



**UNIVERSITÀ
DEGLI STUDI
DI PADOVA**



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN BIOINGEGNERIA

**Simulazione delle interazioni tra cellule dendritiche e cellule T del sistema
immunitario durante un esperimento di labeling enzimatico**

Relatore: Prof. Del Favero Simone

Laureanda: De Rossi Giulia

Correlatore: Ing. Bellato Massimo, Dott.ssa Cimolato Chiara

ANNO ACCADEMICO 2022 – 2023

Data di laurea 10/07/2023

ABSTRACT

In questo elaborato viene trattata la simulazione di un sistema di rilevamento delle interazioni tra le cellule dendritiche e cellule T, componenti del sistema immunitario il cui compito è quello di degradare agenti patogeni entrati nel nostro corpo. La simulazione cerca di riprodurre il funzionamento di una tecnica sviluppata da un gruppo di ricercatori e presente in letteratura chiamata LIPSTIC: tale sistema ingegnerizzato utilizza un fluoroforo come etichetta rilevabile dell'interazione tra le due cellule, sfruttando l'aumento di probabilità di passaggio del fluoroforo da una cellula all'altra in caso di avvenuta interazione.

Tramite l'utilizzo della piattaforma Matlab, il simulatore sviluppato permette di far vedere attraverso un video le cellule che si muovono ed interagiscono tra loro passandosi, il fluoroforo. Sono state quindi stimate quante interazioni e passaggi di fluoroforo avvengono (veri positivi) e calcolate la sensibilità, specificità e precisione del sistema. I risultati ottenuti mostrano che tale simulazione è una buona rappresentazione dell'approccio LIPSTIC.

INDICE

Introduzione	5
Capitolo 1: Il sistema immunitario	7
1.1 Immunità innata e immunità adattativa.....	8
1.2 La cellula dendritica	10
1.3 I linfociti T.....	12
1.4 LIPSTIC: Monitoraggio delle interazioni tra cellule T e cellule dendritiche mediante etichettatura enzimatica intercellulare.....	14
Capitolo 2: Simulatore semplificato ignorando il carico del fluoroforo	18
2.1 Simulatore con le cellule immunitarie distribuite casualmente e riempite di fluoroforo.....	20
2.2 Simulatore con le cellule immunitarie distribuite in modo ordinato e riempite di fluoroforo.....	26
Capitolo 3: Simulatore che include l'interazione con il fluoroforo	31
3.1 Simulatore con le cellule immunitarie e il fluoroforo distribuiti in modo casuale.....	31
3.2 Simulatore con le cellule immunitarie distribuite in modo ordinato e il fluoroforo sparso nel reticolo.....	38
Capitolo 4: Analisi dei risultati	39
4.1 Caso in cui il fluoroforo si esaurisce.....	41
4.2 Caso in cui il fluoroforo non si esaurisce.....	46
Conclusioni	51
Ringraziamenti	53

Bibliografia e sitografia	54
Appendice	57

INTRODUZIONE

In questa tesi si è cercato di simulare le interazioni che ci possono essere tra i linfociti T e le cellule dendritiche, due cellule del sistema immunitario che hanno il compito di degradare agenti patogeni entrati nel nostro corpo, in una procedura che è stata sviluppata da un gruppo di ricercatori e che prende il nome di LIPSTIC in cui il passaggio dell'antigene tra le due cellule è rilevato utilizzando un'etichetta come la biotina o fluoroforo tramite citometria a flusso [1],[2].

I ricercatori hanno sviluppato questa tecnica perché il passaggio del fluoroforo tra sortasi caricata e oligo-guanina è inefficiente mentre il legame tra recettore e ligando che avviene in LIPSTIC è più duraturo ed efficiente. Quindi, poiché la probabilità del passaggio di fluoroforo è tanto maggiore quanto maggiore è il tempo di esposizione o vicinanza tra le cellule, con LIPSTIC si ha più possibilità di passaggio del fluoroforo e di rilevare l'interazione tra le due cellule immunitarie.

In particolare, nel primo capitolo viene trattato cosa si intende per sistema immunitario [3],[4],[5], quali sono le cellule che ne fanno parte evidenziando i ruoli della cellula dendritica [10],[11],[12] e del linfocita T [13],[14]. La cellula dendritica, appartenente all'immunità innata, risposta immunitaria che avviene per prima, presenta l'antigene ai linfociti T: gli antigeni che vengono fagocitati sono esposti sulla membrana esterna della cellula e sono riconosciuti dai linfociti, attivando così la risposta immunitaria per l'antigene presentato.

I linfociti T appartengono all'immunità adattativa che avviene più tardivamente. Ci sono diverse tipologie di linfociti T che si differenziano tra di loro in base alle proteine prodotte e alle funzionalità, molti dei quali vengono attivati dagli antigeni trasportati dalle cellule dendritiche. Iniziano a proliferare in cloni specifici distruggendo l'antigene.

Viene inoltre descritta la procedura di monitoraggio delle interazioni tra le cellule T e dendritiche realizzata da degli studiosi di nome LIPSTIC. Tramite l'utilizzo di un'etichetta rilevabile come il fluoroforo si monitora l'interazione tra la cellula dendritica e la cellula T: dall'interazione tra i due tipi di cellule il fluoroforo passa da una cellula all'altra.

Nel secondo capitolo viene mostrato come ha inizio l'implementazione dei simulatori utilizzando la piattaforma Matlab: ne vengono create quattro versioni, a complessità incrementale, distinguendo il caso in cui le cellule immunitarie sono precaricate col fluoroforo oppure quest'ultimo è sparso nell'ambiente circostante (reticolo). Si ha anche la distinzione del caso in cui le cellule sono disposte in modo ordinato oppure in modo casuale nel reticolo.

In particolare, inizialmente viene spiegato, attraverso lo pseudocodice, il simulatore con le cellule immunitarie disposte in modo casuale e precaricate di fluoroforo. Inoltre, è stato creato un video in cui si vedono le cellule muoversi all'interno del reticolo, interagire tra loro e passarsi il fluoroforo.

Dopo di che si passa al simulatore con le cellule immunitarie disposte in modo ordinato e precaricate di fluoroforo per aiutare il programmatore a rilevare eventuali errori nella procedura di visualizzazione del video.

Nel terzo capitolo si può vedere come viene realizzato il simulatore con le cellule immunitarie e il fluoroforo sparsi casualmente nel reticolo. In questo caso si ha anche il conteggio di varie variabili come i veri positivi, i veri negativi, i falsi positivi ed i falsi negativi, che verranno utilizzati nel capitolo dell'analisi dei risultati. In questo caso viene sfruttata la sola versione del simulatore con le cellule immunitarie disposte in modo casuale e il fluoroforo sparso nel reticolo.

Nell'ultimo capitolo si ha, invece, la visualizzazione dei parametri introdotti nel capitolo precedente, attraverso delle figure riportanti il loro andamento tramite la media, varianza e deviazione standard all'aumentare della durata della simulazione. Si pone particolare attenzione alla variabile che conta i veri positivi che indicano l'evento in cui una cellula T si lega e passa il fluoroforo a una cellula dendritica per la prima volta poiché è il caso oggetto del nostro studio. Per concludere, vengono mostrate delle metriche statistiche delle performance del sistema come la sensibilità, precisione e "F1 score". In particolare, si può vedere la loro dipendenza dal fatto che il fluoroforo si esaurisca o meno, ossia se, quando una cellula immunitaria si carica di fluoroforo, quest'ultimo viene cancellato dal reticolo.

CAPITOLO 1: IL SISTEMA IMMUNITARIO

L'organismo, per mezzo del sistema immunitario, ha la capacità di reagire alla penetrazione di agenti patogeni e di tossine, con la formazione di sostanze difensive e cellule che sono sia sparse nell'organismo, sia concentrate in organi linfatici primari, che sono il midollo osseo e il timo, e secondari, i linfonodi, la milza, le adenoidi e le tonsille. Le sostanze che stimolano questa risposta immunitaria vengono chiamate antigeni e possono essere presenti sulla superficie di diversi agenti patogeni quali virus, batteri e funghi [3].

Il processo di formazione delle cellule immunitarie richiede un certo tempo e, quando sono state prodotte in quantità sufficiente, sono in grado di neutralizzare gli antigeni penetrati nell'organismo. Infatti, un individuo, il cui sangue contenga una certa quantità di cellule immunitarie contro un determinato antigene patogeno, è immune alla sua azione, cioè non si ammala né è destinato ad ammalarsi anche nel caso in cui la quantità di cellule presenti nel suo sangue si sia ridotta; il suo organismo è in grado ormai di rispondere all'antigene più rapidamente di un organismo che venga per la prima volta a contatto con un determinato agente patogeno [3], [6].

Questa azione difensiva, l'immunità, è rigidamente specifica, vale a dire che ogni cellula immunitaria agisce nei confronti di un antigene specifico; il fatto di essere immuni contro un determinato agente patogeno non significa quindi che si è protetti anche contro agenti imparentati con questo o con altri del tutto diversi. Ciò è molto importante per quelle malattie che possono essere causate da tipi diversi dello stesso agente patogeno, come per esempio la poliomielite e l'influenza.

La prima linea di difesa nell'uomo sono le barriere anatomiche ovvero la pelle, la mucosa del tratto respiratorio e la mucosa dell'intestino. Infatti, l'infezione può avvenire attraverso le superfici mucose interne dell'organismo poiché ogni patogeno ha una sua particolare modalità di trasmissione, per esempio il cibo e l'acqua che l'uomo consuma, possono essere contaminati da diversi patogeni.

Al di là delle barriere anatomiche, nel caso ci sia una minaccia più aggressiva e gli agenti infettivi passino queste barriere, si ha l'intervento di proteine solubili, chiamate proteine del complemento (per esempio le proteine antimicrobiche e la proteina C3) [4], [5].

Il successivo meccanismo di protezione è costituito dalle cellule dell'immunità innata. Le principali sono i macrofagi tissutali, i granulociti, le cellule *natural killer* (NK) e le cellule dendritiche [7].

Passando da un meccanismo al successivo avviene il reclutamento di cellule più efficienti che sono i linfociti dell'immunità adattativa rappresentate dalle cellule B (in condizioni opportune producono gli anticorpi) e dai linfociti T. Quest'ultimi maturano nel timo mentre i linfociti B nel midollo osseo

ed entrambi derivano dalle cellule staminali del midollo osseo, progenitrici di tutte le cellule del sangue.

Il nostro sistema immunitario è in grado di discriminare benissimo il *self* dal *non self*. Il *self* è l'insieme di tutte le strutture presenti nel nostro organismo, cioè proteine, cellule e tessuti, mentre il *non self* è tutto ciò che non proviene dal nostro organismo (parassiti, virus, batteri).

Sono però considerati *non self* anche gli organi trapiantati dalla stessa specie, che causano quindi una risposta immunitaria potente che porta al rigetto dell'organo se non si interviene per bloccarla [4].

1.1 IMMUNITA' INNATA E IMMUNITA' ADATTATIVA

La difesa contro i patogeni è assicurata dalle reazioni precoci dell'immunità innata e da quelle più tardive dell'immunità adattativa (vedi figura 1).

L'immunità innata (detta anche naturale o nativa) consiste in meccanismi di difesa cellulare e biochimici preesistenti all'infezione e pronti a reagire con rapidità [8]. Questi meccanismi rispondono ai prodotti dei microbi e alle cellule danneggiate reagendo sostanzialmente in modo identico alle ripetute esposizioni. Le principali caratteristiche dell'immunità innata sono:

- si attiva prima;
- dura da minuti a giorni;
- ha una modalità di riconoscimento molto specifica, che però si limita ad un numero di antigeni relativamente basso (circa mille);
- comprende la fase di infiammazione, l'attivazione delle proteine del complemento, la fagocitosi e la distruzione del patogeno.

Le principali componenti dell'immunità innata sono le barriere fisiche e chimiche, le cellule fagocitarie (neutrofili e macrofagi), le cellule dendritiche e le cellule *natural killer* (NK), le proteine del sangue, tra cui i fattori del sistema del complemento.

A fianco dell'immunità innata esistono altre risposte immunitarie che aumentano in ampiezza e capacità difensive a ogni successiva esposizione a un particolare agente infettivo. Questa forma di immunità, detta anche immunità specifica o acquisita, viene definita adattativa [9]. Le sue principali caratteristiche sono:

- si instaura quando l'immunità innata non è sufficiente a contenere l'infezione, ed è più tardiva;
- inizia poche ore dopo l'infezione, ma quando viene attivata può durare fino a settimane in quanto presenta diverse modalità di azione del sistema immunitario, potendo agire precocemente, cioè dopo poche ore, oppure dopo giorni, o addirittura dopo settimane;

- riconosce un numero enormemente più elevato di patogeni ed ha una capacità di discriminazione molto più alta;
- è caratterizzata dalla memoria immunologica, cioè dalla facoltà di “ricordare” l’infezione rispondendo con maggiore vigore a esposizioni ripetute a uno stesso microbo. Dipende dalle cellule B della memoria che ci proteggono contro la re-infezione, cioè contro l’infezione dallo stesso patogeno. Ci mette qualche settimana a instaurarsi ma può durare per tutta la vita.

I principali componenti sono i linfociti B e T e i loro prodotti di secrezione, come gli anticorpi.

Esistono due tipi di risposte adattative, chiamati immunità umorale e immunità cellulare [4].

L’immunità umorale è mediata da molecole presenti nel sangue e nelle secrezioni mucosali, chiamati anticorpi che sono prodotti da cellule denominate linfociti B. I linfociti B presentano nella loro superficie proteine particolari dette recettori di superficie che interagiscono con gli antigeni. In seguito a questa interazione, inizia la trasformazione dei linfociti B in plasmacellule, produttrici dei 5 principali tipi di anticorpi e in cellule B di memoria. Gli anticorpi riconoscono gli antigeni neutralizzandone l’infettività e identificandoli per la successiva eliminazione da parte di altre cellule. Gli anticorpi possiedono diversi gradi di specializzazione e, a seconda del tipo, possono attivare differenti meccanismi per combattere i microbi (meccanismi effettori).

I linfociti T, dette anche cellule T, sono responsabili dell’immunità cellulare. I microbi intracellulari sopravvivono e proliferano all’interno dei fagociti e di altri tipi cellulari e diventano inaccessibili agli anticorpi circolanti, per questo motivo servono i linfociti T che eliminano i microbi residenti nei fagociti reclutando altri tipi di leucociti in grado di eliminare questi patogeni e di aiutare i linfociti B a produrre anticorpi efficaci.

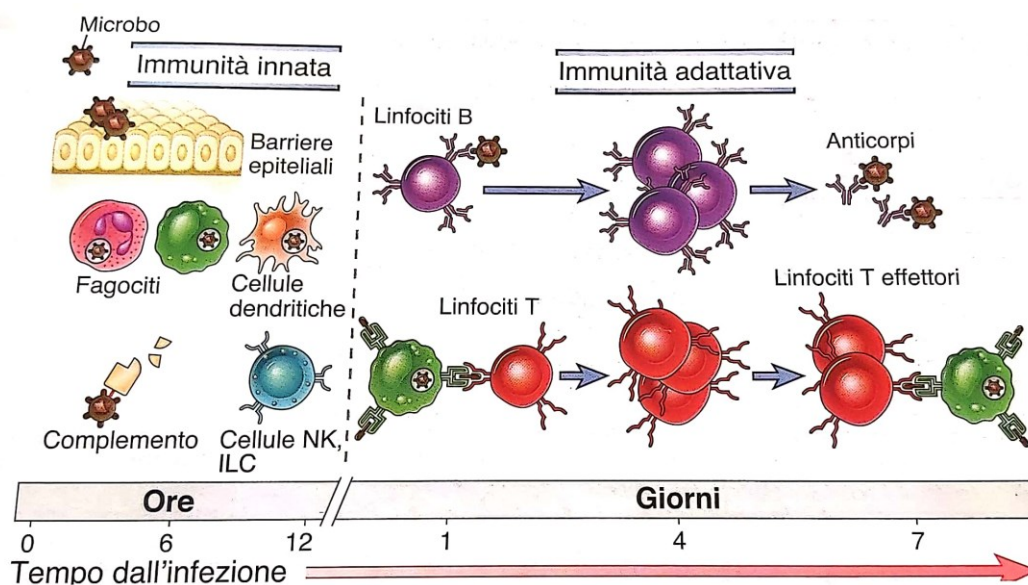


Figura 1. L’immagine mostra le cellule che intervengono nell’immunità innata e adattativa e il tempo dall’infezione. Fonte: [4].

Di seguito si ha un approfondimento delle cellule T e delle cellule dendritiche poiché sono le cellule a cui fa riferimento il progetto di questa tesi.

1.2 LA CELLULA DENDRITICA

La cellula dendritica appartiene all'immunità innata e ha la funzione di presentare l'antigene ai linfociti B e T. Le cellule dendritiche nascono dalle cellule staminali ematopoietiche nel midollo osseo e sono le più importanti dell'insieme delle APC (Antigen presenting cell), cellule che catturano i microbi e altri antigeni e li presentano ai linfociti fornendo segnali in grado di stimolare la proliferazione e la differenziazione dei linfociti stessi. Sono caratterizzate da lunghe protrusioni della membrana e da una spiccata capacità fagocitaria. La maggior parte delle cellule dendritiche appartiene alla linea differenziativa mieloide e deriva da precursori midollari che possono differenziarsi in monociti e granulociti [11].

Esistono diversi tipi di cellule dendritiche, confinate in zone diverse del nostro organismo. Nella pelle troviamo le cellule di Langerhans, cellule dendritiche immature con alta capacità di fagocitosi, originate dalle cellule staminali ematopoietiche presenti nel sacco vitellino o nel fegato fetale [10]. Le cellule dendritiche fungono da sentinella e permettono di trasportare gli antigeni ai linfonodi [4]. I linfonodi, invece, contengono nelle zone T (zone con molti linfociti T) delle cellule dendritiche chiamate interdiganti, mentre nelle zone B (con linfociti B) sono presenti delle cellule dendritiche follicolari. Questi ultimi due tipi cellulari hanno una morfologia molto diversa, il che suggerisce una diversa attività funzionale. Nei tessuti troviamo anche cellule dendritiche classiche, cellule dendritiche plasmacitoidi, cellule dendritiche infiammatorie derivanti tutte dalla cellula staminale emopoietica del midollo osseo [4], [12].

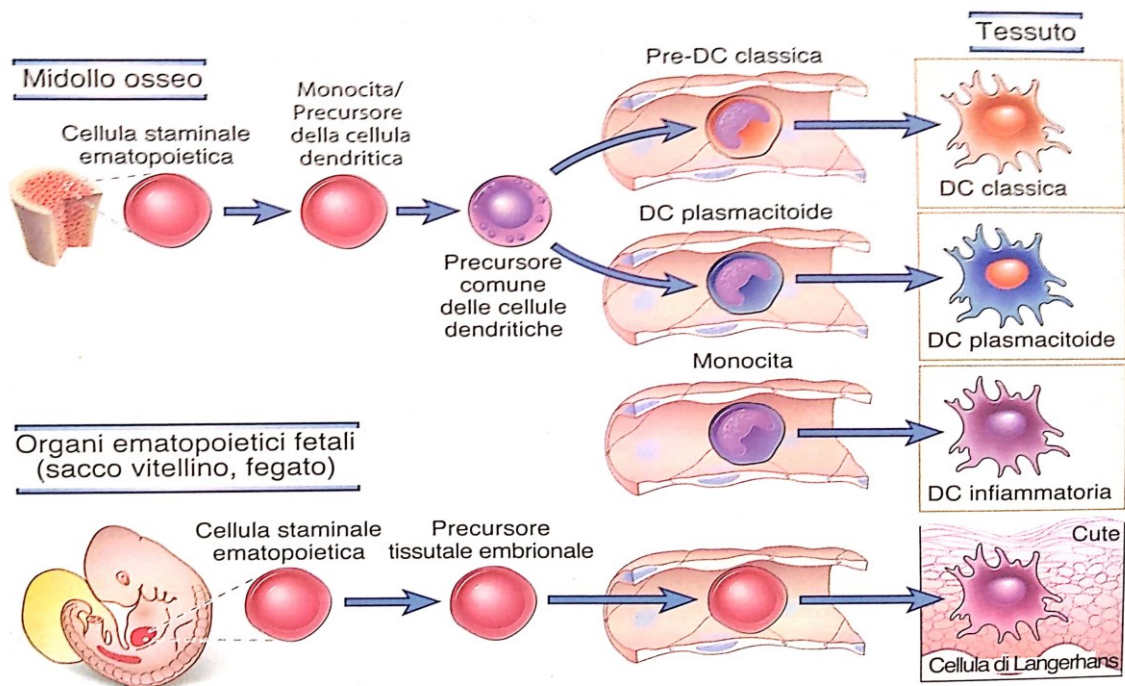


Figura 2. Maturazione delle cellule dendritiche. Fonte:[4].

Prima della maturazione sono presenti nei tessuti non linfoidi, e in particolare dove è più comune incontrare antigeni (per es., la cute e le mucose): in questa fase di immaturità, le cellule dendritiche hanno basso potere di stimolazione dei linfociti T, ma hanno un'alta capacità di riconoscere e processare l'antigene. Il riconoscimento dell'antigene avviene per mezzo di particolari recettori specifici (Pattern recognition receptor, PRR; Toll-like receptor, TLR) con una lunga storia evolutiva, capaci di distinguere diverse strutture molecolari di patogeni [10].

L'incontro con l'antigene dà inizio alla maturazione delle cellule dendritiche, legata anche a segnali chimici molto precisi, e in particolare alla presenza delle molecole di mediatori dell'infiammazione. Dopo l'incontro, l'antigene viene internalizzato per fagocitosi e processato da parte della cellula; quest'ultima migra dal tessuto non linfoide verso gli organi linfatici secondari (linfonodi, milza). Le proteine antigeniche ottenute con la degradazione del patogeno vengono quindi esposte al fine di presentare l'antigene ai linfociti T.

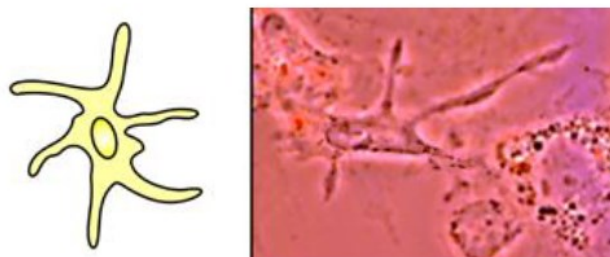


Figura 3. Rappresentazione di una cellula dendritica. Fonte: [4].

1.3 I LINFOCITI T

I linfociti T costituiscono le cellule dell'immunità cellula-mediata, poiché entrano in azione quando l'infezione bypassa le barriere cellulari e diventa intracellulare. I linfociti sono costituiti da sottopopolazioni che si differenziano tra di loro in base al tipo di proteine prodotte e alla funzionalità. Infatti, si distinguono:

- i linfociti T helper: esprimono sulla superficie una proteina chiamata CD4, secernono citochine che sono responsabili per molte delle risposte dell'immunità innata e adattativa, funzionando come "messaggeri" del sistema immunitario. Le citochine prodotte dai linfociti T stimolano la proliferazione e la differenziazione degli stessi linfociti T e di linfociti B. Inoltre, hanno la funzione di attivare i macrofagi [13];
- i linfociti T citotossici: esprimono sulla superficie una proteina chiamata CD8, uccidono le cellule che presentano antigeni estranei *non self*, come cellule infettate da virus o da altri microbi intracellulari;
- i linfociti T regolatori: svolgono il ruolo di inibire le risposte immunitarie [4].

I linfociti T derivano da una cellula staminale midollare, maturano nell'organo linfoide primario, il timo, e successivamente raggiungono attraverso il sangue gli organi linfoidi secondari (linfonodi, milza) [14].

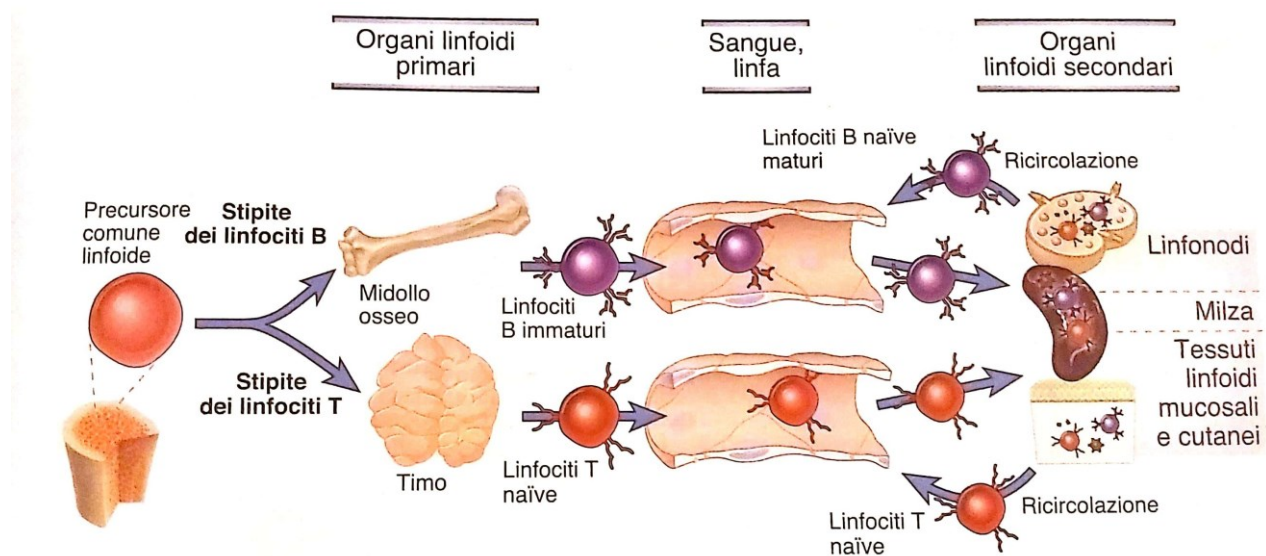


Figura 4. Maturazione dei linfociti T. Fonte:[4].

I linfociti T che escono dal timo e arrivano negli organi secondari sono linfociti maturi, chiamati naïve. Questi sono quiescenti da un punto di vista funzionale e, in seguito all'attivazione da parte di un antigene, proliferano provocando l'espansione dei cloni di linfociti specifici per l'antigene:

questo processo è chiamato espansione clonale. La rapida espansione clonale dei linfociti è necessaria per fronteggiare la capacità dei microbi di proliferare velocemente.

Contemporaneamente all'espansione clonale, i linfociti T stimolati si differenziano in cellule effettrici capaci di eliminare l'antigene. Altri linfociti T attivati si differenziano in cellule della memoria, dotate di lunga sopravvivenza e in grado di garantire risposte rapide ed efficaci in caso di un successivo incontro con lo stesso antigene. Infine, i linfociti T entrano nel sangue e migrano nei siti di infezione [15].

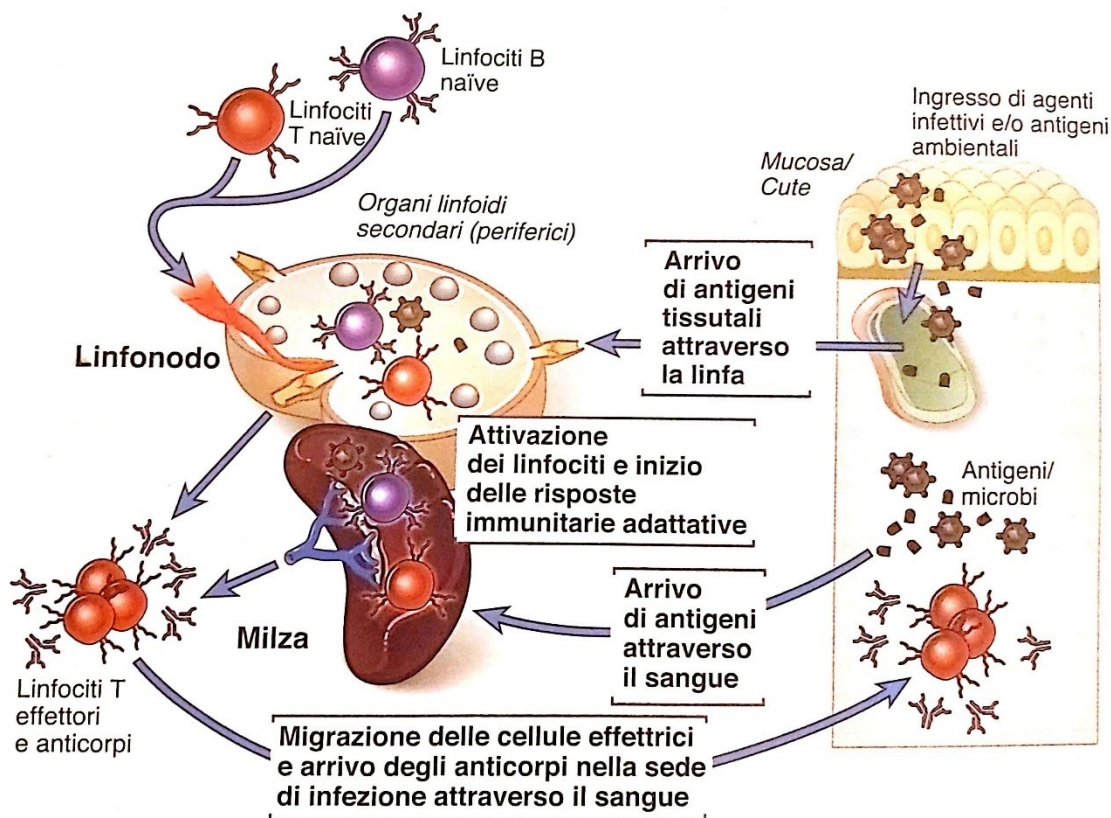


Figura 5. L'immagine mostra le fasi del processo di attivazione dei linfociti. Fonte: [4].

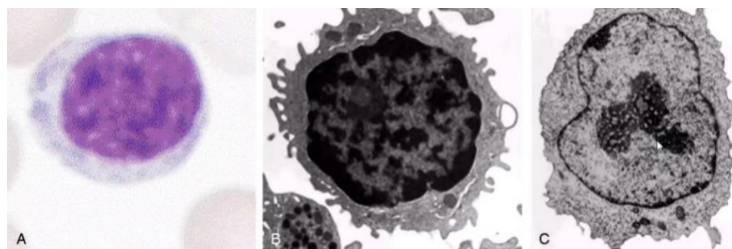


Figura 6. Morfologia dei linfociti T. Fonte: [4].

1.4 LIPSTIC: MONITORAGGIO DELLE INTERAZIONI TRA CELLULE T E CELLULE DENDRITICHE MEDIANTE ETICHETTATURA ENZIMATICA INTERCELLULARE

Gli autori Giulia Pasquali, Aleksey Chudnovskiy et al. [1] hanno ingegnerizzato un sistema per il monitoraggio delle interazioni tra cellule T e cellule dendritiche tramite un approccio che utilizza l'etichettatura cellulare mediata dall'enzima sortasi batterica A. Questo enzima di membrana è stato fuso alle componenti delle sinapsi delle cellule immunitarie, per identificare le interazioni recettore-ligando tra le cellule nei topi vivi, generando un segnale che può essere successivamente rilevato ex vivo mediante citometria a flusso. Questo approccio viene chiamato "Labelling Immune Partnerships by SorTagging Intercellular Contacts (LIPSTIC)". Utilizzando LIPSTIC hanno dimostrato che le interazioni tra le cellule dendritiche e le cellule T durante il priming delle cellule T hanno una fase precoce cognitiva, durante la quale le interazioni tra il recettore CD40 presente nelle cellule dendritiche e la proteina chiamata CD40L o CD40 ligando, espressa soprattutto nelle cellule T attivate, avvengono in modo specifico e necessario per la realizzazione di tale tecnica.

Infatti, i ricercatori hanno sviluppato questa tecnica perché il passaggio del fluoroforo tra sortasi caricata e oligo-guanina è inefficiente mentre il legame tra recettore e ligando che avviene in LIPSTIC è più duraturo ed efficiente. Quindi, poiché la probabilità del passaggio di fluoroforo è tanto maggiore quanto maggiore è il tempo di esposizione o vicinanza tra le cellule, con LIPSTIC si ha più possibilità di passaggio del fluoroforo e di rilevare l'interazione tra le due cellule immunitarie.

In particolare, LIPSTIC si basa sull'etichettatura utilizzando la transpeptidasi sortasi A (SrtA) di *Staphylococcus aureus*. La SrtA trasferisce covalentemente un substrato a cui è agganciato un peptide contenente il motivo di smistamento "LPXTG" a una vicina oligoglicina. Sfruttando questo meccanismo, un ligando e un recettore di interesse sono fusi geneticamente rispettivamente a SrtA o a un tag che consiste in cinque resine gliciniche N-terminali (G5). Viene aggiunto alla SrtA un peptide "LPETG" legato all'estremità N a un'etichetta rilevabile come la biotina o a un fluoroforo. Quando un ligando e un recettore interagiscono, la SrtA catalizza il trasferimento del substrato sul recettore marcato G5. Dopo la separazione delle cellule, la storia dell'interazione è rilevata dalla presenza del fluoroforo sulla superficie della cellula che esprime G5 (si veda figura 7).

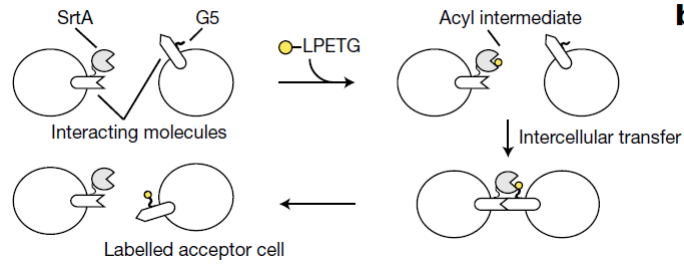


Figura 7. Rappresentazione schematica dell'approccio LIPSTIC. Fonte:[1].

Per testare questo sistema gli autori hanno introdotto in due popolazioni di cellule HEK293T (una specifica linea cellulare immortalizzata spesso usata come modello in biologia) materiale genetico separatamente come G5-CD40 e CD40L-SrtA, hanno mescolato le due popolazioni in presenza del substrato biotinilato di SrtA (biotina-LPETG) per 30 minuti, e poi hanno analizzato le cellule mediante citometria a flusso.

Per dimostrare la specificità, le cellule G5-CD40 sono state incubate con cellule HEK293T a cui è stata aggiunta la SrtA fusa con una variante di CD40L che porta due mutazioni puntiformi che compromettono fortemente il legame con CD40 o con SrtA ancorato alla superficie cellulare dal dominio transmembrana di PDGFR.

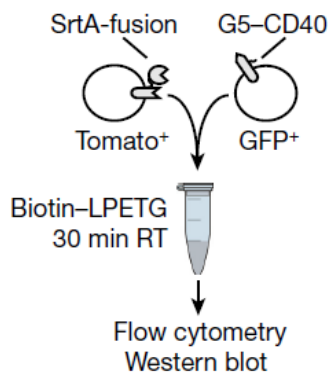


Figura 8. Configurazione sperimentale per analizzare la marcatura intercellulare in cellule HEK293T trasfettate. Fonte:[1].

L'analisi a citometria a flusso ha dimostrato che le cellule G5-CD40 hanno ricevuto la biotina in modo efficiente. Quindi LIPSTIC è un metodo efficace e specifico in grado di marcare le interazioni recettore ligando tra le cellule *in vitro*.

Per determinare se LIPSTIC può funzionare *in vivo* e a livelli endogeni di espressione del recettore e del ligando, gli autori hanno generato topi portatori di alleli CD40G5 e CD40LGSrtA mirati ai loro loci endogeni. Per misurare LIPSTIC durante le interazioni antigene-specifiche tra cellule T e

cellule presentanti l'antigene, gli autori hanno incrociato Cd40lgSrtA con CD4-Cre e OT-II Topi TCR, che esprimono un recettore delle cellule T specifico per il peptide OVA (ovalbumina) di pollo. Hanno coltivato le cellule T CD4 OT-II-SrtA per 6 ore con cellule dendritiche Cd40G5/G5 trattate con OVA o con un peptide LCMV-GP di controllo e hanno aggiunto il substrato con la biotina durante gli ultimi 20 minuti di cultura (si veda figura 9).

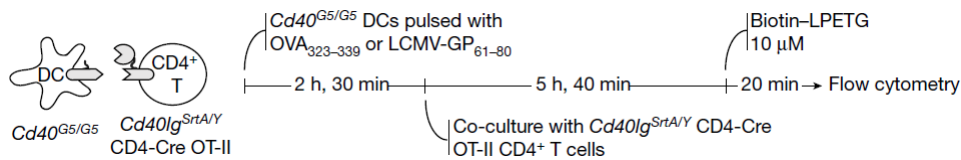


Figura 9. Impostazione sperimentale spiegata sopra. Fonte: [1].

L'etichettatura intercellulare efficiente si è verificata solo quando le cellule dendritiche sono state trattate con il peptide cognato, che è correlato all'induzione dell'espressione di CD40L-SrtA sulle cellule T. La marcatura delle cellule dendritiche è stata fortemente inibita dall'aggiunta di un anticorpo bloccante CD40L, confermando che l'interazione CD40-CD40L è necessaria.

L'etichettatura LIPSTIC è stata dose-responsiva su un intervallo a sei log di concentrazioni di peptidi OVA (si veda figura 10).

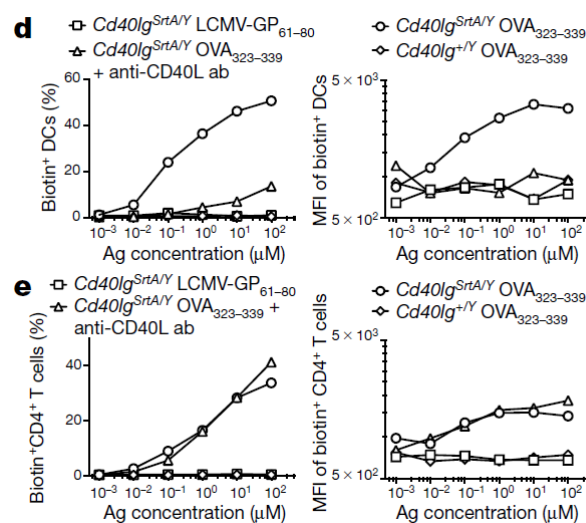


Figura 10. I grafici mostrano la percentuale (a sinistra) e l'intensità media di fluorescenza (MFI, a destra) delle cellule dendritiche (d) e delle cellule T (e). Fonte:[1].

Pertanto, l'etichettatura LIPSTIC negli esperimenti di priming *ex vivo* a breve termine dipende dalle interazioni tra recettore e ligando ed è dose-reattiva in un'ampia gamma di concentrazioni di antigene ed è specifica per le cellule bersaglio che mostrano gli antigeni cognati.

Dopo di che, sempre per determinare se la LIPSTIC può essere utilizzata *in vivo*, gli autori hanno utilizzato un modello di priming delle cellule T ben consolidato in cui le cellule dendritiche Cd40G5/G5 trattate con OVA sono state iniettate per via sottocutanea nel cuscinetto plantare di topi riceventi, seguite 18 ore dopo da un trasferimento endovenoso di cellule T CD4 OT-II-SrtA. Hanno distribuito il substrato LIPSTIC al linfonodo popliteo (PLN) mediante iniezione di un totale

di 300 nmol di biotina-LPETG in sei iniezioni tra 10 e 12 ore dopo il trasferimento delle cellule T (si veda figura 11).

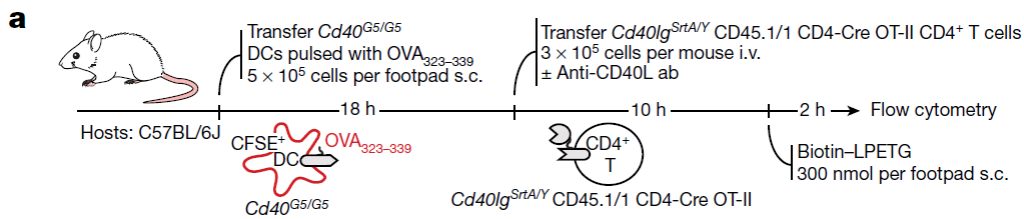


Figura 11. Setup sperimentale. Fonte:[1].

La citometria a flusso delle cellule PLN ha mostrato un'efficiente etichettatura LIPSTIC delle cellule dendritiche trasferite, dipendente dall'espressione di CD40L-SrtA da parte delle cellule T e sensibile al trattamento con un anticorpo bloccante.

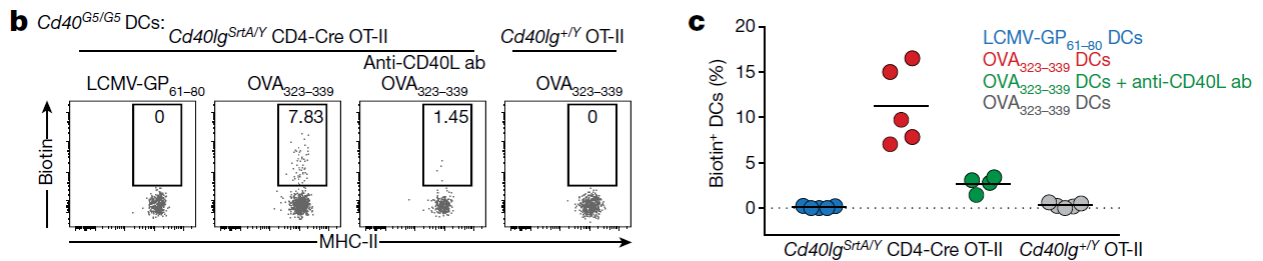


Figura 12. I grafici mostrano a sinistra l'analisi citometrica a flusso delle cellule PLN che mostra la marcatura con biotina del trasferimento delle cellule dendritiche CD40LG5/G5, a destra la percentuale di cellule dendritiche tra quelle trasferite. Fonte: [1].

Per confermare ulteriormente la dipendenza dell'etichettatura LIPSTIC dall'interazione CD40-CD40L hanno sfruttato l'osservazione che, in assenza di un allele CD40G5, le glicine N-terminali endogene sulla superficie cellulare possono funzionare come accettori a bassa efficienza per il substrato. Tale etichettatura era completamente assente quando le cellule dendritiche caricate con Ag erano carenti di CD40, dimostrando ancora una volta che il legame CD40-CD40L è essenziale per l'etichettatura. L'analisi della cinetica di eliminazione dell'etichetta era ancora rilevabile a 4 e 8 ore dopo l'iniezione del substrato.

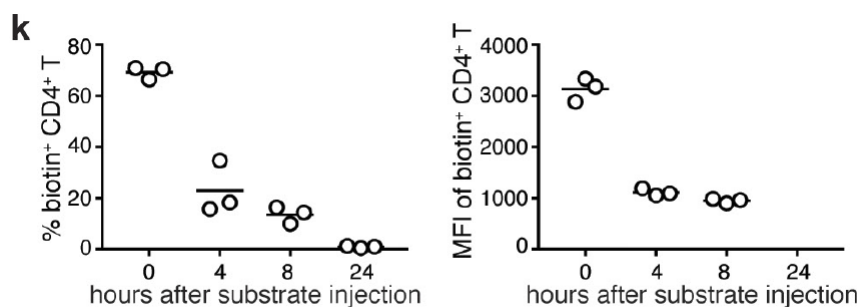


Figura 13. I grafici mostrano a sinistra la percentuale, a destra l'MFI di biotina tra le cellule T. Fonte: [1].

CAPITOLO 2: SIMULATORE SEMPLIFICATO

IGNORANDO IL CARICO DEL FLUOROFORO

In questo capitolo viene spiegato come ha inizio l'implementazione dei vari simulatori.

In particolare, vengono creati quattro simulatori per mostrare, attraverso un video, l'approccio LIPSTIC:

- uno in cui le cellule immunitarie sono distribuite e si muovono in modo causale e le cellule T sono precaricate di fluoroforo in modo infinito (si veda figura 14);
- uno in cui le cellule immunitarie sono distribuite e si muovono in modo casuale e il fluoroforo è distribuito casualmente nel reticolo (si veda figura 14);
- uno in cui le cellule immunitarie sono distribuite e si muovono in modo ordinato e le cellule T sono precaricate di fluoroforo in modo infinito (si veda figura 15);
- uno in cui le cellule immunitarie sono distribuite e si muovono in modo ordinato e il fluoroforo è distribuito casualmente nel reticolo (si veda figura 15);

Il simulatore con le cellule ordinate serve per aiutare il programmatore nel debug.

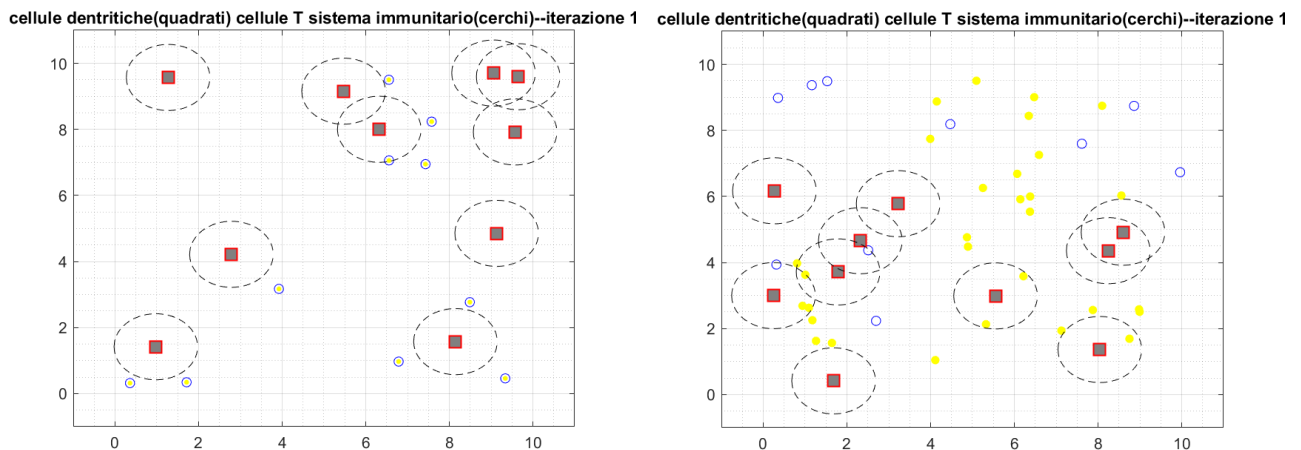


Figura 14. Le immagini mostrano i due simulatori con le cellule immunitarie distribuite in modo casuale: a sinistra le cellule T sono precaricate di fluoroforo mentre a sinistra il fluoroforo è sparso casualmente nel reticolo.

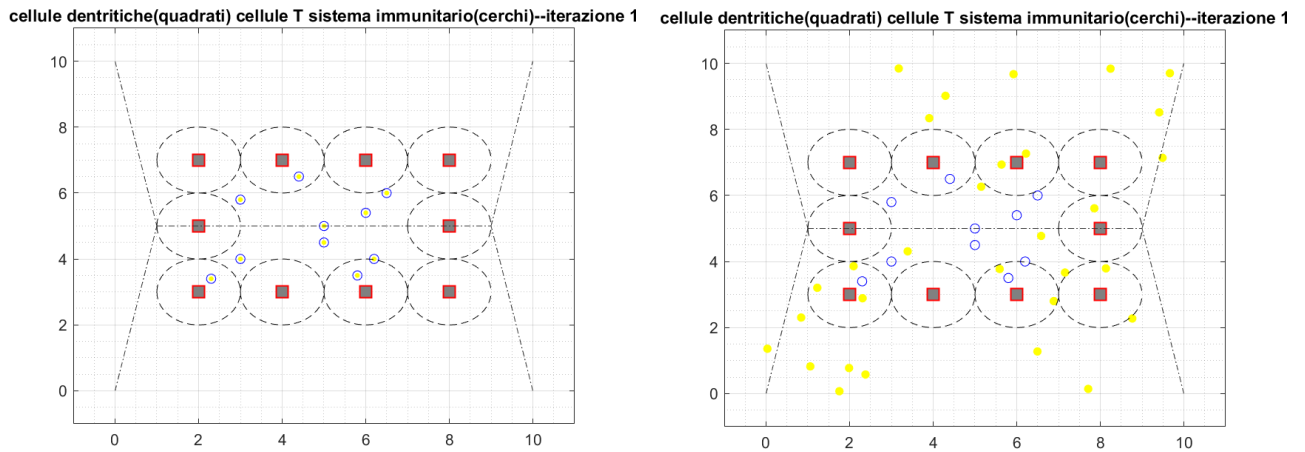


Figura 15. L'immagine mostra i due simulatori con le cellule distribuite in modo ordinato: a sinistra le cellule T sono precaricate di fluoroforo, a destra il fluoroforo è sparso casualmente nel reticolo.

Innanzitutto, vengono stabiliti dei parametri per tutti e quattro i simulatori come il numero di cellule immunitarie (10) e di cellule dendritiche (10), le dimensioni del reticolo in cui le cellule si muovono (10 per l'asse x e 10 per l'asse y, entrambe quantità per il momento adimensionali), il numero di iterazioni del simulatore (all'inizio viene settato a 20 poi verrà aumentato).

Vengono poi fissate la lunghezza dello spostamento delle cellule immunitarie e diverse probabilità che verranno utilizzate nel codice: la probabilità che una cellula immunitaria scambi il fluoroforo con una cellula dendritica, la probabilità che una cellula immunitaria e una cellula dendritica interagiscono, la probabilità che una cellula immunitaria si muova e infine la probabilità che una cellula immunitaria si stacchi da una cellula dendritica.

Vengono inizializzati una "matrice_interazione" e un "vettore_interazione" che serviranno per tenere traccia dell'interazione tra le cellule T e le cellule dendritiche. Il "vettore_interazione" ha tante colonne quanto è il numero di cellule immunitarie, i cui valori saranno uguali a 1 se la cellula T sta interagendo, 0 altrimenti. Nella "matrice_interazione", le righe indicano le iterazioni del simulatore mentre le colonne le cellule T che stanno interagendo. All'iterazione i -esima, se la cellula T j -esima sta interagendo con la cellula dendritica, il valore della matrice nella posizione (i,j) sarà uguale al valore della cellula dendritica altrimenti il valore sarà uguale a 0. Si crea anche un "vettore_scambio_fluoro" con un numero di colonne uguale al numero di cellule T e i cui valori saranno 1 se avverrà lo scambio di fluoroforo tra una cellula T e una dendritica, altrimenti verrà assegnato il valore 0.

2.1 SIMULATORE CON LE CELLULE IMMUNITARIE DISTRIBUITE CASUALMENTE E RIEMPITE DI FLUOROFORO

Per ottenere le coordinate delle cellule immunitarie e dendritiche si fa riferimento a diverse funzioni.

Le cellule dendritiche vengono distribuite in modo casuale nel reticolo attraverso l'uso della funzione "inizializza_cell_dendr_random": si creano due vettori "pos_x_bers" e "pos_y_bers", che contengono le coordinate delle cellule, con un numero di colonne uguale al numero delle cellule dendritiche e i cui valori si ottengono dal comando "rand" di Matlab che restituisce un numero casuale tra zero e uno. Ogni valore ottenuto viene moltiplicato per le dimensioni x e y del reticolo. Si controlla, inoltre, che non ci siano cellule dendritiche sovrapposte andando a vedere se la coordinata x della cellula dendritica i-esima è uguale alla coordinata x della cellula dendritica i-esima +1 e questo lo si fa anche per le coordinate y. Se questo avviene si assegna alla coordinata x della cellula dendritica i-esima lo stesso valore aggiungendo 1 (si veda figura 16).

```
function [pos_x_bers pos_y_bers]=inizializza_cell_dendr_random(x_reticolo,y_reticolo,NumCellDendr)

pos_x_bers=x_reticolo*rand(1,NumCellDendr);
pos_y_bers=y_reticolo*rand(1,NumCellDendr);

%controllo che non ci siano cellule dendritiche sovrapposte

for i=1:NumCellDendr-1
    if pos_x_bers(i)==pos_x_bers(i+1) && pos_y_bers(i)==pos_y_bers(i+1)
        pos_x_bers(i)=pos_x_bers(i)+1;
    end
end
```

Figura 16. Vengono mostrate le righe di codice Matlab usate per creare la funzione "inizializza_cell_dendr_random".

Allo stesso modo si distribuiscono in modo casuale le cellule immunitarie attraverso l'uso della funzione "inizializza_cel_immun": vengono creati due vettori con un numero di colonne uguale al numero di cellule immunitarie con la stessa modalità scritta per le cellule dendritiche (si veda figura 17).

```
function [s, t]=inizializza_cel_immun(x_retic,y_retic,n_cell_imm)

s=x_retic*rand(1,n_cell_imm);
t=y_retic*rand(1,n_cell_imm);

end
```

Figura 17. Vengono mostrate le righe di codice Matlab usate per creare la funzione "inizializza_cel_immun".

Per le coordinate delle cellule immunitarie si utilizza la funzione “step_cell_immun” che contiene le seguenti istruzioni scritte in pseudocodice:

```
for i=1:length(old_x_position)(con questo ciclo for vado a considerare tutte le cellule immunitarie
una alla volta)
    if la cellula immunitaria sta interagendo con una dendritica
        -rimane ferma: coordinata x è uguale al valore del primo vettore dato in input e
        coordinata y è uguale al valore del secondo vettore dato in input
    else
        -calcolo due quantità, “p” e “t”, che indicano lo spostamento in x e in y della cellula
        immunitaria usando il comando “randn” che restituisce un valore casuale ottenuto
        da una distribuzione normale standard.

        -con probabilità “p_movimento”,
        - si assegna alla coordinata x della cellula immunitaria il valore del primo vettore
        dato in input + p
        - si assegna alla coordinata y della cellula immunitaria il valore del secondo
        vettore dato in input + t
    end
end
end
```

Poi si controlla che le cellule immunitarie non escano dal reticolo: si assegna alla coordinata x e alla coordinata y della cellula immunitaria i-esima il valore massimo tra la coordinata stessa e lo zero (così la cellula immunitaria non uscirà dal bordo di sinistra), mentre per verificare che la cellula non esca dal bordo di destra del reticolo si assegna alla coordinata x e y della cellula i-esima il minimo tra la coordinata stessa e il valore massimo del bordo (pari a 10).

Nella seguente figura viene mostrato il codice equivalente allo pseudocodice spiegato precedentemente.

```
function [pos_x_imm,pos_y_imm]=step_cell_immun(old_x_position,old_y_position,prob,cell_interaction,step_length,j)
for i=1:length(old_x_position)
    % se le cellule interagiscono rimangono fermo
    if cell_interaction(j,i)~=0
        pos_x_imm(1,i)=old_x_position(1,i);
        pos_y_imm(1,i)=old_y_position(1,i);
    else
        p=step_length*randn();
        t=step_length*randn();
        probabilita=rand();
        if probabilita<prob
            pos_x_imm(1,i)=old_x_position(1,i)+p;
            pos_y_imm(1,i)=old_y_position(1,i)+t;
        else
            %cellula non si sposta
            pos_x_imm(1,i)=old_x_position(1,i);
            pos_y_imm(1,i)=old_y_position(1,i);
        end
    end
    %controllo che le cellule immunitarie non escano dai bordi del
    %reticolo
    pos_x_imm(1,i)=max(pos_x_imm(1,i),0);
    pos_y_imm(1,i)=max(pos_y_imm(1,i),0);
    pos_x_imm(1,i)=min(pos_x_imm(1,i),10);
    pos_y_imm(1,i)=min(pos_y_imm(1,i),10);
end
end
```

Figura 18. Questa figura mostra le righe di codice Matlab equivalente allo pseudocodice spiegato sopra.

Di seguito si ha un ciclo esterno che itera su un certo numero di iterazioni che descrivono l'andamento temporale del simulatore e avendo quantizzato il tempo in tanti istanti Δt , si ripete l'esperimento su un numero n di istanti Δt .

```
for j=1:T(iterazioni del simulatore)
```

```
    if j==1
```

```
        -mostro in una figura come sono disposte inizialmente le cellule immunitarie e le cellule dendritiche. Vengono visualizzate rispettivamente con dei cerchi blu riempiti di fluoroforo giallo mentre le altre con dei quadrati rossi. Si disegna poi la regione di interazione tra le cellule che equivale a dei cerchi il cui centro coincide con la cellula dendritica. Poi la figura viene salvata nel video.
```

```
    end
```

```
    for indCellimm=1:NumCellimm
```

```
        for indCelldendr=1:NumCellDendr
```

```
            - calcolo la distanza euclidea tra le due cellule utilizzando la funzione "norma" che prende le coordinate x e y della cellula immunitaria e dendritica all'iterazione j-esima e ne esegue la radice quadrata (l'immagine 19 che mostra il codice corrispondente è inserita dopo lo pseudocodice)
```

```
            if distanza_euclidea < soglia fissata
```

```
                -con probabilita "p_interact" (probabilità che una cellula immunitaria interagisca con una cellula dendritica) e contemporaneamente la cellula immunitaria non è legata ad una dendritica
```

```
                    - matrice_interazione(j,indCellimm)=indCelldendr
```

```
                    - vettore_interazione(1,indCellimm)=1 (vuol dire che la cellula immunitaria sta interagendo)
```

```
                -con probabilita "p_distacco"(probabilità che una cellula immunitaria si stacchi da una dendritica) e contemporaneamente la cellula immunitaria è legata ad una dendritica
```

```
                    - vettore_interazione(j,indCellimm)=0;
```

```
                    - matrice_interazione(j,indCellimm)=0;
```



```

    -con probabilità "p_scambio_fluoroforo_dendimm" (probabilità che una
      cellula immunitaria scambi il fluoroforo con una cellula dendritica)
      - vettore_scambio_fluoroforo(1,indCelldendr)=1;
    end
  end
end

```

-dopo di che calcolo le nuove coordinate x e y delle cellule immunitarie usando la funzione "step_cell_immun"

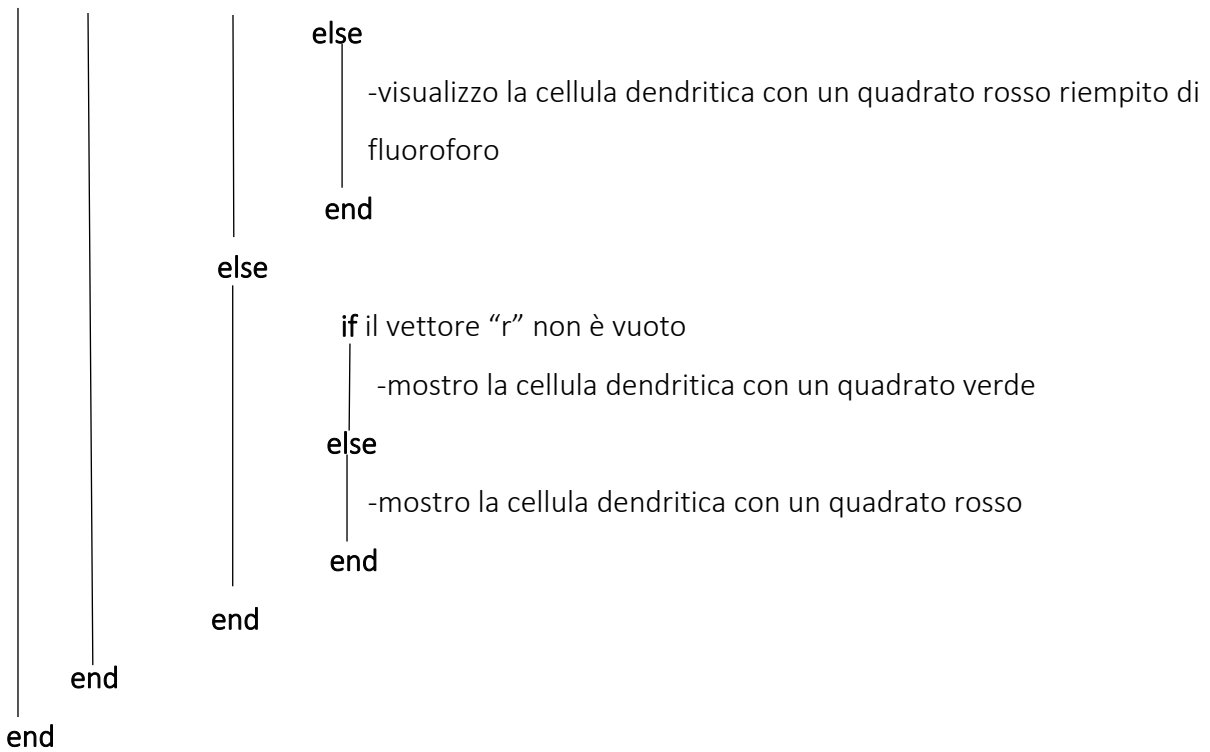
(utilizzo il seguente pseudocodice per creare la figura che mostra le cellule immunitarie riempite di fluoroforo e quelle dendritiche che interagiscono)

```

for indicecellimm=1:NumCellimm
  -r=matrice_interazione(j,indicecellimm)
  if r ≈ 0 (vuol dire che la cellula immunitaria è legata a una dendritica)
    - mostro la cellula immunitaria con un pallino verde riempito di
      fluoroforo giallo. Il legame tra la cellula immunitaria e dendritica viene
      visualizzato con una linea blu che congiunge le due cellule
  else
    - mostro la cellula immunitaria con un pallino blu riempito di fluoroforo
      giallo
  end
end

for indicecellden=1:NumCellDendr
  - con la funzione Matlab "find" viene creato un vettore "r" i cui valori sono diversi da
  zero se la cellula dendritica è legata a una immunitaria
  if la cellula dendritica ha scambiato il fluoroforo con la cellula immunitaria
    if il vettore "r" non è vuoto (vuol dire che la cellula è legata a una cell-
      ula immunitaria)
      - visualizzo la cellula dendritica con un quadrato verde riempito di
      fluoroforo
    end
  end
end

```



Si disegna poi la regione di interazione tra la cellula dendritica e immunitaria che equivale a un cerchio il cui centro coincide con la cellula dendritica. Poi la figura viene salvata nel video (per il codice riferito allo pseudocodice spiegato sopra si rimanda all'appendice).

```

distanza_euclidea=norma(pos_x_imm(j,indCellimm),pos_x_dendr(1,indCelldendr),pos_y_imm(j,indCellimm),pos_y_dendr(1,indCelldendr))

function distanza_euclidea=norma(x1,x2,y1,y2)
    distanza_euclidea= sqrt(((x1-x2)^2)+((y1-y2)^2));
end

```

Figura 19. Si mostrano le righe di codice Matlab spiegate in pseudocodice sopra.

2.2 SIMULATORE CON LE CELLULE IMMUNITARIE DISTRIBUITE IN MODO ORDINATO E RIEMPIE DI FLUOROFORO

Nel simulatore in cui le cellule sono disposte in modo ordinato viene utilizzato lo stesso codice del simulatore con le cellule distribuite in modo casuale ma con delle modifiche a delle funzioni come “inizializza_cell_dendr_random”, “inizializza_cel_immun” e” step_cell_immun”.

Nella funzione “inizializza_cell_dendr_ord” le cellule dendritiche vengono distribuite lungo un rettangolo fissato (si veda figura 20).

```
function [pos_x_dendr ,pos_y_dendr]=inizializza_cell_dendr_ord(numdendr)
pos_x_dendr=zeros(1,numdendr);
pos_y_dendr=zeros(1,numdendr);
pos_x_dendr(1,1)=2;
pos_x_dendr(1,2)=4;
pos_x_dendr(1,3)=6;
pos_x_dendr(1,4)=8;
pos_x_dendr(1,5)=8;
pos_x_dendr(1,6)=8;
pos_x_dendr(1,7)=6;
pos_x_dendr(1,8)=4;
pos_x_dendr(1,9)=2;
pos_x_dendr(1,10)=2;
pos_y_dendr(1,1)=3;
pos_y_dendr(1,2)=3;
pos_y_dendr(1,3)=3;
pos_y_dendr(1,4)=3;
pos_y_dendr(1,5)=5;
pos_y_dendr(1,6)=7;
pos_y_dendr(1,7)=7;
pos_y_dendr(1,8)=7;
pos_y_dendr(1,9)=7;
pos_y_dendr(1,10)=5;
```

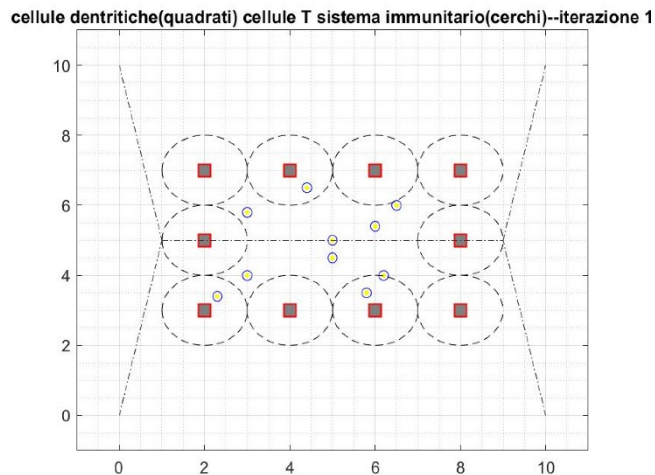


Figura 20. La seguente figura mostra a sinistra il codice Matlab relativo alla funzione “inizializza_cell_dendr_ord”, a destra le cellule dendritiche disposte lungo il rettangolo.

Nella funzione “inizializza_cell_immun_ord” le cellule immunitarie vengono disposte inizialmente dentro al rettangolo costruito dalle cellule dendritiche in posizioni fissate (si veda figura 21).

```
function [s_random ,t_random]=inizializza_cel_immun_ord(NumCellimm)
s_random=zeros(1,NumCellimm);
t_random=zeros(1,NumCellimm);
s_random(1,1)=2.3;
s_random(1,2)=5.8;
s_random(1,3)=6.2;
s_random(1,4)=5;
s_random(1,5)=3;
s_random(1,6)=5;
s_random(1,7)=6;
s_random(1,8)=6.5;
s_random(1,9)=4.4;
s_random(1,10)=3;
t_random(1,1)=3.4;
t_random(1,2)=3.5;
t_random(1,3)=4;
t_random(1,4)=4.5;
t_random(1,5)=4;
t_random(1,6)=5;
t_random(1,7)=5.4;
t_random(1,8)=6;
t_random(1,9)=6.5;
t_random(1,10)=5.8;
```

Figura 21. Viene mostrato il codice Matlab della funzione “inizializza_cell_immun_ord”.

La funzione “step_cell_immun_ord” è utilizzata per calcolare le nuove posizioni delle cellule immunitarie. In particolare, il reticolo del simulatore viene suddiviso in quattro regioni (si veda figura 22) e in base a dove la cellula è disposta, questa si muoverà a destra, sinistra, in alto o in basso.

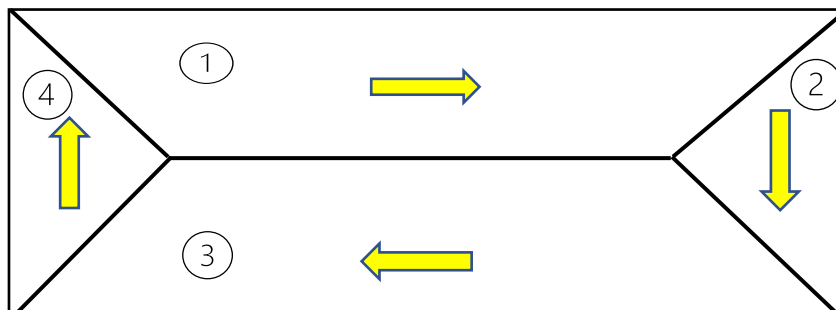


Figura 22. Viene mostrato il reticolo del simulatore diviso in quattro regioni e come le cellule immunitarie si muovono in ogni regione: a destra, in basso, a sinistra e in alto.

Di seguito si ha lo pseudocodice relativo alla funzione “step_cell_immun_ord”:

- si inizializzano due vettori in cui verranno salvate le nuove posizioni x e y delle cellule immunitarie.
- si inizializzano otto valori che corrispondono ai coefficienti angolari e alle intercette delle rette che dividono il reticolo.

```

for i=1:length(old_x_position)
    if la cellula immunitaria sta interagendo con una dendritica
        - rimane ferma: coordinata x è uguale al valore del primo vettore dato in input e
        coordinata y è uguale al valore del secondo vettore dato in input
    else
        -con probabilità “p_step” (probabilità che una cellula immunitaria si muova) e contem-
        poraneamente la cellula immunitaria sta nella regione numero uno del reticolo
            - la cellula si sposta a destra di un valore prefissato
            “step_length_cell_immun”
    
```

```

end
    -con probabilità "p_step" e contemporaneamente la cellula si trova nella regione
    numero due del reticolo
        - la cellula si sposta in basso del valore "step_length_cell_immun"
    -con probabilità "p_step" e contemporaneamente la cellula si trova nella regione
    numero tre del reticolo
        - la cellula si sposta a sinistra del valore "step_length_cell_immun"

    -con probabilità "p_step" e contemporaneamente la cellula si trova nella regione
    numero quattro del reticolo
        - la cellula si sposta in alto del valore "step_length_cell_immun"
    end
end

```

Poi si controlla che le cellule immunitarie non escano dal reticolo. Nella seguente figura viene mostrato il codice equivalente allo pseudocodice spiegato precedentemente.

```

function [pos_x_imm_new, pos_y_imm_new]=step_cell_immun_ord(old_x_position,old_y_position,p_step,cell_interaction_imm,step_length_cellimm,j)
pos_x_imm_new=zeros(1,length(old_x_position));
pos_y_imm_new=zeros(1,length(old_x_position));
m1=-5;
q1=10;
q2=5;
m3=5;
q3=-40;
m4=-5;
q4=50;
m5=5;
for i=1:length(old_x_position)
    % se le cellule interagiscono rimangono ferme

    if cell_interaction_imm(j,i)~=0

        pos_x_imm_new(1,i)=old_x_position(1,i);
        pos_y_imm_new(1,i)=old_y_position(1,i);

    else
        p=rand();
        if p<p_step && old_y_position(1,i)>=q2 && old_y_position(1,i)>=m3.*(old_x_position(1,i))+q3 && old_y_position(1,i)>=m1.*old_x_position(1,i)+q1
            pos_x_imm_new(1,i)=old_x_position(1,i)+step_length_cellimm;
            pos_y_imm_new(1,i)=old_y_position(1,i);

        elseif p<p_step && old_y_position(1,i)<m3.*(old_x_position(1,i))+q3 && old_y_position(1,i)>=m4.*old_x_position(1,i)+q4
            pos_x_imm_new(1,i)=old_x_position(1,i);
            pos_y_imm_new(1,i)=old_y_position(1,i)-step_length_cellimm;
        elseif p<p_step && old_y_position(1,i)<q2 && old_y_position(1,i)<=m5.*(old_x_position(1,i)) && old_y_position(1,i)<m4.*old_x_position(1,i)+q4
            pos_x_imm_new(1,i)=old_x_position(1,i)-step_length_cellimm;
            pos_y_imm_new(1,i)=old_y_position(1,i);

        elseif p<p_step && old_y_position(1,i)>m5.*(old_x_position(1,i)) && old_y_position(1,i)<m1.*old_x_position(1,i)+q1
            pos_x_imm_new(1,i)=old_x_position(1,i);
            pos_y_imm_new(1,i)=old_y_position(1,i)+step_length_cellimm;
        end
        pos_x_imm_new(1,i)=max(pos_x_imm_new(1,i),0);
        pos_y_imm_new(1,i)=max(pos_y_imm_new(1,i),0);
        pos_x_imm_new(1,i)=min(pos_x_imm_new(1,i),10);
        pos_y_imm_new(1,i)=min(pos_y_imm_new(1,i),10);

    end
end
end
end

```

Figura 23. Si mostra le righe di codice Matlab equivalente allo pseudocodice spiegato sopra.

CAPITOLO 3: SIMULATORE CHE INCLUDE L'INTERAZIONE CON IL FLUOROFORO

3.1 SIMULATORE CON LE CELLULE IMMUNITARIE E IL FLUOROFORO DISTRIBUITI IN MODO CASUALE

In questo simulatore, oltre all'inizializzazione dei parametri visti nel capitolo 2, ci sono l'inizializzazione di due vettori che contengono le coordinate x e y del fluoroforo, una "matrice_memoria_interazione", un "vettore_fluoro_cellimm" (vettore di lunghezza pari al numero di cellule immunitarie e che contiene nella posizione i-esima il valore 1 se l'i-esima cellula T ha il fluoroforo, altrimenti 0), un vettore "fluo_passato" (vettore di lunghezza pari al numero di cellule immunitarie e che contiene nella posizione i-esima il valore 1 se l'i-esima cellula T ha passato il fluoroforo a una cellula dendritica, altrimenti 0). In questo simulatore sono stati aggiunti sei vettori (count_X) che indicano delle variabili che vanno a conteggiare degli eventi e che saranno utili nell'analisi dei risultati: i veri positivi **count_TP** (viene contato l'evento in cui una cellula immunitaria si lega o è già legata e passa il fluoroforo alla dendritica), i falsi negativi **count_FN** (viene contato l'evento in cui la cellula immunitaria con il fluoroforo si slega dalla dendritica ma non ha passato il fluoroforo), i falsi negativi "no fluo" **count_FN_nofluo** (evento in cui la cellula immunitaria che non ha il fluoroforo si slega dalla dendritica), i falsi positivi **count_FP** (evento in cui una cellula immunitaria con il fluoroforo passa vicino alla dendritica, non si lega ma passa il fluoroforo alla dendritica), i veri negativi **count_TN** (evento in cui una cellula immunitaria con il fluoroforo passa vicino alla dendritica, non si lega e non passa il fluoroforo), i veri negativi "no fluo" **count_TN_nofluo** (evento in cui una cellula immunitaria non carica di fluoroforo passa vicino alla dendritica, non si lega), i veri positivi successivi **count_TP_successivi** (evento in cui una cellula dendritica che ha già il fluoroforo riceve altro fluoroforo da altre cellule immunitarie cariche).

Poi, come visto nel capitolo 2.1, vengono usate le funzioni "inizializza_cell_dendr_random" per calcolare le coordinate delle cellule dendritiche, "inizializza_cell_immun" e "step_cell_imm" per ottenere le coordinate iniziali delle cellule immunitarie.

Si procede, poi, con il seguente pseudocodice:

```
for j=1:T(numero iterazioni simulatore)
```

```
    if j==1
```

```
        - mostro in una figura come sono distribuite le cellule dendritiche e come sono disposte inizialmente le cellule immunitarie. Viene plottato il fluoroforo con dei pallini gialli e anche la regione di interazione.
```

```
    else
```

```
        -matrice_interazione(j,:)=matrice_interazione(j-1,:); (ricopio tutte le colonne della matrice corrispondenti alla riga j-1 nelle colonne corrispondenti alla riga j-esima)
```

```
        -matrice_memoria_interazione(j,:)=matrice_memoria_interazione(j-1,:);
```

```
        (le prossime righe di codice servono per andare a copiare nella posizione (1,j) dei vari vettori, il valore che si ha nell'iterazione (1,j-1) in modo da non perdere il valore del conteggio dei vari indici ottenuto fino all'iterazione j-1)
```

```
        -count_TP(1,j)=count_TP(1,j-1);
```

```
        -count_FP(1,j)=count_FP(1,j-1);
```

```
        -count_TN(1,j)=count_TN(1,j-1);
```

```
        -count_FN(1,j)=count_FN(1,j-1);
```

```
        -count_TN_nofluo(1,j)=count_TN_nofluo(1,j-1);
```

```
        -count_FN_nofluo(1,j)=count_FN_nofluo(1,j-1);
```

```
        -count_TP_successivi(1,j)=count_TP_successivi(1,j-1);
```

```
    end
```

```
    -Inizializzo due vettori vuoti "x_fluoroforo_new" e "y_fluoroforo_new"
```

```
    for indCellimm=1:NumCellimm
```

```
        if la cellula immunitaria è già legata
```

```
            if la cellula immunitaria ha il fluoroforo
```

```
                if riesce a farlo passare
```

```
                    -vettore_fluoro_cell_imm(1,indCellimm)=0;
```

```
                    -r=matrice_interazione(j,indCellimm);
```

```

        -if vettore_scambio_fluoro(1,r)==1(se la cellula dendritica ha già il fluoro-
        |   foro)
        |   -count_TP_successivi(1,j)=count_TP_successivi(1,j)+1;
        |   end
        |   -vettore_scambio_fluoro(1,r)=1;
        |   -fluo_passato(1,indCellimm)=1;
        |   -count_TP(1,j)=count_TP(1,j)+1;
        |   end
    end
    if si slega
        |   if ha passato il fluoroforo
        |   |   -matrice_interazione(j,indCellimm)=0;
        |   |   -fluo_passato(1,indCellimm)=0;
        |   |   else(se non ha passato il fluoroforo)
        |   |   |   if aveva il fluoroforo
        |   |   |   |   -count_FN(1,j)=count_FN(1,j)+1;
        |   |   |   |   -matrice_interazione(j,indCellimm)=0;
        |   |   |   else(se non aveva il fluoroforo)
        |   |   |   |   -count_FN_nofluo(1,j)=count_FN_nofluo(1,j)+1;
        |   |   |   |   -matrice_interazione(j,indCellimm)=0;
        |   |   |   end
        |   |   end
        |   end
        |   else(se non si slega)
        |   |   -non faccio nulla
        |   end
    else(se non è già legata)
        |   for indCelldend=1:NumCellDendr
        |   |   if è vicino a una dendritica

```

```

if si lega
    -matrice_interazione(j,indCellimm)=indCelldend;
    -matrice_memoria_interazione(j,indCelldend)=matrice_memo
        -ria_interazione(j,indCelldend)+1;
end
if ha il fluoroforo
    if passa il fluoroforo a una dendritica
        -vettore_fluoro_cellimm(1,indCellimm)=0;
        -vettore_scambio_fluoro(1,indCelldend)=1;
        if la cellula immunitaria non è legata a una dendritica
            -count_FP(1,j)=count_FP(1,j)+1;
        else
            -fluo_passato(1,indCellimm)=1;
            -count_TP(1,j)=count_TP(1,j)+1;
            if la dendritica ha già il fluoroforo
                count_TP_successivi(1,j)=count_TP_successivi
                    (1,j)+1;
            end
        end
    end
    else(se non passa il fluoroforo)
        if non è legata a una dendritica
            -count_TN(1,j)=count_TN(1,j)+1;
        end
    end
end
else(se non ha il fluoroforo)
    if non è legata a una dendritica
        -count_TN_nofluo(1,j)=count_TN_nofluo(1,j)+1;
    end
end
end
end
end
end
end

```

end

-dopo di che calcolo le nuove coordinate x e y delle cellule immunitarie usando la funzione "step_cell_immun"

```
for k=1:NumCellimm
```

```
    for s=1:length(x_fluoroforo)
```

```
        - calcolo la distanza tra la cellula immunitaria e il fluoroforo attraverso l'uso  
        della funzione "norma"
```

```
        -if "distanza_fluoro_cell" < quantità fissata "threshold_fluoroforo_cellimm" &&
```

```
            con probabilità "p_fluoroforo" (probabilità che una cellula immunitaria senza  
            fluoroforo si carichi di fluoroforo)
```

```
            -vettore_fluoro_cellimm(1,k)=1
```

```
            -x_fluoroforo(1,s)=NaN
```

```
            -y_fluoroforo(1,s)=NaN
```

```
        end
```

```
    end
```

```
end
```

(il seguente pseudocodice serve per creare i vettori delle coordinate x e y del fluoroforo a cui abbiamo tolto quei fluorofori che le cellule immunitarie hanno preso)

```
for f=1:length(x_fluoroforo)
```

```
    if il valore x_fluoroforo(1,f) è diverso da NaN
```

```
        -z = x_fluoroforo(1,f)
```

```
        - x_fluoroforo_new=[x_fluoroforo_new z]
```

```
    end
```

```
    if il valore y_fluoroforo(1,f) è diverso da NaN
```

```
        -p=y_fluoroforo(1,f)
```

```
        -y_fluoroforo_new=[y_fluoroforo_new p]
```

```
    end
```

```
end
```

```
- x_fluoroforo=x_fluoroforo_new
```

```
-y_fluoroforo= y_fluoroforo_new
```

(utilizzo il seguente pseudocodice per creare la figura che mostra il fluoroforo sparso nel reticolo, le cellule immunitarie e quelle dendritiche che interagiscono)

```
for indicecellimm=1:NumCellimm
```

```
    -salvo nella variabile "r" il valore della "matrice_interazione" all'iterazione j-esima
```

```
    if vettore_fluoro_cellimm(1,indicecellimm)==1
```

```
        if  $r \cong 0$  ( vuol dire che la cellula immunitaria è legata a una dendritica)
```

```
            - mostro la cellula immunitaria con un pallino verde riempito di fluoroforo  
            giallo. Il legame tra la cellula immunitaria e dendritica viene  
            visualizzato con una linea blu che congiunge le due cellule
```

```
        else
```

```
            - mostro la cellula immunitaria con un pallino blu riempito di fluoroforo
```

```
        end
```

```
    else
```

```
        if  $r \cong 0$ 
```

```
            -mostro la cellula immunitaria con un pallino verde vuoto. Il legame tra la  
            cellula immunitaria e dendritica viene visualizzato con una linea blu che  
            congiunge le due cellule
```

```
        else
```

```
            -mostro la cellula immunitaria con un pallino blu vuoto
```

```
        end
```

```
    end
```

```
end
```

```
-disegno il fluoroforo sparso nel reticolo
```

```
for indicecellden=1:NumCellDendr
```

```
    - con la funzione Matlab "find" viene creato un vettore "r" i cui valori sono diversi da  
    zero se la cellula dendritica è legata a una immunitaria
```

```
    if la cellula dendritica ha scambiato il fluoroforo con la cellula immunitaria
```

```
        if  $\text{matrice\_memoria\_interazione}(j,\text{indicecellden}) \cong 0$  (vuol dire che la cellula dendri-  
        tica ha già interagito con una cellula T)
```

```
            -visualizzo la cellula dendritica con una stella nera riempita di fluoroforo  
            giallo
```

```

else
    if il vettore "r" non è vuoto (vuol dire che la cellula è legata a una cellula
        immunitaria)
        - visualizzo la cellula dendritica con un quadrato verde riempito di
        fluoroforo
    else
        -visualizzo la cellula dendritica con un quadrato rosso riempito di
        fluoroforo
    end
end
else(la cellula dendritica non ha scambiato il fluoroforo con la cellula immunitaria)
    if matrice_memoria_interazione(j,indicellden)  $\cong$  0
        -visualizzo la cellula dendritica con una stella nera vuota
    else
        if il vettore "r" non è vuoto (vuol dire che la cellula è legata a una cel-
            lula immunitaria)
            -visualizzo la cellula dendritica con un quadrato verde vuoto
        else
            -visualizzo la cellula dendritica con un quadrato rosso vuoto
        end
    end
end
end
end
end
end

```

Si disegna poi la regione di interazione tra la cellula dendritica e immunitaria che equivale a un cerchio il cui centro coincide con la cellula dendritica. Poi la figura viene salvata nel video (per il codice relativo allo pseudocodice spiegato precedentemente si veda l'appendice).

3.2 SIMULATORE CON LE CELLULE IMMUNITARIE DISTRIBUITE IN MODO ORDINATO E IL FLUOROFORO SPARSO NEL RETICOLO

Per questo simulatore viene utilizzato lo stesso codice del simulatore con le cellule distribuite in modo casuale e il fluoroforo sparso nel reticolo con delle modifiche che riguardano l'uso delle funzioni "inicializza_cell_dendr_ord", "inicializza_cell_immun_ord" e "step_cell_immun_ord" spiegate nel simulatore con le cellule distribuite in modo ordinato (per il codice si veda l'appendice).

CAPITOLO 4: ANALISI DEI RISULTATI

L'obiettivo di questo capitolo è quello di commentare e mostrare attraverso delle immagini l'andamento dei risultati (veri positivi, veri negativi, falsi positivi, falsi negativi) con la rispettiva media, varianza e deviazione standard.

Inoltre, verranno mostrati dei parametri usati in statistica come la sensibilità, specificità, precisione, "F1 score".

Innanzitutto, si è partiti utilizzando il codice descritto nel capitolo 3.1, a cui si è tolta la parte che serviva per ottenere le varie figure da inserire nel video. Si è settato il numero di iterazioni del simulatore (1000) e il numero di simulazioni considerate (100).

Poi, oltre alle matrici già create che contengono i valori dei veri positivi totali ("TP"), veri negativi ("TN"), falsi positivi ("FP"), falsi negativi ("FN"), sono state realizzate altre matrici: la matrice veri positivi visibili ("TP_visibili") che conta solo l'evento in cui una cellula immunitaria passa per la prima volta il fluoroforo a una cellula dendritica che non ha il fluoroforo. Questa matrice viene realizzata andando a sottrarre alla matrice dei veri positivi totali, la matrice dei veri positivi successivi (matrice che tiene conto dell'evento in cui una cellula immunitaria passa il fluoroforo alla cellula dendritica che ha già ricevuto il fluoroforo). Poi viene creata la matrice dei falsi negativi ("FN_tot_2") che equivale alla matrice dei falsi negativi a cui abbiamo aggiunto i veri positivi successivi e i falsi negativi "no fluo".

Con le funzioni Matlab "mean", "var", "std" si sono calcolate le equivalenti media, varianza, deviazione standard delle varie matrici e poi vengono costruite le figure che mostrano i risultati. Per ottenere i valori dei parametri statistici sensibilità, specificità, precisione e "F1 score" si è rimandato alle loro definizioni: la sensibilità e la specificità descrivono matematicamente l'accuratezza di un test che segnala la presenza o l'assenza di una condizione. Se gli individui che hanno la condizione sono considerati "positivi" e quelli che non l'hanno sono considerati "negativi", allora la sensibilità è una misura di quanto bene un test può identificare i veri positivi e la specificità è una misura di quanto bene un test può identificare i veri negativi [16].

Ossia la sensibilità è la probabilità di un risultato positivo del test, condizionata dal fatto che l'individuo sia veramente positivo. La specificità è la probabilità di un risultato negativo del test, condizionata dal fatto che l'individuo sia veramente negativo. Vengono calcolate con le seguenti formule:

$$\text{sensibilità} = \frac{TP}{TP + FN}$$

$$\text{specificità} = \frac{TN}{TN + FP}$$

La precisione, invece, indica quanto vicini o quanto ripetibili siano i risultati e a quanto costantemente qualcosa è strettamente esatto. Viene calcolata con la seguente formula:

$$\text{precisione} = \frac{TP}{TP + FP}$$

L' "F1 score" è una misura dell'accuratezza di un test. La formula per calcolarlo è :

$$F1\ score = \frac{2 * TP}{2 * TP + FP + FN}$$

Poiché le formule usano i veri positivi e i falsi negativi si è distinto il caso in cui vengono considerati i veri positivi totali (veri positivi visibili più veri positivi successivi) dal caso in cui si considerano solo i veri positivi visibili e i falsi negativi comprendono i veri positivi successivi. Dalle figure realizzate si è notato che il possibile e graduale esaurimento di fluoroforo influenza i risultati. Per esaurimento del fluoroforo si intende il fatto che ogni volta che una cellula immunitaria vuota passa vicino al fluoroforo che è sparso nel reticolo, si carica e il fluoroforo in quella posizione viene cancellato mentre nel caso di non esaurimento il fluoroforo anche se viene preso dalla cellula immunitaria rimane disponibile nel reticolo per altre cellule immunitarie.

4.1 CASO IN CUI IL FLUOROFORO SI ESAURISCE

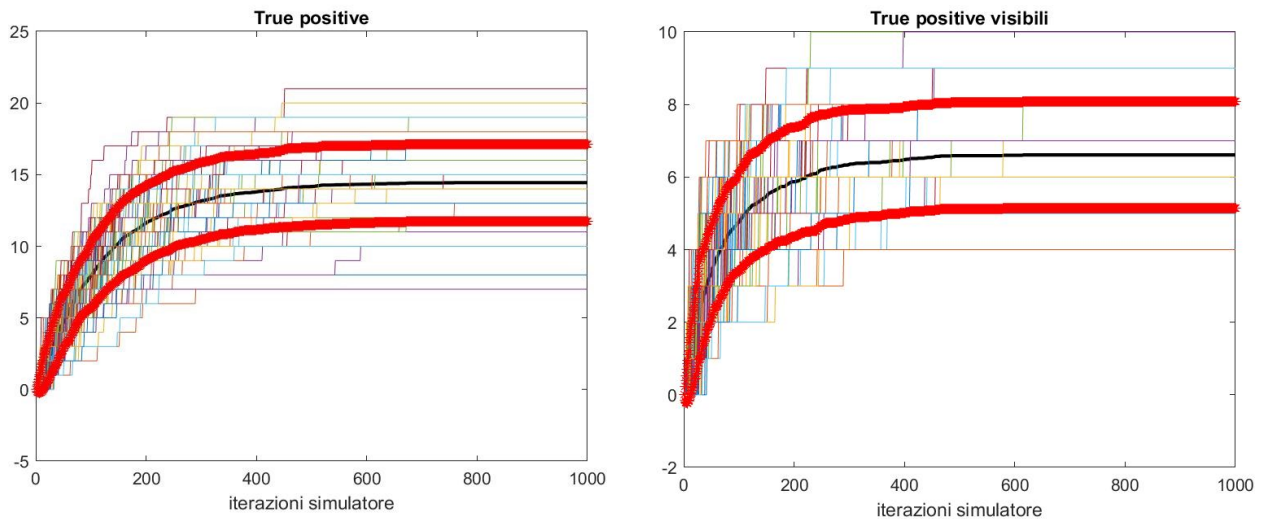


Figura 24. A sinistra grafico dei veri positivi totali al variare delle iterazioni del simulatore. A destra grafico dei veri positivi visibili. Ogni linea colorata mostra l'andamento dei TP per ogni simulazione del simulatore. La linea nera è la media dei valori, le due linee rosse sono $\text{media} \pm \text{deviazione standard}$.

Dai grafici si può vedere che considerando i veri positivi totali le curve raggiungono un valore costante più alto rispetto al caso in cui si considerano solo i veri positivi visibili. Questo avviene perché i veri positivi visibili possono raggiungere un valore massimo uguale a 10 che corrisponde al numero di cellule dendritiche che possono ricevere il fluoroforo come prima volta. Nei veri positivi totali, invece, poiché comprendono anche i veri positivi successivi i valori aumentano.

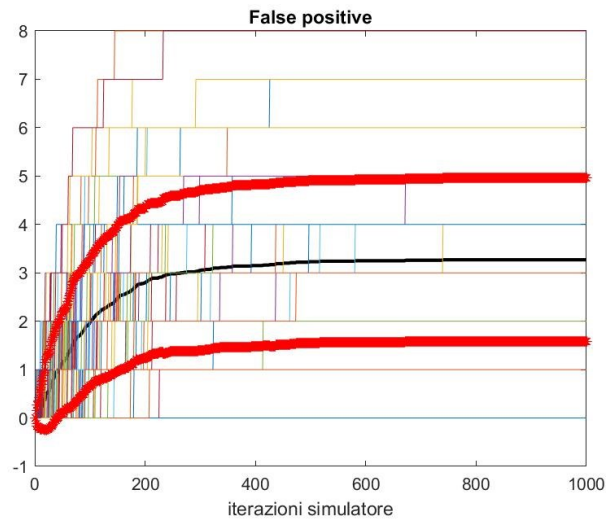


Figura 25. Grafico dei falsi positivi totali al variare delle iterazioni del simulatore. Ogni linea colorata mostra l'andamento dei FP per ogni simulazione del simulatore. La linea nera è la media dei valori, le due linee rosse sono $\text{media} \pm \text{deviazione standard}$.

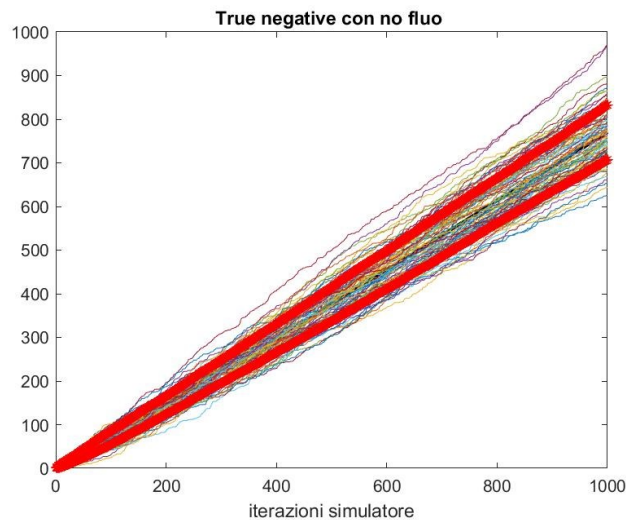


Figura 26. Grafico dei veri negativi totali al variare delle iterazioni del simulatore. Ogni linea colorata mostra l'andamento dei TN per ogni simulazione del simulatore. La linea nera è la media dei valori, le due linee rosse sono $\text{media} \pm \text{deviazione standard}$.

Nell'immagine sopra i falsi positivi aumentano nel tempo raggiungendo un valore costante in media pari a 3. I veri negativi, invece, hanno un andamento crescente nel tempo, raggiungendo un valore in media circa pari a 700 in corrispondenza di 1000 iterazioni del simulatore.

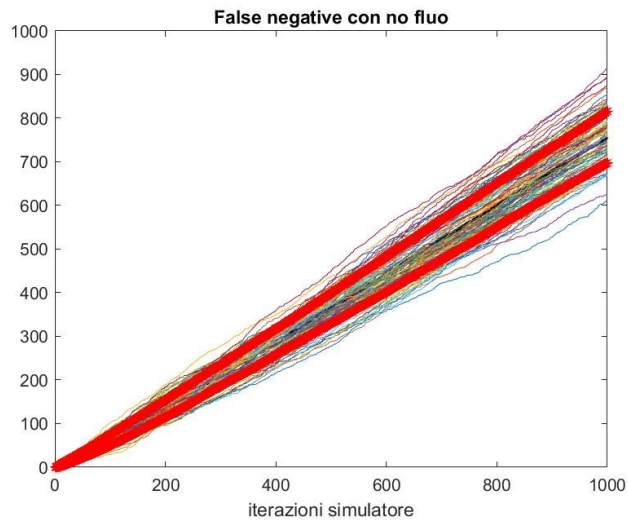


Figura 27. Grafico dei falsi negativi totali al variare delle iterazioni del simulatore. Ogni linea colorata mostra l'andamento dei FN per ogni simulazione del simulatore. La linea nera è la media dei valori, le due linee rosse sono $\text{media} \pm \text{deviazione standard}$.

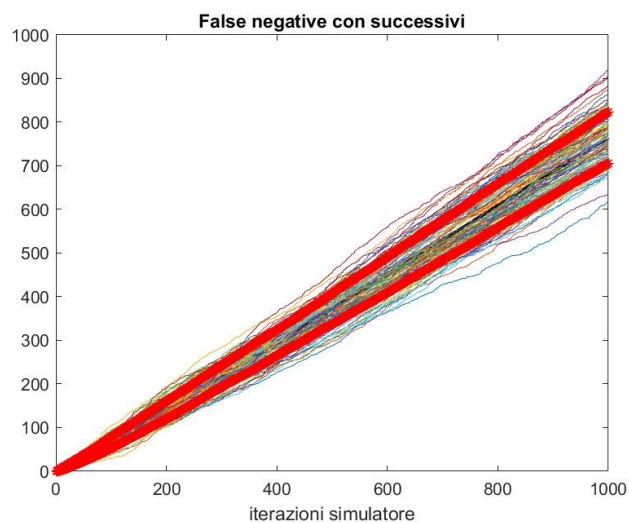


Figura 28. Grafico dei falsi negativi a cui si sono aggiunti i veri positivi successivi, al variare delle iterazioni del simulatore. Ogni linea colorata mostra l'andamento dei FN per ogni simulazione del simulatore. La linea nera è la media dei valori, le due linee rosse sono $\text{media} \pm \text{deviazione standard}$.

Dai grafici sopra non si nota molta differenza nel considerare i falsi negativi totali oppure i falsi negativi a cui abbiamo aggiunto i veri positivi successivi.

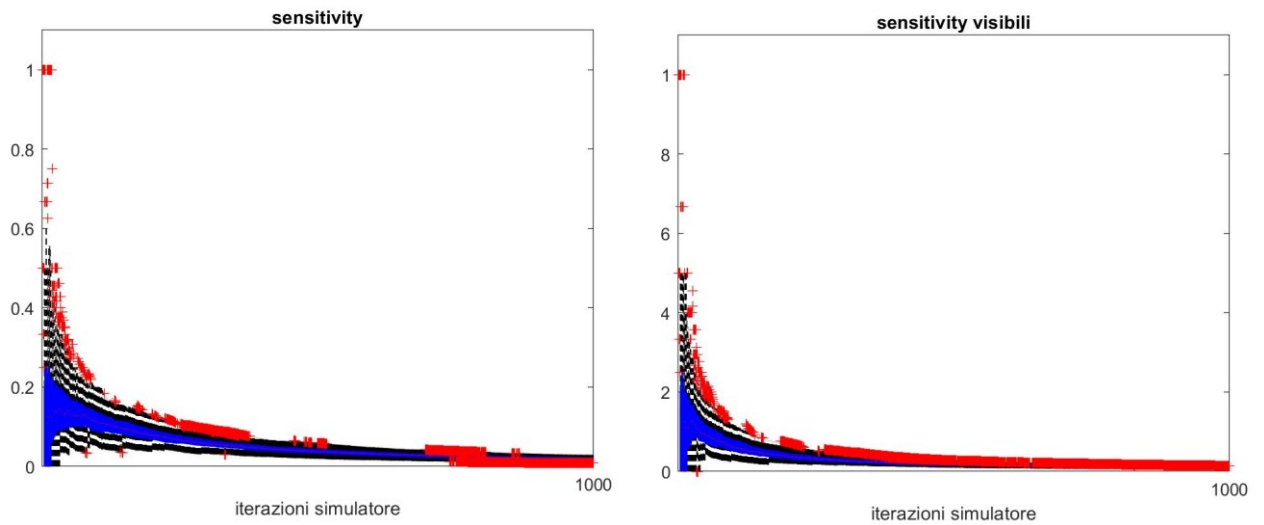


Figura 29. A sinistra si trova il grafico della sensibilità andando a considerare i veri positivi totali e i falsi negativi senza i veri positivi successivi. A destra si trova il grafico della sensibilità considerando i veri positivi visibili. La parte blu del grafico indica come varia la varianza della sensibilità.

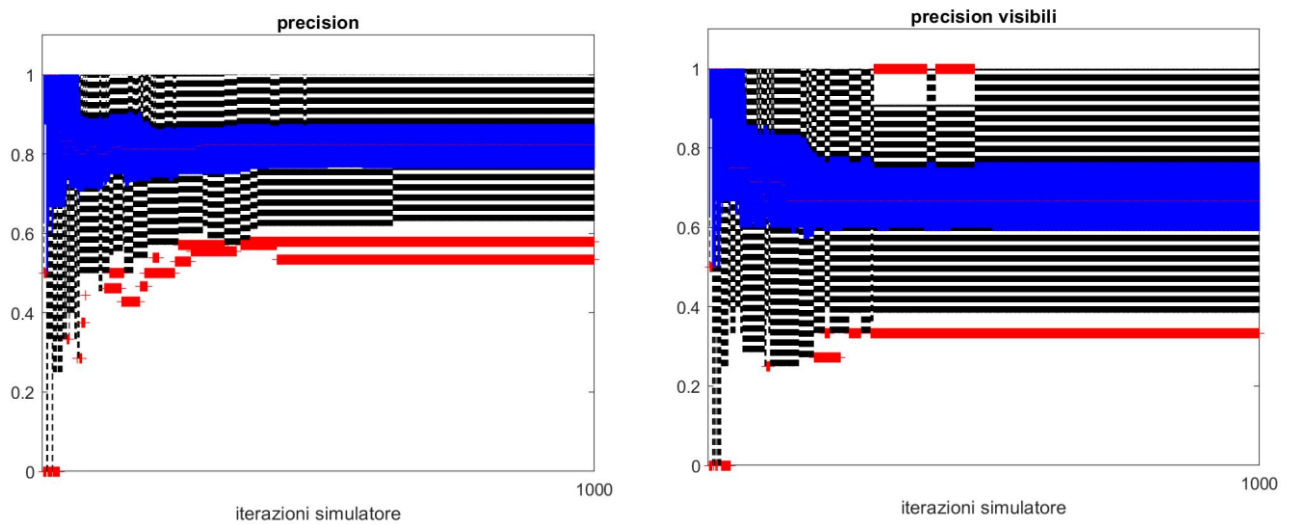


Figura 30. Grafico della precisione andando a considerare i veri positivi totali, a sinistra, grafico della precisione considerando i veri positivi visibili, a destra. La parte blu del grafico indica come varia la varianza della precisione.

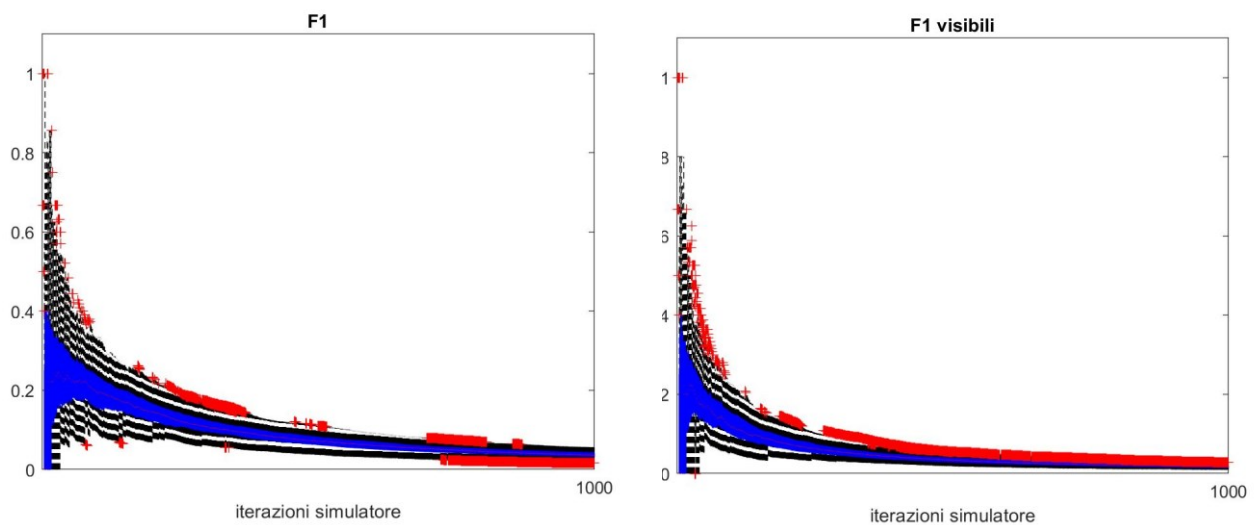


Figura 31. Grafico del “F1 score” andando a considerare i veri positivi totali e i falsi negativi senza i veri positivi successivi, a sinistra. A destra viene mostrato il grafico del “F1 score” andando a considerare i veri positivi visibili. La parte blu del grafico indica come varia la varianza del “F1 score”.

Dai grafici dei falsi negativi, della sensibilità, della precisione e del “F1 score” non si notano grosse differenze considerando i veri positivi totali oppure i veri positivi visibili. La sensibilità ha un andamento a campana che man mano decresce arrivando quasi al valore 0 poiché i veri positivi diminuiscono nel tempo. La precisione, invece, assume un valore in media costante nel tempo (0.8).

4.2 CASO IN CUI IL FLUOROFORO NON SI ESAURISCE

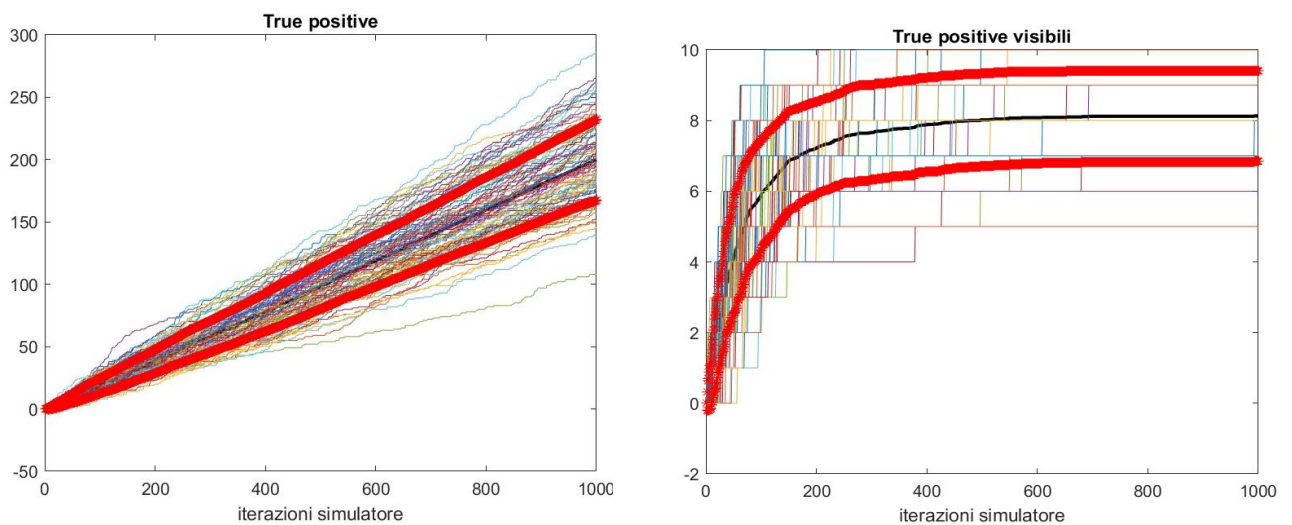


Figura 32. Grafico dei veri positivi totali al variare delle iterazioni del simulatore, a sinistra. Grafico dei veri positivi visibili al variare delle iterazioni del simulatore, a destra. Ogni linea colorata mostra l'andamento dei TP per ogni simulazione del simulatore. La linea nera è la media dei valori, le due linee rosse sono $\text{media} \pm \text{deviazione standard}$.

Dai grafici dei veri positivi totali si nota un andamento delle curve molto diverso rispetto ai veri positivi visibili. Infatti, i veri positivi totali continuano ad aumentare al variare delle iterazioni del simulatore mentre i veri positivi visibili raggiungono un plateau in media di valore 8. Questo avviene perché, nel grafico dei veri positivi totali il fluoroforo rimane sempre disponibile alle cellule immunitarie e quindi il valore dei veri positivi successivi aumenta sempre di più.

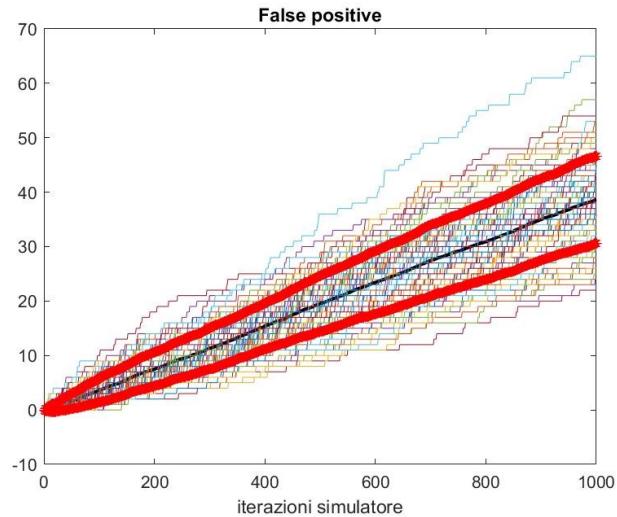


Figura 33. Grafico dei falsi positivi al variare delle iterazioni del simulatore. Ogni linea colorata mostra l'andamento dei FP per ogni simulazione del simulatore. La linea nera è la media dei valori, le due linee rosse sono $\text{media} \pm \text{deviazione standard}$.

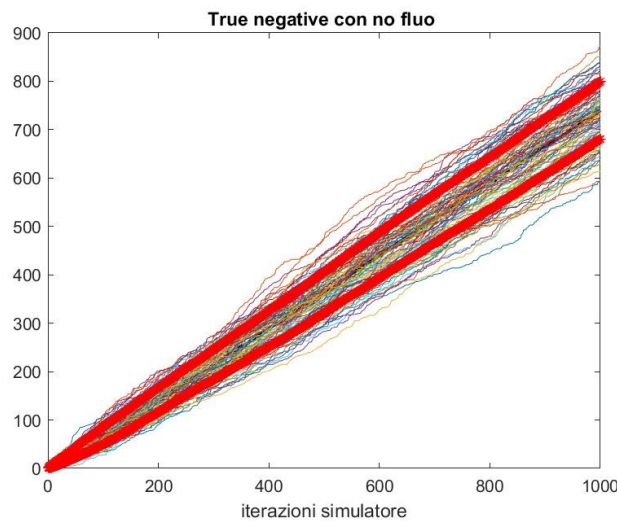


Figura 34. Grafico dei veri negativi totali al variare delle iterazioni del simulatore. Ogni linea colorata mostra l'andamento dei TN per ogni simulazione del simulatore. La linea nera è la media dei valori, le due linee rosse sono $\text{media} \pm \text{deviazione standard}$.

I grafici mostrano un andamento crescente nel tempo sia per i falsi positivi sia per i veri negativi totali.

I falsi positivi raggiungono un valore massimo in media di circa 30 mentre i veri negativi di circa 700.

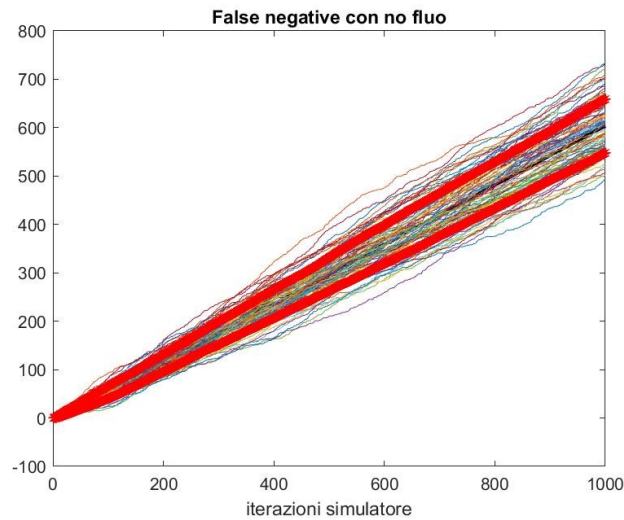


Figura 35. Grafico dei falsi negativi totali. Ogni linea colorata mostra l'andamento dei FN per ogni simulazione del simulatore. La linea nera è la media dei valori, le due linee rosse sono $\text{media} \pm \text{deviazione standard}$.

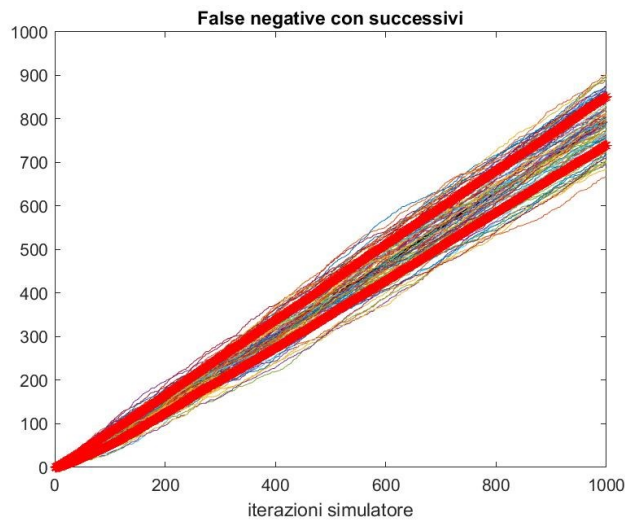


Figura 36. Grafico dei falsi negativi a cui si sono aggiunti i veri positivi successivi, al variare delle iterazioni del simulatore. Ogni linea colorata mostra l'andamento dei FN per ogni simulazione del simulatore. La linea nera è la media dei valori, le due linee rosse sono $\text{media} \pm \text{deviazione standard}$.

Dai grafici sopra non si nota molta differenza nel considerare i falsi negativi totali oppure i falsi negativi a cui abbiamo aggiunto i veri positivi successivi.

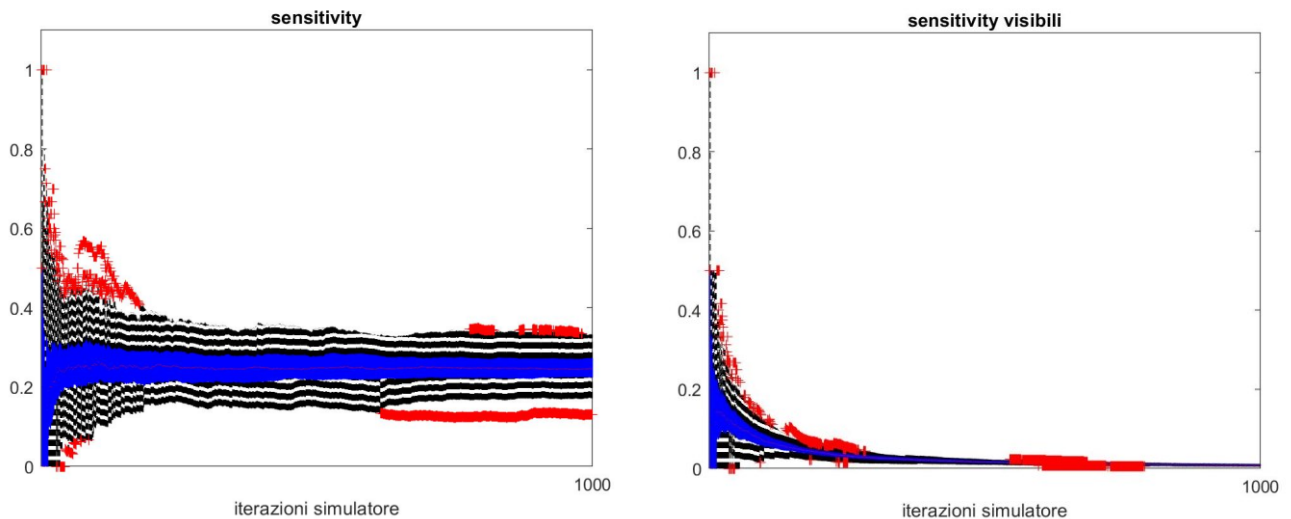


Figura 37. Grafico della sensibilità andando a considerare i veri positivi totali e i falsi negativi senza i veri positivi successivi, a sinistra. Grafico della sensibilità andando a considerare i veri positivi visibili, a destra. La parte blu del grafico indica come varia la varianza della sensibilità.

L'andamento della sensibilità è molto diverso tra i due casi. Quando si considerano i veri positivi totali la sensibilità raggiunge un valore intorno alle 0.3, poiché dalla formula aumentano con il tempo sia i veri successivi totali che i falsi negativi, mentre utilizzando i veri positivi visibili la sensibilità diminuisce fino a raggiungere quasi il valore zero, poiché i veri positivi visibili raggiungono un valore costante e i falsi negativi aumentano sempre di più.

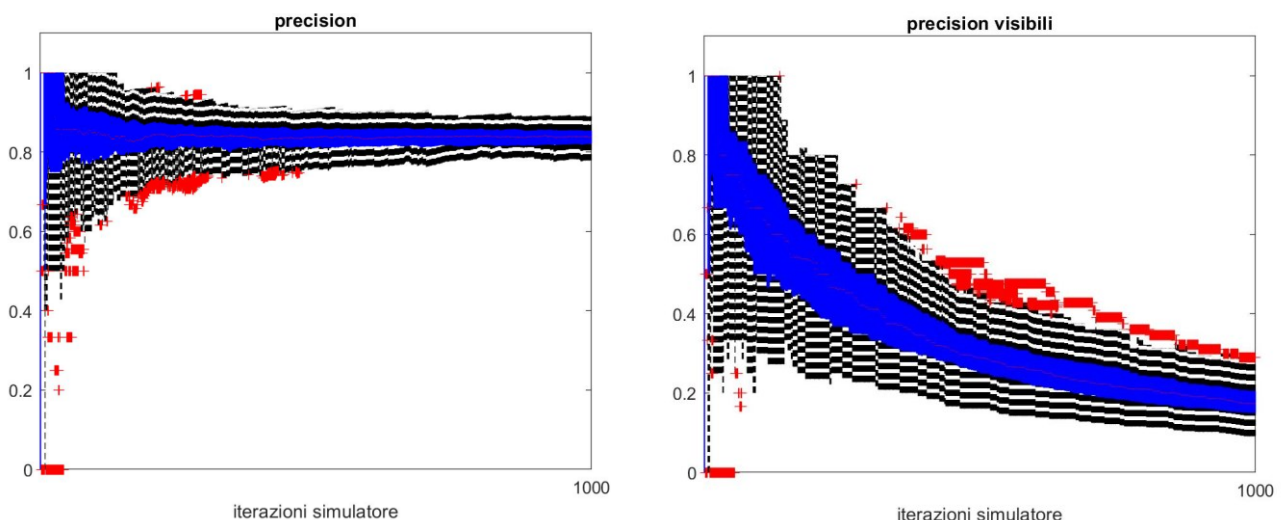


Figura 38. Grafico della precisione andando a considerare i veri positivi totali, a sinistra. Grafico della precisione andando a considerare i veri positivi visibili, a destra. La parte blu del grafico indica come varia la varianza della precisione.

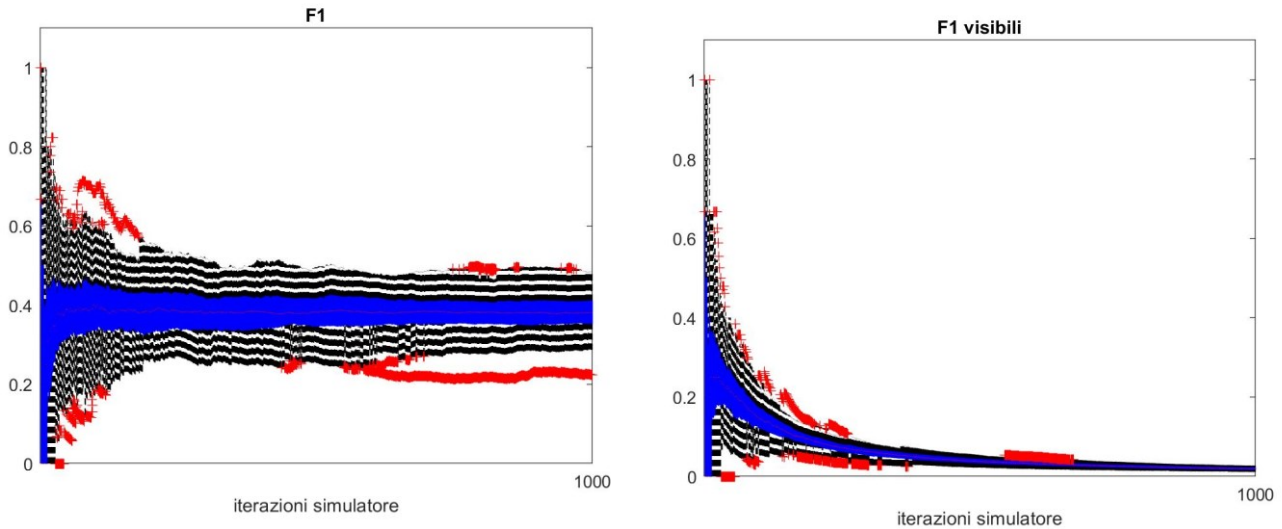


Figura 39. Grafico del “F1 score” andando a considerare i veri positivi totali e i falsi negativi senza i veri positivi successivi, a sinistra. Grafico del “F1 score” considerando i veri positivi visibili, a destra. La parte blu del grafico indica come varia la varianza del “F1 score”.

Sia per la precisione che per il “F1 score” si ha la stessa differenza tra gli andamenti dei due casi.

Considerando i veri positivi totali la precisione assume un valore in media costante circa pari a 0.8 mentre l’“F1 score” il valore costante è circa pari allo 0.4.

Considerando i veri positivi visibili, invece, si ha un andamento decrescente nel tempo sia per la precisione che per l’“F1 score”.

Nei problemi fortemente sbilanciati dove la quantità di interesse sono gli elementi positivi che accadono poche volte, le metriche basate sugli eventi negativi, per esempio specificity, false negative rate, accuracy, negative predictive rate, false positive rate, non sono particolarmente informative [17].

CONCLUSIONI

Lo scopo del presente lavoro di tesi è stato quello di voler spiegare in cosa consiste l'approccio LIPSTIC cercando, attraverso l'uso della piattaforma Matlab, di simulare l'interazione tra le cellule del sistema immunitario, in particolare le cellule T e le cellule dendritiche.

Per introdurre l'argomento è stato analizzato il sistema immunitario da un punto di vista teorico, tentando di comprenderne gli aspetti più generali, dopo di che si è cercato di studiare in modo più approfondito le proprietà e funzionalità delle cellule dendritiche e cellule T del sistema immunitario.

In primo luogo, si è studiato cosa si intende per approccio LIPSTIC, ossia un metodo per monitorare le interazioni tra i due tipi di cellule immunitarie basato sull'uso di un'etichetta rilevabile come la biotina o fluoroforo: quando le due cellule interagiscono avviene il passaggio del fluoroforo da una cellula all'altra.

In secondo luogo, si è cercato di realizzare l'interazione tra le cellule attraverso la creazione di quattro simulatori in cui si possono vedere le cellule immunitarie muoversi in un reticolo e interagire tra di loro.

In particolare, si è distinto il caso in cui le cellule immunitarie sono distribuite in modo casuale nel reticolo dal caso in cui sono distribuite in modo ordinato.

Il fluoroforo è stato aggiunto al reticolo successivamente, prima inserendolo all'interno delle cellule immunitarie poi distribuito casualmente nel reticolo. In quest'ultimo caso, le cellule T dovevano prima caricarsi di fluoroforo e poi interagendo con la cellula dendritica passarlo a quella più vicina.

Dopo di che, sono stati conteggiati i veri positivi, veri negativi, falsi positivi, falsi negativi che sono serviti per l'analisi dei risultati: dalle figure ottenute si può vedere l'andamento di questi parametri al variare delle iterazioni del simulatore.

Si è posta più attenzione all'andamento dei veri positivi "visibili" poiché si riferiscono a tutti quegli eventi in cui una cellula T carica di fluoroforo, si lega e passa il fluoroforo a una cellula dendritica per la prima volta. Questi casi sono molto importanti perché appartengono all'oggetto del nostro studio.

L'andamento dei veri positivi "visibili", infatti è crescente nel tempo e arriva ad un massimo di 10 poiché è il numero massimo di cellule dendritiche a cui una cellula T può passare il fluoroforo.

Per concludere, sono state calcolate delle metriche statistiche come la sensibilità, specificità e precisione.

È di maggior importanza la sensibilità, probabilità di un risultato positivo al test (legame e passaggio di fluoroforo come prima volta) che, considerando sempre come veri positivi quelli “visibili”, decresce nel tempo poiché man mano che passa il tempo tutte le cellule dendritiche ricevono il fluoroforo e quindi non ci sarà più un esito positivo al test.

Infine, i risultati ottenuti mostrano che tale simulazione è una buona rappresentazione dell’approccio LIPSTIC. Inoltre, attraverso questo simulatore sarà possibile andare a studiare come la variazione di alcuni parametri strutturali del sistema, ad esempio l’affinità ligando recettore o fluoroforo sortasi (codificate come probabilità di evento) possano influenzare il funzionamento del sistema e quindi portare a eventuali future ottimizzazioni di LIPSTIC.

RINGRAZIAMENTI

Questa tesi è il giusto finale di un percorso di studi fatto di gioie e successi, ma anche di delusioni che mi hanno aiutata a crescere e a superare i miei limiti.

Ci tengo a ringraziare innanzitutto coloro che hanno reso realizzabile questa esperienza, ovvero i miei genitori, che mi hanno sempre mostrato il loro supporto e per i quali provo un'immensa ammirazione.

Ringrazio i miei fratelli, che con le loro esperienze personali e lavorative hanno saputo darmi dei consigli molto utili.

Ringrazio Michele, che è al mio fianco da ben dieci anni, per tutto il supporto, per le avventure passate insieme e per essermi accanto ogni singolo giorno.

Ringrazio amici, colleghi universitari in particolare Debora per le ore di spensieratezza e vita passate insieme.

Ringrazio il Professor Del Favero, l'Ingegnere Bellato e Chiara Cimolato, per avermi guidato e supportato durante la realizzazione della tesi e per avermi dato la possibilità di imparare nuove competenze.

BIBLIOGRAFIA E SITOGRAFIA

- [1] Pasqual, G., Chudnovskiy, A., Tas, J. M., Agudelo, M., Schweitzer, L. D., Cui, A., ... & Victora, G. D. (2018). Monitoring T cell–dendritic cell interactions in vivo by intercellular enzymatic labelling. *Nature*, 553(7689), 496-500.
- [2] McKinnon, K. M. (2018). Flow cytometry: an overview. *Current protocols in immunology*, 120(1), 5-1.
- [3] Rothenberg, R. E., Minelli, L., & Caslini-Cavalli, A. (1998). *Enciclopedia della medicina Garzanti*. Garzanti.
- [4] Lichtman, A. H., Abbas, A. K., & Pillai, S. (2015). *Immunologia cellulare e molecolare*. Elsevier Italia.
- [5] Delves, P. J., & Roitt, I. M. (2000). The immune system. *New England journal of medicine*, 343(1), 37-49.
- [6] Yatim, K. M., & Lakkis, F. G. (2015). A brief journey through the immune system. *Clinical journal of the American Society of Nephrology: CJASN*, 10(7), 1274.
- [7] Dott.ssa S.Ramondo. *Il sistema immunitario* **URL:**
https://www.disputer.unich.it/sites/st13/files/sistema_immunitario.pdf accesso in data 18/06/2023
- [8] Shaw, A. C., Joshi, S., Greenwood, H., Panda, A., & Lord, J. M. (2010). Aging of the innate immune system. *Current opinion in immunology*, 22(4), 507-513.
- [9] Cooper, M. D., & Alder, M. N. (2006). The evolution of adaptive immune systems. *Cell*, 124(4), 815-822.

- [10] Bell, D., Young, J. W., & Banchereau, J. (1999). Dendritic cells. *Advances in immunology*, 72, 255-324.
- [11] Banchereau, J., Briere, F., Caux, C., Davoust, J., Lebecque, S., Liu, Y. J., ... & Palucka, K. (2000). Immunobiology of dendritic cells. *Annual review of immunology*, 18(1), 767-811.
- [12] Enciclopedia Treccani-Enciclopedia della Scienza e della Tecnica (2008). *Cellula dendritica* **URL:** https://www.treccani.it/enciclopedia/cellula-dendritica_%28Enciclopedia-della-Scienza-e-della-Tecnica%29 accesso in data 14/06/2023
- [13] Romagnani, S. (2000). The role of lymphocytes in allergic disease. *Journal of Allergy and Clinical Immunology*, 105(3), 399-408.
- [14] Enciclopedia Treccani-Enciclopedia della scienza e della Tecnica (2008). *I linfociti T*. **URL:** https://www.treccani.it/enciclopedia/linfociti-t_%28Enciclopedia-della-Scienza-e-della-Tecnica%29 accesso in data 14/06/2023
- [15] Peter J. Delves , PhD, University College London, London, UK (2021). *Componenti cellulari del sistema immunitario*. **URL:** <https://www.msmanuals.com/it-it/professionale/immunologia-malattie-allergiche/biologia-del-sistema-immunitario/componenti-cellulari-del-sistema-immunitario> accesso in data 14/06/2023
- [16] Saah, A. J., & Hoover, D. R. (1998, April). Sensitivity and specificity revisited: significance of the terms in analytic and diagnostic language. In *Annales de Dermatologie et de Venereologie* (Vol. 125, No. 4, pp. 291-294).

[17] Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, *10*(3), e0118432.

APPENDICE

CODICE DEL SIMULATORE CON LE CELLULE IMMUNITARIE DISTRIBUITE IN MODO CASUALE E RIEMPITE DI FLUOROFORO

```
close all
clear all
clc

%% inizializzazione parametri generali del simulatore

NumCellDendr=10; %cellule bersaglio
NumCellimm=10; %cellule sistema immunitario

%dimensioni reticolo bidimensionale

x=10;
y=10;

%iterazioni simulatore

T=20;

% lunghezza spostamento delle cellule immunitarie

step_length_cellimm=0.5;

%info sul video

outputVideo=VideoWriter('cellule_in_movimento.avi');
outputVideo.FrameRate=2;
open(outputVideo);

%soglia

threshold=1;

% varie probabilità

p_scambio_fluoroforo_dendimm=0.6; % probabilità che una cellula immunitaria
scambi il fluoroforo con cellula dendritica
p_interact=0.6; % probabilità che due cellule interagiscono
p_step=1; % probabilità che una cellula immunitaria si muova
p_distacco=0.1; % probabilità che una cellula immunitaria si stacchi dalla
cellula dendritica

%% inizializzazione matrice e vettore di interazione

matrice_interazione=zeros(T,NumCellimm);
vettore_interazione=zeros(1,NumCellimm);

%% inizializzazione vettore del scambio fluoroforo
```

```

vettore_scambio_fluoro=zeros(1,NumCellDendr);

%% inizializzazione random cellule bersaglio e cellule sistema immunitario

%distribuzione random delle cellule bersaglio

[pos_x_dendr, pos_y_dendr]=inizializza_cell_dendr_random(x,y,NumCellDendr);

%distribuzione random delle cellule del sistema immunitario
j=1;
[s_random, t_random]=inizializza_cel_immun(x,y,NumCellimm);
[pos_x_imm,
pos_y_imm]=step_cell_immun(s_random,t_random,p_step,matrice_interazione,step_len
gth_cellimm,j);

%% visualizzazione delle cellule bersaglio e cellule immunitarie in una figura
for j=1:T

    if j==1
        I=plot(s_random,t_random,'bo');
        hold on
        plot(pos_x_dendr,pos_y_dendr,'gs','LineWidth',1,'MarkerSize',10,'Marke
rEdgeColor','r','MarkerFaceColor',[0.5,0.5,0.5]);
        sz = 10;
        scatter(s_random,t_random,sz,'y','filled');

        for i=1:NumCellDendr

            x_cerchio=pos_x_dendr(1,i)-threshold:0.01:pos_x_dendr(1,i)+threshold;
            raggio=threshold;
            for indice=1:length(x_cerchio)
                y_cerchio_2(indice)=-sqrt(threshold.^2-(x_cerchio(indice)-
pos_x_dendr(1,i)).^2)+pos_y_dendr(1,i);
                y_cerchio(indice)=sqrt(threshold.^2-(x_cerchio(indice)-
pos_x_dendr(1,i)).^2)+pos_y_dendr(1,i);
            end
            plot(x_cerchio,y_cerchio,'k--');
            hold on
            plot(x_cerchio,y_cerchio_2,'k--');
            end
            grid on
            grid minor
            xlim([-0.1.*x x.*1.1]);
            ylim([-0.1.*y y.*1.1]);
            title(['cellule dentritiche(quadrati) cellule T sistema
immunitario(cerchi)--iterazione ' num2str(j)]);

            file_name=['iterazione'];
            saveas(I,file_name,'png');
            img=imread(file_name,'png');
            writeVideo(outputVideo,img);
        end

    close all
%%verifico se le cellule immunitarie e bersaglio interagiscono
for indCellimm= 1:NumCellimm
    for indCelldendr=1:NumCellDendr

```

```

distanza_euclidea=norma(pos_x_imm(j,indCellimm),pos_x_dendr(1,indCelldendr),pos_
y_imm(j,indCellimm),pos_y_dendr(1,indCelldendr));

    if distanza_euclidea < threshold
        prob=rand();
        if prob<p_interact && vettore_interazione(1,indCellimm)==0

            matrice_interazione(j,indCellimm)=indCelldendr;
            vettore_interazione(1,indCellimm)=1;
        end

        p_scambio=rand();
        if p_scambio< p_scambio_fluoroforo_dendimm
            vettore_scambio_fluoro(1,indCelldendr)=1;
        end

        probabilita=rand();
        if probabilita<p_distacco && vettore_interazione(1,indCellimm)~=0

            matrice_interazione(j,indCellimm)=0;
            vettore_interazione(1,indCellimm)=0;
        end

    end
end
end

%spostamento delle cellule sistema immunitario

[pos_x_imm(j+1,:),
pos_y_imm(j+1,:)] = step_cell_immun(pos_x_imm(j,:),pos_y_imm(j,:),p_step,matrice_i
nterazione,step_length_cellimm,j);

% visualizzazione cellule immunitarie
for indicecellimm=1:NumCellimm

    r=matrice_interazione(j,indicecellimm);
    if r~=0

plot(pos_x_imm(j,indicecellimm),pos_y_imm(j,indicecellimm),'go');
        hold on
        plot([pos_x_imm(j,indicecellimm)
pos_x_dendr(1,r)], [pos_y_imm(j,indicecellimm) pos_y_dendr(1,r)], '-b')
        sz = 10;

scatter(pos_x_imm(j,indicecellimm),pos_y_imm(j,indicecellimm),sz,'y','filled');

    else
        I=plot(pos_x_imm(j,indicecellimm),pos_y_imm(j,indicecellimm),'bo');
        hold on
        sz = 10;

scatter(pos_x_imm(j,indicecellimm),pos_y_imm(j,indicecellimm),sz,'y','filled');

    end
end

%visualizzazione cellule dendritiche

```

```

    for indicecellden=1:NumCellDendr
        r=find(matrice_interazione(j,:)==indicecellden);
        if vettore_scambio_fluoro(1,indicecellden)==1
            if length(r)>0

plot(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),'gs','LineWidth',
1,'MarkerSize',10,'MarkerEdgeColor','g','MarkerFaceColor',[0.5,0.5,0.5]);
            sz = 10;

scatter(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),sz,'y','filled
');

                else

plot(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),'gs','LineWidth',
1,'MarkerSize',10,'MarkerEdgeColor','r','MarkerFaceColor',[0.5,0.5,0.5]);
            sz = 10;

scatter(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),sz,'y','filled
');

                end
            else
                if length(r)>0

plot(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),'gs','LineWidth',
1,'MarkerSize',10,'MarkerEdgeColor','g','MarkerFaceColor',[0.5,0.5,0.5]);
                else

plot(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),'gs','LineWidth',
1,'MarkerSize',10,'MarkerEdgeColor','r','MarkerFaceColor',[0.5,0.5,0.5]);
                end
            end
        end

% visualizzazione regione di interazione
for i=1:NumCellDendr

    x_cerchio=pos_x_dendr(1,i)-threshold:0.01:pos_x_dendr(1,i)+threshold;
    raggio=threshold;
    for indice=1:length(x_cerchio)
        y_cerchio_2(indice)=-sqrt(threshold.^2-(x_cerchio(indice)-
pos_x_dendr(1,i)).^2)+pos_y_dendr(1,i);
        y_cerchio(indice)=sqrt(threshold.^2-(x_cerchio(indice)-
pos_x_dendr(1,i)).^2)+pos_y_dendr(1,i);
    end
    plot(x_cerchio,y_cerchio,'k--');
    hold on
    plot(x_cerchio,y_cerchio_2,'k--');
end

grid on
grid minor
xlim([-0.1.*x x.*1.1]);
ylim([-0.1.*y y.*1.1]);
title(['cellule dendritiche(quadrati) cellule T sistema immunitario(cerchi)--
iterazione ' num2str(j)]);

%salvataggio delle figure a ogni iterazione del simulatore nel video
file_name=['iterazione'];
saveas(I,file_name,'png');
img=imread(file_name,'png');

```

```

    writeVideo(outputVideo,img);
end
close(outputVideo);
% visualizzazione del video
imshow('cellule_in_movimento.avi');

```

CODICE DEL SIMULATORE CON LE CELLULE DISTRIBUITE IN MODO ORDINATO E RIEMPITE DI FLUOROFORO

```

close all
clear all
clc

%% inizializzazione parametri generali del simulatore

NumCellDendr=10; %cellule bersaglio
NumCellimm=10; %cellule sistema immunitario

%dimensioni reticolo bidimensionale

x=10;
y=10;

%iterazioni simulatore

T=10;

% lunghezza spostamento delle cellule immunitarie

step_length_cellimm=0.5;

%info sul video

outputVideo=VideoWriter('cellule_in_movimento.avi');
outputVideo.FrameRate=2;
open(outputVideo);

%soglia

threshold=1;

% varie probabilità

p_scambio_fluoroforo_dendimm=0.8; % probabilità che una cellula immunitaria
scambi il fluoroforo con cellula dendritica
p_interact=0.6; % probabilità che due cellule interagiscono
p_step=1; % probabilità che una cellula immunitaria si muova
p_distacco=0.1; % probabilità che una cellula immunitaria si stacchi dalla
cellula dendritica

%% inizializzazione matrice di interazione e vettore iterazione

matrice_interazione=zeros(T,NumCellimm);
vettore_interazione=zeros(1,NumCellimm);
%% inizializzazione vettore del scambio fluoroforo

```

```

vettore_scambio_fluoro=zeros(1,NumCellDendr);

%% inizializzazione cellule bersaglio e cellule sistema immunitario

%distribuzione random delle cellule bersaglio

[pos_x_dendr, pos_y_dendr]=inizializza_cell_dendr_ord(NumCellDendr);

%distribuzione random delle cellule del sistema immunitario
j=1;
[s_random, t_random]=inizializza_cel_immun_ord(NumCellimm);
[pos_x_imm
, pos_y_imm]=step_cell_immun_ord(s_random,t_random,p_step,matrice_interazione,ste
p_length_cellimm,j);

%% visualizzazione delle cellule bersaglio e cellule immunitarie in una figura
for j=1:T

    if j==1
        I=plot(s_random,t_random,'bo');
        hold on

plot(pos_x_dendr,pos_y_dendr,'gs','LineWidth',1,'MarkerSize',10,'MarkerEdgeColor
','r','MarkerFaceColor',[0.5,0.5,0.5]);
        sz = 10;
        scatter(s_random,t_random,sz,'y','filled');
        plot([1 9],[5 5], '-.','color',0.2*[1 1 1])
        plot([0 1],[0 5], '-.','color',0.2*[1 1 1])
        plot([0 1],[10 5], '-.','color',0.2*[1 1 1])
        plot([10 9],[0 5], '-.','color',0.2*[1 1 1])
        plot([9 10],[5 10], '-.','color',0.2*[1 1 1])

        for i=1:NumCellDendr

            x_cerchio=pos_x_dendr(1,i)-threshold:0.01:pos_x_dendr(1,i)+threshold;
            raggio=threshold;
            for indice=1:length(x_cerchio)
                y_cerchio_2(indice)=-sqrt(threshold.^2-(x_cerchio(indice)-
pos_x_dendr(1,i)).^2)+pos_y_dendr(1,i);
                y_cerchio(indice)=sqrt(threshold.^2-(x_cerchio(indice)-
pos_x_dendr(1,i)).^2)+pos_y_dendr(1,i);
            end
            plot(x_cerchio,y_cerchio,'k--');
            hold on
            plot(x_cerchio,y_cerchio_2,'k--');
            end
            grid on
            grid minor
            xlim([-0.1.*x x.*1.1]);
            ylim([-0.1.*y y.*1.1]);
            title(['cellule dendritiche(quadrati) cellule T sistema
immunitario(cerchi)--iterazione ' num2str(j)]);

            file_name=['iterazione'];
            saveas(I,file_name,'png');
            img=imread(file_name,'png');
            writeVideo(outputVideo,img);
        end
end

```

```

close all
%verifico se le cellule immunitarie e bersaglio interagiscono
for indCellimm= 1:NumCellimm
    for indCelldendr=1:NumCellDendr

distanza_euclidea=norma(pos_x_imm(j,indCellimm),pos_x_dendr(1,indCelldendr),pos_
y_imm(j,indCellimm),pos_y_dendr(1,indCelldendr));

        if distanza_euclidea < threshold
            prob=rand();
            if prob<p_interact && vettore_interazione(1,indCellimm)==0

                matrice_interazione(j,indCellimm)=indCelldendr;
                vettore_interazione(1,indCellimm)=1;
            end
            p_scambio=rand();
            if p_scambio< p_scambio_fluoroforo_dendimm
                vettore_scambio_fluoro(1,indCelldendr)=1;
            end
            probabilita=rand();
            if probabilita<p_distacco && vettore_interazione(1,indCellimm)~=0

                matrice_interazione(j,indCellimm)=0;
                vettore_interazione(1,indCellimm)=0;
            end
        end
    end
end

%spostamento delle cellule sistema immunitario

[pos_x_imm(j+1,:),
pos_y_imm(j+1,:)] = step_cell_immun_ord(pos_x_imm(j,:),pos_y_imm(j,:),p_step,matri
ce_interazione,step_length_cellimm,j);

% visualizzazione cellule immunitarie

for indicecellimm=1:NumCellimm

    r=matrice_interazione(j,indicecellimm);
    if r~=0

I=plot(pos_x_imm(j,indicecellimm),pos_y_imm(j,indicecellimm),'go');
        hold on
        plot([pos_x_imm(j,indicecellimm)
pos_x_dendr(1,r)], [pos_y_imm(j,indicecellimm) pos_y_dendr(1,r)], '-b')
        sz = 10;

scatter(pos_x_imm(j,indicecellimm),pos_y_imm(j,indicecellimm),sz,'y','filled');

    else
        I=plot(pos_x_imm(j,indicecellimm),pos_y_imm(j,indicecellimm),'bo');
        hold on
        sz = 10;

scatter(pos_x_imm(j,indicecellimm),pos_y_imm(j,indicecellimm),sz,'y','filled');

```



```

        end
    end
    %visualizzazione cellule dendritiche

    for indicecellden=1:NumCellDendr
        r=find(matrice_interazione(j,:)==indicecellden);
        if vettore_scambio_fluoro(1,indicecellden)==1
            if length(r)>0

plot(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),'gs','LineWidth',
1,'MarkerSize',10,'MarkerEdgeColor','g','MarkerFaceColor',[0.5,0.5,0.5]);
            sz = 10;

scatter(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),sz,'y','filled
');

                else

plot(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),'gs','LineWidth',
1,'MarkerSize',10,'MarkerEdgeColor','r','MarkerFaceColor',[0.5,0.5,0.5]);
            sz = 10;

scatter(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),sz,'y','filled
');

                end
            else
                if length(r)>0

plot(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),'gs','LineWidth',
1,'MarkerSize',10,'MarkerEdgeColor','g','MarkerFaceColor',[0.5,0.5,0.5]);
                else

plot(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),'gs','LineWidth',
1,'MarkerSize',10,'MarkerEdgeColor','r','MarkerFaceColor',[0.5,0.5,0.5]);
                end
            end
        end
    end

    % visualizzazione divisione del reticolo

plot([1 9],[5 5],'-.','color',0.2*[1 1 1])
plot([0 1],[0 5],'-.','color',0.2*[1 1 1])
plot([0 1],[10 5],'-.','color',0.2*[1 1 1])
plot([10 9],[0 5],'-.','color',0.2*[1 1 1])
plot([9 10],[5 10],'-.','color',0.2*[1 1 1])

    % visualizzazione regione di interazione cellule

for i=1:NumCellDendr

    x_cerchio=pos_x_dendr(1,i)-threshold:0.01:pos_x_dendr(1,i)+threshold;
    raggio=threshold;
    for indice=1:length(x_cerchio)
        y_cerchio_2(indice)=-sqrt(threshold.^2-(x_cerchio(indice)-
pos_x_dendr(1,i)).^2)+pos_y_dendr(1,i);
        y_cerchio(indice)=sqrt(threshold.^2-(x_cerchio(indice)-
pos_x_dendr(1,i)).^2)+pos_y_dendr(1,i);
    end
    plot(x_cerchio,y_cerchio,'k--');
    hold on
end

```

```

    plot(x_cerchio,y_cerchio_2,'k--');
end

grid on
grid minor
xlim([-0.1.*x x.*1.1]);
ylim([-0.1.*y y.*1.1]);
title(['cellule dendritiche(quadrati) cellule T sistema immunitario(cerchi)--
iterazione ' num2str(j)]);

%salvataggio delle figure a ogni iterazione del simulatore nel video
file_name=['iterazione'];
saveas(I,file_name,'png');
img=imread(file_name,'png');
writeVideo(outputVideo,img);
end
close(outputVideo);
% visualizzazione del video
imshow('cellule_in_movimento.avi');

```

CODICE DEL SIMULATORE CON LE CELLULE IMMUNITARIE E IL FLUOROFORO DISTRIBUITI IN MODO CASUALE

```
close all
clc

%% inizializzazione parametri generali del simulatore

NumCellDendr=10; %cellule dendritiche
NumCellimm=10; %cellule sistema immunitario

%dimensioni reticolo bidimensionale

x=10;
y=10;

%iterazioni simulatore

T=40;

% lunghezza spostamento delle cellule immunitarie

step_length_cellimm=0.5;

%info sul video

outputVideo=VideoWriter('cellule_in_movimento.avi');
outputVideo.FrameRate=2;
open(outputVideo);

%soglie

threshold=1;

threshold_fluoroforo_cellimm=0.5;

% varie probabilità

p_scambio_fluoroforo_dendimm=0.1; % probabilità che una cellula immunitaria
scambi il fluoroforo con cellula dendritica
p_interact=0.5; % probabilità che due cellule interagiscono
p_step=1; % probabilità che una cellula immunitaria si muova
p_distacco=0.2; % probabilità che una cellula immunitaria si stacchi dalla
cellula dendritica
p_fluoroforo=1; %probabilità che una cellula immunitaria senza fluoroforo si
carichi di fluoroforo

%% inizializzazione fluoroforo

x_fluoroforo=x*rand(1,30);
y_fluoroforo=y*rand(1,30);

%% inizializzazione matrice e vettore di interazione

matrice_interazione=zeros(T,NumCellimm);

matrice_memoria_interazione=zeros(T,NumCellDendr);
```

```

%% inizializzazione vettore del scambio fluoroforo

vettore_scambio_fluoro=zeros(1,NumCellDendr);

%% inizializzazione vettore fluoroforo

vettore_fluoro_cellimm=zeros(1,NumCellimm);

%% vettori per contare TP,FP,TN,FN,TN_nofluo,FN_nofluo,TP_successivi
count_TP=zeros(1,T);
count_FN=zeros(1,T);
count_TN=zeros(1,T);
count_FP=zeros(1,T);
count_TN_nofluo=zeros(1,T);
count_FN_nofluo=zeros(1,T);
count_TP_successivi=zeros(1,T);

%% inizializzazione vettore fluo_passato

fluo_passato=zeros(1,NumCellimm);

%% inizializzazione random cellule bersaglio e cellule sistema immunitario

%distribuzione random delle cellule bersaglio

[pos_x_dendr, pos_y_dendr]=inizializza_cell_dendr_random(x,y,NumCellDendr);

%distribuzione random delle cellule del sistema immunitario
j=1;
[s_random, t_random]=inizializza_cel_immun(x,y,NumCellimm);
[pos_x_imm
, pos_y_imm]=step_cell_immun(s_random,t_random,p_step,matrice_interazione,step_le
ngth_cellimm,j);

%% visualizzazione delle cellule bersaglio e cellule immunitarie in una figura

for j=1:T

    if j==1

        I=plot(s_random,t_random,'bo');
        hold on

plot(pos_x_dendr,pos_y_dendr,'gs','LineWidth',1,'MarkerSize',10,'MarkerEdgeColor
','r','MarkerFaceColor',[0.5,0.5,0.5]);
        sz = 30;
        scatter(x_fluoroforo,y_fluoroforo,sz,'y','filled');

        for i=1:NumCellDendr

            x_cerchio=pos_x_dendr(1,i)-threshold:0.01:pos_x_dendr(1,i)+threshold;
            raggio=threshold;
            for indice=1:length(x_cerchio)
                y_cerchio_2(indice)=-sqrt(threshold.^2-(x_cerchio(indice)-
pos_x_dendr(1,i)).^2)+pos_y_dendr(1,i);

```

```

        y_cerchio(indice)=sqrt(threshold.^2-(x_cerchio(indice)-
pos_x_dendr(1,i)).^2)+pos_y_dendr(1,i);
    end
    plot(x_cerchio,y_cerchio,'k--');
    hold on
    plot(x_cerchio,y_cerchio_2,'k--');
    end
    grid on
    grid minor
    xlim([-0.1.*x x.*1.1]);
    ylim([-0.1.*y y.*1.1]);
    title(['cellule dentritiche(quadrati) cellule T sistema
immunitario(cerchi)--iterazione ' num2str(j)]);

    file_name=['iterazione'];
    saveas(I,file_name,'png');
    img=imread(file_name,'png');
    writeVideo(outputVideo,img);
else

    % copiatura di tutte le colonne dell'iterazione j-1 nell'iterazione
    % j dei seguenti vettori e matrici

    matrice_interazione(j,:)=matrice_interazione(j-1,:);
    matrice_memoria_interazione(j,:)=matrice_memoria_interazione(j-1,:);
    count_TP(1,j)=count_TP(1,j-1);
    count_FP(1,j)=count_FP(1,j-1);
    count_TN(1,j)=count_TN(1,j-1);
    count_FN(1,j)=count_FN(1,j-1);
    count_TN_nofluo(1,j)=count_TN_nofluo(1,j-1);
    count_FN_nofluo(1,j)=count_FN_nofluo(1,j-1);
    count_TP_successivi(1,j)=count_TP_successivi(1,j-1);
end

close all

% inizializzazione vettori per il fluoroforo

x_fluoroforo_new=[];
y_fluoroforo_new=[];

% codice che serve per contare i fari indici
% TP,TN,FP,FN,TN_nofluo,FN_nofluo,TP_successivi
for indCellimm= 1:NumCellimm
    if matrice_interazione(j,indCellimm)~=0
        if vettore_fluoro_cellimm(1,indCellimm)==1

            p_scambio=rand();

            if p_scambio< p_scambio_fluoroforo_dendimm

                vettore_fluoro_cellimm(1,indCellimm)=0;
                r=matrice_interazione(j,indCellimm);
                if vettore_scambio_fluoro(1,r)==1
                    count_TP_successivi(1,j)=count_TP_successivi(1,j)+1;
                end
                vettore_scambio_fluoro(1,r)=1;
                fluo_passato(1,indCellimm)=1;
                count_TP(1,j)=count_TP(1,j)+1;
            end
        end
    end
end

```

```

probabilita=rand();

if probabilita<p_distacco
    if fluo_passato(1,indCellimm)==1
        matrice_interazione(j,indCellimm)=0;
        fluo_passato(1,indCellimm)=0;

    else
        if vettore_fluoro_cellimm(1,indCellimm)==1
            count_FN(1,j)=count_FN(1,j)+1;
            matrice_interazione(j,indCellimm)=0;
        else
            count_FN_nofluo(1,j)=count_FN_nofluo(1,j)+1;
            matrice_interazione(j,indCellimm)=0;
        end
    end
end
else
    for indCelldend=1:NumCellDendr

distanza_euclidea=norma(pos_x_imm(j,indCellimm),pos_x_dendr(1,indCelldend),pos_y
_imm(j,indCellimm),pos_y_dendr(1,indCelldend));

        if distanza_euclidea < threshold &&
matrice_interazione(j,indCellimm)==0
            prob=rand();
            if prob<p_interact
                matrice_interazione(j,indCellimm)=indCelldend;

matrice_memoria_interazione(j,indCelldend)=matrice_memoria_interazione(j,indCell
dend)+1;

            end
            if vettore_fluoro_cellimm(1,indCellimm)==1
                p_scambio=rand();

                if p_scambio< p_scambio_fluoroforo_dendimm

                    vettore_fluoro_cellimm(1,indCellimm)=0;

                    if matrice_interazione(j,indCellimm)==0
                        count_FP(1,j)=count_FP(1,j)+1;
                    else
                        fluo_passato(1,indCellimm)=1;
                        count_TP(1,j)=count_TP(1,j)+1;
                        if vettore_scambio_fluoro(1,indCelldend)==1
                            count_TP_successivi(1,j)=count_TP_succes
sivi(1,j)+1;
                        end
                    end
                    vettore_scambio_fluoro(1,indCelldend)=1;
                else
                    if matrice_interazione(j,indCellimm)==0
                        count_TN(1,j)=count_TN(1,j)+1;
                    end
                end
            else
                if matrice_interazione(j,indCellimm)==0
                    count_TN_nofluo(1,j)=count_TN_nofluo(1,j)+1;
                end
            end
        end
    end
end

```

```

end
end
end
end
end

end

%spostamento delle cellule sistema immunitario

[pos_x_imm(j+1,:),
pos_y_imm(j+1,:)] = step_cell_immun(pos_x_imm(j,:),pos_y_imm(j,:),p_step,matrice_i
nterazione,step_length_cellimm,j);

%creazione nuovi vettori del fluoroforo
for k=1:NumCellimm
    for s=1:length(x_fluoroforo)

distanza_fluoro_cell=norma(x_fluoroforo(1,s),pos_x_imm(j,k),y_fluoroforo(1,s),po
s_y_imm(j,k));
        probabilita_fluo=rand();
        if distanza_fluoro_cell < threshold_fluoroforo_cellimm &&
probabilita_fluo < p_fluoroforo
            vettore_fluoro_cellimm(1,k)=1;
            x_fluoroforo(1,s)=NaN;
            y_fluoroforo(1,s)=NaN;
        end
    end
end
end

for f=1:length(x_fluoroforo)
    if ~isnan(x_fluoroforo(1,f))
        z=x_fluoroforo(1,f);
        x_fluoroforo_new=[x_fluoroforo_new z];
    end
    if ~isnan(y_fluoroforo(1,f))
        p=y_fluoroforo(1,f);
        y_fluoroforo_new= [y_fluoroforo_new p];
    end
end
x_fluoroforo=x_fluoroforo_new;

y_fluoroforo=y_fluoroforo_new;

%visualizzazione cellule immunitarie

for indicecellimm=1:NumCellimm

    r=matrice_interazione(j,indicecellimm);
    if vettore_fluoro_cellimm(1,indicecellimm)==1
        if r~=0

I=plot(pos_x_imm(j,indicecellimm),pos_y_imm(j,indicecellimm),'go');
            hold on
            plot([pos_x_imm(j,indicecellimm)
pos_x_dendr(1,r)], [pos_y_imm(j,indicecellimm) pos_y_dendr(1,r)], '-b')
            sz = 10;

scatter(pos_x_imm(j,indicecellimm),pos_y_imm(j,indicecellimm),sz,'y','filled');
        else

```

```

I=plot(pos_x_imm(j,indicecellimm),pos_y_imm(j,indicecellimm),'bo');
    hold on
    sz = 10;

scatter(pos_x_imm(j,indicecellimm),pos_y_imm(j,indicecellimm),sz,'y','filled');
    end
    else
        if r~=0

I=plot(pos_x_imm(j,indicecellimm),pos_y_imm(j,indicecellimm),'go');
    hold on
        plot([pos_x_imm(j,indicecellimm)
pos_x_dendr(1,r)], [pos_y_imm(j,indicecellimm) pos_y_dendr(1,r)], '-b')

        else

I=plot(pos_x_imm(j,indicecellimm),pos_y_imm(j,indicecellimm),'bo');
    hold on

        end

    end
end

%visualizzazione fluoroforo sparso

sz = 30;
scatter(x_fluoroforo,y_fluoroforo,sz,'y','filled');

% visualizzazione cellule dendritiche

for indicecellden=1:NumCellDendr
    r=find(matrice_interazione(j,:)==indicecellden);
    if vettore_scambio_fluoro(1,indicecellden)==1
        if matrice_memoria_interazione(j,indicecellden)~=0

plot(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),'kpentagram','Lin
eWidth',2);
            sz = 10;

scatter(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),sz,'y','filled
');
                else
                    if ~isempty(r)

plot(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),'gs','LineWidth',
1,'MarkerSize',10,'MarkerEdgeColor','g','MarkerFaceColor',[0.5,0.5,0.5]);
                        sz = 10;

scatter(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),sz,'y','filled
');
                            else

plot(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),'gs','LineWidth',
1,'MarkerSize',10,'MarkerEdgeColor','r','MarkerFaceColor',[0.5,0.5,0.5]);
                                sz = 10;

```



```

scatter(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),sz,'y','filled
');
        end
    end

    else
        if matrice_memoria_interazione(j,indicecellden)~=0

plot(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),'kpentagram','Lin
eWidth',2);
        else
            if ~isempty(r)

plot(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),'gs','LineWidth',
1,'MarkerSize',10,'MarkerEdgeColor','g','MarkerFaceColor',[0.5,0.5,0.5]);
            else

plot(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),'gs','LineWidth',
1,'MarkerSize',10,'MarkerEdgeColor','r','MarkerFaceColor',[0.5,0.5,0.5]);
            end
        end
    end

end

%visualizzazione regione di interazione
for i=1:NumCellDendr

    x_cerchio=pos_x_dendr(1,i)-threshold:0.01:pos_x_dendr(1,i)+threshold;
    raggio=threshold;
    for indice=1:length(x_cerchio)
        y_cerchio_2(indice)=-sqrt(threshold.^2-(x_cerchio(indice)-
pos_x_dendr(1,i)).^2)+pos_y_dendr(1,i);
        y_cerchio(indice)=sqrt(threshold.^2-(x_cerchio(indice)-
pos_x_dendr(1,i)).^2)+pos_y_dendr(1,i);
    end
    plot(x_cerchio,y_cerchio,'k--');
    hold on
    plot(x_cerchio,y_cerchio_2,'k--');
end

grid on
grid minor
xlim([-0.1.*x x.*1.1]);
ylim([-0.1.*y y.*1.1]);
title(['cellule dentritiche(quadrati) cellule T sistema immunitario(cerchi)--
iterazione ' num2str(j)]);

%salvataggio delle figure a ogni iterazione del simulatore nel video
file_name=['iterazione'];
saveas(I,file_name,'png');
img=imread(file_name,'png');
writeVideo(outputVideo,img);
end
close(outputVideo);
% visualizzazione del video
imshow('cellule_in_movimento.avi');

```

CODICE DEL SIMULATORE CON LE CELLULE IMMUNITARIE DISTRIBUITE IN MODO ORDINATO E IL FLUOROFORO SPARSO

```
close all
clear all

%% inizializzazione parametri generali del simulatore

NumCellDendr=10; %cellule bersaglio
NumCellimm=10; %cellule sistema immunitario

%dimensioni reticolo bidimensionale

x=10;
y=10;

%iterazioni simulatore

T=20;

% lunghezza spostamento delle cellule immunitarie

step_length_cellimm=0.5;

%info sul video

outputVideo=VideoWriter('cellule_in_movimento.avi');
outputVideo.FrameRate=2;
open(outputVideo);

%soglia usata per l'interazione tra una cellula immunitaria e una dendritica

threshold=1;

% soglia usata per il caricamento del fluoroforo in una cellula immunitaria

threshold_fluoroforo_cellimm=0.5;

% varie probabilità

p_scambio_fluoroforo_dendimm=0.6; % probabilità che una cellula immunitaria
scambi il fluoroforo con cellula dendritica
p_interact=1; % probabilità che due cellule interagiscono
p_step=1; % probabilità che una cellula immunitaria si muova
p_distacco=0.2; % probabilità che una cellula immunitaria si stacchi dalla
cellula dendritica
p_fluoroforo=0.8; %probabilità che una cellula immunitaria senza fluoroforo si
carichi di fluoroforo

%% inizializzazione fluoroforo

x_fluoroforo=x*rand(1,30);
y_fluoroforo=y*rand(1,30);

%% inizializzazione matrice e vettore di interazione

matrice_interazione=zeros(T,NumCellimm);
```

```

vettore_interazione=zeros(1,NumCellimm);
matrice_memoria_interazione=zeros(T,NumCellDendr);

%% inizializzazione vettore del scambio fluoroforo

vettore_scambio_fluoro=zeros(1,NumCellDendr);

%% inizializzazione vettore fluoroforo

vettore_fluoro_cellimm=zeros(1,NumCellimm);

%% vettori per contare TP,FP,TN,FN,TN_nofluo,FN_nofluo,TP_successivi

count_TP=zeros(1,T);
count_FN=zeros(1,T);
count_TN=zeros(1,T);
count_FP=zeros(1,T);
count_TN_nofluo=zeros(1,T);
count_FN_nofluo=zeros(1,T);
count_TP_successivi=zeros(1,T);

%% inizializzazione vettore fluo_passato

fluo_passato=zeros(1,NumCellimm);

%% inizializzazione random cellule bersaglio e cellule sistema immunitario

%distribuzione random delle cellule bersaglio

[pos_x_dendr, pos_y_dendr]=inizializza_cell_dendr_ord(NumCellDendr);

%distribuzione random delle cellule del sistema immunitario
j=1;
[s_random, t_random]=inizializza_cel_immun_ord(NumCellimm);
[pos_x_imm
,pos_y_imm]=step_cell_immun_ord(s_random,t_random,p_step,matrice_interazione,ste
p_length_cellimm,j);

%% visualizzazione delle cellule bersaglio e cellule immunitarie in una figura

for j=1:T

    if j==1
        I=plot(s_random,t_random,'bo');
        hold on

plot(pos_x_dendr,pos_y_dendr,'gs','LineWidth',1,'MarkerSize',10,'MarkerEdgeColor
','r','MarkerFaceColor',[0.5,0.5,0.5]);
        sz = 30;
        scatter(x_fluoroforo,y_fluoroforo,sz,'y','filled');
        plot([1 9],[5 5], '-.', 'color',0.2*[1 1 1])
        plot([0 1],[0 5], '-.', 'color',0.2*[1 1 1])
        plot([0 1],[10 5], '-.', 'color',0.2*[1 1 1])
        plot([10 9],[0 5], '-.', 'color',0.2*[1 1 1])
        plot([9 10],[5 10], '-.', 'color',0.2*[1 1 1])
        for i=1:NumCellDendr

```

```

        x_cerchio=pos_x_dendr(1,i)-threshold:0.01:pos_x_dendr(1,i)+threshold;
        raggio=threshold;
        for indice=1:length(x_cerchio)
            y_cerchio_2(indice)=-sqrt(threshold.^2-(x_cerchio(indice)-
pos_x_dendr(1,i)).^2)+pos_y_dendr(1,i);
            y_cerchio(indice)=sqrt(threshold.^2-(x_cerchio(indice)-
pos_x_dendr(1,i)).^2)+pos_y_dendr(1,i);
        end
        plot(x_cerchio,y_cerchio,'k--');
        hold on
        plot(x_cerchio,y_cerchio_2,'k--');
        end
        grid on
        grid minor
        xlim([-0.1.*x x.*1.1]);
        ylim([-0.1.*y y.*1.1]);
        title(['cellule dentritiche(quadrati) cellule T sistema
immunitario(cerchi)--iterazione ' num2str(j)]);

        file_name=['iterazione'];
        saveas(I,file_name,'png');
        img=imread(file_name,'png');
        writeVideo(outputVideo,img);
    else
        %copiatura di tutte le colonne dell'iterazione j-1 nell'iterazione j
        %dei seguenti vettori e matrici
        matrice_interazione(j,:)=matrice_interazione(j-1,:);
        matrice_memoria_interazione(j,:)=matrice_memoria_interazione(j-1,:);
        count_TP(1,j)=count_TP(1,j-1);
        count_FP(1,j)=count_FP(1,j-1);
        count_TN(1,j)=count_TN(1,j-1);
        count_FN(1,j)=count_FN(1,j-1);
        count_TN_nofluo(1,j)=count_TN_nofluo(1,j-1);
        count_FN_nofluo(1,j)=count_FN_nofluo(1,j-1);
        count_TP_successivi(1,j)=count_TP_successivi(1,j-1);
    end

    close all
    %inizializzazione per il fluoroforo
    x_fluoroforo_new=[];
    y_fluoroforo_new=[];

% codice per contare i vari indici
% TP,TN,FP,FN,TN_nofluo,FN_nofluo,TP_successivi
for indCellimm= 1:NumCellimm
    if matrice_interazione(j,indCellimm)~=0
        if vettore_fluoro_cellimm(1,indCellimm)==1

            p_sambio=rand();

            if p_sambio< p_sambio_fluoroforo_dendimm

                vettore_fluoro_cellimm(1,indCellimm)=0;
                r=matrice_interazione(j,indCellimm);
                if vettore_sambio_fluoro(1,r)==1
                    count_TP_successivi(1,j)=count_TP_successivi(1,j)+1;
                end
                vettore_sambio_fluoro(1,r)=1;
                fluo_passato(1,indCellimm)=1;
                count_TP(1,j)=count_TP(1,j)+1;
            end
        end
    end
end

```

```

end

probabilita=rand();

if probabilita<p_distacco
    if fluo_passato(1,indCellimm)==1
        matrice_interazione(j,indCellimm)=0;
        fluo_passato(1,indCellimm)=0;

    else
        if vettore_fluoro_cellimm(1,indCellimm)==1
            count_FN(1,j)=count_FN(1,j)+1;
            matrice_interazione(j,indCellimm)=0;
        else
            count_FN_nofluo(1,j)=count_FN_nofluo(1,j)+1;
            matrice_interazione(j,indCellimm)=0;
        end
    end
end
else
    for indCelldend=1:NumCellDendr

distanza_euclidea=norma(pos_x_imm(j,indCellimm),pos_x_dendr(1,indCelldend),pos_y
_imm(j,indCellimm),pos_y_dendr(1,indCelldend));

        if distanza_euclidea < threshold &&
matrice_interazione(j,indCellimm)==0
            prob=rand();
            if prob<p_interact
                matrice_interazione(j,indCellimm)=indCelldend;

matrice_memoria_interazione(j,indCelldend)=matrice_memoria_interazione(j,indCell
dend)+1;

            end
            if vettore_fluoro_cellimm(1,indCellimm)==1
                p_scambio=rand();

                if p_scambio< p_scambio_fluoroforo_dendimm

                    vettore_fluoro_cellimm(1,indCellimm)=0;

                    if matrice_interazione(j,indCellimm)==0
                        count_FP(1,j)=count_FP(1,j)+1;
                    else
                        fluo_passato(1,indCellimm)=1;
                        count_TP(1,j)=count_TP(1,j)+1;
                        if vettore_scambio_fluoro(1,indCelldend)==1

count_TP_successivi(1,j)=count_TP_successivi(1,j)+1;
                            end
                        end
                        vettore_scambio_fluoro(1,indCelldend)=1;
                    else
                        if matrice_interazione(j,indCellimm)==0
                            count_TN(1,j)=count_TN(1,j)+1;
                        end
                    end
                else
                    if matrice_interazione(j,indCellimm)==0

```

```

count_TN_nofluo(1,j)=count_TN_nofluo(1,j)+1;
end
end
end
end
end

end

end

%spostamento delle cellule sistema immunitario

[pos_x_imm(j+1,:),
pos_y_imm(j+1,:)] = step_cell_immun_ord(pos_x_imm(j,:),pos_y_imm(j,:),p_step,matri
ce_interazione,step_length_cellimm,j);

% creazione nuovi vettori del fluoroforo
for k=1:NumCellimm
    for s=1:length(x_fluoroforo)

distanza_fluoro_cell=norma(x_fluoroforo(1,s),pos_x_imm(j,k),y_fluoroforo(1,s),po
s_y_imm(j,k));
        probabilita_fluo=rand();
        if distanza_fluoro_cell < threshold_fluoroforo_cellimm &&
probabilita_fluo < p_fluoroforo
            vettore_fluoro_cellimm(1,k)=1;
            x_fluoroforo(1,s)=NaN;
            y_fluoroforo(1,s)=NaN;
        end
    end
end

for f=1:length(x_fluoroforo)
    if ~isnan(x_fluoroforo(1,f))
        z=x_fluoroforo(1,f);
        x_fluoroforo_new=[x_fluoroforo_new z];
    end
    if ~isnan(y_fluoroforo(1,f))
        p=y_fluoroforo(1,f);
        y_fluoroforo_new= [y_fluoroforo_new p];
    end
end
x_fluoroforo=x_fluoroforo_new;

y_fluoroforo=y_fluoroforo_new;

%visualizzazione cellule immunitarie
for indicecellimm=1:NumCellimm

    r=matrice_interazione(j,indicecellimm);
    if vettore_fluoro_cellimm(1,indicecellimm)==1
        if r~=0

I=plot(pos_x_imm(j,indicecellimm),pos_y_imm(j,indicecellimm),'go');
        hold on
        plot([pos_x_imm(j,indicecellimm)
pos_x_dendr(1,r)], [pos_y_imm(j,indicecellimm) pos_y_dendr(1,r)], '-b')
        sz = 10;

scatter(pos_x_imm(j,indicecellimm),pos_y_imm(j,indicecellimm),sz,'y','filled');
        else

```

```

I=plot(pos_x_imm(j,indicecellimm),pos_y_imm(j,indicecellimm),'bo');
    hold on
    sz = 10;

scatter(pos_x_imm(j,indicecellimm),pos_y_imm(j,indicecellimm),sz,'y','filled');
    end
    else
        if r~=0

I=plot(pos_x_imm(j,indicecellimm),pos_y_imm(j,indicecellimm),'go');
    hold on
        plot([pos_x_imm(j,indicecellimm)
pos_x_dendr(1,r)], [pos_y_imm(j,indicecellimm) pos_y_dendr(1,r)], '-b')

        else

I=plot(pos_x_imm(j,indicecellimm),pos_y_imm(j,indicecellimm),'bo');
    hold on

        end

    end
end

%visualizzazione fluoroforo sparso
sz = 30;
scatter(x_fluoroforo,y_fluoroforo,sz,'y','filled');

%visualizzazione cellule dendritiche
for indicecellden=1:NumCellDendr
    r=find(matrice_interazione(j,:)==indicecellden);
    if vettore_scambio_fluoro(1,indicecellden)==1
        if ~isempty(r)

plot(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),'gs','LineWidth',
1,'MarkerSize',10,'MarkerEdgeColor','g','MarkerFaceColor',[0.5,0.5,0.5]);
        sz = 10;

scatter(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),sz,'y','filled
');

        else

plot(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),'gs','LineWidth',
1,'MarkerSize',10,'MarkerEdgeColor','r','MarkerFaceColor',[0.5,0.5,0.5]);
        sz = 10;

scatter(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),sz,'y','filled
');

        end
    else
        if ~isempty(r)

plot(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),'gs','LineWidth',
1,'MarkerSize',10,'MarkerEdgeColor','g','MarkerFaceColor',[0.5,0.5,0.5]);
        else

plot(pos_x_dendr(1,indicecellden),pos_y_dendr(1,indicecellden),'gs','LineWidth',
1,'MarkerSize',10,'MarkerEdgeColor','r','MarkerFaceColor',[0.5,0.5,0.5]);
        end
    end
end

```

```

        end
    end
%visualizzazione divisione del reticolo
plot([1 9],[5 5], '-.', 'color', 0.2*[1 1 1])
plot([0 1],[0 5], '-.', 'color', 0.2*[1 1 1])
plot([0 1],[10 5], '-.', 'color', 0.2*[1 1 1])
plot([10 9],[0 5], '-.', 'color', 0.2*[1 1 1])
plot([9 10],[5 10], '-.', 'color', 0.2*[1 1 1])

%visualizzazione regione di interazione
for i=1:NumCellDendr

    x_cerchio=pos_x_dendr(1,i)-threshold:0.01:pos_x_dendr(1,i)+threshold;
    raggio=threshold;
    for indice=1:length(x_cerchio)
        y_cerchio_2(indice)=-sqrt(threshold.^2-(x_cerchio(indice)-
pos_x_dendr(1,i)).^2)+pos_y_dendr(1,i);
        y_cerchio(indice)=sqrt(threshold.^2-(x_cerchio(indice)-
pos_x_dendr(1,i)).^2)+pos_y_dendr(1,i);
    end
    plot(x_cerchio,y_cerchio,'k--');
    hold on
    plot(x_cerchio,y_cerchio_2,'k--');
end

grid on
grid minor
xlim([-0.1.*x x.*1.1]);
ylim([-0.1.*y y.*1.1]);
title(['cellule dendritiche(quadrati) cellule T sistema immunitario(cerchi)--
iterazione ' num2str(j)]);

%salvataggio delle figure a ogni iterazione del simulatore nel video
file_name=['iterazione'];
saveas(I,file_name,'png');
img=imread(file_name,'png');
writeVideo(outputVideo,img);
end
close(outputVideo);
% visualizzazione del video
implay('cellule_in_movimento.avi');

```


CODICE PER ANALIZZARE I RISULTATI

```
close all
clc

%% inizializzazione parametri generali del simulatore

NumCellDendr=10; %cellule dendritiche
NumCellimm=10; %cellule sistema immunitario

%dimensioni reticolo bidimensionale

x=10;
y=10;

%iterazioni simulatore

T=1000;
%numero simulazioni

N=100;

% lunghezza spostamento delle cellule immunitarie

step_length_cellimm=0.5;

%soglia

threshold=1;

threshold_fluoroforo_cellimm=0.5;

% varie probabilità

p_scambio_fluoroforo_dendimm=0.1; % probabilità che una cellula immunitaria
scambi il fluoroforo con cellula dendritica
p_interact=0.5; % probabilità che due cellule interagiscono
p_step=0.9; % probabilità che una cellula immunitaria si muova
p_distacco=0.2; % probabilità che una cellula immunitaria si stacchi dalla
cellula dendritica
p_fluoroforo=0.9; %probabilità che una cellula immunitaria senza fluoroforo si
carichi di fluoroforo

matrice_TP=zeros(N,T);
matrice_TN=zeros(N,T);
matrice_FP=zeros(N,T);
matrice_FN=zeros(N,T);
matrice_FN_nofluo=zeros(N,T);
matrice_TN_nofluo=zeros(N,T);
matrice_TP_successivi=zeros(N,T);
media_TP=[];
media_FP=[];
media_TN=[];
media_FN=[];
media_TN_fluo=[];
media_FN_nofluo=[];
matrice_TP_visibili=zeros(N,T);

for i_sim=1:N
%% inizializzazione fluoroforo
```

```

x_fluoroforo=x*rand(1,30);
y_fluoroforo=y*rand(1,30);

%% inizializzazione matrice e vettore di interazione

matrice_interazione=zeros(T,NumCellimm);

matrice_memoria_interazione=zeros(T,NumCellDendr);

%% inizializzazione vettore del scambio fluoroforo

vettore_scambio_fluoro=zeros(1,NumCellDendr);

%% inizializzazione vettore fluoroforo

vettore_fluoro_cellimm=zeros(1,NumCellimm);

%% vettori per contare TP,FP,TN,FN
count_TP=zeros(1,T);
count_FN=zeros(1,T);
count_TN=zeros(1,T);
count_FP=zeros(1,T);
count_TN_nofluo=zeros(1,T);
count_FN_nofluo=zeros(1,T);
count_TP_successivi=zeros(1,T);

fluo_passato=zeros(1,NumCellimm);

%% inizializzazione random cellule bersaglio e cellule sistema immunitario

%distribuzione random delle cellule bersaglio

[pos_x_dendr, pos_y_dendr]=inizializza_cell_dendr_random(x,y,NumCellDendr);

%distribuzione random delle cellule del sistema immunitario
j=1;
[s_random, t_random]=inizializza_cel_immun(x,y,NumCellimm);
[pos_x_imm
, pos_y_imm]=step_cell_immun(s_random,t_random,p_step,matrice_interazione,step_le
ngth_cellimm,j);

%% visualizzazione delle cellule bersaglio e cellule immunitarie in una figura

for j=1:T

    if j==1

    else
        matrice_interazione(j,:)=matrice_interazione(j-1,:);
        matrice_memoria_interazione(j,:)=matrice_memoria_interazione(j-1,:);
        count_TP(1,j)=count_TP(1,j-1);
        count_FP(1,j)=count_FP(1,j-1);
        count_TN(1,j)=count_TN(1,j-1);
    end
end

```

```

count_FN(1,j)=count_FN(1,j-1);
count_TN_nofluo(1,j)=count_TN_nofluo(1,j-1);
count_FN_nofluo(1,j)=count_FN_nofluo(1,j-1);
count_TP_successivi(1,j)=count_TP_successivi(1,j-1);
end

close all

x_fluoroforo_new=[];
y_fluoroforo_new=[];

%verifico se le cellule immunitarie e bersaglio interagiscono
for indCellimm= 1:NumCellimm
    if matrice_interazione(j,indCellimm)~=0
        if vettore_fluoro_cellimm(1,indCellimm)==1

            p_scambio=rand();

            if p_scambio< p_scambio_fluoroforo_dendimm

                vettore_fluoro_cellimm(1,indCellimm)=0;
                r=matrice_interazione(j,indCellimm);
                if vettore_scambio_fluoro(1,r)==1
                    count_TP_successivi(1,j)=count_TP_successivi(1,j)+1;
                end
                vettore_scambio_fluoro(1,r)=1;
                fluo_passato(1,indCellimm)=1;
                count_TP(1,j)=count_TP(1,j)+1;
            end
        end
    end

    probabilita=rand();

    if probabilita<p_distacco
        if fluo_passato(1,indCellimm)==1
            matrice_interazione(j,indCellimm)=0;
            fluo_passato(1,indCellimm)=0;

        else
            if vettore_fluoro_cellimm(1,indCellimm)==1
                count_FN(1,j)=count_FN(1,j)+1;
                matrice_interazione(j,indCellimm)=0;
            else
                count_FN_nofluo(1,j)=count_FN_nofluo(1,j)+1;
                matrice_interazione(j,indCellimm)=0;
            end
        end
    end
end
else
    for indCelldend=1:NumCellDendr

distanza_euclidea=norma(pos_x_imm(j,indCellimm),pos_x_dendr(1,indCelldend),pos_y
_imm(j,indCellimm),pos_y_dendr(1,indCelldend));

        if distanza_euclidea < threshold &&
matrice_interazione(j,indCellimm)==0
            prob=rand();
            if prob<p_interact
                matrice_interazione(j,indCellimm)=indCelldend;
            end
        end
    end
end

```

```

matrice_memoria_interazione(j,indCelledend)=matrice_memoria_interazione(j,indCell
dend)+1;

                end
                if vettore_fluoro_cellimm(1,indCellimm)==1
                    p_sambio=rand();

                    if p_sambio< p_sambio_fluoroforo_dendimm

                        vettore_fluoro_cellimm(1,indCellimm)=0;

                        if matrice_interazione(j,indCellimm)==0
                            count_FP(1,j)=count_FP(1,j)+1;
                        else
                            fluo_passato(1,indCellimm)=1;
                            count_TP(1,j)=count_TP(1,j)+1;
                            if vettore_sambio_fluoro(1,indCelledend)==1
count_TP_successivi(1,j)=count_TP_successivi(1,j)+1;
                                end
                                end
                                vettore_sambio_fluoro(1,indCelledend)=1;
                            else
                                if matrice_interazione(j,indCellimm)==0
                                    count_TN(1,j)=count_TN(1,j)+1;
                                end
                                end
                                else
                                    if matrice_interazione(j,indCellimm)==0
                                        count_TN_nofluo(1,j)=count_TN_nofluo(1,j)+1;
                                    end
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end

```

```
%spostamento delle cellule sistema immunitario
```

```

[pos_x_imm(j+1,:),
pos_y_imm(j+1,:)] = step_cell_immun(pos_x_imm(j,:),pos_y_imm(j,:),p_step,matrice_i
nterazione,step_length_cellimm,j);

for k=1:NumCellimm
    for s=1:length(x_fluoroforo)

distanza_fluoro_cell=norma(x_fluoroforo(1,s),pos_x_imm(j,k),y_fluoroforo(1,s),po
s_y_imm(j,k));
        probabilita_fluo=rand();
        if distanza_fluoro_cell< threshold_fluoroforo_cellimm &&
probabilita_fluo<p_fluoroforo
            vettore_fluoro_cellimm(1,k)=1;
            x_fluoroforo(1,s)=NaN;
            y_fluoroforo(1,s)=NaN;
        end
    end
end

```

```

end

for f=1:length(x_fluoroforo)
    if ~isnan(x_fluoroforo(1,f))
        z=x_fluoroforo(1,f);
        x_fluoroforo_new=[x_fluoroforo_new z];
    end
    if ~isnan(y_fluoroforo(1,f))
        p=y_fluoroforo(1,f);
        y_fluoroforo_new= [y_fluoroforo_new p];
    end
end
x_fluoroforo=x_fluoroforo_new;

y_fluoroforo=y_fluoroforo_new;

end

matrice_TP(i_sim,:)=count_TP;
matrice_TN(i_sim,:)=count_TN;
matrice_FP(i_sim,:)=count_FP;
matrice_FN(i_sim,:)=count_FN;
matrice_FN_nofluo(i_sim,:)=count_FN_nofluo;
matrice_TN_nofluo(i_sim,:)=count_TN_nofluo;
matrice_TP_successivi(i_sim,:)=count_TP_successivi;

end

matrice_TP_visibili=matrice_TP-matrice_TP_successivi;
matrice_FN_camb=matrice_FN+matrice_TP_successivi;
matrice_TN_tot=matrice_TN+matrice_TN_nofluo;
matrice_FN_tot=matrice_FN+matrice_FN_nofluo;
matrice_FN_tot_2=matrice_FN_camb+matrice_FN_nofluo;

media_TP=mean(matrice_TP,1);
media_FP=mean(matrice_FP,1);
media_TN=mean(matrice_TN,1);
media_FN=mean(matrice_FN,1);
media_FN_nofluo=mean(matrice_FN_nofluo,1);
media_TN_nofluo=mean(matrice_TN_nofluo,1);
media_TP_visibili=mean(matrice_TP_visibili,1);
media_FN_tot_2=mean(matrice_FN_tot_2);
media_TN_tot=mean(matrice_TN_tot);
media_FN_tot=mean(matrice_FN_tot);
vettore_iterazioni=1:T;
w=0;
varianza_TP=var(matrice_TP,w,1);
varianza_FP=var(matrice_FP,w,1);
varianza_TN=var(matrice_TN,w,1);
varianza_FN=var(matrice_FN,w,1);
varianza_FN_nofluo=var(matrice_FN_nofluo,w,1);
varianza_TN_nofluo=var(matrice_TN_nofluo,w,1);
varianza_TP_visibili=var(matrice_TP_visibili,w,1);
varianza_FN_tot_2=var(matrice_FN_tot_2,w,1);
varianza_FN_tot=var(matrice_FN_tot,w,1);
varianza_TN_tot=var(matrice_TN_tot,w,1);
s_dev_TP=std(matrice_TP,w,1);
s_dev_FP=std(matrice_FP,w,1);
s_dev_TN=std(matrice_TN,w,1);
s_dev_FN=std(matrice_FN,w,1);
s_dev_FN_nofluo=std(matrice_FN_nofluo,w,1);
s_dev_TN_nofluo=std(matrice_TN_nofluo,w,1);

```

```

s_dev_TP_visibili=std(matrice_TP_visibili,w,1);
s_dev_FN_tot_2=std(matrice_FN_tot_2,w,1);
s_dev_FN_tot=std(matrice_FN_tot,w,1);
s_dev_TN_tot=std(matrice_TN_tot,w,1);

figure
A=plot(vettore_iterazioni,media_TP,'k','LineWidth',2);
hold on
plot(vettore_iterazioni,matrice_TP);
plot(vettore_iterazioni,(media_TP+s_dev_TP),'*r');
plot(vettore_iterazioni,(media_TP-s_dev_TP),'*r');
title('True positive')
xlabel('iterazioni simulatore');
saveas(A,'TP_totali_NOESAURIMENTO.jpeg');
figure
B=plot(vettore_iterazioni,media_FP,'k','LineWidth',2);
hold on
plot(vettore_iterazioni,matrice_FP);
plot(vettore_iterazioni,(media_FP+s_dev_FP),'*r');
plot(vettore_iterazioni,(media_FP-s_dev_FP),'*r');
title('False positive')
xlabel('iterazioni simulatore');
saveas(B,'FP_totali_NOESAURIMENTO.jpeg');
figure
C=plot(vettore_iterazioni,media_TN,'k','LineWidth',2);
hold on
plot(vettore_iterazioni,matrice_TN);
plot(vettore_iterazioni,(media_TN+s_dev_TN),'*r');
plot(vettore_iterazioni,(media_TN-s_dev_TN),'*r');
title('True negative')
xlabel('iterazioni simulatore');
saveas(C,'TN_singoli_NOESAURIMENTO.jpeg');
figure
D=plot(vettore_iterazioni,media_FN,'k','LineWidth',2);
hold on
plot(vettore_iterazioni,matrice_FN);
plot(vettore_iterazioni,(media_FN+s_dev_FN),'*r');
plot(vettore_iterazioni,(media_FN-s_dev_FN),'*r');
title('False Negative')
xlabel('iterazioni simulatore');
saveas(D,'FN_singoli_NOESAURIMENTO.jpeg');
figure
E=plot(vettore_iterazioni,media_TN_nofluo,'k','LineWidth',2);
hold on
plot(vettore_iterazioni,matrice_TN_nofluo);
plot(vettore_iterazioni,(media_TN_nofluo+s_dev_TN_nofluo),'*r');
plot(vettore_iterazioni,(media_TN_nofluo-s_dev_TN_nofluo),'*r');
title('True negative no fluo')
xlabel('iterazioni simulatore');
saveas(E,'TN_nofluo_NOESAURIMENTO.jpeg');
figure
F=plot(vettore_iterazioni,media_FN_nofluo,'k','LineWidth',2);
hold on
plot(vettore_iterazioni,matrice_FN_nofluo);
plot(vettore_iterazioni,(media_FN_nofluo+s_dev_FN_nofluo),'*r');
plot(vettore_iterazioni,(media_FN_nofluo-s_dev_FN_nofluo),'*r');
title('False negative no fluo')
xlabel('iterazioni simulatore');
saveas(F,'FN_nofluo_NOESAURIMENTO.jpeg');
figure
G=plot(vettore_iterazioni,media_TP_visibili,'k','LineWidth',2);
hold on

```

```

plot(vettore_iterazioni,matrice_TP_visibili);
plot(vettore_iterazioni,(media_TP_visibili+s_dev_TP_visibili),'*r');
plot(vettore_iterazioni,(media_TP_visibili-s_dev_TP_visibili),'*r');
title('True positive visibili')
xlabel('iterazioni simulatore');
saveas(G,'TP_visibili_NOESAURIMENTO.jpeg');
figure
H=plot(vettore_iterazioni,media_FN_tot_2,'k','LineWidth',2);
hold on
plot(vettore_iterazioni,matrice_FN_tot_2);
plot(vettore_iterazioni,(media_FN_tot_2+s_dev_FN_tot_2),'*r');
plot(vettore_iterazioni,(media_FN_tot_2-s_dev_FN_tot_2),'*r');
title('False negative con successivi')
xlabel('iterazioni simulatore');
saveas(H,'FN_con_nofluo_TPsuccessivi_NOESAURIMENTO.jpeg');
figure
I=plot(vettore_iterazioni,media_FN_tot,'k','LineWidth',2);
hold on
plot(vettore_iterazioni,matrice_FN_tot);
plot(vettore_iterazioni,(media_FN_tot+s_dev_FN_tot),'*r');
plot(vettore_iterazioni,(media_FN_tot-s_dev_FN_tot),'*r');
title('False negative con no fluo')
xlabel('iterazioni simulatore');
saveas(I,'FN_con no fluo_NOESAURIMENTO.jpeg');
figure
L=plot(vettore_iterazioni,media_TN_tot,'k','LineWidth',2);
hold on
plot(vettore_iterazioni,matrice_TN_tot);
plot(vettore_iterazioni,(media_TN_tot+s_dev_TN_tot),'*r');
plot(vettore_iterazioni,(media_TN_tot-s_dev_TN_tot),'*r');
title('True negative con no fluo ')
xlabel('iterazioni simulatore');
saveas(L,'TN_connofluo_NOESAURIMENTO.jpeg');

figure
subplot(4,2,1)
plot(vettore_iterazioni,media_TP,'k','LineWidth',2);
hold on
plot(vettore_iterazioni,matrice_TP);
plot(vettore_iterazioni,(media_TP+s_dev_TP),'*r');
plot(vettore_iterazioni,(media_TP-s_dev_TP),'*r');
title('True positive')
xlabel('iterazioni simulatore');
xticks([0:50:T])
subplot(4,2,2)
plot(vettore_iterazioni,media_TP_visibili,'k','LineWidth',2);
hold on
plot(vettore_iterazioni,matrice_TP_visibili);
plot(vettore_iterazioni,(media_TP_visibili+s_dev_TP_visibili),'*r');
plot(vettore_iterazioni,(media_TP_visibili-s_dev_TP_visibili),'*r');
title('True positive visibili')
xlabel('iterazioni simulatore');
xticks([0:50:T])
subplot(4,2,3)
plot(vettore_iterazioni,media_FP,'k','LineWidth',2);
hold on
plot(vettore_iterazioni,matrice_FP);
plot(vettore_iterazioni,(media_FP+s_dev_FP),'*r');
plot(vettore_iterazioni,(media_FP-s_dev_FP),'*r');
title('False positive')
xlabel('iterazioni simulatore');
xticks([0:50:T])
subplot(4,2,5)

```

```

plot(vettore_iterazioni,media_TN,'k','LineWidth',2);
hold on
plot(vettore_iterazioni,matrice_TN);
plot(vettore_iterazioni,(media_TN+s_dev_TN),'*r');
plot(vettore_iterazioni,(media_TN-s_dev_TN),'*r');
title('True negative')
xlabel('iterazioni simulatore');
xticks([0:50:T])
subplot(4,2,7)
plot(vettore_iterazioni,media_FN,'k','LineWidth',2);
hold on
plot(vettore_iterazioni,matrice_FN);
plot(vettore_iterazioni,(media_FN+s_dev_FN),'*r');
plot(vettore_iterazioni,(media_FN-s_dev_FN),'*r');
title('False Negative')
xlabel('iterazioni simulatore');
xticks([0:50:T])
subplot(4,2,8)
plot(vettore_iterazioni,media_FN_tot_2,'k','LineWidth',2);
hold on
plot(vettore_iterazioni,matrice_FN_tot_2);
plot(vettore_iterazioni,(media_FN_tot_2+s_dev_FN_tot),'*r');
plot(vettore_iterazioni,(media_FN_tot_2-s_dev_FN_tot),'*r');
title('False negative con visibili')
xlabel('iterazione simulatore');
xticks([0:50:T]);

sensitivity= matrice_TP./(matrice_TP+matrice_FN_tot);
specificity=matrice_TN_tot./(matrice_TN_tot+matrice_FP);
precision=matrice_TP./(matrice_TP+matrice_FP);
false_positive_rate=matrice_FP./(matrice_FP+matrice_TN_tot);
accurancy=(matrice_TP+matrice_TN_tot)./(matrice_TP+matrice_TN_tot+matrice_FP+matrice_FN_tot);
F1=(2.*matrice_TP)./(2.*matrice_TP+matrice_FP+matrice_FN_tot);
NPV=matrice_TN_tot./(matrice_TN_tot+matrice_FN_tot);
FNR=matrice_FN_tot./(matrice_FN_tot+matrice_TP);

sensitivity_visibili= matrice_TP_visibili./(matrice_TP+matrice_FN_tot_2);
precision_visibili=matrice_TP_visibili./(matrice_TP_visibili+matrice_FP);
accurancy_visibili=(matrice_TP_visibili+matrice_TN_tot)./(matrice_TP_visibili+matrice_TN_tot+matrice_FP+matrice_FN_tot_2);
F1_visibili=(2.*matrice_TP_visibili)./(2.*matrice_TP_visibili+matrice_FP+matrice_FN_tot_2);
FNR_visibili=matrice_FN_tot_2./(matrice_FN_tot_2+matrice_TP_visibili);

figure

M=boxplot(sensitivity);
title('sensitivity');
xticks([0 1000])
xticklabels({'0','1000'})
xlabel('iterazioni simulatore')
ylim([min(sensitivity,[],'all')*-0.1 max(sensitivity,[],'all')*1.1]);

figure
N=boxplot(specificity);
title('specificity');
xticks([0 1000])
xticklabels({'0','1000'})
xlabel('iterazioni simulatore')

```



```

ylim([min(specificity, [], 'all')*-0.1 max(specificity, [], 'all')*1.1]);

figure
O=boxplot(precision);
title('precision');
xticks([0 1000])
xticklabels({'0', '1000'})
xlabel('iterazioni simulatore')
ylim([min(precision, [], 'all')*-0.1 max(precision, [], 'all')*1.1]);

figure
P=boxplot(false_positive_rate);
title('false positive rate');
xticks([0 1000])
xticklabels({'0', '1000'})
xlabel('iterazioni simulatore')
ylim([min(false_positive_rate, [], 'all')*-0.1
max(false_positive_rate, [], 'all')*1.1]);

figure
Q=boxplot(accuracy);
title('accuracy');
xticks([0 1000])
xticklabels({'0', '1000'})
xlabel('iterazioni simulatore')
ylim([min(accuracy, [], 'all')*-0.1 max(accuracy, [], 'all')*1.1]);

figure
R=boxplot(F1);
title('F1');
xticks([0 1000])
xticklabels({'0', '1000'})
xlabel('iterazioni simulatore')
ylim([min(F1, [], 'all')*-0.1 max(F1, [], 'all')*1.1]);

figure
S=boxplot(NPV);
title('negative predictive value');
xticks([0 1000])
xticklabels({'0', '1000'})
xlabel('iterazioni simulatore')
ylim([min(NPV, [], 'all')*-0.1 max(NPV, [], 'all')*1.1]);

figure
T=boxplot(FNR);
title('false negative rate');
xticks([0 1000])
xticklabels({'0', '1000'})
xlabel('iterazioni simulatore')
ylim([min(FNR, [], 'all')*-0.1 max(FNR, [], 'all')*1.1]);

figure
U=boxplot(sensitivity_visibili);
title('sensitivity visibili');
xticks([0 1000])
xticklabels({'0', '1000'})
xlabel('iterazioni simulatore')
ylim([min(sensitivity, [], 'all')*-0.1 max(sensitivity, [], 'all')*1.1]);

```

```

figure
V=boxplot(precision_visibili);
title('precision visibili');
xticks([0 1000])
xticklabels({'0','1000'})
xlabel('iterazioni simulatore')
ylim([min(precision_visibili,[],'all')*-0.1
max(precision_visibili,[],'all')*1.1]);

```

```

figure
W=boxplot(accuracy_visibili);
title('accuracy visibili');
xticks([0 1000])
xticklabels({'0','1000'})
xlabel('iterazioni simulatore')
ylim([min(accuracy_visibili,[],'all')*-0.1
max(accuracy_visibili,[],'all')*1.1]);

```

```

figure
X=boxplot(F1_visibili);
title('F1 visibili');
xticks([0 1000])
xticklabels({'0','1000'})
xlabel('iterazioni simulatore')
ylim([min(F1_visibili,[],'all')*-0.1 max(F1_visibili,[],'all')*1.1]);

```

```

figure
Y=boxplot(FNR_visibili);
title('false negative rate visibili');
xticks([0 1000])
xticklabels({'0','1000'})
xlabel('iterazioni simulatore')
ylim([min(FNR_visibili,[],'all')*-0.1 max(FNR_visibili,[],'all')*1.1]);

```