindi, il loro codice sorgente non è liberamente accessibile. I software analizzati in questa sezione sono accomunati dal fatto di essere gratuiti, ma presentano tool e funzionalità disponibili solo a pagamento. In genere, gli sviluppatori di queste applicazioni rilasciano gratuitamente il visualizzatore di immagini istologiche, solitamente con limitate funzioni di annotazione, mentre gli strumenti di analisi e segmentazione sono accessibili e utilizzabili solo acquistando una licenza commerciale.

2.2.1 Aperio ImageScope

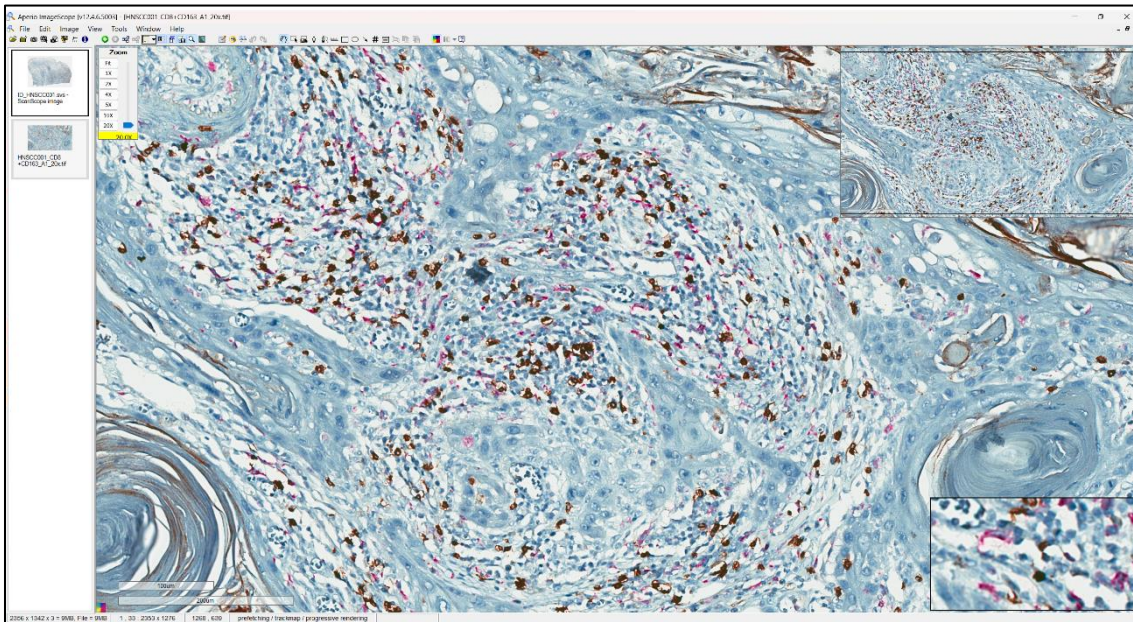


Figura 2.7 - Interfaccia utente di Aperio ImageScope

Aperio ImageScope è un software facente parte della piattaforma *Aperio eSlide Manager* sviluppata da *Leica Biosystems* per la gestione di immagini istologiche. *ImageScope* rappresenta lo strumento per la visualizzazione e l'analisi delle immagini ed è disponibile gratuitamente per il download dal sito di *Leica Biosystems*. Questo software mette a disposizione svariate funzionalità di annotazione per evidenziare, misurare o annotare aree specifiche nelle immagini e la possibilità di eseguire algoritmi di analisi quantitativa e qualitativa sulle regioni annotate. Tra le sue features principali troviamo: funzionalità per la collaborazione e condivisione dei progetti di analisi, strumenti di elaborazione delle immagini avanzata, integrabilità con altri sistemi e software in ambito medico, supporto multi-piattaforma, funzionalità di videoconferenza *eSlide* remota, strumenti di crittografia e restrizioni di accesso per la sicurezza dei dati [11].

Limitazioni principali

- **Annotazione manuale:** *ImageScope* mette a disposizione unicamente strumenti per l'annotazione manuale di ROI all'interno dell'immagine.
- **Algoritmi di analisi:** *ImageScope* non offre algoritmi integrati per l'analisi istologica, ma essi andranno acquistati a parte o aggiunti autonomamente.
- **Segmentazione cellulare:** il software non offre una funzione per la segmentazione automatica di cellule all'interno dell'immagine.

- **Software freely-available:** non trattandosi di un software open-source, non esiste la possibilità di modifica delle funzionalità o estensione attraverso plugin.
- **Supporto formato DICOM:** *ImageScope* non supporta l'apertura e la visualizzazione di file di formato DICOM.

2.2.2 NIS-Elements Viewer

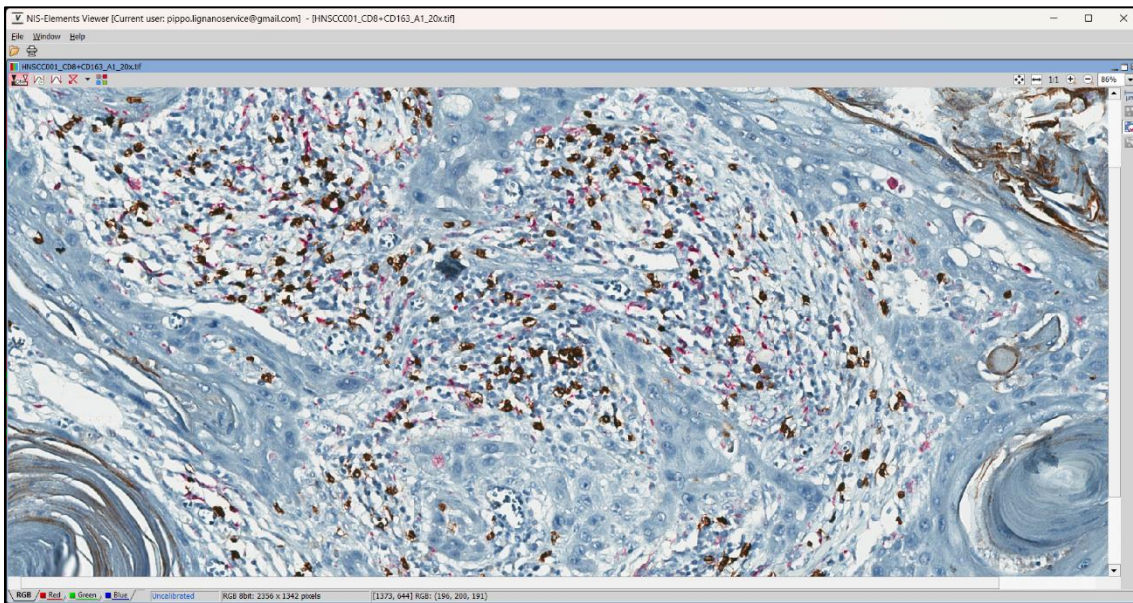


Figura 2.8 - Interfaccia utente di NIS-Elements Viewer

NIS-Elements Viewer è un programma standalone gratuito sviluppato e distribuito da *Nikon* per visualizzare immagini e dataset di microscopia; offre le stesse modalità di visualizzazione e selezione delle immagini messi a disposizione dal software commerciale *NIS-Elements*, come ad esempio visualizzazione dei volumi con rendering 3D, visualizzazione in sequenza di dati temporali e visualizzazione di immagini stratificate. *NIS-Elements Viewer*, quindi, offre solo funzionalità di visualizzazione delle immagini; per quanto riguarda tutti gli strumenti di annotazione dei campioni e analisi dei dati, essi sono contenuti nel pacchetto core di *NIS-Elements* [12].

Limitazioni principali

- **Funzionalità limitate:** *NIS-Elements Viewer* offre unicamente funzionalità di visualizzazione di immagini.
- **Formato file supportati:** il visualizzatore supporta solamente il caricamento di immagini di formato ND2, TIFF e JPEG, mentre altri (e.g. SVS) non sono supportati.

2.2.3 Sedeen Viewer

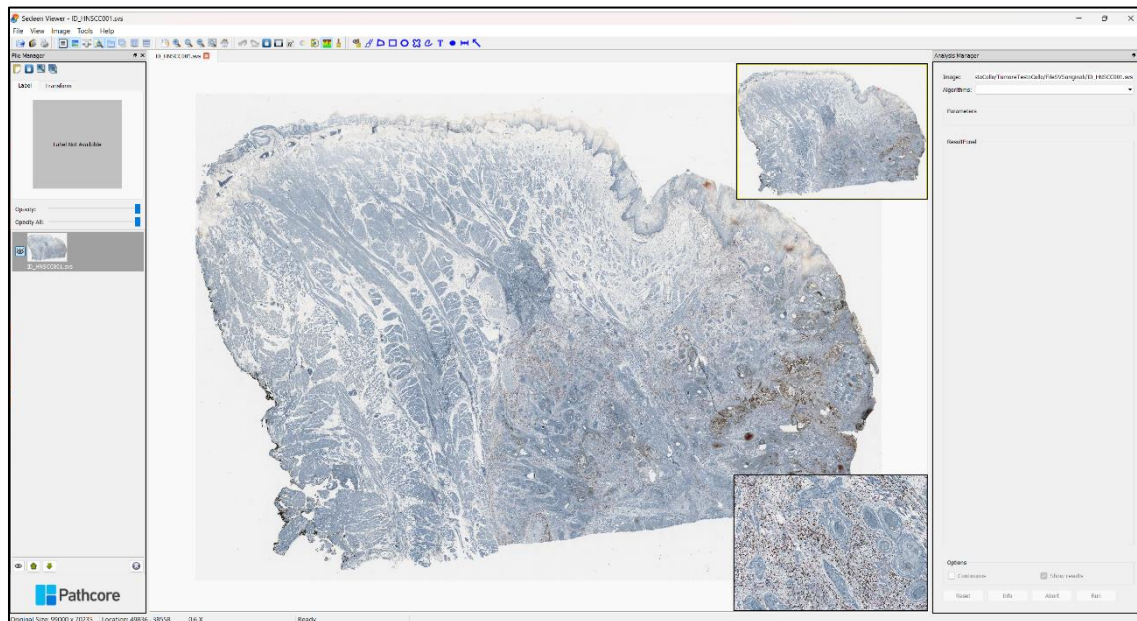


Figura 2.9 - Interfaccia utente di Sedeen Viewer

Sedeen Viewer è un visualizzatore di immagini di campioni istologici sviluppato da *Pathcore*, che offre supporto per diversi formati di immagini istologiche di grandi dimensioni, tra cui *Aperio SVS*, *TIFF* e *JPEG-2000*. *Sedeen* fornisce una vasta gamma di strumenti di annotazione, tra cui un tool di disegno a mano libera e forme poligonali. È inoltre possibile esportare le regioni annotate in formato *XML* per consentirne l'apertura con altri programmi. Il software fornisce anche funzionalità per l'elaborazione delle immagini, come correzione della gamma dei colori e applicazione di pseudo-colori alle immagini in scala di grigi. *Sedeen* integra alcuni limitati strumenti di analisi, come ad esempio algoritmi di *thresholding* e di analisi dei marcatori colorati; è però possibile sviluppare e aggiungere i propri algoritmi di analisi attraverso *Pathcore SDK* [13].

Limitazioni principali

- **Analisi limitata:** sebbene integri algoritmi di analisi, le operazioni eseguibili sono poche e limitate e non è possibile effettuare flussi di analisi complessi.

2.2.4 SlideViewer

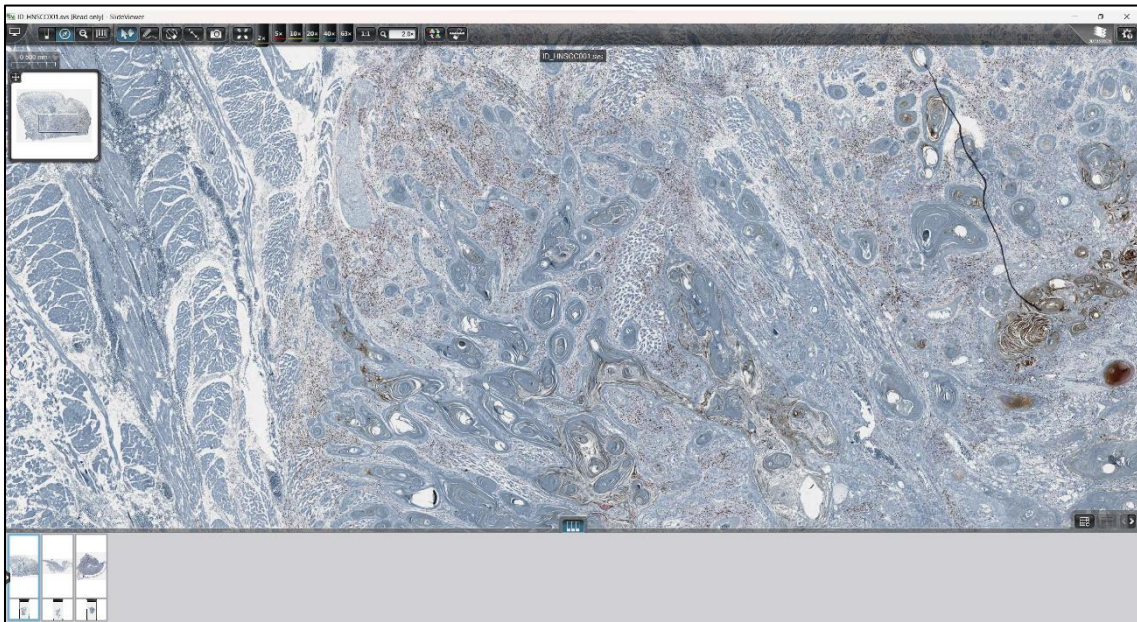


Figura 2.10 - Interfaccia utente di SlideViewer

SlideViewer (precedentemente conosciuto come *CaseViewer*) è un software per la visualizzazione di immagini istologiche sviluppato da *3DHISTECH* e rilasciato gratuitamente con la possibilità di aggiungere funzionalità aggiuntive a pagamento. Nella sua versione base, il software è un visualizzatore di immagini dei principali formati di istologia, con strumenti per l'annotazione di regioni e la misurazione di determinate aree; per eventuali funzionalità di analisi avanzate è richiesta una licenza commerciale. Tra le features dell'applicazione troviamo la capacità *MultiView*, che permette il caricamento di più slide contemporaneamente, la visualizzazione dei vari livelli di stack delle slide e l'anteprima delle immagini prima della loro effettiva apertura. In *SlideViewer* è possibile effettuare la classificazione manuale dei nuclei cellulari tramite il plugin gratuito *Marker Counter*. Questo strumento consente di definire diverse categorie di classificazione e di selezionare manualmente le cellule appartenenti a ciascuna categoria. In seguito, è possibile estrarre misurazioni e dati statistici. Questo software supporta i formati di immagini istologiche più diffusi (e.g. SVS, DICOM, TIFF), però unicamente per la visualizzazione in modalità read-only; per poter essere annotate o analizzate con *Marker Counter*, le immagini dovranno essere prima convertite in formato MRXS utilizzando un convertitore fornito da *3DHISTECH* [14].

Limitazioni principali

- **Funzionalità limitate:** la versione base di *CaseViewer* mette a disposizione funzionalità limitate, in particolare per quanto riguarda l'analisi a singola cellula, i cui strumenti sono disponibili solo con una licenza commerciale.
- **Segmentazione cellulare:** non è presente uno strumento per la segmentazione automatica di cellule all'interno dell'immagine, ma solo una funzionalità per la classificazione manuale.
- **Estensibilità:** trattandosi di un software non open-source, non esistono possibilità per integrare il codice sorgente o sviluppare estensioni.
- **Supporto formati:** i formati SVS, DICOM e TIFF sono supportati solo in modalità read-only.

2.3 Software commerciali

2.3.1 Amira

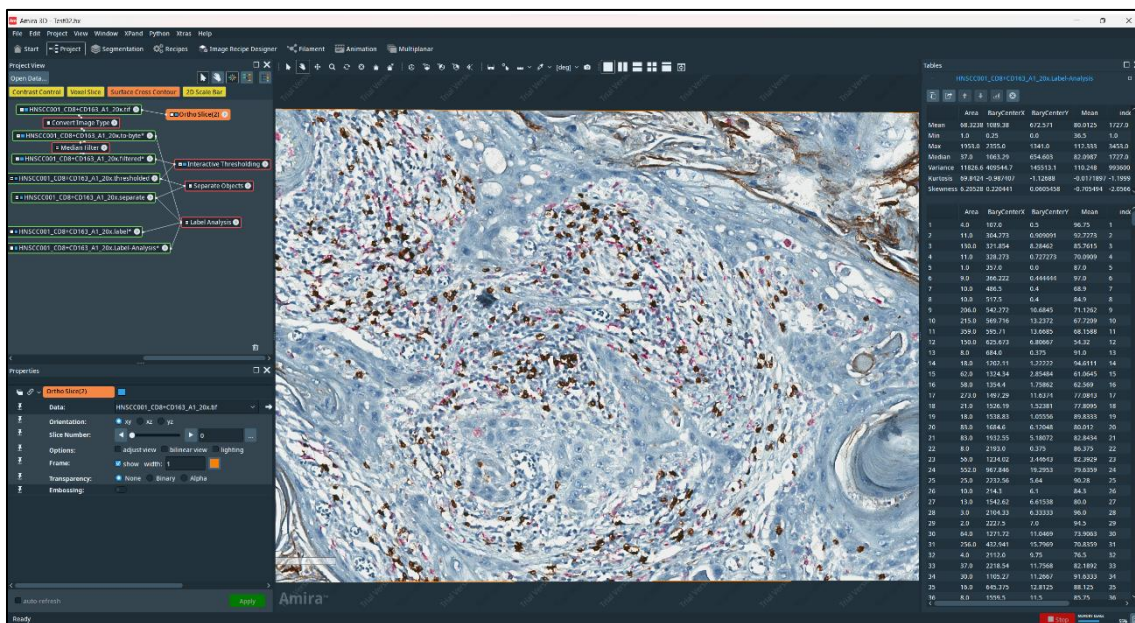


Figura 2.11 - Interfaccia utente di Amira

Amira è una soluzione software sviluppata dalla *ThermoFisher Scientific* per la visualizzazione, l'analisi e la comprensione di immagini nel campo della life science e della ricerca biomedica, provenienti da molteplici modalità di imaging (e.g. microscopia ottica ed elettronica). *Amira* offre potenti strumenti e funzionalità per la gestione di dati multidimensionali complessi, permettendo agli utenti di visualizzare e manipolare dati provenienti da immagini 2D, 3D, 4D e 5D. Gli strumenti di analisi a disposizione consentono agli utenti di estrarre informazioni quantitative e

qualitative dai loro dati. Questi strumenti includono filtri per l'elaborazione dell'immagine, algoritmi avanzati di segmentazione per identificare e isolare regioni di interesse, strumenti di misurazione per quantificare le dimensioni e le proprietà delle strutture, e molto altro ancora. Le opportunità di analisi offerte da questo software sono veramente numerose, però gli utenti alle prime armi potrebbero sentirsi sopraffatti dall'ampia gamma di strumenti e dalla loro complessità; comunque, l'interfaccia utente integra numerosi tutorial e guide, che coprono dalle operazioni di base fino ai workflow più complessi. Per la gestione dei flussi di lavoro, *Amira* utilizza un ambiente grafico per la definizione di ogni singola operazione con i suoi relativi parametri; non sono richieste quindi conoscenze di programmazione pregresse. Per quanto riguarda la segmentazione cellulare, *Amira* offre algoritmi sia automatici che interattivi, specializzati principalmente per l'applicazione su dati multidimensionali. Sono inclusi anche dei moduli per la classificazione automatica attraverso moduli di deep learning [15].

Limitazioni principali

- **Supporto formato SVS:** *Amira* non supporta nativamente il formato dei file SVS: per poter essere visualizzati, essi andranno prima convertiti in un altro formato (e.g. TIFF).
- **Annotazione di ROI:** il procedimento per l'annotazione manuale di regioni di interesse risulta essere più complesso e meno intuitivo rispetto ad altri tool.
- **Gestione di grandi dataset:** l'analisi di data set molto grandi può risultare inefficiente e richiedere molto tempo. Potrebbe essere richiesto di scomporre immagini di grandi dimensioni in sezioni più piccole.
- **Costo:** *Amira* può essere molto costoso, in particolar modo per istituzioni o utenti con budget limitati. In base alle features richieste, il prezzo del software varia tra i 4000 e i 9000 dollari.

2.3.2 Arivis

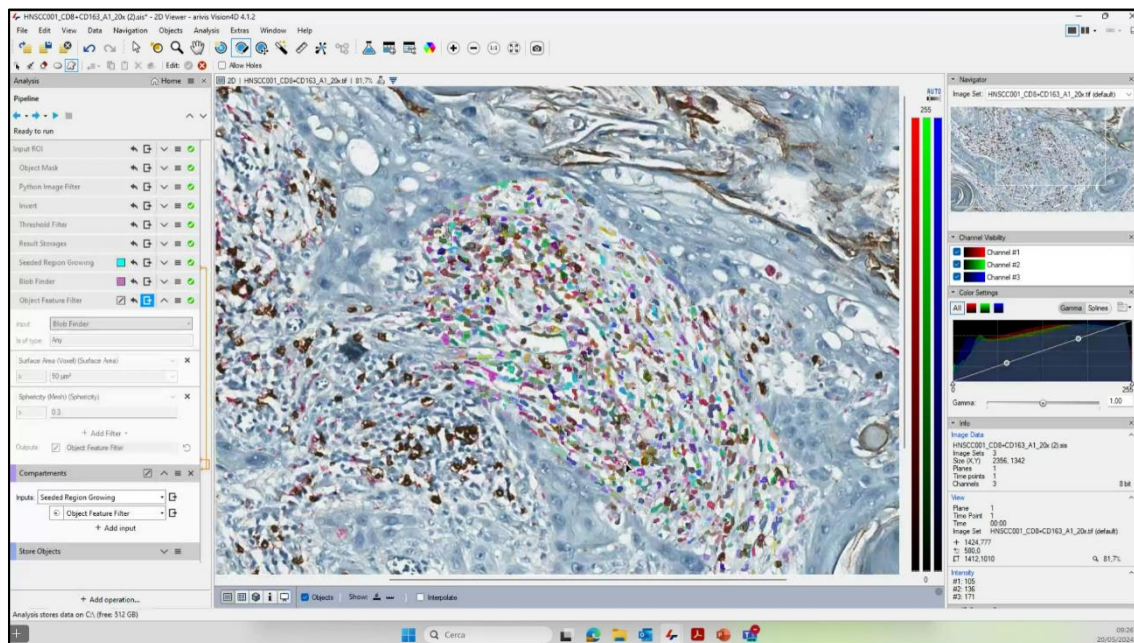


Figura 2.12 - Interfaccia utente di Arivis

Arivis Pro è un software modulare per l'analisi di immagini scientifiche sviluppato e distribuito da *Zeiss*. Permette di lavorare con immagini multicanale 2D, 3D e 4D di dimensioni virtualmente illimitate, in quanto i dataset vengono caricati nella memoria RAM in maniera dinamica in base ai dati che l'utente visualizza. Il software fornisce un'interfaccia utente flessibile e intuitiva per creare flussi di lavoro personalizzati per i casi di studio più specifici, oppure pipeline predefinite per casi d'uso comuni. *Arivis* offre un'infrastruttura altamente scalabile, non dipendente dalle risorse di sistema locali, e consente di elaborare e quantificare qualsiasi tipo di dati di immagine multidimensionale. È possibile eseguire analisi complesse e sfruttare funzionalità per l'elaborazione delle immagini e per la segmentazione automatica utilizzando modelli di machine learning. Esiste anche la piattaforma *Arivis Cloud* (precedentemente conosciuta come *Apeer*), un'infrastruttura web che consente di sfruttare strumenti di analisi delle immagini tramite intelligenza artificiale grazie al toolkit *Zeiss Arivis AI*, senza bisogno di installazioni o scrittura di codice [15].

Limitazioni principali

- **Costo:** una licenza commerciale per l'utilizzo di *Arivis* risulta essere troppo costosa per un utente singolo.
- **Analisi:** il flusso di lavoro per la creazione di analisi complesse risulta essere meno intuitivo agli utenti non esperti rispetto ad altri software.

2.3.3 HALO

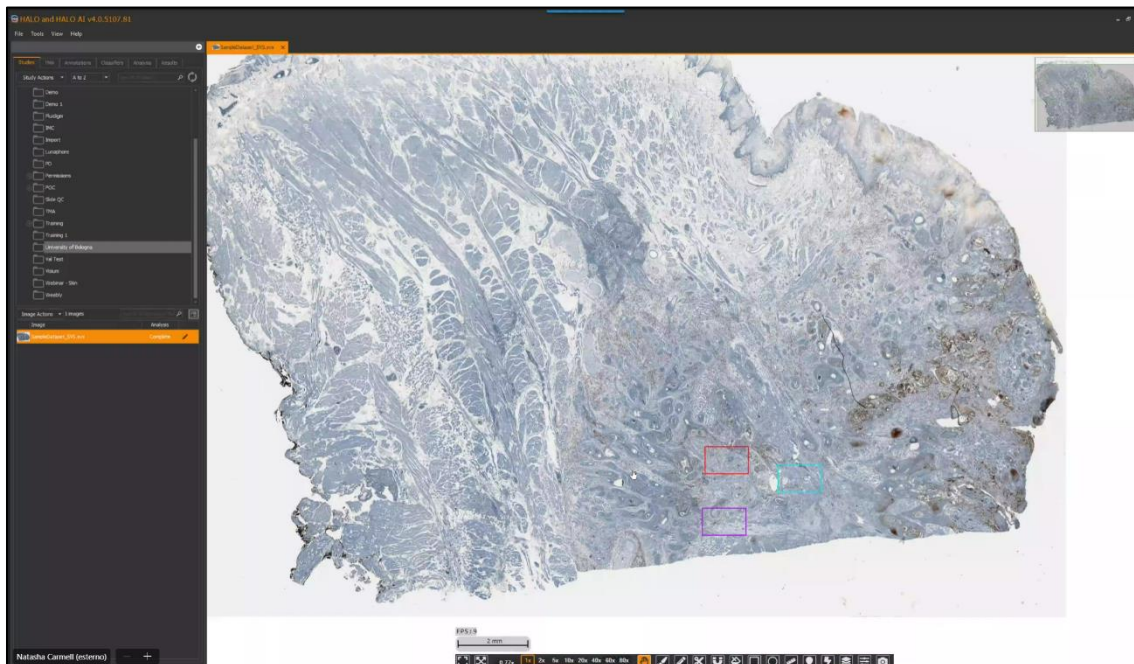


Figura 2.13 – Interfaccia utente di HALO

HALO è una piattaforma software per l'analisi avanzata di immagini istologiche e per la patologia digitale, progettata da *Indica Labs* per supportare la ricerca in campo biomedico. *HALO* offre un'interfaccia intuitiva e facile da usare anche per gli utenti meno esperti che permette l'esecuzione di analisi senza il bisogno di creare algoritmi da zero o impostare flussi di lavoro complessi. L'architettura dell'applicazione si basa su moduli progettati appositamente per le diverse necessità di analisi dei campioni, garantendo così flessibilità, scalabilità e prestazioni ottimali. È possibile utilizzare reti di deep learning pre-addestrate per la segmentazione automatica delle cellule in immagini sia a campo chiaro che a fluorescenza, oppure è possibile addestrare i propri modelli di intelligenza artificiale attraverso il modulo *HALO AI*. Il modulo *Multiplex IHC* si rivela essere la scelta più idonea per il caso di studio di questa Tesi. Consente di analizzare simultaneamente fino a cinque colorazioni di marcatori diversi all'interno delle strutture cellulari in immagini a campo chiaro. Per ottimizzare l'analisi, è inoltre possibile integrare il modulo *HALO AI* per la segmentazione cellulare. *HALO* semplifica le operazioni di analisi offrendo una vasta gamma di strumenti di annotazione, tra cui un tool di annotazione a mano libera e funzionalità di intelligenza artificiale per annotare simultaneamente regioni appartenenti alla stessa categoria. Durante il processo di analisi, è possibile ottimizzare in tempo reale i parametri di analisi sfruttando la funzionalità di real-time tuning di *HALO*: questa permette di selezionare una piccola area dell'immagine all'interno della quale verrà mostrato un feedback in diretta sui parametri di analisi. Una volta completata l'analisi, *HALO* offre diverse funzionalità per la visualizzazione,

l'esportazione e il salvataggio dei dati e delle misurazioni ottenute, sia a livello statistico generale che per le singole cellule. È inoltre possibile salvare le impostazioni di analisi per eseguire un'analisi di batch su una grande quantità di immagini in modo semplice e veloce. *HALO* supporta la maggior parte dei formati di immagini più diffusi, sia proprietari che standard, e gestisce anche dati di dimensioni molto grandi (>100GB). Tuttavia, è importante notare che il software non dispone di sistemi di ottimizzazione per il caricamento delle immagini, quindi per lavorare con immagini di dimensioni molto grandi è consigliabile utilizzare una macchina con specifiche hardware adeguate. *HALO* non integra nativamente un sistema per l'estendibilità con algoritmi o funzionalità personalizzate, è però possibile richiedere l'aggiunta di estensioni specifiche contattando il team di sviluppo [16].

Limitazioni principali

- **Curva di apprendimento:** *HALO* richiede un periodo di apprendimento per poter essere utilizzato a pieno dai nuovi utenti.
- **Costo:** una licenza biennale di *HALO* che include tre moduli di analisi e *HALO AI* viene a costare 37,500 euro.
- **Richiesta di risorse:** *HALO* può richiedere significative risorse hardware e software per funzionare in modo ottimale

2.3.4 Imaris



Figura 2.14 - Interfaccia utente di ImarisViewer

Imaris è una soluzione software supportata dall'intelligenza artificiale, progettata per l'analisi interattiva di immagini di microscopia. Sviluppato da *Bitplane AG*, una società del gruppo *Oxford Instruments*, questo strumento si concentra principalmente sulla gestione di dati in tre e quattro dimensioni, ma fornisce anche limitate funzionalità per l'analisi di immagini bidimensionali. *Imaris* consente l'esecuzione di analisi quantitative complesse, quali la segmentazione di varie strutture biologiche, la misurazione delle proprietà cellulari e il tracciamento delle particelle. Diversi pacchetti di *Imaris* sono disponibili, ciascuno con caratteristiche specifiche progettate per soddisfare le esigenze riscontrate in diverse discipline delle life sciences; in particolare, le funzionalità per la segmentazione e l'analisi cellulare sono contenute nel package *Imaris for Cell Biologists*, che fornisce anche un interfacciamento bidirezionale da *Imaris* a linguaggi di programmazione classici (*Matlab, Java, Python*) per l'integrazione delle funzionalità attraverso algoritmi personalizzati. Tuttavia, poiché questo software si focalizza prevalentemente sull'imaging in tre e quattro dimensioni, le sue capacità di analisi di immagini istologiche in due dimensioni sono limitate. Una delle poche funzionalità è il tool di segmentazione cellulare tramite machine learning, il quale richiede un certo livello di training e quindi non può venire automatizzato direttamente. *Imaris* è un software estremamente sofisticato e ricco di strumenti che coprono una vasta gamma di necessità delle life sciences, che però per la semplice segmentazione cellulare in regioni di interesse di immagini di provini istologici risulta essere fin troppo avanzato [17].

Limitazioni principali

- **Annotazione di ROI:** non è possibile annotare manualmente regioni di interesse specifiche, per analizzare porzioni specifiche di un'immagine essa andrà tagliata o ridimensionata
- **Costo:** per una licenza permanente varia tra i 14,000 e i 49,000 dollari, in base ai pacchetti e alle funzionalità incluse.

2.4 Tabelle comparative

Tabella 1 - Comparazione dei software open source per l'analisi a singola cellula di immagini istologiche (X = si, O = no)

	<i>CellProfiler</i>	<i>Cytomine</i>	<i>Digital Slide Archive</i>	<i>Icy</i>	<i>ImageJ/Fiji</i>	<i>QuPath</i>
VERSIONE						
Prima release	2005	2010	2017	2011	1997	2016
Versione attuale	4.2.6	3.2.3	1.1.0	2.5.2	1.54	0.5.0
DOCUMENTAZIONE						
Linguaggio	<i>Python</i>	<i>Python, Java</i>	<i>Python</i>	<i>Java</i>	<i>Java</i>	<i>Java</i>
Guida utente	X	X	X	X	X	X
Sito web	X	X	X	X	X	X
Open source	X	X	X	X	X	X
USABILITA'						
SVS/DICOM/TIFF	X/X/X	X/X/X	X/O/X	X/X/X	X/X/X	X/X/X
Supporto dati 3D	X (plugin)	O	O	X	X (plugin)	O
Programmazione richiesta	X	O	O	O	O	O
GUI user-friendly	O	X	X	X	X	X
Multiplatforma	X	X	O	X	X	X
FUNZIONALITA'						
Importazione di ROI	X (maschera)	O	O	X (.xml)	X (maschera)	X (GeoJSON)
Esportazione di ROI	X (maschera)	O	O	X (.xml)	X (maschera)	X (GeoJSON)
Segmentazione automatica	X	X	X	X (plugin)	X (plugin)	X
Segmentazione manuale	X	X	X	X	X	X
Segmentazione semiautomatica	X	X	X	X	X	X
Classificazione automatica	X	X	O	X (plugin)	X (plugin)	X
Classificazione manuale	X	X	X	X	X	X
Estensibilità	X	X	X	X	X	X

Tabella 2 - Comparazione dei software freely-available per l'analisi a singola cellula di immagini istologiche (X = sì, O = no)

	<i>Apertio ImageScope</i>	<i>NIS-Elements Viewer</i>	<i>Sedeen Viewer</i>	<i>SlideViewer</i>
VERSIONE				
Prima release	2011	2007	2013	2012
Versione attuale	12.4.6	5.21.00	5.4.4	2.7
DOCUMENTAZIONE				
Linguaggio	C++	C++	//	C++
Guida utente	X	X	X	X
Sito web	X	X	X	X
Open source	O	O	O	O
USABILITA'				
SVS/DICOM/TIFF	X/O/X	O/O/X	X/O/X	X/X/X
Supporto dati 3D	O	X	O	X
Programmazione richiesta	O	O	O	O
GUI user-friendly	X	X	X	X
Multiplatforma	Windows	X	Windows	X
FUNZIONALITÀ				
Importazione di ROI	O	O	O	O
Esportazione di ROI	O	O	O	O
Segmentazione automatica	O	O	O	O
Segmentazione manuale	X	O	X	X
Segmentazione semiautomatica	O	O	O	O
Classificazione automatica	O	O	O	O
Classificazione manuale	O	O	O	X
Estensibilità	O	O	X	O

Tabella 3 - Comparazione dei software commerciali per l'analisi a singola cellula di immagini istologiche (X = sì, O = no)

	<i>Amira</i>	<i>Arvis Pro</i>	<i>HALO</i>	<i>Imaris</i>
VERSIONE				
Prima release	1999	2002	2012	1993
Versione attuale	2023.2	4.1.2	4.0	10.1
DOCUMENTAZIONE				
Linguaggio	//	//	C++	//
Guida utente	X	X	X	X
Sito web	X	X	X	X
Open source	O	O	O	O
USABILITA'				
SVS/DICOM/TIFF	O/X/X	X/X/X	X/X/X	X/X/X
Supporto dati 3D	X	X	O	X
Programmazione richiesta	O	O	O	O
GUI user-friendly	X	X	X	X
Multiplatforma	X	Windows	X	X
FUNZIONALITÀ				
Importazione di ROI	X (maschera)	X (maschera)	X	X
Esportazione di ROI	X (maschera)	X (maschera)	X	X
Segmentazione automatica	X	X	X	X
Segmentazione manuale	O	X	X	O
Segmentazione semiautomatica	X	O	X	O
Classificazione automatica	X	X	X	X
Classificazione manuale	O	O	X	O
Estensibilità	O	X (<i>Python, MATLAB</i>)	O	X

Tabella 4 - Software per l'analisi a singola cellula di immagini istologiche: link per il download

Software open-source	
<i>CellProfiler</i>	https://cellprofiler.org/releases
<i>Cytomine</i>	https://github.com/cytomine/Cytomine-community-edition/releases/tag/CE2023.1
<i>Digital Slide Archive</i>	https://github.com/DigitalSlideArchive/digital_slide_archive/blob/master/devops/dsa/
<i>Icy</i>	icy.bioimageanalysis.org/download/
<i>ImageJ/Fiji</i>	https://imagej.net/software/fiji/downloads
<i>QuPath</i>	https://qupath.github.io/
Software freely-available	
<i>Aperio ImageScope</i>	https://www.leicabiosystems.com/it-it/digital-pathology/gestione/aperio-imagescope/
<i>NIS Elements Viewer</i>	https://www.microscope.healthcare.nikon.com/it_EU/products/software/nis-elements/viewer
<i>Sedeen Viewer</i>	https://pathcore.com/sedeen
<i>SlideViewer</i>	https://www.3dhistech.com/research/software-downloads/

Tabella 5 - Software commerciali per l'analisi a singola cellula di immagini istologiche: link ufficiali

Software commerciali	
<i>Amira</i>	https://www.thermofisher.com/it/en/home/electron-microscopy/products/software-em-3d-vis/amira-software/cell-biology.html
<i>Arivis Pro</i>	https://www.arivis.com
<i>HALO</i>	https://indicalab.com/halo/
<i>Imaris</i>	https://imaris.oxinst.com

3 *QuPath* e modalità di scrittura plugin

Dopo aver esaminato la letteratura per comprendere quali software per l'analisi di immagini istologiche sono ad oggi a disposizione sul mercato, andiamo ad approfondire più nel dettaglio l'applicativo *QuPath*. La scelta di voler approfondire proprio questa piattaforma si deve al fatto che ad oggi è probabilmente la soluzione open-source più utilizzata da Life Scientists, ed uno degli scopi di questa Tesi vuole essere proprio quello di fornire un testo unico che possa aiutare a migliorare le loro capacità di utilizzo di questo tool.

3.1 Requisiti di sistema

- **Sistema Operativo:** *QuPath* é compatibile con Windows, Linux e macOS (OS X 10.7.4 o successivi).
- **Memoria RAM:** Almeno 8 GB.
- **Spazio su disco:** Almeno 500 MB di spazio libero.

3.2 Installazione

Visita il sito ufficiale di *QuPath* all'indirizzo <https://qupath.github.io/> e, nella sezione download, scegli la versione appropriata in base al tuo sistema operativo.

3.2.1 Windows

Su sistema operativo Windows, ci sono due possibili opzioni per l'installazione:

- **Windows installer (msi)** per un'installazione tradizionale sul dispositivo, così che l'applicazione appaia nel menu Start.
- **Windows portable (exe)** per scaricare direttamente un file eseguibile, senza necessità di ulteriori installazioni.

3.2.2 macOS

Anche su sistema operativo macOS abbiamo due opzioni a disposizione:

- **macOS Intel (pkg)** funziona su tutti i dispositivi mac più recenti, sia con processore Intel che con processore Apple silicon. In caso di dubbio, questa è l'opzione standard da scegliere.
- **macOS Apple silicon (pkg)** funziona solo sui dispositivi mac più recenti con processori Apple silicon (ad esempio processori M1, M2 o M3). Il grande vantaggio è che funziona in modo sostanzialmente più veloce ed efficiente rispetto alla versione per Intel. Lo svantaggio è che i nuovi processori Apple silicon non sono ancora supportati dalla libreria *Bio-formats*, quindi alcuni tipi di file (come le immagini .czi da Zeiss Axioscans) non verranno aperti correttamente.

3.2.3 Linux

Su sistema operativo Linux sarà sufficiente scaricare ed estrarre il file .tar.xz. Sarà poi necessario usare il comando “`chmod u+x /path/to/QuPath/bin/QuPath`” per rendere l'applicazione eseguibile.

3.3 Modalità d'uso

Di seguito vengono illustrate le funzionalità base dell'applicazione. Le immagini dell'interfaccia utente mostrate in questa sezione sono state catturate utilizzando un dispositivo con sistema operativo Windows 11.

3.3.1 Esecuzione di *QuPath*

Eseguendo l'applicazione, verrà presentata la schermata di benvenuto. Qui sono presenti link utili per la documentazione, il forum della community e la pagina *GitHub* del software. È inoltre possibile cambiare il tema dell'interfaccia utente (chiaro o scuro) e verificare la presenza di aggiornamenti. Una volta pronti per l'utilizzo del software, sarà sufficiente premere sul pulsante “Get started!”. Deselezionando il box “Show this on startup” nell'angolo in basso a destra della finestra, la schermata di benvenuto non verrà più mostrata nelle prossime esecuzioni.

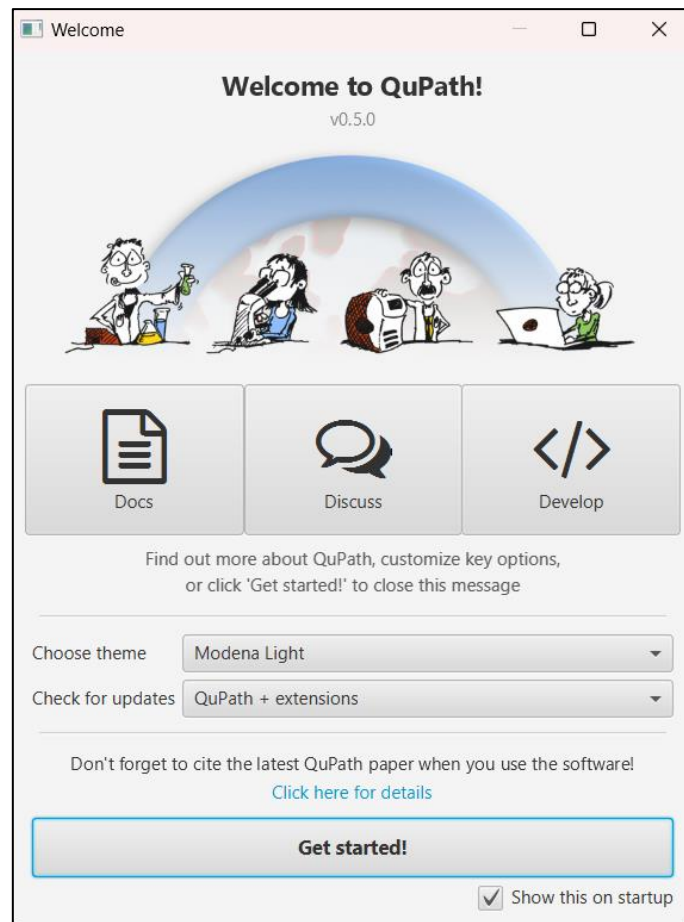


Figura 3.1 - Schermata di benvenuto di QuPath

3.3.2 Finestra principale

In Figura 3.1 é mostrata la schermata principale di *QuPath*. Andiamo a vederne i suoi elementi principali:

- In alto è presente una **barra dei menu** con diverse voci. Ogni voce apre un sottomenu a tendina che permette di svolgere operazioni diverse all'interno dell'applicazione, come caricare file, modificare la vista della finestra o svolgere operazioni di analisi cellulare.
- Appena sotto la barra dei menu, troviamo un'ulteriore **barra degli strumenti**, che comprende diversi pulsanti utili per operare direttamente con l'immagine, come ad esempio tool per l'annotazione digitale.
- Sulla sinistra è presente un **pannello di analisi** che svolge diverse funzionalità: gestione dei progetti, informazioni sull'immagine correntemente visualizzata, gestione delle annotazioni, gerarchia degli oggetti individuati, gestione del flusso di lavoro.
- Infine, il **pannello di visualizzazione** centrale è quello dove verrà visualizzata l'immagine una volta caricata.

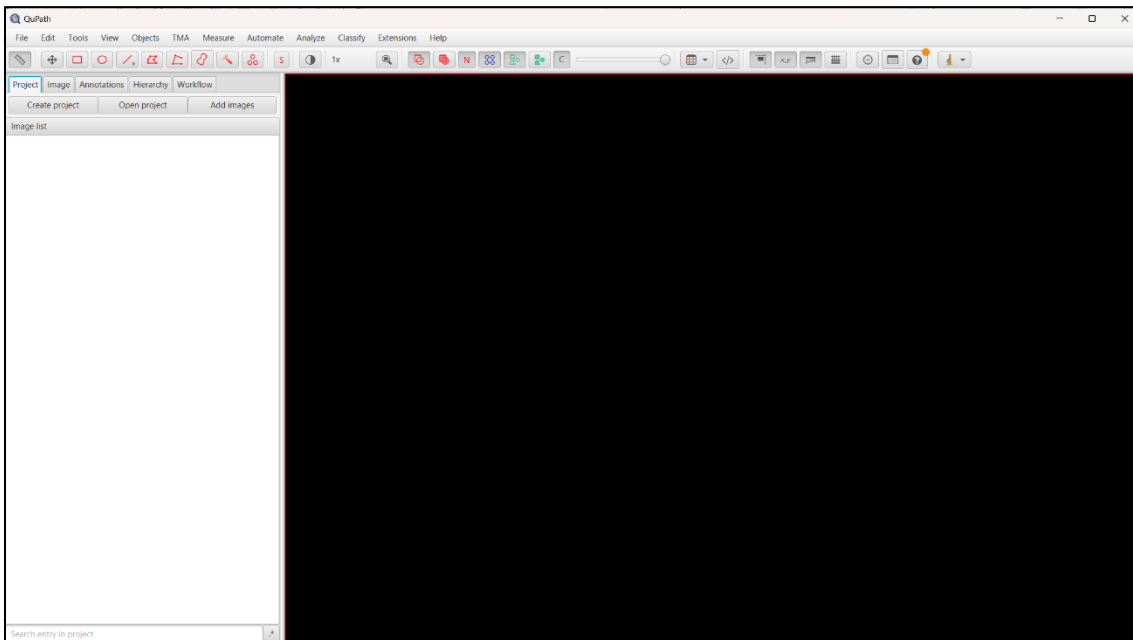


Figura 3.2 - Finestra principale di QuPath

3.3.3 Caricamento di un'immagine

Per caricare un'immagine sarà sufficiente cliccare sulla voce “File” nella barra dei menu e successivamente su “Open...”. Così facendo si aprirà una finestra di esplorazione dei file, attraverso la quale sarà possibile passare in rassegna le cartelle del sistema e selezionare una o più immagini da caricare. Un'alternativa più semplice è trascinare il file che si vuole aprire direttamente sulla finestra principale; questo funziona per tutti i tipi di file che *QuPath* supporta, non solo per le immagini.

Quando si carica un'immagine si apre una finestra (Figura 3.3) per selezionare il tipo di immagine. Cliccando su “Show details” in basso a sinistra si avranno più informazioni relative al tipo di immagine correntemente selezionato. Tra le varie opzioni va scelta la corrispondenza più vicina per l'immagine che si vuole caricare. *QuPath* può stimare automaticamente il tipo di immagine; per fare ciò va impostata l'opzione “Auto estimate” per “Set image type” andando su “Edit” e poi “Preferences”. Va tenuto in considerazione che il rilevamento automatico potrebbe sbagliare, quindi è importante non farci troppo affidamento.

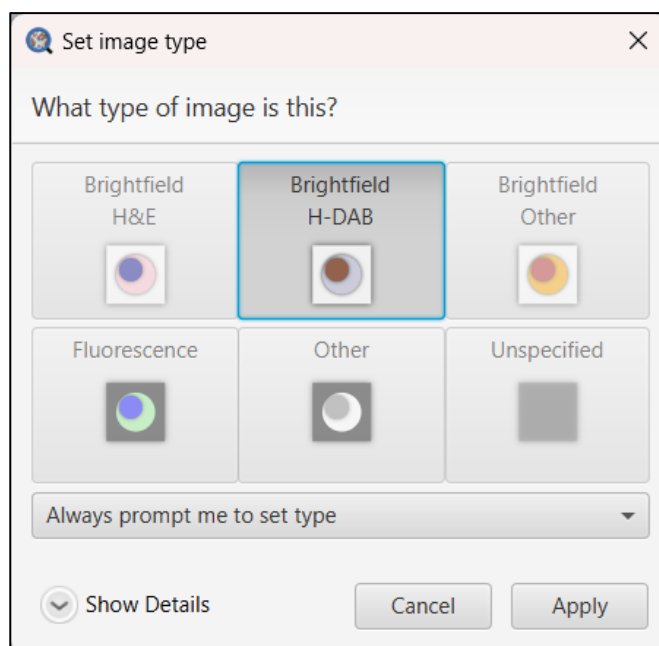


Figura 3.3 - Finestra di selezionamento tipologia di immagine

Nel pannello di analisi sulla sinistra, cliccando sulla scheda “Image” si visualizzeranno le proprietà dell’immagine caricata (Figura 3.4). Da qui sarà anche possibile cambiare la tipologia dell’immagine, cliccando due volte sul tipo di immagine correntemente impostato.

Project Image Annotations Hierarchy Workflow				
Name		Value		
Name	ID_HNSCC001.svs			
URI	file://C:/Users/pippo/Desktop/Tesi/Mat...			
Pixel type	uint8 (rgb)			
Magnification	40.0			
Width	99000 px (25007.40 µm)			
Height	70235 px (17741.36 µm)			
Dimensions (CZT)	3 x 1 x 1			
Pixel width	0.2526 µm			
Pixel height	0.2526 µm			
Uncompressed size	19.4 GB			
Server type	OpenSlide			
Pyramid	1 4 16 32			
Metadata changed	No			
Image type	Brightfield (H-DAB)			
Stain 1	Hematoxylin: 0.651 0.701 0.29			
Stain 2	DAB: 0.269 0.568 0.778			
Stain 3	Residual: 0.633 -0.713 0.302			
Background	255 255 255			

Figura 3.4 – Scheda Image del pannello di analisi

3.3.4 Oggetti in *QuPath*

Il concetto di “oggetto” è fondamentale per l’analisi a singola cellula in *QuPath*. Un oggetto può essere considerato in modo semplice come un qualcosa all’interno di un’immagine che *QuPath* è in grado di identificare, classificare e misurare.

Consideriamo due tipologie di oggetti fondamentali:

- **Annotazioni:** solitamente vengono create dall’utente, disegnandole direttamente sull’immagine
- **Detections (rilevamenti):** tipicamente vengono creati direttamente da *QuPath*, ad esempio rilevando automaticamente le cellule.

Nella maggior parte dei casi, in un’immagine si avrà un numero limitato di annotazioni, ma potenzialmente si potranno avere milioni di detections; per questo motivo *QuPath* le gestisce e mostra in modo leggermente diverso, anche se mantengono molte similarità tra loro.

3.3.5 Disegnare annotazioni

Per creare delle annotazioni, queste andranno disegnate sull’immagine. È possibile disegnare annotazioni di diverse forme: rettangolo, ellissi, poligono o a mano libera. Per creare un’annotazione, sarà sufficiente selezionare la forma voluta dalla barra degli strumenti e poi cliccare e trascinare il puntatore sull’area dell’immagine che si vuole annotare. Di default, dopo avere disegnato un’annotazione *QuPath* selezionerà in automatico lo strumento “Move”, per evitare la creazione fortuita di altre annotazioni.

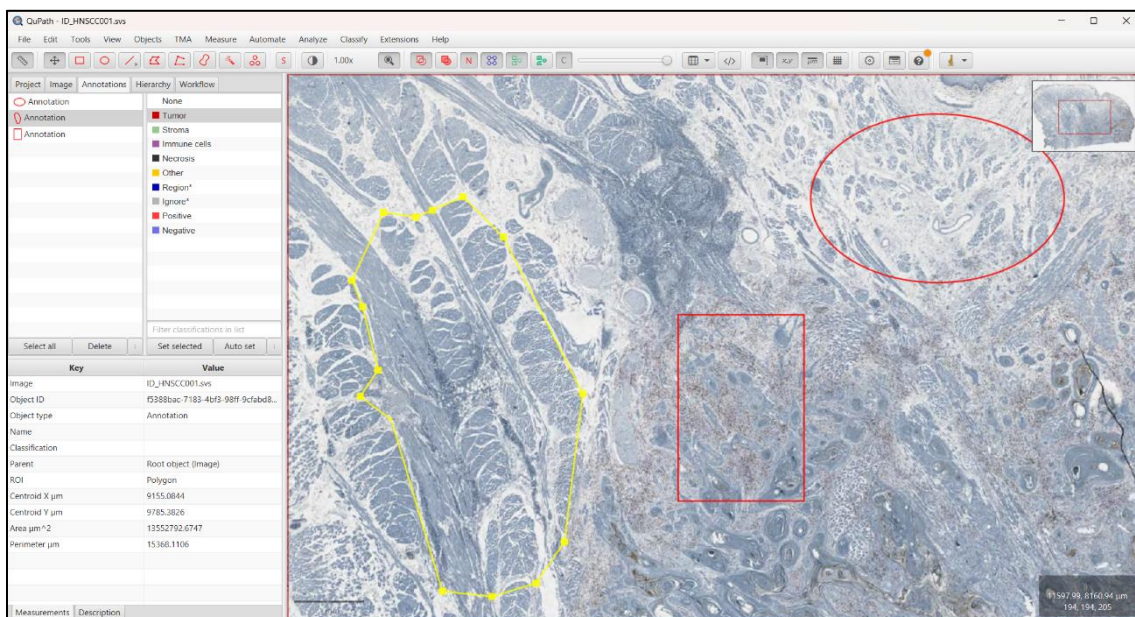


Figura 3.5 - Annotazioni rettangolare, ellittica e poligonale disegnate su un’immagine

Quando si disegna un'annotazione, esse viene automaticamente selezionata. Nella [Figura 3.5](#) si può notare che l'oggetto selezionato (l'annotazione poligonale) è mostrata in giallo. Per selezionare un altro oggetto bisogna prima tornare allo strumento "Move" e poi fare doppio clic sull'oggetto desiderato.

Nel pannello di analisi, all'interno della scheda "Annotations" ([Figura 3.6](#)) troviamo una lista di tutte le annotazioni disegnate sull'immagine. Da questa lista è possibile impostare le proprietà delle annotazioni. Per fare ciò basterà selezionare la classificazione adatta dalla colonna sulla destra e poi cliccare su "Set class". In alternativa è anche possibile aggiungere delle classificazioni definite dall'utente, cliccando con il tasto destro sull'annotazione desiderata e poi selezionando "Set properties". Dalla finestra che comparirà si potrà scegliere qualsiasi nome e colore arbitrari. Nella parte inferiore della scheda si trova una tabella che mostra le misurazioni per l'oggetto correntemente selezionato.

Key	Value
Image	ID_HNSCC001.svs
Object ID	ea6cf4d9-01ae-4af7-a04f-75d20b...
Object type	Annotation
Name	
Classification	Immune cells
Parent	Root object (Image)
ROI	Ellipse
Centroid X μm	15950.6796
Centroid Y μm	6810.8538
Area μm^2	7363806.6803
Perimeter μm	9873.8066

Figura 3.6 - Scheda "Annotations"

Per rimuovere un'annotazione, esistono diverse possibilità: direttamente dalla lista nella scheda “Annotations” oppure, nel visualizzatore, selezionando l'oggetto da eliminare e premendo “Delete” o “Backspace” sulla tastiera oppure, sempre dopo aver selezionato l'oggetto, premendo il tasto destro e selezionando l'opzione “Delete object”.

3.3.6 Rilevamento cellule

Vediamo ora gli oggetti di tipo rilevamento. Questi oggetti vengono creati direttamente da *QuPath*, per esempio rilevando automaticamente le cellule presenti all'interno di una regione. Per prima cosa bisogna selezionare un'annotazione e poi eseguire il comando di rilevamento automatico cliccando, dalla barra dei menu, su “Analyze” → “Cell detection” → “Cell detection”. Così facendo si aprirà una finestra (Figura 3.7) contenente una lista di parametri per adattare il rilevamento a immagini diverse; le impostazioni default comunque funzionano bene per la maggior parte dei casi. Premendo il bottone “Run” in fondo alla finestra, dopo qualche secondo di elaborazione, appariranno oggetti rappresentanti le cellule nell'area selezionata (Figura 3.8).

Cell detection

Setup parameters

Detection image: Hematoxylin OD

Requested pixel size: 0.5 μm

Nucleus parameters

Background radius: 8 μm

Use opening by reconstruction

Median filter radius: 0 μm

Sigma: 1.5 μm

Minimum area: 10 μm²

Maximum area: 400 μm²

Intensity parameters

Threshold: 0.1

Max background intensity: 2

Split by shape

Exclude DAB (membrane staining)

Cell parameters

Cell expansion: 5 μm

Include cell nucleus

General parameters

Smooth boundaries

Make measurements

Run

Figura 3.7 - Parametri per il rilevamento di cellule

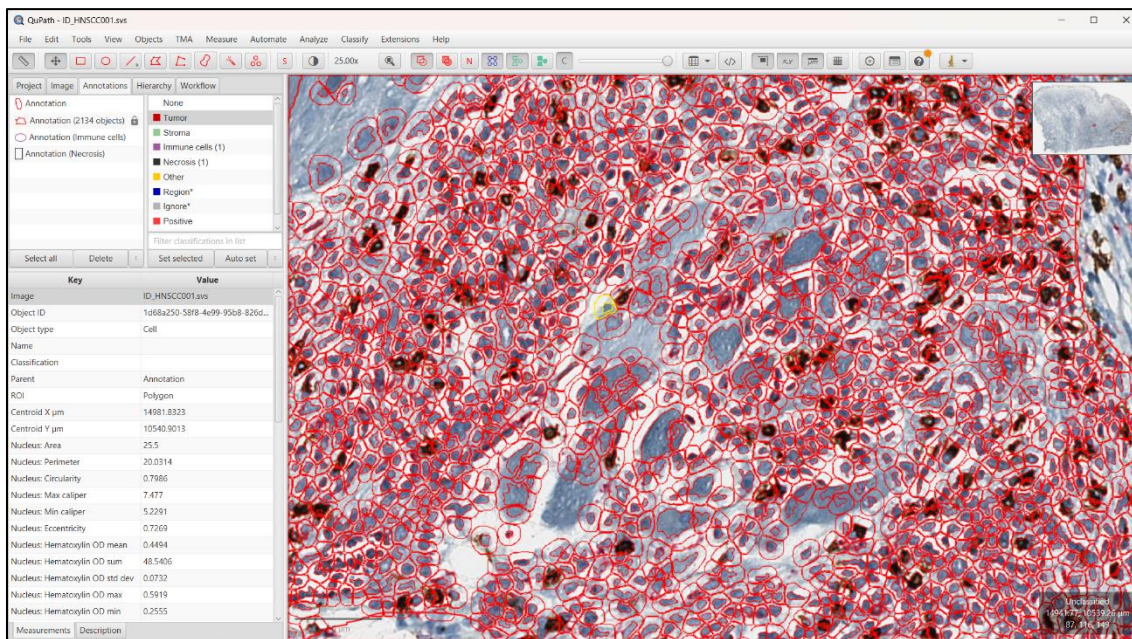


Figura 3.8 - Cellule rilevate, utilizzando parametri di rilevamento default

Come le annotazioni, anche gli oggetti di tipo rilevamento hanno delle misurazioni. Selezionando una cellula le sue misurazioni vengono visualizzate all'interno del pannello di analisi, nella stessa tabella delle annotazioni.

3.3.7 Salvataggio

Una volta finito, è possibile salvare quanto svolto finora cliccando, dalla barra dei menu, "File" → "Save as...". Così facendo verrà creato un file .qpdata, che è il formato standard di *QuPath* per salvare gli oggetti e altri meta-dati relativi all'immagine. È fortemente consigliato l'utilizzo dei progetti piuttosto che il salvataggio di file .qpdata individuali.

3.3.8 Gestione dei progetti

La creazione e l'utilizzo di un progetto fornisce numerose funzionalità:

- Raggruppamento di immagini diverse
- Cambiare facilmente tra le immagini
- Organizzare in modo coerente i file di dati, assieme a script, classificatori e altro
- Visualizzare informazioni essenziali su un'immagine, senza la necessità di aprirla completamente

La creazione di un progetto per ogni processo di analisi è sempre molto consigliata; va sottolineato che alcuni comandi e script funzionano solamente all'interno del contesto di un progetto.

Il primo passo per la creazione di un progetto sta nel creare una cartella vuota da qualche parte nel proprio computer. Per impostare questa cartella come directory del nuovo progetto si potrà trascinare la cartella direttamente sull'interfaccia utente di *QuPath*, oppure cliccare “New project” nel pannello di analisi e selezionare la cartella dalla finestra di esplorazione risorse che compare. La cartella per il nuovo progetto deve essere obbligatoriamente vuota. Una volta creato il progetto è possibile aggiungere immagini trascinandole su *QuPath* o attraverso “File” → “Project...” → “Add images”. Per passare da un'immagine all'altra sarà sufficiente cliccare sulle miniature nell'elenco delle immagini del progetto nella scheda “Project” del pannello di analisi.

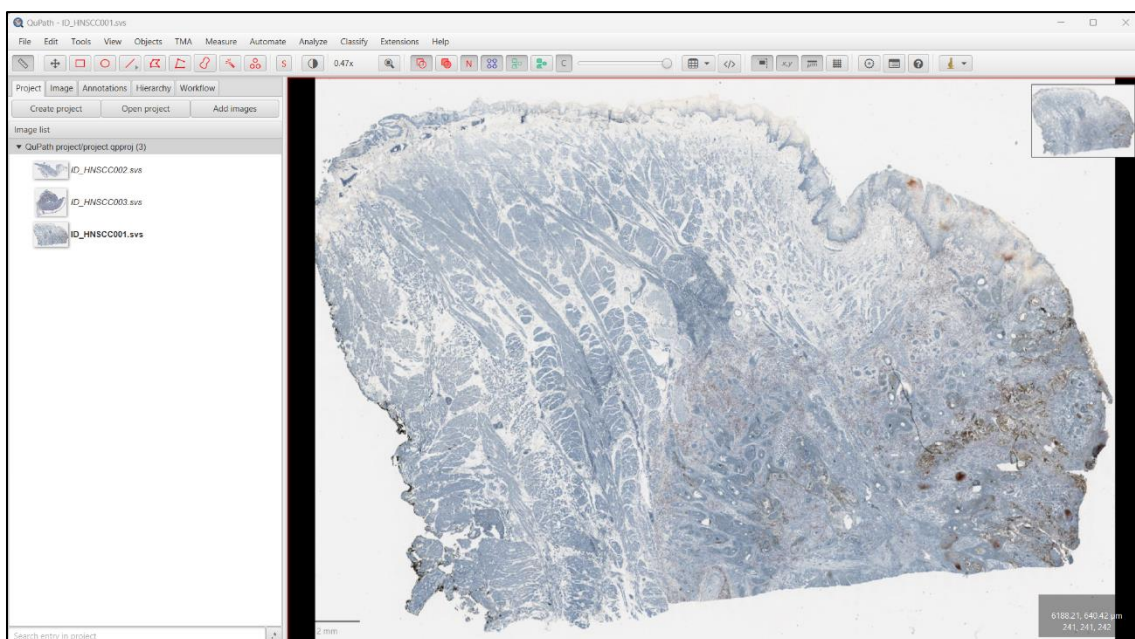


Figura 3.9 - Progetto contenente tre immagini diverse

3.4 Scripting

3.4.1 Workflow

In *QuPath* il concetto di workflow rappresenta una sequenza di operazioni che sono state applicate all'immagine, inclusi anche i parametri utilizzati nei comandi. Questo strumento fornisce quindi una modalità per tenere traccia e standardizzare l'analisi di un'immagine, rendendo possibile la verifica di come dei dati siano stati ottenuti e l'applicazione dello stesso tipo di analisi su altre immagini. *QuPath* salva automaticamente i comandi eseguiti nella scheda “**Workflow**” all'interno del pannello di analisi, per tenere traccia del lavoro svolto. Questo storico dei comandi eseguiti è un esempio di workflow di tutto quello che è stato applicato all'immagine aperta.

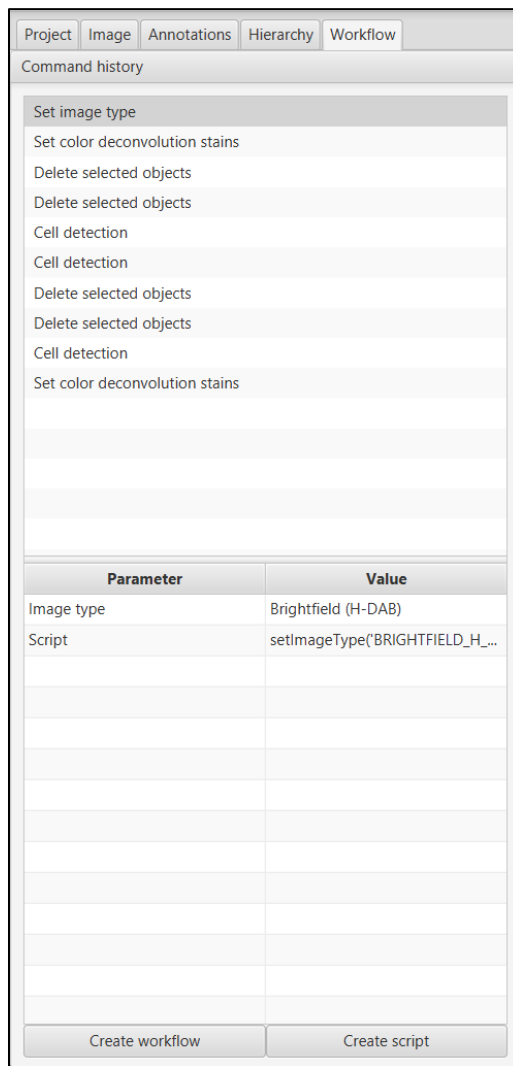
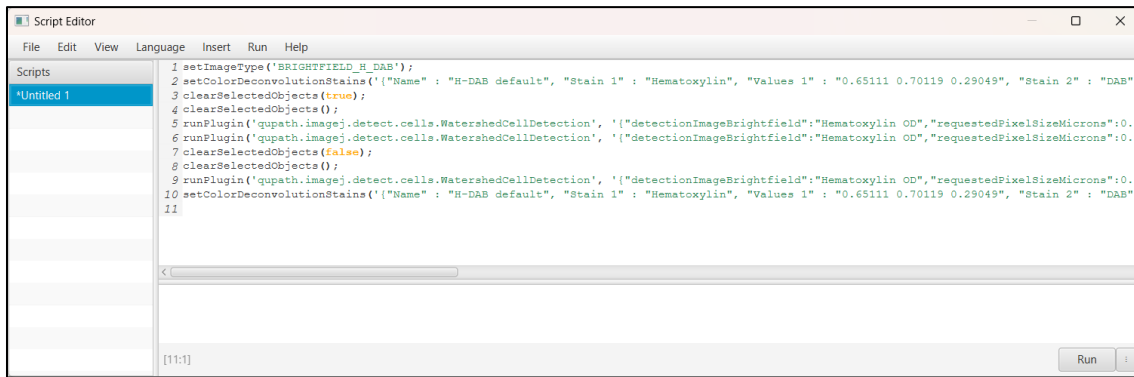


Figura 3.10 - Esempio di uno storico dei comandi

Cliccando sul pulsante “Create workflow” è possibile creare un nuovo workflow basato sull’attuale storico dei comandi, con una differenza: questo nuovo workflow potrà essere modificato a piacimento. Il nuovo workflow viene aperto in una nuova finestra, attraverso la quale si potranno effettuare modifiche cliccando con il tasto destro del mouse sulle voci dei comandi.

3.4.2 Dai workflow agli script

Gli script in *QuPath* rendono possibile la riproducibilità delle analisi e l’automazione di analisi su immagini multiple. È possibile generare automaticamente uno script partendo da un workflow cliccando sul pulsante “Create script” nel pannello workflow; così facendo si aprirà una nuova finestra di “Script editor” (Figura 3.11).



```
1 setImageType('BRIGHTFIELD_H_DAB');
2 setColorDeconvolutionStains(['Name': "H-DAB default", "Stain 1": "Hematoxylin", "Values 1": "0.65111 0.70119 0.29049", "Stain 2": "DAB",
3 clearSelectedObjects(true);
4 clearSelectedObjects();
5 runPlugin('qupath.imagej.detect.cells.WatershedCellDetection', ['detectionImageBrightfield':"Hematoxylin OD", "requestedPixelSizeMicrons":0.5
6 runPlugin('qupath.imagej.detect.cells.WatershedCellDetection', ['detectionImageBrightfield':"Hematoxylin OD", "requestedPixelSizeMicrons":0.5
7 clearSelectedObjects(false);
8 clearSelectedObjects();
9 runPlugin('qupath.imagej.detect.cells.WatershedCellDetection', ['detectionImageBrightfield':"Hematoxylin OD", "requestedPixelSizeMicrons":0.5
10 setColorDeconvolutionStains(['Name': "H-DAB default", "Stain 1": "Hematoxylin", "Values 1": "0.65111 0.70119 0.29049", "Stain 2": "DAB",
11
```

Figura 3.11 - Script generato da un workflow

Per mandare in esecuzione uno script sarà sufficiente aprire l'immagine sulla quale si vuole effettuare l'analisi e il relativo script e poi cliccare sul pulsante "Run" nella finestra "Script editor". Per eseguire lo stesso script su più immagini, per prima cosa bisognerà avere un progetto contenente tutte le immagini che si vogliono analizzare. In seguito, dal menu a tendina della finestra dello script, basterà cliccare su "Run" → "Run for project".

3.4.3 Groovy



Figura 3.12 - Logo di Groovy [18]

Gli script in *QuPath* sono scritti in linguaggio *Groovy*. *Groovy* è un linguaggio di programmazione ad oggetti per la piattaforma *Java*, ma alternativo al linguaggio *Java*, molto utilizzato come linguaggio di scripting. *Groovy* è un linguaggio tipizzato dinamicamente che si integra in maniera trasparente con librerie e altro codice *Java*. Una delle sue caratteristiche principali è la sintassi che risulta più flessibile e meno verbosa rispetto a *Java*, semplice e facile da imparare anche per chi non è esperto nella programmazione più avanzata [19].

3.4.4 Script personalizzati

Per creare uno script su *QuPath* partendo da zero, bisogna selezionare dal menu a tendina “Automate” → “Show script editor”; così facendo si aprirà una finestra dove poter scrivere il proprio script. È possibile visualizzare la documentazione delle API di *QuPath* attraverso il seguente link: <https://qupath.github.io/javadoc/docs/>.

Per poter accedere ai metodi statici usati per lo scripting in *QuPath*, è possibile selezionare l’opzione “Include default imports” dal menu a tendina “Run”. Facendo così, *QuPath* include la seguente riga all’inizio dello script:

```
1. import static qupath.lib.gui.scripting.QPEx.*
```

3.5 Estensioni

Attraverso le estensioni è possibile aggiungere nuove funzionalità a *QuPath*, oppure modificarne di esistenti. Le estensioni sono scritte in linguaggio *Java* e vengono distribuite come file Jar. Le estensioni offrono ancora più possibilità rispetto agli script, permettendo per esempio la modifica o la rimozione anche di funzionalità standard di *QuPath*, ma sono leggermente più complesse da sviluppare.

Un’estensione è semplicemente un file Jar che va posizionato in una directory delle estensioni nel classpath di *QuPath*. Concettualmente un’estensione è molto simile ad un plugin di *ImageJ*, con un’unica differenza: mentre i plugin di *ImageJ* aggiungono semplicemente dei comandi al programma, le estensioni permettono una modifica più flessibile di *QuPath*. Durante l’avvio, *QuPath* controllerà la directory delle estensioni e, nel caso in cui ce ne fossero di disponibili, procederà ad installarle una alla volta. Per installare direttamente un’estensione, è anche possibile trascinare il relativo file Jar sulla finestra principale di *QuPath*; dopo aver creato una nuova directory delle estensioni (se non esiste già una), l’estensione verrà installata. Per rimuovere un’estensione sarà sufficiente rimuovere il relativo file Jar dalla directory delle estensioni; il cambiamento avrà effetto dopo aver riavviato *QuPath*.

3.5.1 Creazione di un’estensione

Creare una nuova estensione per *QuPath* prevede tre operazioni principali:

- 1- scrivere una classe *Java* che implementi l'interfaccia *qupath.lib.gui.extensions.QuPathExtension*
- 2- creare un file di testo *META-INF/services/qupath.lib.gui.extensions.QuPathExtension* contenente un'unica riga che fornisce il nome completo della classe creata nel passo 1
- 3- creare un file Jar contenente sia la classe *Java* che il file di testo

I passaggi 2 e 3 sono necessari poiché l'architettura delle estensioni in *QuPath* si basa sul *Service Provider Framework* di *Java* per l'estendibilità. Il compito principale risulta quindi l'implementazione dell'interfaccia *QuPathExtension*:

```

1. package qupath.lib.gui.extensions;
2.
3. import qupath.lib.common.GeneralTools;
4. import qupath.lib.common.Version;
5. import qupath.lib.gui.QuPathGUI;
6.
7. /**
8.  * Simple interface for QuPath extensions.
9.  * <p>
10. * This allows dynamic discovery of new extensions.
11. *
12. * @author Pete Bankhead
13. *
14. */
15. public interface QuPathExtension {
16.
17.     /**
18.      * Install the extension for a QuPathGUI instance.
19.      * This generally involves adding new commands to appropriate menus.
20.      * Note that if an extension is only expected to be compatible with a specific QuPath
21.      * version
22.      * this method provides an opportunity to test version compatibility before making any
23.      * changes.
24.      *
25.      * @param qupath
26.      * @see QuPathGUI#getVersion()
27.      *
28.      * @implNote When multiple extensions are present, the order in which they will be
29.      * installed is undefined.
30.      */
31.     public void installExtension(QuPathGUI qupath);
32.
33.     /**
34.      * A readable name for the extension.
35.      *
36.      * @return
37.      */
38.     public String getName();
39.
40.     /**
41.      * A short description of the extension for displaying in the main GUI.
42.      * <p>
43.      * This could also contain licensing information.
44.      *
45.      * @return
46.      */
47.     public String getDescription();

```

```

47.
48.     /**
49.      * Get a QuPath version for which this extension was written.
50.      * <p>
51.      * This is used to provide an explanation if the extension could not be loaded.
52.      * It has a default implementation that returns {@link Version#UNKNOWN} to allow
53.      * backwards compatibility.
54.      * however it strongly recommended to return the actual QuPath version against which
55.      * the extension was developed and tested.
56.      * @return a semantic version corresponding to a QuPath version, e.g. "0.3.0".
57.      * @see Version
58.      */
59.     public default Version getQuPathVersion() {
60.         return Version.UNKNOWN;
61.     }
62.     /**
63.      * Get the version of the current extension.
64.      * @return
65.      * @implNote the default implementation looks for any package version associated with
66.      * the implementing class.
67.      * returning {@link Version#UNKNOWN} if none can be found.
68.      * @see GeneralTools#getPackageVersion(Class)
69.      */
70.     public default Version getVersion() {
71.         var packageVersion = GeneralTools.getPackageVersion(getClass());
72.         return Version.parse(packageVersion);
73.     }
74. }
75.

```

Il nome e la descrizione possono essere virtualmente qualsiasi cosa, vengono usati solamente a fini di visualizzazione. Il metodo `installExtension(QuPathGUI qupath)` é quello importante, poiché è quello che permette all'estensione di effettuare modifiche (come l'aggiunta di comandi). Questo metodo viene chiamato una sola volta da *QuPath*. Attraverso questo metodo è possibile modificare l'interfaccia di *QuPath*; la maggior parte delle volte risulta utile aggiungere voci nei menu per l'interazione con l'estensione. Il seguente codice mostra un esempio in cui viene aggiunta una nuova voce al menu per lanciare uno script editor personalizzato:

```

1. @Override
2.     public void installExtension(QuPathGUI qupath) {
3.
4.         // Get a reference to a menu, creating it if necessary
5.         Menu menu = qupath.getMenu("Automate>Rich script editor", true);
6.
7.         // Create a new MenuItem, which shows a new script editor when selected
8.         MenuItem item = new MenuItem("Show Rich script editor");
9.         item.setOnAction(e -> {
10.             new RichScriptEditor(qupath).showNewScript();
11.         });
12.
13.         // Add to the menu
14.         menu.getItems().add(item);
15.
16.     }
17.

```

Il metodo *getMenu(String name)* viene utilizzato per ottenere un particolare menu o sottomenu (attraverso l'utilizzo di '>'), in modo da poter aggiungere la nuova voce dove è più appropriato.

4 Algoritmi in *QuPath* per la Segmentazione di singole cellule all'interno di ROI

Il processo di segmentazione consiste nell'identificazione e isolamento di particolari regioni di interesse all'interno di immagini istologiche. Nel contesto di questa Tesi poniamo particolare interesse nell'individuazione e segmentazione di cellule in *QuPath*.

4.1 Segmentazione automatica

Attraverso l'utilizzo di algoritmi avanzati di segmentazione, *QuPath* permette l'automatizzazione del riconoscimento di diverse componenti cellulari e strutture anatomiche, semplificando i procedimenti di analisi e migliorandone l'efficienza e la precisione.

4.1.1 Segmentazione *Watershed*

Alla base della segmentazione automatica implementata da *QuPath* troviamo l'algoritmo di segmentazione *Watershed* (spartiacque). Questo algoritmo si basa sul fatto che una qualsiasi immagine in scala di grigi possa venire considerata come una superficie topografica, dove l'intensità luminosa di ogni punto denota la sua altezza: le zone ad intensità più alta rappresentano i picchi, mentre quelle dove l'intensità è più bassa rappresentano le valli. Concettualmente si riempiono le valli isolate con acqua colorata e, man mano che il livello di acqua si alza le valli si fondono. Ogni volta che avviene una di queste fusioni viene inserita una barriera. Il processo continua fino a che tutti i picchi vengono sommersi, e le barriere risultanti forniranno la segmentazione dell'immagine. Va sottolineato che, nella segmentazione automatica in *QuPath*, la trasformazione *Watershed* sta alla base dell'implementazione, ma l'algoritmo finale è molto più complesso e progettato per funzionare su una vasta gamma di immagini diverse (incluse sia immagini in campo chiaro che a fluorescenza) [20].

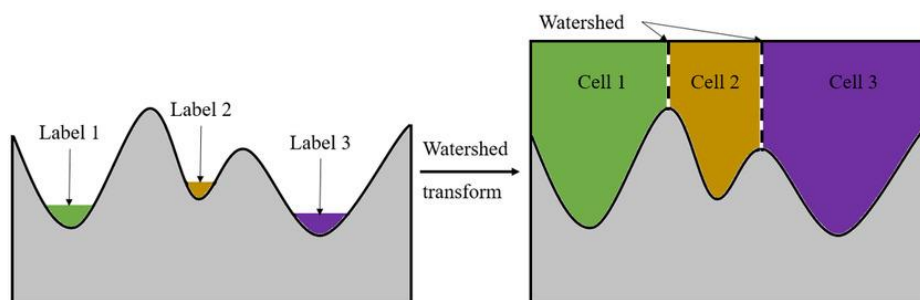


Figura 4.1 - Diagramma di esempio di segmentazione *Watershed*

4.1.2 Stain separation

La stain separation (separazione delle colorazioni) è un procedimento molto importante per migliorare il processo di segmentazione automatica e ottenere dati più consistenti. Nel contesto dell'istopatologia, le “colorazioni” si riferiscono ai coloranti utilizzati nella preparazione dei campioni istologici per “macchiare” e quindi evidenziare specifici elementi cellulari o tissutali. La stain separation consiste quindi nella separazione, in questo caso digitale, le diverse colorazioni al fine di migliorare l'analisi dettagliata del campione.

Per la separazione delle colorazioni in immagini di tipo brightfield, *QuPath* utilizza il metodo di deconvoluzione del colore [21]. Fondamentalmente, questo metodo permette la separazione digitale di fino a tre colorazioni da un'immagine RGB. Per farlo, è necessario conoscere due cose: i valori dello sfondo per ciascun canale RGB (rosso, verde, blu) e un vettore di colorazione, che caratterizza il colore preciso per ogni tinta da separare. Il comando “Brightness/Contrast” nella barra degli strumenti permette la visualizzazione dei vari canali separati. Se la separazione prevede solo due colorazioni, ne viene generata una terza in automatico, definita “Residual”.

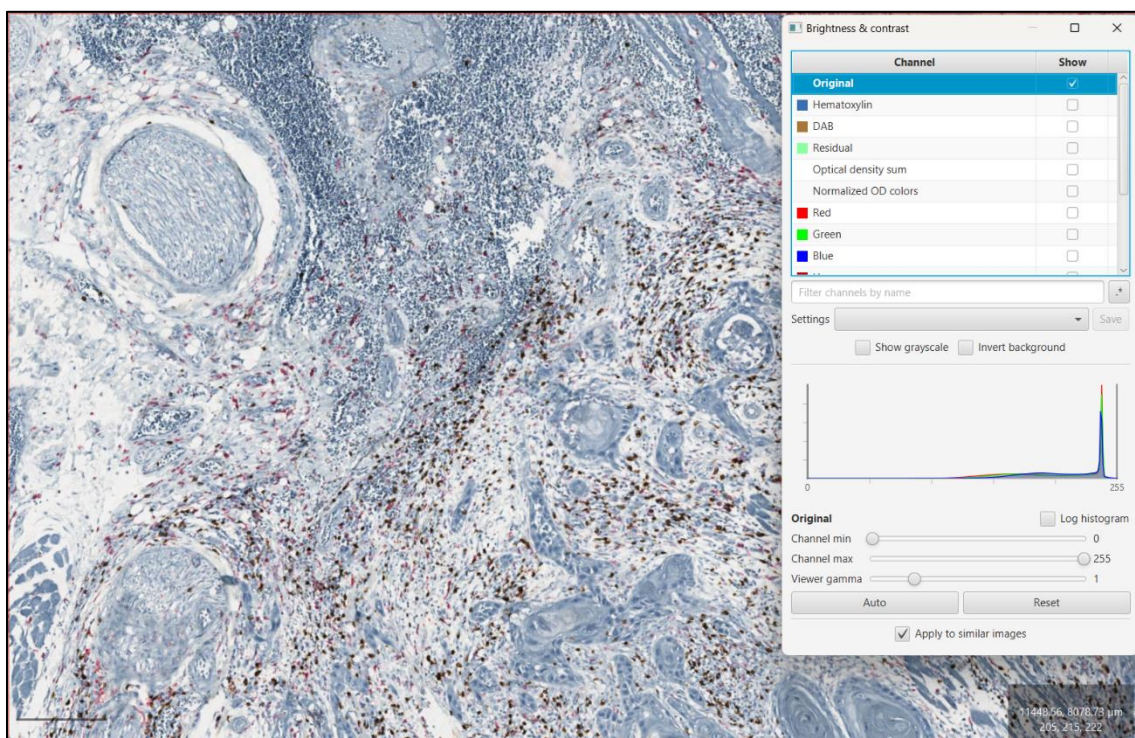


Figura 4.2 - Immagine originale

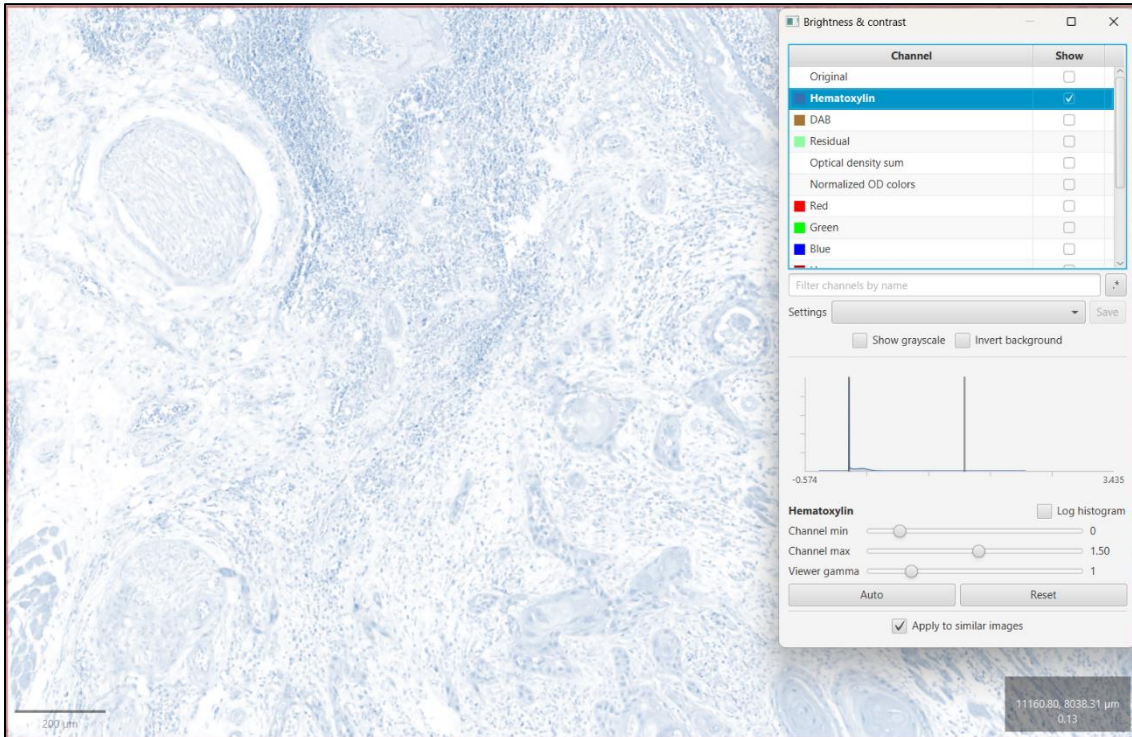


Figura 4.3 - Separazione della colorazione ematossilina

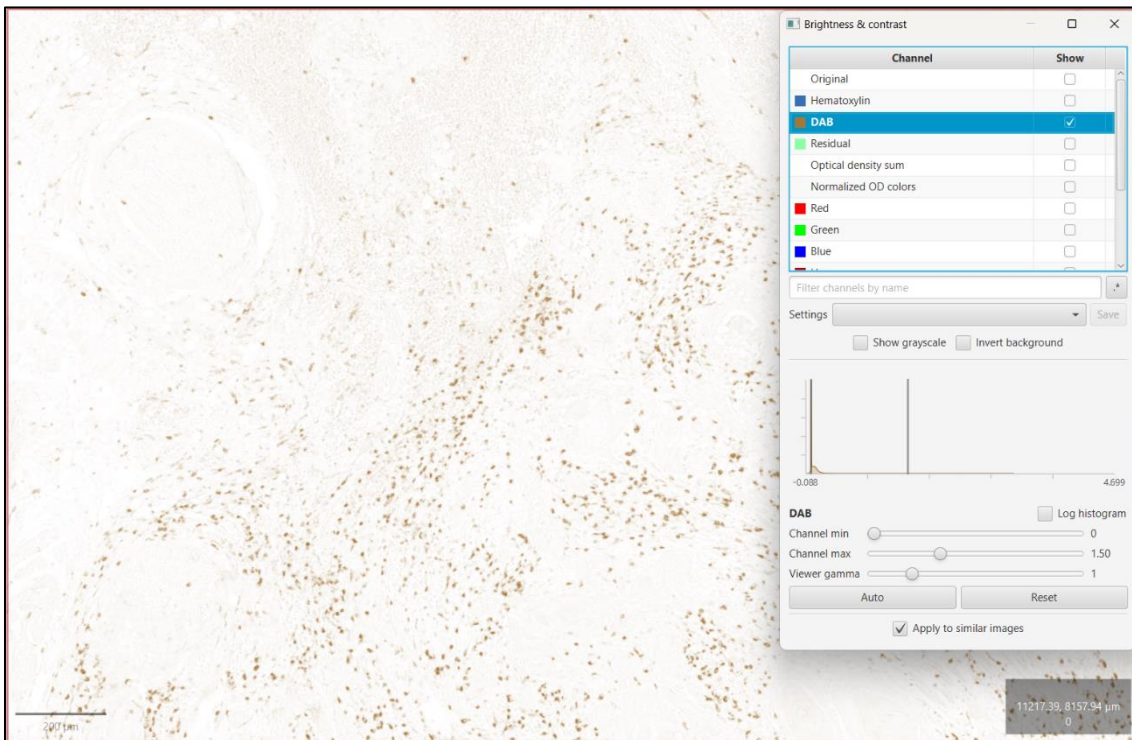


Figura 4.4 - Separazione della colorazione DAB

4.1.3 Stima dei vettori di colorazione

I vettori di colorazione sono gli elementi fondamentali per la deconvoluzione dei colori, poiché forniscono una rappresentazione normalizzata del colore di ogni tinta pura nell'immagine, senza tenere conto dell'intensità della colorazione. Quando viene impostata un'immagine come tipologia

brightfield, *QuPath* imposterà dei vettori di colorazione default per caratterizzare ematossilina, eosina o DAB. Tuttavia, per alcuni processi (in particolar modo la segmentazione di cellule), si potrebbero ottenere risultati migliori impostando esplicitamente i vettori di colorazione più adatti alla colorazione effettiva dell'immagine. Nel caso in cui ci siano esattamente due colorazioni da separare, *QuPath* mette a disposizione un metodo per stimare i vettori di colorazione, attraverso il comando “Analyze” → “Estimate stain vectors”. Come prima cosa bisogna scegliere una regione rappresentativa dell'immagine contenente chiari esempi delle colorazioni che si vogliono separare e possibilmente contenente anche una porzione di sfondo; una volta scelta questa regione, andrà racchiusa in un'annotazione rettangolare.

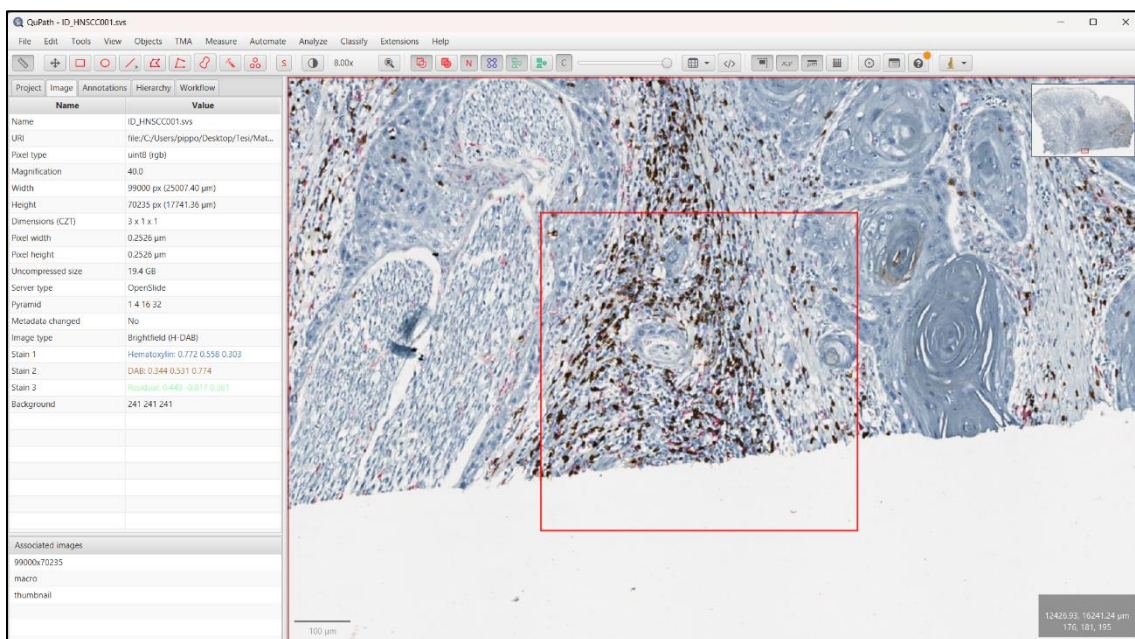


Figura 4.5 - Annotazione della regione per stima automatica dei vettori di colorazione

Una volta creata l'annotazione, è possibile lanciare il comando “Estimate stain vectors”. Se lo sfondo contenuto nella regione annotata non dovesse coincidere con i valori di sfondo che utilizza *QuPath*, basterà confermare l'aggiornamento dei nuovi valori per lo sfondo nella finestra di dialogo che verrà aperta (Figura 4.6).

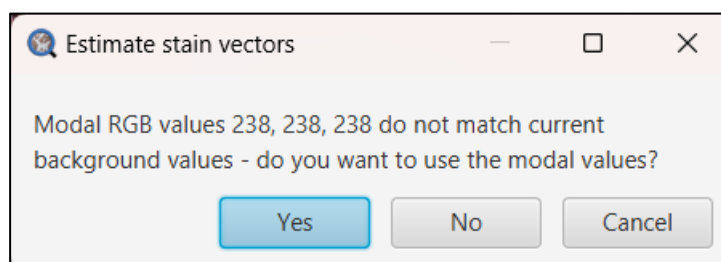


Figura 4.6 - Prompt per l'aggiornamento dei valori di sfondo

A questo punto *QuPath* genera dei diagrammi di dispersione per visualizzare le relazioni tra i valori del rosso, del verde e del blu per ogni pixel (Figura 4.7). Vengono creati tre diagrammi separati, che mostrano ciascuno un colore in relazione agli altri due. Le due righe colorate rappresentano i vettori di colorazione attuali.



Figura 4.7 - Diagrammi di dispersione per stima dei vettori di colorazione

Idealmente, i vettori dovrebbero racchiudere precisamente la maggior parte dei punti a dispersione; vediamo però che in questo caso non è così. Cliccando sul pulsante “Auto”, *QuPath* sceglierà automaticamente i vettori migliori utilizzando i dati della regione precedentemente annotata. Eventualmente è anche possibile modificare manualmente i parametri per scegliere i vettori a proprio piacimento.

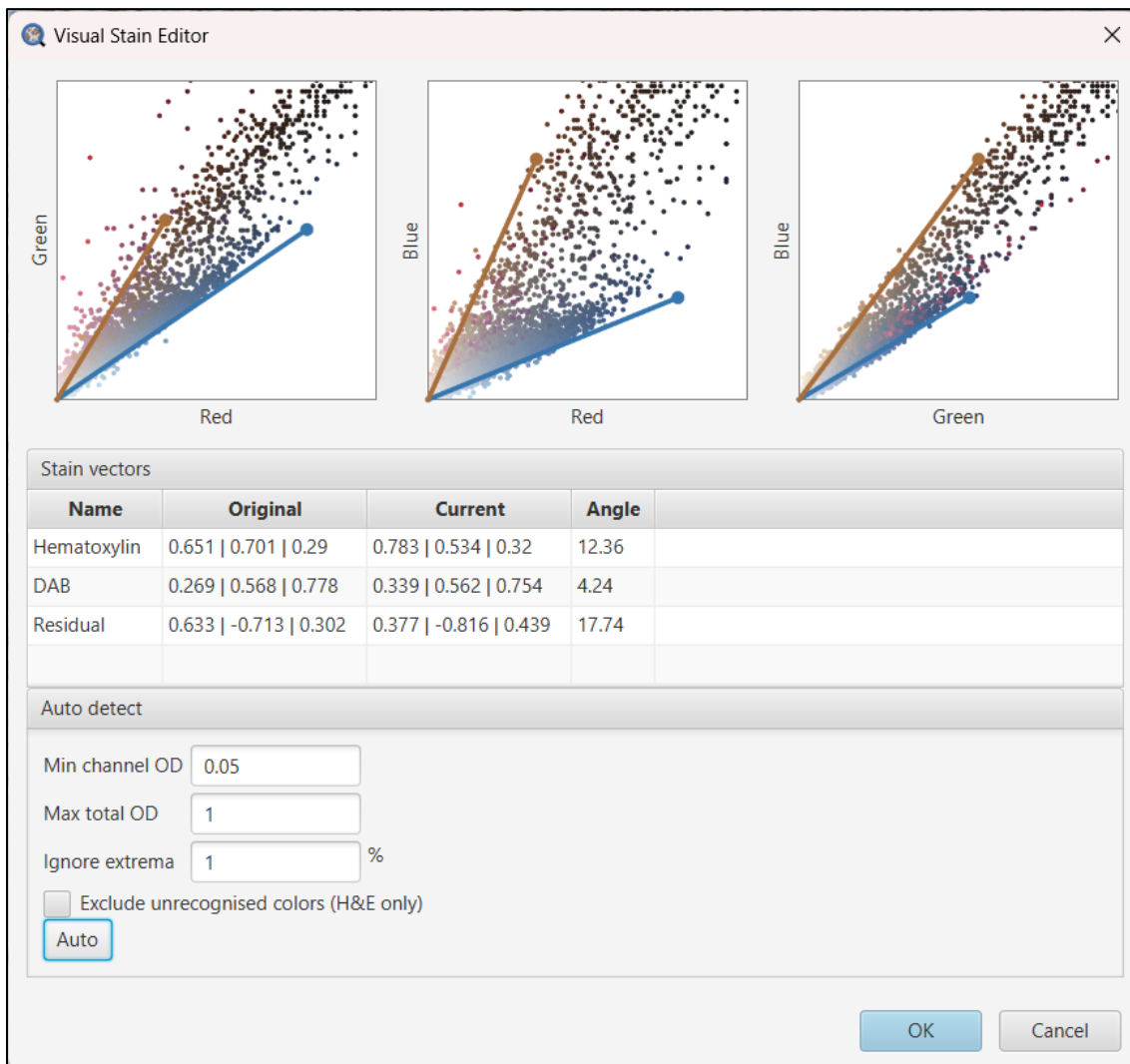


Figura 4.8 - Vettori di colorazione scelti automaticamente da QuPath

Premendo su “OK”, sarà possibile inserire un nome per i nuovi vettori di colorazione calcolati. Una volta fatto ciò, *QuPath* ritornerà alla visualizzazione dell’immagine correntemente aperta, sulla quale sono stati applicati i nuovi vettori di colorazione per la separazione delle tinte. Attraverso lo storico dei comandi, risulta semplice generare uno script che imposti i valori dei nuovi vettori di colorazione, per poi eseguirlo su tutte le immagini contenute nel progetto attraverso il comando “Run” → “Run for project”.

```

1. setImageType('BRIGHTFIELD_H_DAB');
2. setColorDeconvolutionStains({'Name' : "H-DAB estimated", "Stain 1" : "Hematoxylin",
    "Values 1" : "0.78307 0.5335 0.31965", "Stain 2" : "DAB",
    "Values 2" : "0.33912 0.56234 0.75417",
    "Background" : " 241 241 241"});
3.

```

4.1.4 Segmentazione cellulare

Vediamo ora la segmentazione automatica di cellule, con la possibilità anche di ottenere riscontri negativi o positivi basati su determinati parametri per il riconoscimento di cellule tumorali. Come primo passo, bisogna annotare una regione relativamente contenuta del campione sulla quale si vuole effettuare l'analisi, utilizzando uno degli strumenti di annotazione citati nei capitoli precedenti. Una volta fatto questo, è possibile eseguire il comando “Analyze” → “Cell detection” → “Positive cell detection”, che aprirà una finestra di dialogo dove poter impostare le opzioni per il rilevamento (Figura 4.9). I valori di default solitamente vanno bene come punto di partenza, con la possibilità di modificarli in base alle diverse necessità.

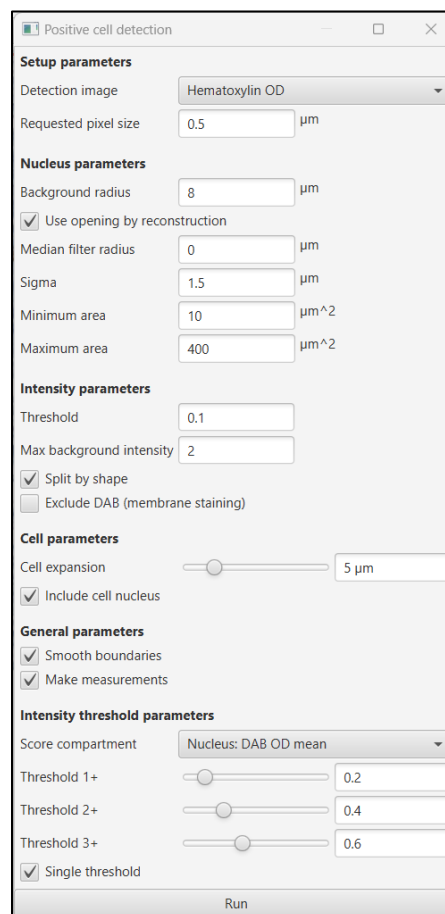


Figura 4.9 - Parametri per l'individuazione di cellule positive

Le ultime voci rappresentano i parametri secondo i quali le cellule individuate debbano essere considerate positive o negative. Il valore “Nucleus: DAB OD mean” per l'impostazione “Score compartment” indica che la scelta sarà basata sulla colorazione media di DAB (marrone) del nucleo, che è adatto per il nostro caso. Selezionando la casella “Single threshold” si avrà una categorizzazione binaria tra match positivi o negativi; deselezionandola sarà possibile definire tre

differenti soglie di valori per categorizzare le cellule in base all'intensità di colorazione, ossia negativa, debolmente positiva, moderatamente positiva e fortemente positiva. Nel caso di una categorizzazione binaria, *QuPath* utilizzerà il primo valore di soglia. Una volta premuto sul pulsante “Run”, otterremo il risultato della segmentazione come mostrato in [Figura 4.10](#), con le cellule rilevate positivamente colorate in rosso e quelle negative in blu. Nel nostro caso, attraverso il pannello “Annotations”, vediamo che sono stati rilevati 3031 oggetti di tipo cellula, con una percentuale di match positivi del 29.1983%.

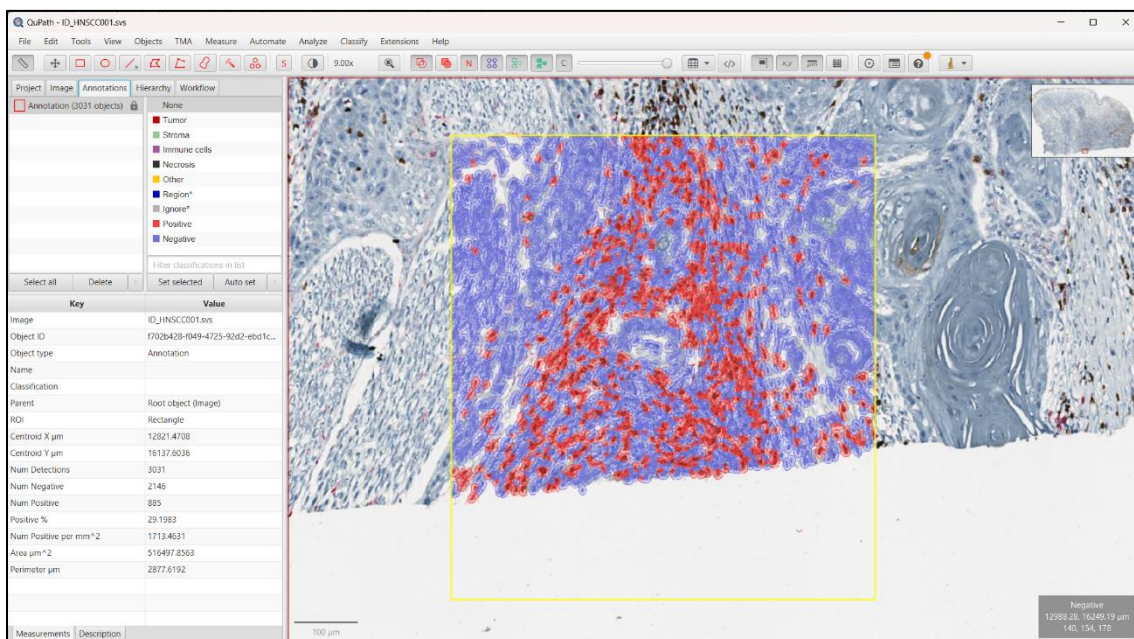


Figura 4.10 - Risultato del rilevamento di cellule positive

4.1.5 Misurazioni

Una volta ultimata la segmentazione delle cellule, è possibile analizzare ogni cellula singolarmente. Infatti, cliccando due volte su una qualsiasi cellula individuata, sarà possibile visionarne le varie misurazioni nel pannello a sinistra ([Figura 4.11](#)). Qui possiamo trovare dei dati significativi per l'analisi di ogni singola cellula, come area e perimetro del nucleo e l'intensità della colorazione.

Key	Value
Image	ID_HNSCC001.svs
Object ID	a8ad0c1b-35ee-41a5-b022-3c94...
Object type	Cell
Name	
Classification	Positive
Parent	Annotation
ROI	Polygon
Centroid X μm	13045.4096
Centroid Y μm	16031.033
Nucleus: Area	27.5
Nucleus: Perimeter	25.6424
Nucleus: Circularity	0.5256
Nucleus: Max caliper	11.1947
Nucleus: Min caliper	3.0157
Nucleus: Eccentricity	0.9692
Nucleus: Hematoxylin OD mean	0.6037
Nucleus: Hematoxylin OD sum	65.8039
Nucleus: Hematoxylin OD std dev	0.2
Nucleus: Hematoxylin OD max	0.9586
Nucleus: Hematoxylin OD min	0.2321
Nucleus: Hematoxylin OD range	0.7265
Nucleus: DAB OD mean	0.9603
Nucleus: DAB OD sum	104.6703
Nucleus: DAB OD std dev	0.2376
Nucleus: DAB OD max	1.3889
Nucleus: DAB OD min	0.4132
Nucleus: DAB OD range	0.9757
Nucleus: Area	116.75

Figura 4.11 - Misurazioni di una delle cellule individuate

Un ulteriore metodo per visualizzare le misurazioni delle cellule è il comando “Measure” → “Show detection measurements”. Questo comando aprirà una nuova finestra (Figura 4.12) contenente una tabella con le misurazioni di tutte le cellule; da qui sarà possibile generare istogrammi (“Show histograms”), ordinare la tabella in base alle colonne, selezionare singole cellule ed esportare le misurazioni in formato CSV.

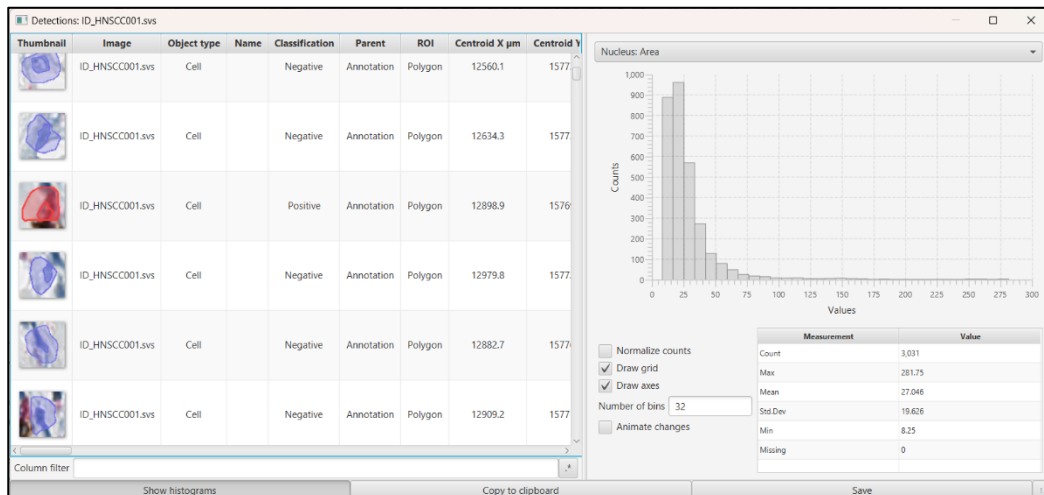


Figura 4.12 - Tabella delle misurazioni

4.2 Segmentazione semi-automatica

Una volta segmentate automaticamente le cellule, è possibile modificare e affinare la qualità del risultato andando a rimuovere o aggiungere cellule ai rilevamenti. Per fare ciò basterà utilizzare il comando “Convert detections to points” all’interno dello strumento “Points” nella barra degli strumenti (quello con i tre pallini rossi). Questo comando genererà un’annotazione puntiforme editabile, partendo dagli oggetti rilevati dalla segmentazione. Tramite la finestra “Points” sarà ora possibile modificare la segmentazione, andando ad aggiungere o rimuovere punti alle annotazioni.

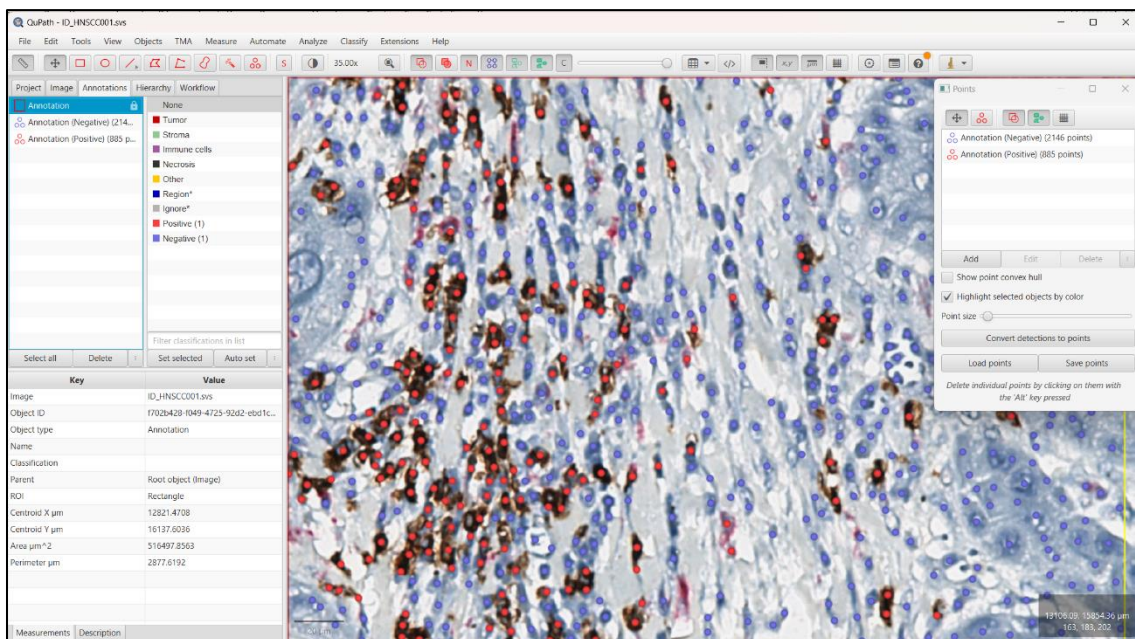


Figura 4.13 - Cellule segmentate convertite in annotazioni puntiformi

4.3 Segmentazione manuale

QuPath offre anche la possibilità di segmentare e contare manualmente le cellule presenti all’interno di un’area dell’immagine, attraverso la creazione di un’annotazione puntiforme. Per fare ciò, bisognerà cliccare sullo strumento “Point” nella barra degli strumenti, che aprirà un pannello denominato “Points” (Figura 4.14). Sarà quindi possibile andare a cliccare con il mouse sulle cellule che si vogliono contare direttamente sull’immagine. Cliccando poi sul pulsante “Add” nel pannello, verrà creata una nuova annotazione per categorizzare in maniera diversa un altro insieme di cellule. Facendo doppio clic su un’annotazione nel pannello “Points” sarà possibile configurarne un nome descrittivo e un colore per la differenziazione. Va considerato che questa tipologia di segmentazione manuale offre meno informazioni rispetto all’analisi delle cellule rispetto alla segmentazione

automatica, ma risulta più flessibile ed è consigliato usarla nei casi in cui si voglia effettuare solamente un conteggio delle cellule presenti in una porzione contenuta del campione.

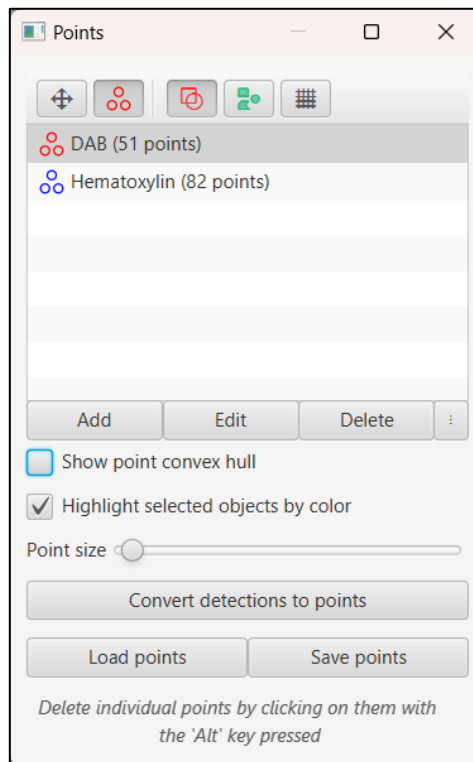


Figura 4.14 - Pannello per le annotazioni puntiformi

5 Analisi quantitativa sperimentale

5.1 Descrizione caso di studio

In questo capitolo verrà esaminata da vicino l'applicazione dei diversi software per l'analisi di immagini istologiche citati nel capitolo 2, concentrandosi su un caso di studio specifico che comprende la selezione di una specifica ROI che risulti ripetibile e il conteggio totale dei nuclei al suo interno. Questo approccio mira a fornire una valutazione dettagliata delle capacità di questi strumenti software nell'identificare e quantificare strutture e caratteristiche rilevanti presenti nel tessuto biologico, con un'attenzione particolare alla riproducibilità e all'accuratezza delle misurazioni. Attraverso un'analisi quantitativa sperimentale delle misurazioni ottenute utilizzando i diversi software, si mira a comprendere meglio i vantaggi e le limitazioni di tali strumenti nell'ambito dell'analisi istologica, con possibili implicazioni per la pratica clinica e la ricerca biomedica. In particolare, verranno sperimentati e analizzati i software che permettono di svolgere le seguenti operazioni:

1. Definizione di una ROI attraverso uno strumento di disegno a mano libera
2. Segmentazione automatica e conteggio dei nuclei totali all'interno della ROI definita

I risultati prodotti dai diversi software saranno confrontati con una ground truth ottenuta attraverso il conteggio manuale dei nuclei, effettuato mediante un tool di conteggio denominato "tally counter". Come ROI da analizzare è stata scelta una regione a forma di "fagiolino" avente area di circa 0.086 mm^2 e perimetro circa 1.23 mm , mostrata in Figura 5.1.

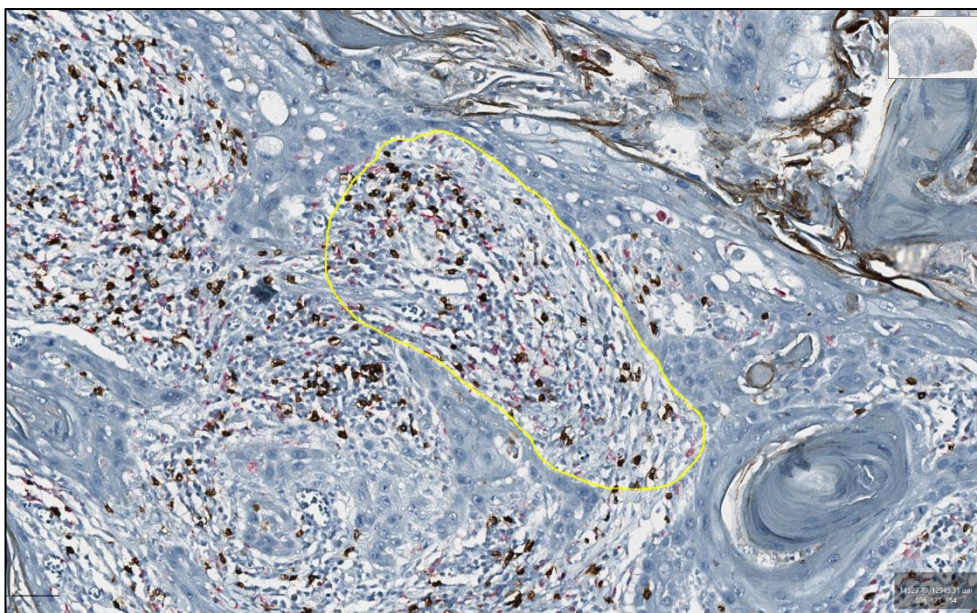


Figura 5.1 - ROI di riferimento per l'analisi sperimentale

La ROI di riferimento (Figura 5.1) è stata definita e disegnata utilizzando il software *QuPath*, per poi venire esportata. Per ottenere la maggior coerenza possibile durante la fase di sperimentazione, nei software che lo permettono questa ROI è stata importata direttamente, mentre per gli altri è stata ridisegnata utilizzando gli strumenti a disposizione, con la maggior precisione possibile. In particolare:

- *Amira* – ROI importata mediante maschera binaria.
- *Arivis* – ROI definita con strumento a mano libera poiché il software è stato testato durante una demo.
- *CellProfiler* – ROI importata mediante maschera binaria.
- *HALO* – ROI definita con strumento a mano libera poiché il software è stato testato durante una demo.
- *ImageJ/Fiji* – ROI importata mediante maschera binaria.
- *NIS-Elements* – ROI definita con strumento a mano libera.
- *QuPath* – ROI di riferimento definita con strumento a mano libera ed esportata come maschera binaria.

5.2 Descrizione provino istologico

L'immagine utilizzata per questo studio è stata acquisita presso l'istituto oncologico IRCCS IRST di Meldola. Questa immagine, fornita in forma anonimizzata per garantire la riservatezza dei pazienti coinvolti nella ricerca, è il frutto di un meticoloso processo di preparazione e acquisizione. Si tratta di una scansione ad alta risoluzione effettuata a 40x di un intero campione istologico proveniente da una biopsia tumorale della categoria testa-collo. Il campione è stato colorato utilizzando tre diversi marcatori chimici: ematossilina (blu), DAB (*3,3'-Diaminobenzidine*, marrone) e AP-Red (*Alkaline Phosphatase Red, fosfatasi alcalina*, rosso); questa colorazione consente una dettagliata analisi delle strutture cellulari e tissutali, come si può notare in Figura 5.3. Lo strumento utilizzato per la cattura dell'immagine è lo scanner *Aperio CS2* di *Leica Biosystems*.



Figura 5.2 - Provino istologico di tumore testa-collo

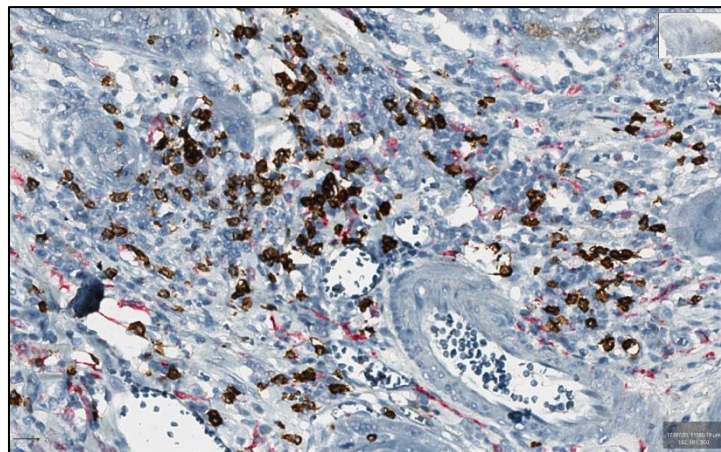


Figura 5.3 - Sezione del provino dove si possono notare le tre colorazioni dei marcatori

5.3 Descrizione ground truth

Per definire la ground truth con la quale verranno confrontati i risultati sperimentali ottenuti, è stato utilizzato il tool di annotazione “Points” di *QuPath*, che permette di creare un’annotazione composta da oggetti puntiformi e quindi di selezionare manualmente e contare tutti i nuclei all’interno della ROI definita. L’utente che ha svolto l’analisi manuale e ha definito la ground truth è stato precedentemente addestrato al riconoscimento delle strutture cellulari e all’identificazione dei diversi segnali di colorazione da parte di un operatore esperto. L’addestramento si è svolto su immagini simili ma distinte da quella su cui si è svolta l’analisi sperimentale.

5.3.1 Ground truth nuclei totali

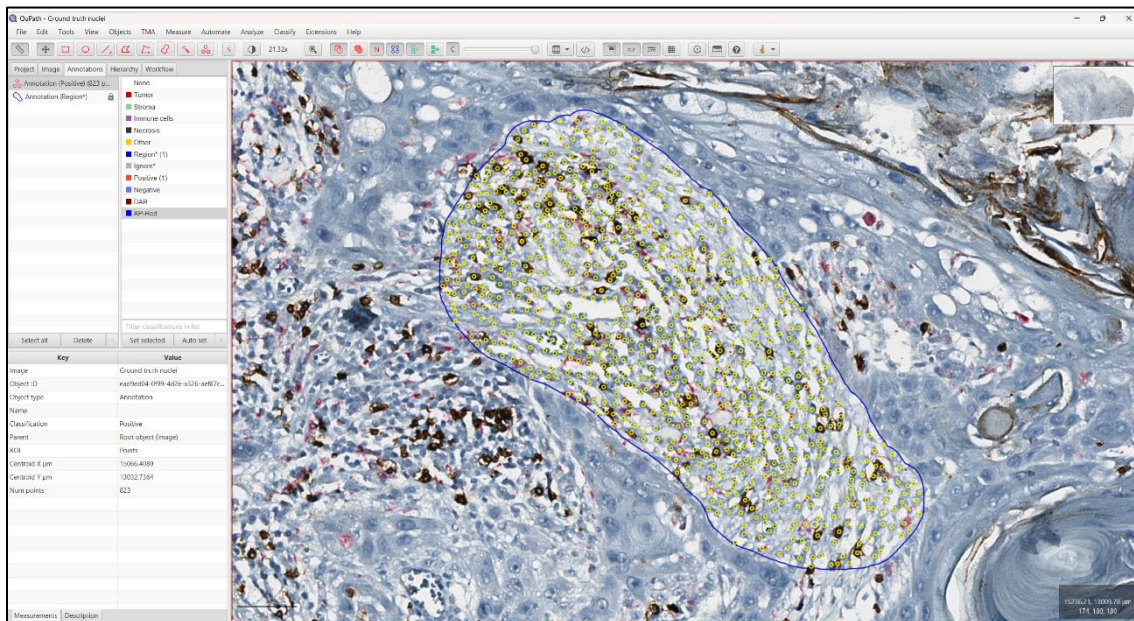


Figura 5.4 - Annotazione per il conteggio manuale dei nuclei cellulari all'interno della ROI

Dall'analisi manuale per la definizione della ground truth sono stati individuati 823 nuclei cellulari, come si può verificare in Figura 5.4, nella tabella delle misurazioni sulla sinistra dell'interfaccia utente.

5.3.2 Ground truth con classificazione marcatori

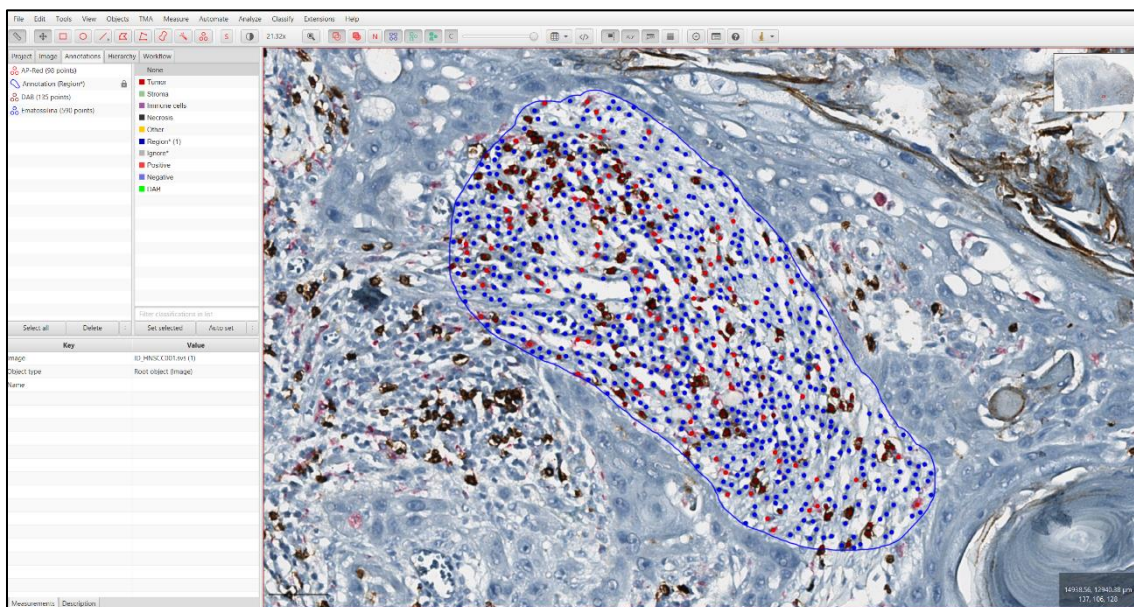


Figura 5.5 - Annotazioni per il conteggio manuale con classificazione dei nuclei

L'analisi manuale ha ottenuto i seguenti risultati per quanto riguarda la definizione della ground truth con classificazione dei marcatori: 590 nuclei colorati da ematossilina, 135 nuclei DAB, 98 nuclei AP-Red, per un totale di 823 nuclei cellulari.

5.4 Conteggio totale: risultati sperimentali

5.4.1 Tool idonei

Software open source:

- *CellProfiler* – **incluso**.
- *Cytomine* – **escluso**. Il software non offre una funzionalità per il conteggio automatico delle cellule.
- *Digital Slide Archive* – **escluso**. L'algoritmo di rilevamento automatico su questo software permette solamente di definire una ROI rettangolare nella quale venire eseguito; inoltre viene effettuato il rilevamento ma non il conteggio e l'extrapolazione di dati.
- *Icy* – **escluso**. Nonostante offra diverse funzionalità per l'analisi, non è possibile effettuare il rilevamento automatico su immagini istologiche a campo chiaro.
- *ImageJ/Fiji* – **incluso**.
- *QuPath* – **incluso**.

Software *freely-available*:

- *Aperio ImageScope* – **escluso**. Trattandosi di un visualizzatore di immagini i cui algoritmi consentono solo di extrapolare la conta dei pixel e non la conta delle cellule.
- *NIS-Elements Viewer* – **incluso**, ma nella sua versione commerciale.
- *Sedeen Viewer* – **escluso**. Sebbene offra diversi algoritmi di analisi, non è possibile eseguire il conteggio automatico.
- *SlideViewer* – **escluso**. Il software offre solo funzionalità per il conteggio manuale delle cellule.

Software commerciali:

- *Amira* – **incluso**.
- *Arivis* – **incluso**.
- *HALO* – **incluso**.

- *Imaris* – **escluso**. Il software offre funzionalità altamente avanzate, ma si concentra sull'analisi di dataset in tre e quattro dimensioni e di immagini a fluorescenza.

5.4.2 Amira

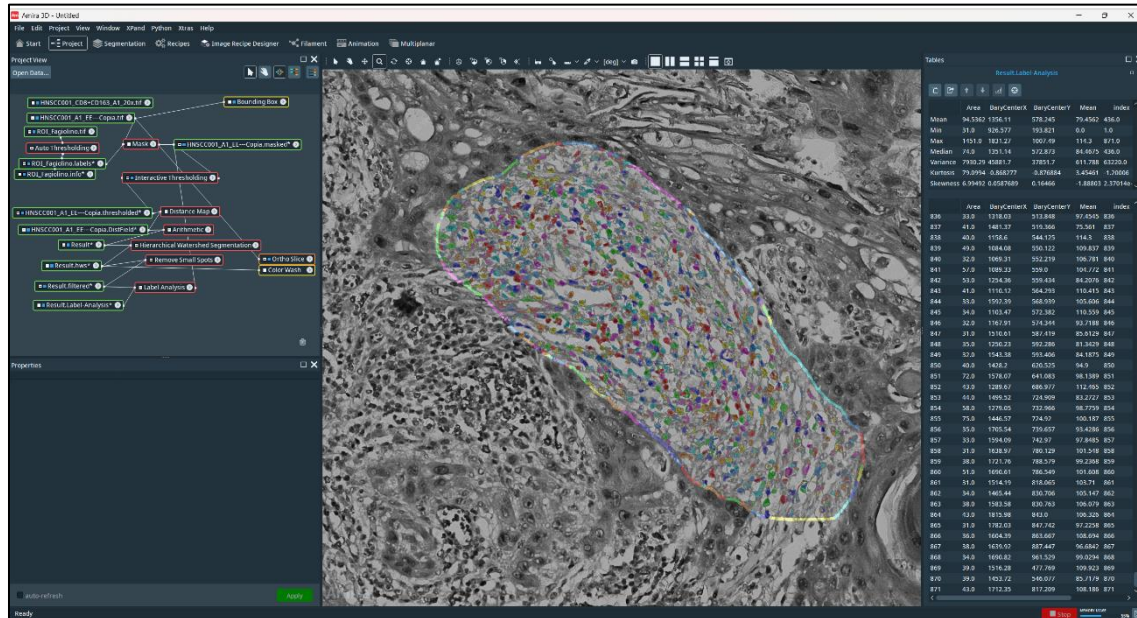


Figura 5.6 - Conteggio dei nuclei con Amira

L'analisi effettuata con *Amira* ha portato ad un conteggio di 871 nuclei totali. Il processo di analisi ha previsto l'applicazione di una maschera binaria per selezionare la regione di interesse, un'operazione di thresholding, una segmentazione Watershed per separare i nuclei uniti tra loro ed una rimozione degli elementi troppo piccoli.

I principali parametri utilizzati sono i seguenti:

- Valori di soglia (0 – 120)
- Modalità segmentazione Watershed (DARK)
- Range globale segmentazione (1,00 – 6,75)
- Livello di dettaglio segmentazione (0,6)
- Dimensione minima per rimozione small spots (30 pixel)

5.4.3 Arivis

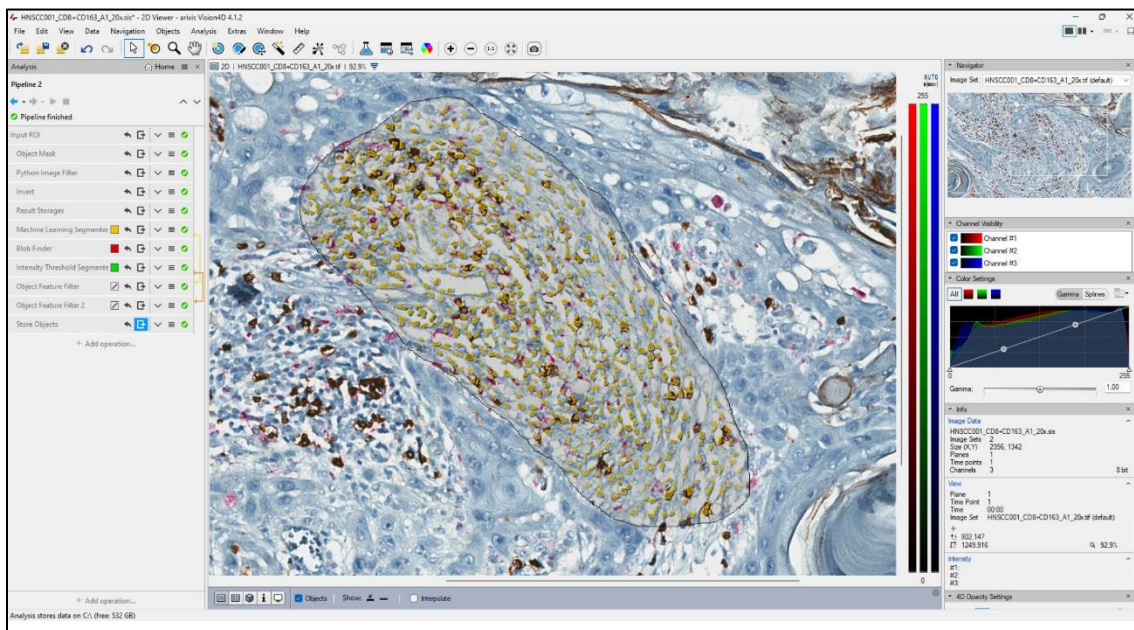


Figura 5.7 - Cell Detection in Arivis

La pipeline di analisi utilizzata su *Arivis* ha individuato un totale di 645 nuclei cellulari. Questo flusso di analisi prevede una segmentazione basata su machine learning, seguita da un algoritmo *blob finder* per l'individuazione dei nuclei. I risultati ottenuti sono stati poi filtrati per superficie e circolarità per scartare falsi positivi e corpuscoli.

5.4.4 CellProfiler

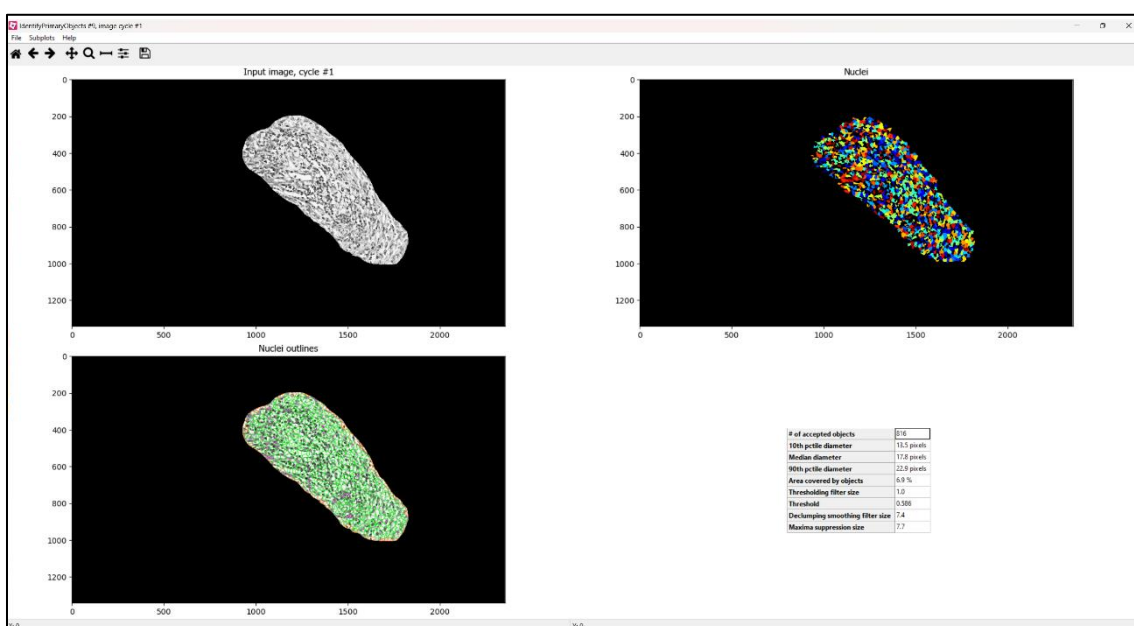


Figura 5.8 - Conteggio cellule in CellProfiler

Il conteggio delle cellule effettuato tramite CellProfiler ha rilevato un totale di 816 oggetti. La pipeline di analisi utilizzata comprende l'identificazione della ROI di interesse mediante i moduli *IdentifyObjectsManually* e *ConvertObjectsToImage*, seguita dall'individuazione e dal conteggio delle cellule tramite il modulo *IdentifyPrimaryObjects*.

I parametri impostati sono i seguenti:

- diametro tipico degli oggetti in pixel (min 11, max 80)
- eliminazione degli oggetti al di fuori dell'intervallo dei diametri (SI)
- eliminazione degli oggetti che toccano il bordo dell'immagine (SI)
- metodo di thresholding (Minimum Cross-Entropy)
- scala di smoothing del threshold (1.3488)
- fattore di correzione del threshold (1.0)
- metodo per distinguere gli oggetti ammassati (INTENSITY)
- metodo per tracciare le linee di divisione tra gli oggetti ammassati (PROPAGATE).

5.4.5 HALO

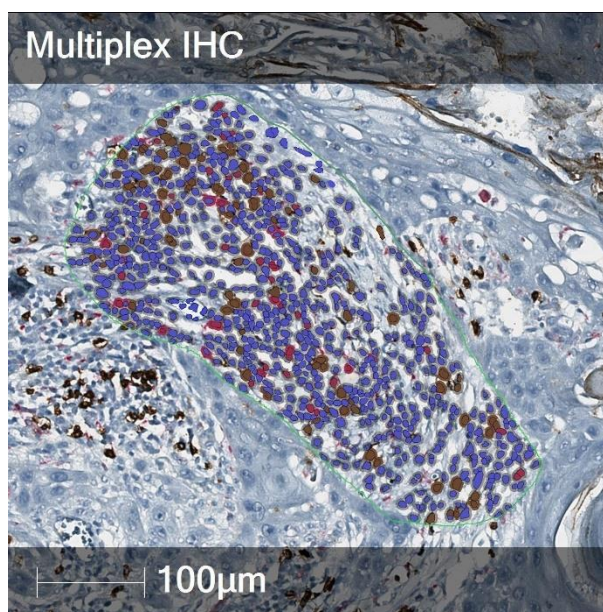


Figura 5.9 - Cell detection in HALO

Il conteggio delle cellule tramite *HALO* è stato effettuato utilizzando la funzionalità di riconoscimento automatico fornita dal software. Tale procedura si avvale di una rete neurale precedentemente addestrata su immagini simili ma non identiche a quelle impiegate per il test in questione. Alla fine, è stato individuato un totale di 815 cellule.

5.4.6 ImageJ/Fiji

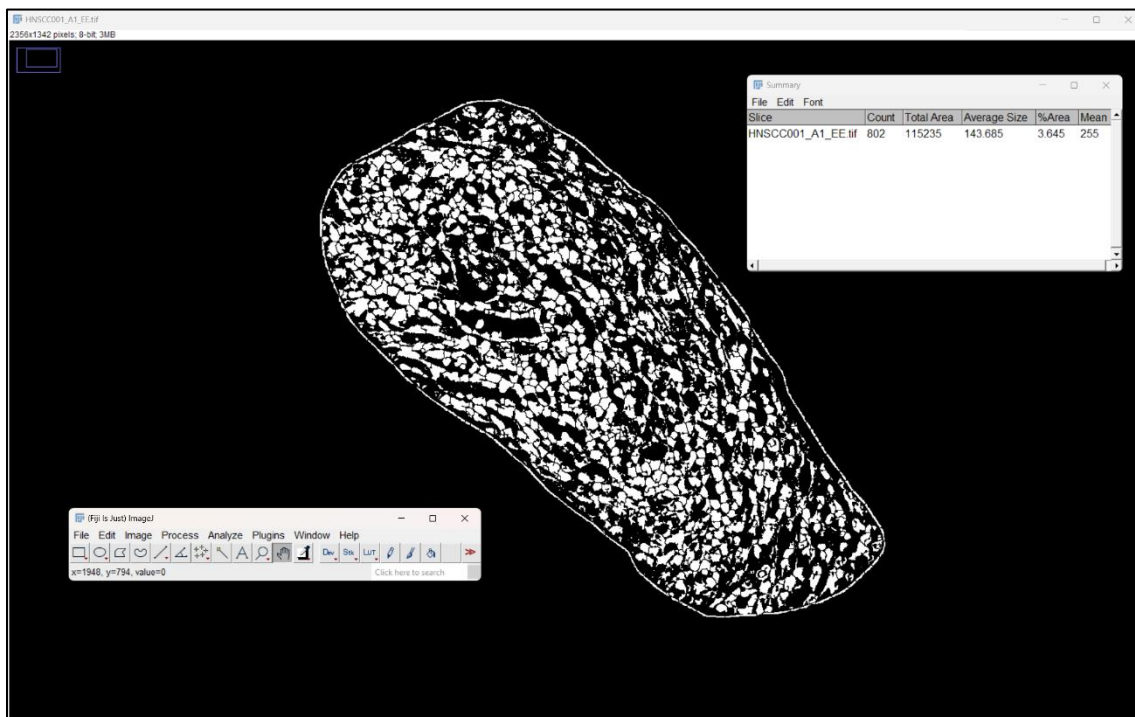


Figura 5.10 - Cell count con ImageJ/Fiji

Il processo di conteggio delle cellule tramite *ImageJ/Fiji* coinvolge diverse fasi. Inizialmente, l'immagine è stata sottoposta a deconvoluzione per rimuovere i canali rosso e verde. Successivamente, è stata delineata l'area di interesse utilizzando lo strumento di selezione a mano libera e da questa è stata creata una maschera binaria. Tale maschera, dopo essere stata invertita, è stata quindi sottratta dall'immagine originale. Successivamente, l'immagine è stata convertita in un formato binario, applicando una soglia compresa tra 0 e 175. L'algoritmo di Watershed è stato quindi applicato per la segmentazione delle cellule. Infine, attraverso il plugin *Particle Analysis*, è stato eseguito il conteggio delle cellule totali, risultando in un totale di 802.

I principali parametri utilizzati sono i seguenti:

- area minima (60 pixel²)
- circolarità (0,00 – 1,00)
- esclusione delle cellule ai bordi (SI)

5.4.7 NIS-Elements

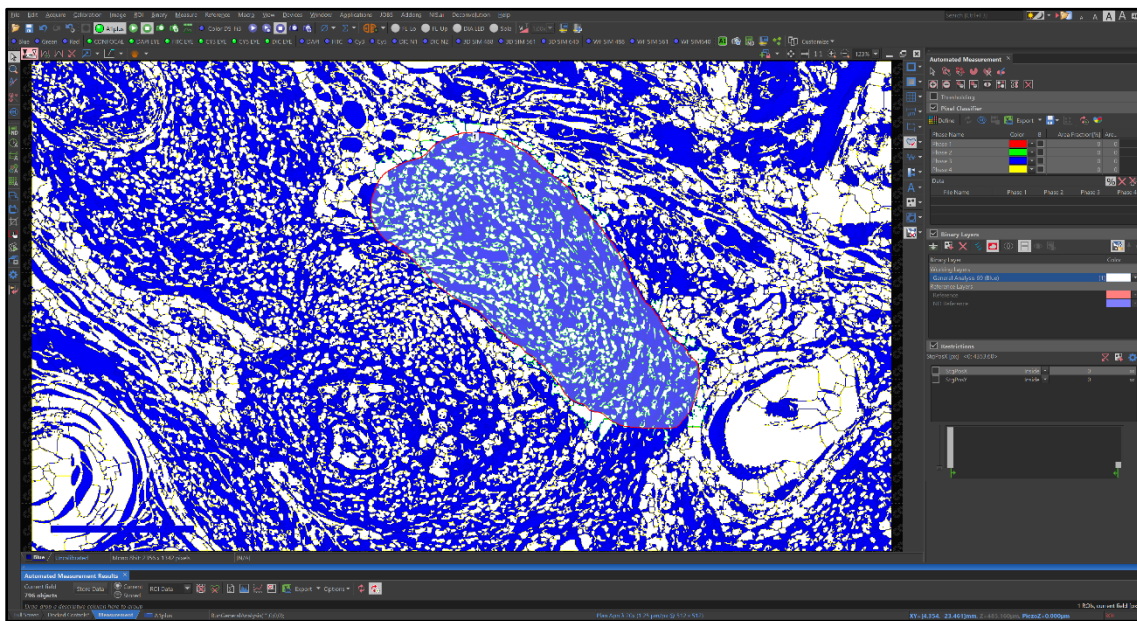


Figura 5.11 - Cell detection in NIS-Elements

Per eseguire il conteggio automatico dei nuclei in *NIS-Elements*, si è proceduto rimuovendo prima i canali di colorazione rosso e verde dall'immagine mediante la tecnica di color deconvolution. L'algoritmo di conteggio ha identificato complessivamente 796 cellule.

I parametri adottati includono:

- valori di soglia (0 – 225)
- *Smooth* (2x)
- *Clean* (2x)
- Riempimento dei buchi (NO)
- *Separate* (2x)

5.4.8 QuPath

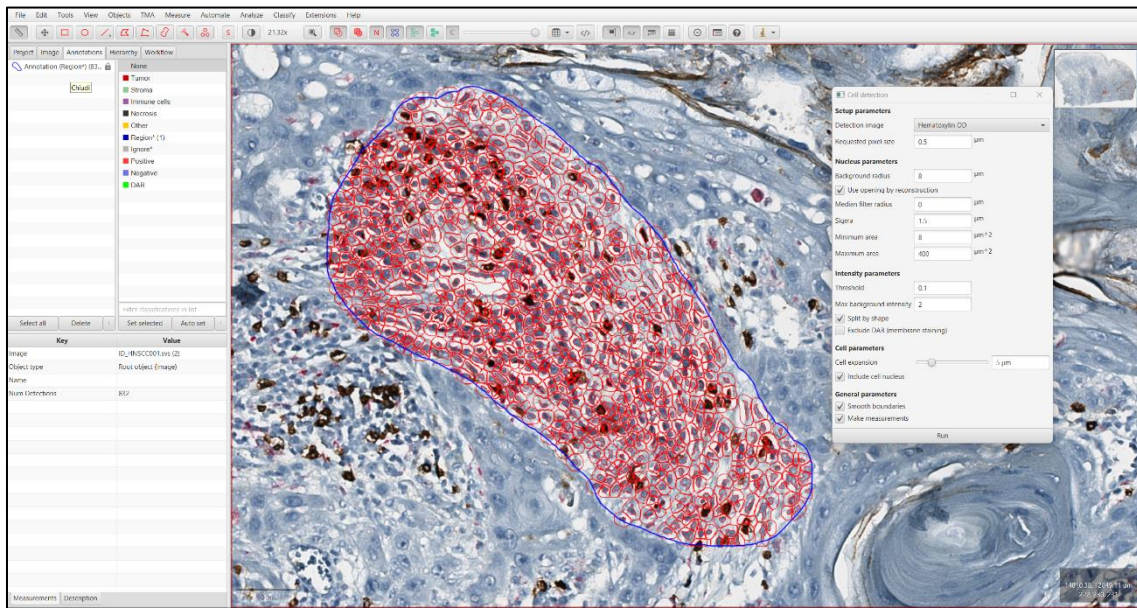


Figura 5.12 - Cell detection in QuPath

La rilevazione automatica in *QuPath* ha conteggiato 832 cellule totali.

Sono stati utilizzati i seguenti parametri per la rilevazione:

- dimensione dei pixel ($0.5 \mu\text{m}$)
- raggio di sfondo del nucleo ($8 \mu\text{m}$)
- raggio di filtro mediano ($0 \mu\text{m}$)
- sigma ($1.5 \mu\text{m}$)
- area minima del nucleo ($8 \mu\text{m}^2$)
- area massima del nucleo ($400 \mu\text{m}^2$)
- valore di soglia (0.1)
- intensità massima di sfondo (2.0)
- espansione cellulare ($5 \mu\text{m}$)

5.4.9 Discussione dei risultati

Tabella 6 - Confronto dei risultati sperimentali: conteggio nuclei totali

Software	Conteggio nuclei	Errore assoluto	Rank
<i>Ground truth</i>	823	//	//
<i>Amira</i>	871	5.83%	6
<i>Arivis</i>	645	21.62%	7
<i>CellProfiler</i>	816	0.85%	1
<i>HALO</i>	815	0.97%	2
<i>ImageJ/Fiji</i>	802	2.55%	4
<i>NIS-Elements</i>	796	3.28%	5
<i>QuPath</i>	832	1.09%	3

Per analizzare e confrontare i risultati, per ogni software testato è stato calcolato l'errore assoluto percentuale del risultato ottenuto rispetto al valore di ground truth, utilizzando la seguente formula:

$$\frac{|valore\ ottenuto - ground\ truth|}{ground\ truth} \times 100$$

Dall'analisi dei risultati, emerge che i software più precisi nel conteggio dei nuclei totali all'interno di una regione di interesse risultano essere *CellProfiler*, *HALO* e *QuPath*, con un errore assoluto di circa 1% rispetto alla ground truth. Va sottolineato però che anche le soluzioni di *Amira*, *ImageJ/Fiji* e *NIS-Elements* hanno ottenuto dei risultati accettabili, con un errore inferiore al 6%.

5.5 Conteggio con classificazione dei marcatori: risultati sperimentali

5.5.1 Tool idonei

Software open source:

- *CellProfiler* – **escluso**. Non vi sono funzionalità per la classificazione automatica dei marcatori.
- *Cytomine* – **escluso**. Il software non offre una funzionalità per il conteggio automatico delle cellule.

- *Digital Slide Archive* – **escluso**. L'algoritmo di rilevamento automatico su questo software permette solamente di definire una ROI rettangolare nella quale venire eseguito; inoltre viene effettuato il rilevamento ma non il conteggio e l'estrapolazione di dati.
- *Icy* – **escluso**. Nonostante offra diverse funzionalità per l'analisi, non è possibile effettuare il rilevamento automatico su immagini istologiche a campo chiaro.
- *ImageJ/Fiji* – **incluso**.
- *QuPath* – **incluso**.

Software *freely-available*:

- *Aperio ImageScope* – **escluso**. Trattandosi di un visualizzatore di immagini i cui algoritmi consentono solo di estrapolare la conta dei pixel e non la conta delle cellule.
- *NIS-Elements Viewer* – **incluso**, ma nella sua versione commerciale.
- *Sedeen Viewer* – **escluso**. Sebbene offra diversi algoritmi di analisi, non è possibile eseguire il conteggio automatico.
- *SlideViewer* – **escluso**. Il software offre solo funzionalità per il conteggio manuale delle cellule.

Software commerciali:

- *Amira* – **incluso**.
- *Arivis* – **incluso**.
- *HALO* – **incluso**.
- *Imaris* – **escluso**. Il software offre funzionalità altamente avanzate, ma si concentra sull'analisi di dataset in tre e quattro dimensioni e di immagini a fluorescenza.

5.5.2 Amira

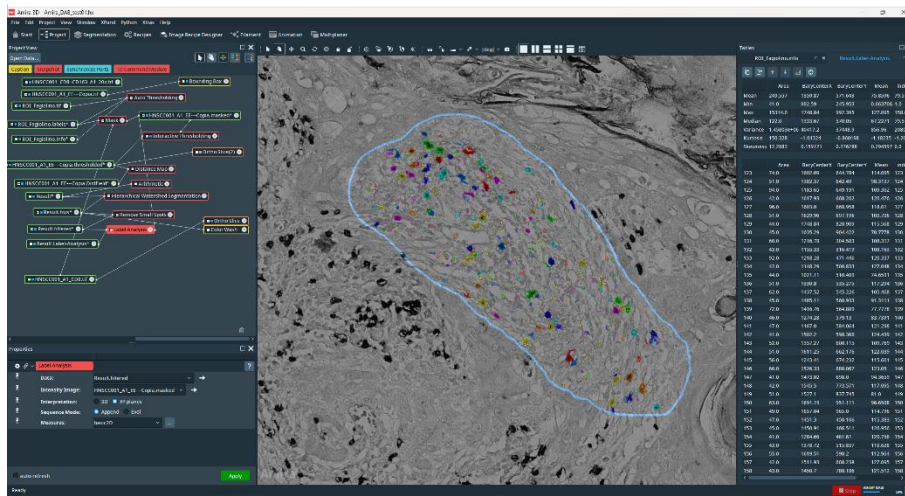


Figura 5.13 - Conteggio nuclei DAB con Amira

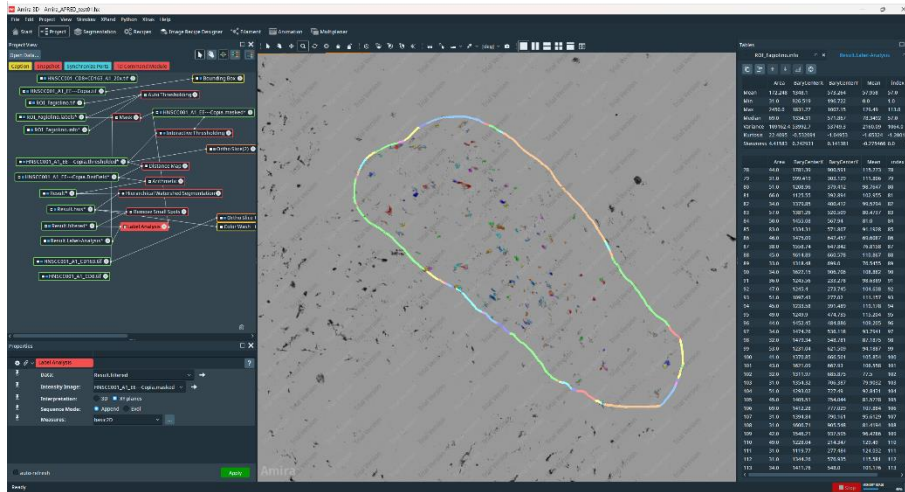


Figura 5.14 - Conteggio nuclei AP-Red con Amira

Per classificare e conteggiare i nuclei con *Amira* è stato prima necessario applicare un passaggio di color deconvolution all'immagine originale, ottenendo così due diverse immagini in scala di grigi sulle quali svolgere l'analisi. Il processo di analisi è analogo a quello utilizzato per il conteggio dei nuclei totali, quindi prevede l'applicazione di una maschera binaria per la selezione della regione di interesse, un'operazione di thresholding, una segmentazione Watershed e la rimozione delle macchie più piccole.

L'analisi del marcatore DAB ha ottenuto un conteggio di 158 nuclei, utilizzando i seguenti parametri principali:

- Valori di soglia (0 – 140)
- Modalità segmentazione Watershed (DARK)
- Range globale segmentazione (1,00 – 6,75)

- Livello di dettaglio segmentazione (0,8)
- Dimensione minima per rimozione small spots (40 pixel)

L'analisi del marcatore AP-Red ha invece ottenuto un conteggio di 113 nuclei, con i seguenti parametri principali:

- Valori di soglia (0 – 150)
- Modalità segmentazione Watershed (DARK)
- Range globale segmentazione (1,00 – 6,75)
- Livello di dettaglio segmentazione (0,6)
- Dimensione minima per rimozione small spots (30 pixel)

5.5.3 Arivis

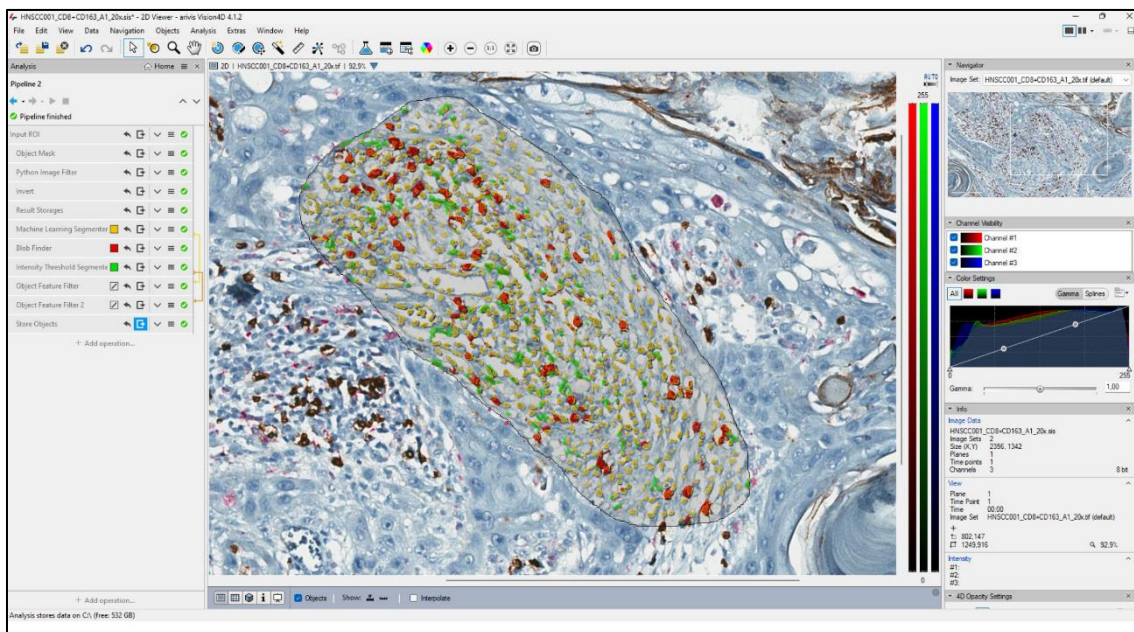


Figura 5.15 - Classificazione marcatori con Arivis

Con *Arivis* è possibile individuare e classificare diversi marcatori all'interno dell'immagine utilizzando un passaggio di color deconvolution nel flusso di analisi. I risultati ottenuti sono i seguenti: 168 nuclei marcati DAB, 120 nuclei marcati AP-Red.

5.5.4 HALO

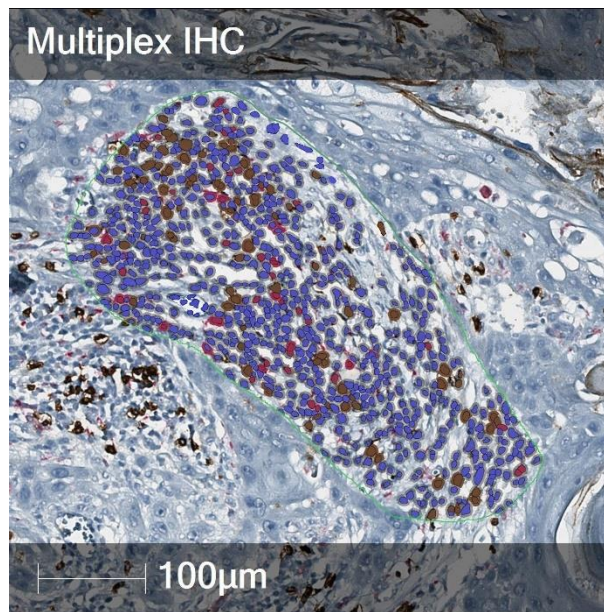


Figura 5.16 - Classificazione marcatori con HALO

La classificazione automatica dei marcatori fornita da *HALO* ha ottenuto i seguenti conteggi: 94 nuclei marcati DAB, 51 nuclei marcati AP-Red. La classificazione è stata eseguita utilizzando la funzionalità di riconoscimento automatico basato su machine learning fornita da *HALO*.

5.5.5 ImageJ/Fiji

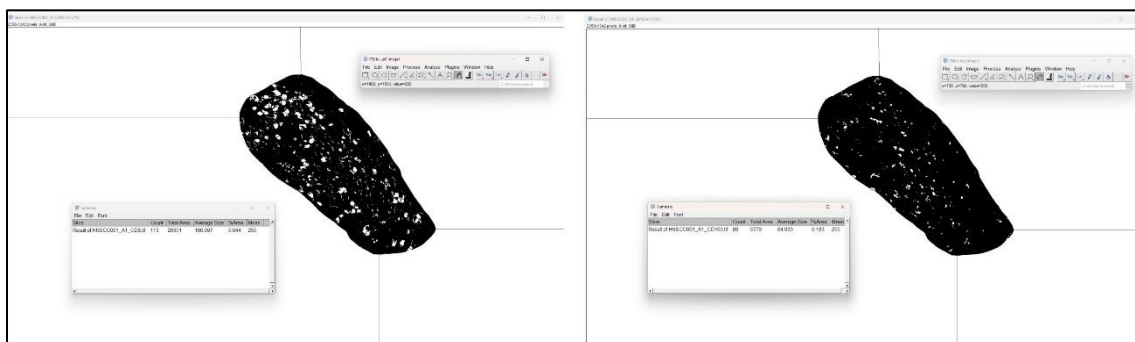


Figura 5.17 - Classificazione marcatori con ImageJ/Fiji

Similmente a quanto fatto per il conteggio dei nuclei totali utilizzando *ImageJ/Fiji*, l'immagine è stata prima sottoposta a un processo di color deconvolution per separare i tre diversi canali di colorazione. L'analisi del marcatore DAB ha ottenuto un conteggio di 113 nuclei e sono stati utilizzati i seguenti parametri:

- Valori di soglia (0 – 140)
- Dimensione (min 60 pixel², max 1000 pixel²)

- Circolarità (0,00 – 1,00)

Per quanto riguarda il marcatore AP-Red, il software ha conteggiato un totale di 89 nuclei. I parametri utilizzati sono i seguenti:

- Valori di soglia (0 – 175)
- Dimensione (min 30 pixel², max 1000 pixel²)
- Circolarità (0,00 – 1,00)

5.5.6 *NIS-Elements*

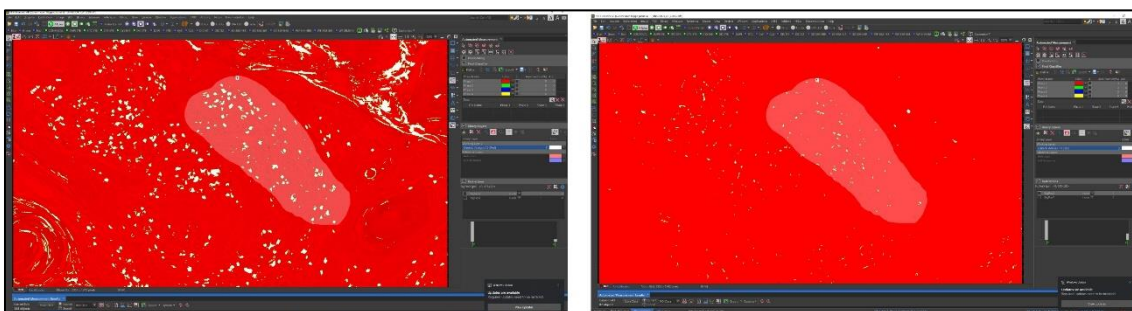


Figura 5.18 - Classificazione marcatori con *NIS-Elements*

Anche con *NIS-Elements* è necessario processare l'immagine mediante color deconvolution per poter analizzare e conteggiare i diversi marcatori. Per quanto riguarda il marcatore DAB, sono state conteggiati 128 nuclei, utilizzando questi parametri:

- Valori di soglia (0 – 215)
- *Smooth* (2x)
- *Clean* (2x)
- Riempimento dei buchi (SI)
- *Separate* (2x)

Per il marcatore AP-Red sono stati individuati 87 nuclei, e nell'analisi sono stati applicati i seguenti parametri:

- Valori di soglia (0 – 230)
- *Smooth* (2x)
- *Clean* (2x)
- Riempimento dei buchi (SI)
- *Separate* (2x)

5.5.7 QuPath

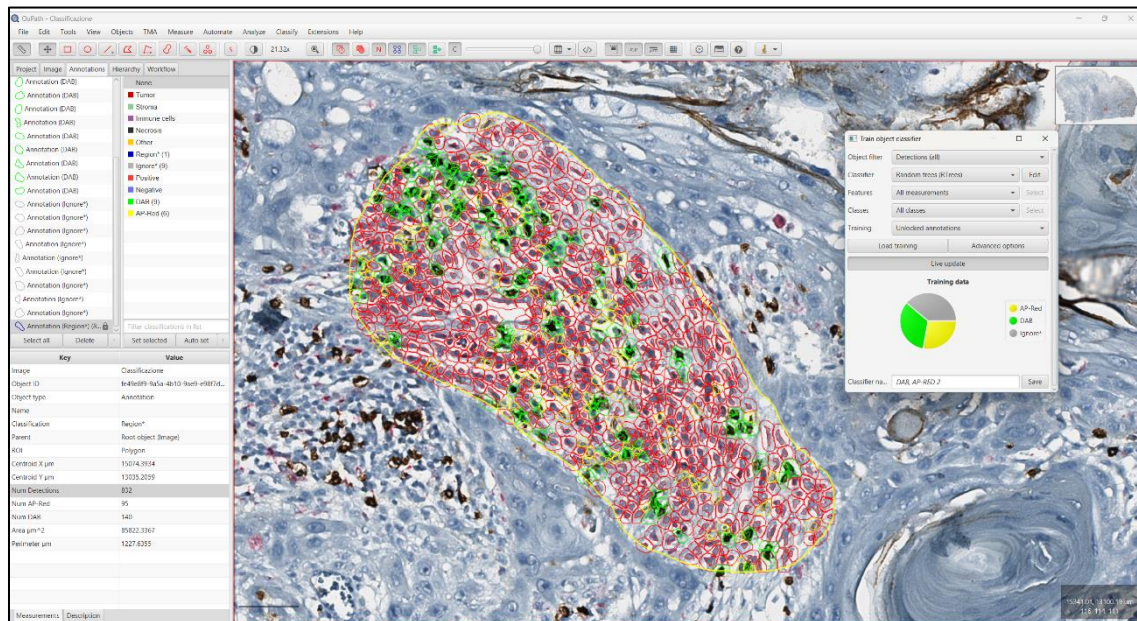


Figura 5.19 - Classificazione marcatori con QuPath

QuPath offre uno strumento di classificazione delle cellule basato su una rete neurale. Per utilizzarlo, è necessario prima creare e addestrare un classificatore di oggetti. Questo processo inizia con la definizione e l'annotazione manuale di una selezione di nuclei all'interno dell'immagine, classificandoli in base ai marcatori che li colorano. Il classificatore utilizzerà questi dati forniti dall'utente per creare un modello di riconoscimento dei marcatori. Una volta completato l'addestramento, il classificatore sarà in grado di rilevare automaticamente i marcatori in qualsiasi regione annotata dall'utente, dove sia stato eseguito un rilevamento automatico delle cellule. Nel nostro caso, l'addestramento è stato effettuato classificando manualmente una trentina di nuclei, circa dieci per ciascun marcatore. I risultati finali ottenuti sono stati: 142 nuclei con marcatore DAB e 101 nuclei con marcatore AP-Red.

5.5.8 Discussione dei risultati

Tabella 7 - Confronto dei risultati sperimentali: classificazione DAB

Software	Conteggio DAB	Errore assoluto	Rank
<i>Ground truth</i>	135	//	//
<i>Amira</i>	158	17,03%	4
<i>Arivis</i>	168	24.44%	5
<i>HALO</i>	94	30.37%	6
<i>ImageJ/Fiji</i>	113	16.30%	3
<i>NIS-Elements</i>	128	5.18%	1
<i>QuPath</i>	142	5.18%	1

Tabella 8 - Confronto dei risultati sperimentali: classificazione AP-Red

Software	Conteggio AP-Red	Errore assoluto	Rank
<i>Ground truth</i>	98	//	//
<i>Amira</i>	113	15,31%	4
<i>Arivis</i>	120	22.45%	5
<i>HALO</i>	51	47.96%	6
<i>ImageJ/Fiji</i>	89	9.18%	2
<i>NIS-Elements</i>	87	11.22%	3
<i>QuPath</i>	101	3.06%	1

Analogamente a come svolto per il conteggio dei nuclei totali, è stato calcolato per ogni software l'errore assoluto percentuale sia per la classificazione del marcatore DAB che per la classificazione del marcatore AP-Red. La formula utilizzata per il calcolo dell'errore è la seguente:

$$\frac{|valore\ ottenuto - ground\ truth|}{ground\ truth} \times 100$$

Analizzando i risultati sperimentali ottenuti, emerge che *QuPath* risulta essere il software più preciso per la classificazione dei marcatori e conteggio dei nuclei, con errori assoluti del 5,18% per il marcatore DAB e del 3,06% per il marcatore AP-Red. Risultati accettabili sono stati ottenuti

anche da *NIS-Elements* (5,18% DAB, 11,22% AP-Red) e da *ImageJ/Fiji* (16,30% DAB, 9,18% AP-Red). Per quanto riguarda gli altri software, gli errori risultano troppo elevati per essere considerati precisi ed affidabili.

6 Conclusioni e sviluppi futuri

Questo studio è nato da una specifica necessità del gruppo di ricerca "*Data Science for Health*" (DS4H): analizzare e approfondire le diverse soluzioni software disponibili per lo svolgimento di concrete operazioni di analisi su immagini di provini istologici, con una particolare attenzione all'applicativo *QuPath*. Il gruppo di ricerca richiedeva a sostegno di alcuni biologi una soluzione software che permettesse di svolgere analisi quantitative (e.g. conteggio di cellule, estrapolazione di misurazioni) in regioni di interesse determinate dall'operatore, e che fosse intuitiva e facile da utilizzare per gli utenti non esperti in campo informatico.

In questa Tesi ho ricercato, studiato e testato i principali software in commercio per l'analisi di immagini istologiche, considerando i requisiti specifici richiesti dai ricercatori di DS4H. Successivamente, mi sono concentrato esclusivamente su *QuPath*, descrivendone tutte le funzionalità in una sorta di guida utente essenziale per il suo utilizzo. Infine, seguendo un vero e proprio processo sperimentale, ho analizzato un caso di studio reale utilizzando i diversi software elencati in precedenza, ottenendo dati empirici per confrontarne e valutarne la precisione e l'affidabilità.

Tabella 9 - Ranking precisione software per errore medio

Rank	Software	Errore medio
1	<i>QuPath</i>	3.11%
2	<i>NIS-Elements</i>	6.56%
3	<i>ImageJ/Fiji</i>	9.34%
4	<i>Amira</i>	12.72%
5	<i>Arivis</i>	22.84%
6	<i>HALO</i>	26.43%

I risultati sperimentali, sintetizzati nella [Tabella 8](#), dimostrano che *QuPath* è lo strumento più preciso per il caso di utilizzo esposto dai ricercatori di DS4H, seguito da *NIS-Elements* e *ImageJ/Fiji*, che hanno ottenuto dei buoni risultati, con errori medi inferiori al 10%. *QuPath* si

distingue anche per la sua facilità d'uso per utenti non esperti in campo informatico, in particolare per la semplicità con cui è possibile svolgere analisi complesse come il rilevamento automatico dei nuclei o il training di una rete neurale per la classificazione automatica dei marcatori. Di non minore importanza, *QuPath* è un software open-source, quindi gratuito e facilmente estendibile tramite plugin o script personalizzati.

È importante sottolineare che in questa valutazione sono stati considerati principalmente i requisiti e le funzionalità richiesti dal gruppo di ricerca. Altri software (e.g. *Arivis*, *HALO*) offrono strumenti avanzati per analisi molto più complesse, come lo studio di dati multidimensionali, il tracciamento delle cellule nel tempo e l'analisi supportata da intelligenza artificiale. In conclusione, *QuPath* risulta essere lo strumento più adatto per le necessità dei ricercatori di DS4H, considerando anche usabilità e costo.

Questa ricerca contribuisce alla letteratura esistente fornendo una panoramica dettagliata delle diverse tecnologie digitali che supportano il lavoro di analisi e studio dei ricercatori nel campo delle *Life Sciences*, rappresentando un utile strumento a supporto della scelta del software più adatto in base alle specifiche necessità. Questo lavoro non vuole tuttavia essere una versione definitiva su questo argomento. Saranno necessarie ulteriori ricerche per esplorare diversi casi di studio e sperimentare altre funzionalità non coperte in questa Tesi, in modo da avere un quadro più completo dello stato dell'arte per quanto riguarda l'analisi digitale di provini istologici.

In sintesi, dopo aver fornito una panoramica generale sui vari software attualmente disponibili, questo studio dimostra che, per lo specifico caso di studio proposto dal gruppo di ricerca DS4H, *QuPath* risulta essere lo strumento più preciso, affidabile e facile da utilizzare.

Al momento, l'analisi svolta si è focalizzata esclusivamente sul conteggio delle cellule, senza considerare l'accuratezza delle maschere di segmentazione delle singole cellule. In altre parole, non è stata eseguita un'analisi approfondita delle maschere per valutare la presenza di falsi positivi o falsi negativi. L'attuale valutazione è quindi limitata al semplice conteggio e non riflette l'accuratezza o la precisione della segmentazione cellulare. Lo sviluppo futuro di questa ricerca prevede un'analisi della qualità e della precisione delle maschere di segmentazione mediante metriche pixel-based, come l'indice di Jaccard per la similarità [22].

I risultati ottenuti sono stati presentati alla seguente conferenza:

- Poster presentation: “Comparative analysis of commercial, freely-available, and open-source software for single-cell analysis within a histological image ROI”, Filippo Piccinini, Filippo Pilutti, Lorenzo Rigoni, Mariachiara Stellato, Marcella Tazzari, Enrico Giampieri, Daniel Remondini, Gastone Castellani, Antonella Carbonaro. PhD Summer School on Mathematics and Machine Learning for Image Analysis, June 4-12, 2024, University of Bologna, Bologna, Italy.

Inoltre, un abstract riassuntivo del lavoro svolto è stato recentemente accettato in forma di Oral Presentation alla seguente conferenza:

- Oral presentation: “Comparative analysis of commercial, freely-available, and open-source software for single-cell analysis within a histological image ROI”, Filippo Piccinini, Filippo Pilutti, Lorenzo Rigoni, Mariachiara Stellato, Marcella Tazzari, Enrico Giampieri, Daniel Remondini, Gastone Castellani, Antonella Carbonaro. XXIII International Conference on Mechanics in Medicine and Biology (ICMMB), September 11-13, 2024, Bruxelles, Belgium.

e sarà in futuro oggetto di pubblicazione scientifica.

Bibliografia

- [1] Marchetti, M. Segmentazione *automatica di regioni in immagini istologiche* (Doctoral dissertation)
- [2] Rolls, G. (2012) Process of fixation and the nature of fixatives. *Fixation*
- [3] <https://www.uottawa.ca/research-innovation/histology/services/histological-staining#:~:text=Histological%20staining%20is%20used%20to,categories%3A%20Routine%20and%20Special%20Stains>.
- [4] Carpenter, A. E., Jones, T. R., Lamprecht, M. R., Clarke, C., Kang, I. H., Friman, O., ... & Sabatini, D. M. (2006). CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome biology*, 7, 1-11.
- [5] Marée, R. A. P. H. A. Ë. L., Rollus, L., Stevens, B., Hoyoux, R., Louppe, G., Vandaele, R., ... & Wehenkel, L. (2016). Cytomine: An open-source software for collaborative analysis of whole-slide images. *Diagnostic Pathology*, 1(8).
- [6] Gutman, D. A., Khalilia, M., Lee, S., Nalisnik, M., Mullen, Z., Beezley, J., ... & Cooper, L. A. (2017). The digital slide archive: a software platform for management, integration, and analysis of histology for cancer research. *Cancer research*, 77(21), e75-e78.
- [7] de Chaumont, F., Dallongeville, S., & Olivo-Marin, J. C. (2011, March). ICY: A new open-source community image processing software. In *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro* (pp. 234-237). IEEE.
- [8] Schindelin, J., Rueden, C. T., Hiner, M. C., & Eliceiri, K. W. (2015). The ImageJ ecosystem: An open platform for biomedical image analysis. *Molecular reproduction and development*, 82(7-8), 518-529.
- [9] Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., ... & Cardona, A. (2012). Fiji: an open-source platform for biological-image analysis. *Nature methods*, 9(7), 676-682.
- [10] Bankhead, P., Loughrey, M. B., Fernández, J. A., Dombrowski, Y., McArt, D. G., Dunne, P. D., ... & Hamilton, P. W. (2017). QuPath: Open source software for digital pathology image analysis. *Scientific reports*, 7(1), 1-7.
- [11] <https://www.leicabiosystems.com/it-it/digital-pathology/gestione/aperio-imagescope/>
- [12] Why, N. I. S. (2022). Nikon offers total software solution covering image capture, archiving, and analysis. *Imaging*, 9100(02), C9100-12.
- [13] <https://pathcore.com/sedeen>
- [14] <https://www.3dhistech.com/solutions/caseviewer/>

- [15] Stalling, D., Westerhoff, M., & Hege, H. C. (2005). Amira: A highly interactive system for visual data analysis. *The visualization handbook*, 38, 749-767.
- [16] <https://indicalab.com/halo/>
- [17] <https://imaris.oxinst.com>
- [18] <https://groovy-lang.org/>
- [19] Koenig, D., Glover, A., King, P., Laforge, G., & Skeet, J. (2007). *Groovy in action*. Manning Publications Co..
- [20] Beucher, S. (1994). Watershed, hierarchical segmentation and waterfall algorithm. *Mathematical morphology and its applications to image processing*, 69-76.
- [21] Ruifrok, A. C., & Johnston, D. A. (2001). Quantification of histochemical staining by color deconvolution. *Analytical and quantitative cytology and histology*, 23(4), 291-299.
- [22] Bertels, J., Eelbode, T., Berman, M., Vandermeulen, D., Maes, F., Bisschops, R., & Blaschko, M. B. (2019). Optimizing the dice score and jaccard index for medical image segmentation: Theory and practice. In *Medical Image Computing and Computer Assisted Intervention—MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part II 22* (pp. 92-100). Springer International Publishing.

Ringraziamenti

Desidero esprimere la mia sincera gratitudine a tutte le persone che mi hanno sostenuto durante la stesura di questa Tesi e il mio percorso universitario.

In primis, vorrei ringraziare la Prof.ssa Antonella Carbonaro dell'Università di Bologna, che mi ha seguito nel ruolo di Relatrice durante la preparazione e lo svolgimento di questa Tesi, per avermi dato l'opportunità di lavorare a questo progetto multidisciplinare e di poter contribuire a un ambito molto stimolante come quello del *"Data science for Health"*.

Desidero inoltre ringraziare la Dott.ssa Marcella Tazzari, principal investigator della IRST *"Immunotherapy Cell Therapy and Biobank (ITCB)"* unit, ed il Prof. Gastone Castellani, direttore della Scuola di Specializzazione in Fisica Medica dell'Università di Bologna, per aver reso possibile questo progetto fornendo l'utilizzo delle immagini provenienti da campioni istologici di pazienti reali acquisiti presso l'Istituto Romagnolo per lo Studio dei Tumori "Dino Amadori" IRCCS di Meldola.

Uno speciale ringraziamento va al Dott. Filippo Piccinini per il suo ruolo di correlatore in questa Tesi. La sua costante presenza e disponibilità, i suoi preziosi suggerimenti e il continuo supporto sono stati essenziali per raggiungere gli obiettivi che ci eravamo preposti. Ha condiviso in maniera chiara ed esauriente le sue vaste conoscenze nel campo della microscopia e dell'oncologia, rendendo più comprensibili argomenti e procedure complessi. La sua gentilezza, competenza e professionalità hanno contribuito ad arricchire non solo questo progetto di Tesi, ma anche me come studente e come persona, e per questo gliene sono grato.

Vorrei dedicare un pensiero e un ringraziamento alla mia ragazza Isabel, ai miei amici e a tutti coloro che mi sono stati accanto e mi hanno sostenuto durante tutto il percorso universitario e la stesura di questa Tesi. Queste persone mi hanno aiutato a prendere la vita con più leggerezza e a dare il giusto peso alle cose.

Infine, il ringraziamento più importante va alla mia famiglia, che dal primo all'ultimo momento mi ha sempre supportato incondizionatamente, nonostante tutte le difficoltà e gli ostacoli che abbiamo dovuto affrontare. I loro sacrifici, la loro pazienza e il loro amore mi hanno dato la forza di superare molte sfide, sia in campo accademico che nella vita. È grazie a loro che oggi sono la persona che

sono, e di questo non me ne dimenticherò mai. Il loro esempio di dedizione e resilienza mi ispira ogni giorno, e porterò sempre con me i valori e le lezioni che mi hanno trasmesso. Spero in futuro di poter ripagare un giorno tutto ciò che hanno fatto per me e continuare a renderli orgogliosi.