Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Evaluation of Machine Learning Algorithms for the Discovery of Tumor Treatments

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΒΑΣΙΛΕΙΟΥ Π. ΣΤΑΥΡΟΠΟΥΛΟΥ

Επιβλέποντες:	Χαρίλαος Ακασιάδης	Συνεργάτης Έρευνας Ε.Κ.Ε.Φ.Ε. "Δημόκριτος"
	Αλέξανδρος Αρτίκης	Αναπλ. Καθηγητής Πα.Πει., Ερευνητής Ε.Κ.Ε.Φ.Ε. "Δημόκριτος"
	Γεώργιος Στάμου	Αναπλ. Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2021

Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Evaluation of Machine Learning Algorithms for the Discovery of Tumor Treatments

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΒΑΣΙΛΕΙΟΥ Π. ΣΤΑΥΡΟΠΟΥΛΟΥ

Επιβλέποντες:	Χαρίλαος Ακασιάδης	Συνεργάτης Έρευνας Ε.Κ.Ε.Φ.Ε. "Δημόκριτος"
	Αλέξανδρος Αρτίκης	Αναπλ. Καθηγητής Πα.Πει., Ερευνητής Ε.Κ.Ε.Φ.Ε. "Δημόκριτος"
	Γεώργιος Στάμου	Αναπλ. Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 5η Οκτωβρίου 2021.

.....	.....	.....
Γεώργιος Στάμου	Ανδρέας Γεώργιος Σταφυλοπάτης	Στέφανος Κόλλιας
Αναπλ. Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

(Υπογραφή)

.....
Βασίλειος Π. Σταυρόπουλος
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η τεχνητή νοημοσύνη χρησιμοποιείται συχνά για να ερμηνεύσει και να αναλύσει μεγάλους όγκους περίπλοκων δεδομένων από διάφορα γνωστικά πεδία. Σε ένα από αυτά, την υπολογιστική βιολογία, η εξερεύνηση τυχόν αποτελεσματικών φαρμακευτικών συνθέσεων για θεραπείες του καρκίνου μέσω προσομοιώσεων απαιτεί πολύ χρόνο και την παράλληλη χρήση πληθώρας υπολογιστικών πόρων για να εκπληρωθεί. Στην παρούσα διπλωματική, εφαρμόζουμε μια μοντέρνα μέθοδο ενεργητικής μάθησης για τον χαρακτηρισμό ενός νέου χώρου καρκινικών θεραπειών, που περιέχει υποσχόμενες θεραπείες για τον περιορισμό καρκινικών κυττάρων, κάνοντας χρήση ενός ανασχεδιασμένου προσομοιωτή για in silico πειραματισμούς. Επιπλέον, εξετάζουμε την εφαρμογή διάφορων μεθόδων συσταδοποίησης και βελτιστοποίησης και συγκρίνουμε την επίδοση τους σε πειραματικές δοκιμές σε υπερ-υπολογιστικό περιβάλλον. Ο βασικός στόχος είναι ο χαρακτηρισμός περιοχών του χώρου θεραπειών σε αυτές που περιέχουν αποτελεσματικές θεραπείες, και σε αυτές που δεν περιέχουν ενδιαφέρουσες περιπτώσεις, ώστε να καθοδηγηθεί η σχετική έρευνα σε πιο στοχευμένα και αποτελεσματικά πειράματα σε ασθενείς. Τα πειραματικά αποτελέσματα αποδεικνύουν ότι η μέθοδος επιτυγχάνει έναν αρκετά ποιοτικό χαρακτηρισμό του χώρου θεραπειών. Επιπλέον, γίνεται αντιληπτό από τα αποτελέσματα ότι η εφαρμογή διαφορετικών μεθόδων συσταδοποίησης και βελτιστοποίησης στην μέθοδο επηρεάζει τον αριθμό των απαιτούμενων προσομοιώσεων και την ποιότητα του χαρακτηρισμού του χώρου θεραπειών.

Λέξεις Κλειδιά: Ενεργητική Μάθηση, Γενετικοί Αλγόριθμοι, Αναζήτηση με προσομοιωμένη ανόπτηση, Προσομοιώσεις καρκινικών κυττάρων, Υπολογιστική βιολογία

Abstract

Machine learning is regularly used to interpret and analyze information from large and complex datasets originating from numerous fields. In one of those fields, namely Bioinformatics, the exploration of potentially beneficial drug configurations for tumor treatments via simulations requires multiple processing units to be used in parallel and a considerable amount of time to be completed. In this thesis, we apply a state-of-the-art model exploration active learning workflow for the characterization of a new drug configuration parameter space, using a redesigned simulator for *in silico* experiments. Moreover, we incorporate different clustering and optimization approaches in the original workflow and compare their performance in simulation trials on high-performance computing infrastructure. The overall goal is to divide the parameter space into regions that contain effective and ineffective treatments, and thus guide the related research towards more focused and effective real-world trials. Experimental results demonstrate that the workflow achieves a fine characterization of the treatment parameter space. Moreover, results indicate that the incorporation of different clustering and optimization algorithms in the workflow affects the quality of the treatment space characterization and the number of required simulations.

Keywords: Active Learning, Genetic Algorithms, Simulated Annealing, Tumor Simulations, Computational Biology

Ευχαριστίες

Θα ήθελα, εν πρώτοις, να ευχαριστήσω τους επιβλέποντες μου από το Ε.Κ.Ε.Φ.Ε. "Δημόκριτος", τον Δρ. Χαρίλαο Ακασιάδη και τον Δρ. Αλέξανδρο Αρτίκη, η καθοδήγηση των οποίων έπαιξε καίριο ρόλο στην ολοκλήρωση της παρούσας διπλωματικής. Οι εβδομαδιαίες συναντήσεις μας και οι συμβουλές τους με βοήθησαν να αντιμετωπίσω τα εμπόδια που συναντούσα αλλά και με ενέπνευσαν να εμπλουτίσω τις γνώσεις μου γύρω από το γνωστικό αντικείμενο της Τεχνητής Νοημοσύνης. Επίσης, θα ήθελα να ευχαριστήσω τον υποψήφιο διδάκτορα Ευάγγελο Μιχελιουδάκη, οι προτάσεις του οποίου ήταν πολύτιμες για τον προσανατολισμό της διπλωματικής.

Θέλω, επίσης, να ευχαριστήσω τον καθηγητή κ. Γιώργο Στάμου της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του ΕΜΠ για την εμπιστοσύνη που μου επέδειξε για την εκπόνηση της παρούσας διπλωματικής. Οι διαλέξεις του ιδίου αλλά και των υπολοίπων μελών του εργαστηρίου Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης αποτέλεσαν εφαλτήριο για την ενασχόλησή μου με τον συναρπαστικό τομέα της Τεχνητής Νοημοσύνης.

Επιπλέον, θα ήθελα να ευχαριστήσω τους φίλους μου, συνάδελφους και μη, για τις αξέχαστες αναμνήσεις που μου προσέφεραν.

Κλείνοντας, θα ήθελα να ευχαριστήσω τους γονείς μου και τον αδερφό μου για την αμέριστη στήριξη που μου έδειξαν καθόλη την διάρκεια των σπουδών μου, και όχι μόνο.

Contents

List of Figures

List of Tables

Εκτενής Περίληψη

Εισαγωγή

Κίνητρο

Ο καρκίνος αποτελεί μια από τις συχνότερες αιτίες θανάτου παγκοσμίως, καθώς αποτελεί υπεύθυνο για περίπου 10 εκατομμύρια από αυτούς σύμφωνα με τον Παγκόσμιο Οργανισμό Υγείας. [1] Η ανακάλυψη αποτελεσματικών θεραπειών ενάντια στον καρκίνο είναι στόχος ύψιστης προτεραιότητας για την ιατρική ερευνητική κοινότητα. Ωστόσο, η απαίτηση μεγάλων χρηματοδοτήσεων και η χρονοβόρα διαδικασία μελέτης και επαλήθευσης νέων θεραπειών αποτελούν σημαντικά εμπόδια στην επίτευξη του αναφερθέντος στόχου γρήγορα και αποτελεσματικά. Επιπλέον, η δυναμική και απρόβλεπτη φύση της αρρώστιας δεν επιτρέπει, σε αρκετές περιπτώσεις, την διενέργεια διεξοδικών κλινικών ερευνών. Η υπολογιστική βιολογία έχει συνεισφέρει στην ανακάλυψη νέων θεραπειών [1, 2] παρέχοντας μοντέλα τα οποία έχουν ως στόχο την περιγραφή της συμπεριφοράς των καρκινικών κυττάρων εντός του ανθρώπινου οργανισμού. Τέτοιου είδους μοντέλα επιτρέπουν στους ερευνητές την εξέταση της επίδοσης των θεραπειών χωρίς να είναι απαραίτητη η διενέργεια κλινικών πειραμάτων, στα οποία ενδεχομένως να τεθεί η ζωή των συμμετεχόντων σε κίνδυνο. Επιπλέον, η διενέργεια πειραμάτων μέσω των μοντέλων αυτών επιτρέπει στους ερευνητές να εξετάσουν μεγαλύτερο όγκο πιθανών θεραπειών και να προχωρήσουν πιο στοχευμένα στις κλινικές δοκιμές, μειώνοντας έτσι αρκετά τον αριθμό των θεραπειών που αποτυγχάνουν στα τελικά στάδια των δοκιμών.

Ωστόσο, η διενέργεια πειραμάτων μέσω τέτοιων μοντέλων είναι, συνήθως, μια χρονοβόρα διαδικασία, η οποία απαιτεί πληθώρα υπολογιστικών πόρων. Επομένως, η πολυπλοκότητα των μοντέλων καθιστά πολύ δύσκολη- ως και αδύνατη- την εξέταση όλων των δυνατών θεραπειών εξαντλητικά. Η χρήση μεθόδων της Μηχανικής Μάθησης, ωστόσο, μπορεί να οδηγήσει σε στοχευμένη μείωση των απαιτούμενων προσομοιώσεων, αλλά και να προσφέρει ένα ασφαλές μέσο πρόβλεψης της επίδοσης των θεραπειών. Συγκεκριμένα, οι Ozik κ.ά. [3]

[1] `https://www.who.int/news-room/fact-sheets/detail/cancer`

παρουσιάσαν μια μέθοδο που έχει ως στόχο την ανακάλυψη των χαρακτηριστικών ελπιδοφόρων θεραπειών, αναγνωρίζοντας τα εύρη των παραμέτρων των θεραπειών που αναμένεται να έχουν τα προσδοκώμενα αποτελέσματα. Η μέθοδος, στην οποία θα αναφερόμαστε εφεξής ως 'πρωτότυπη μέθοδος', αποτελείται από δύο μέρη. Σε αμφότερα τα μέρη δειγματοληπτούνται και αξιολογούνται υποσύνολα του χώρου παραμέτρων των θεραπειών. Το πρώτο μέρος αποτελείται από έναν αλγόριθμο Ενεργητικής Μάθησης [4] και στοχεύει στην αναγνώριση των περιοχών του χώρου παραμέτρων των θεραπειών οι οποίες περιέχουν ελπιδοφόρες θεραπείες. Στο πρώτο μέρος, δείγματα σημείων του χώρου που εξετάζεται χρησιμοποιούνται για να εκπαιδευτεί ένας ταξινομητής. Ο ταξινομήτης χαρακτηρίζει τα σημεία για τα οποία έχει την μεγαλύτερη αβεβαιότητα, τα οποία χρήζουν αξιολόγησης, προτού εκπαιδευτεί εκ΄ νέου. Ωστόσο, ο αριθμός των αβέβαιων σημείων είναι αρκετά μεγάλος, γεγονός που δυσχεραίνει την αξιολόγηση τους. Για να ξεπεραστεί αυτό, τα σημεία συσταδοποιούνται βάσει των χωρικών χαρακτηριστικών τους και μόνο μερικά αντιπροσωπευτικά σημεία επιλέγονται τελικά για να αξιολογηθούν μέσω προσομειώσεων. Στο δεύτερο μέρος γίνεται χρήση ενός Γενετικού Αλγόριθμου για την ανακάλυψη της σύνθεσης των πιο αποτελεσματικών θεραπειών. Συγκεκριμένα, ο στόχος του Γενετικού Αλγόριθμου είναι η εύρεση της πιο αποτελεσματικής θεραπείας. Η κατάσταση εκκίνησης του Γενετικού Αλγορίθμου είναι είτε τυχαία είτε λαμβάνει υπόψιν τα αποτελέσματα του πρώτου μέρους. Στο δεύτερο σενάριο, ο αρχικός πληθυσμός του Γενετικού Αλγόριθμου συμπεριλαμβάνει θεραπείες που έχουν αξιολογηθεί ως οι πιο αποτελεσματικές από τον αλγόριθμο Ενεργητικής Μάθησης. Η παραπάνω μέθοδος αποτελεί ένα πολύτιμο όπλο στην φαρέτρα των ερευνητών, αφού τους επιτρέπει να διενεργούν στοχευμένα και αποτελεσματικότερα κλινικά πειράματα.

Συνεισφορά

Στην παρούσα διπλωματική εφαρμόζουμε την πρωτότυπη μέθοδο σε ένα διαφορετικό πειραματικό περιβάλλον. Συγκεκριμένα, εξετάζουμε την επίδραση της πρωτεΐνης Tumor Necrosis Factor (TNF) [5]. Η πρωτεΐνη TNF παίζει σημαντικό ρόλο στην μετάδοση σημάτων μεταξύ των κυττάρων και μπορεί να οδηγήσει στον θάνατο των καρκινικών. Επιπλέον, χρησιμοποιούμε το PhysiBoSSv2 [2] για την διενέργεια των πειραμάτων, το οποίο αποτελεί μια επέκταση του προσομοιωτή PhysiCell [6], που χρησιμοποιήθηκε στη πρωτότυπη μέθοδο. Το PhysiBoSSv2 περιέχει ένα μοντέλο πολλαπλών κλιμάκων βασισμένο σε πράκτορες για την προσομοίωση της ανάπτυξης των καρκινικών κυττάρων. Στην μέθοδο, το συγκεκριμένο μοντέλο χρησιμοποιείται για να μελετηθεί η επίδραση που

[2]https://github.com/bsc-life/PhysiBoSSv2

έχουν οι συνθέσεις των εξεταζόμενων θεραπειών στην διάδοση των καρκινικών κυττάρων. Αμφότερα τα μέρη της μεθόδου δειγματοληπτούν υποσύνολα των δυνατών θεραπευτικών συνθέσεων, οι οποίες κωδικοποιούνται ως τιμές σε συγκεκριμένες παράμετρους, με στόχο την αξιολόγησή τους, μέσω του προαναφερθέντος μοντέλου. Η αξιολόγηση των θεραπειών πραγματοποιείται σε υπερ-υπολογιστικό περιβάλλον.

Η συσταδοποίηση των αβέβαιων σημείων, στο πρώτο κομμάτι της προτεινόμενης μεθόδου, επηρεάζει σημαντικά τον αριθμό των απαιτούμενων προσομοιώσεων, αλλά παίζει και καίριο ρόλο στην ποιότητα των τελικών αποτελεσμάτων. Επομένως, διαφορετικοί αλγόριθμοι συσταδοποίησης ενδέχεται να οδηγούν σε αρκετά διαφορετικά αποτελέσματα. Επιπλέον, σε αρκετές περιπτώσεις μπορεί ο προσδιορισμός των απαιτούμενων υπερ-παραμέτρων των εκάστοτε αλγορίθμων συσταδοποίησης να είναι αρκετά δύσκολος. Στην πρωτότυπη μέθοδο χρησιμοποιείται ο αλγόριθμος K-Means για την συσταδοποίηση των πιο αβέβαιων σημείων. Ωστόσο, ο K-Means αποτυγχάνει στην αναγνώριση συστάδων με αυθαίρετα σχήματα και στην απομόνωση του θορύβου. Επιπλέον, αδυνατεί στον προσδιορισμό του αριθμού των συστάδων βάσει της κατανομής των σημείων του συνόλου δεδομένων. Οι παραπάνω περιορισμοί οδηγούν σε παραμόρφωση των τελικών συστάδων, αφού θορυβώδη σημεία λαμβάνουν μέρος στην διαδικασία και μεγαλύτερες συστάδες κατακερματίζονται σε μικρότερες (ή αντίστροφα) για να αναγνωριστεί ο ορισμένος αριθμός συστάδων. Αυτό έχει ως αποτέλεσμα να εκτελούνται σε αρκετές περιπτώσεις επιπλέον ή και λιγότερες από τις απαιτούμενες προσομοιώσεις ή να προσομοιώνονται θεραπείες που δεν έχουν την μέγιστη πληροφοριακή αξία. Στα πλαίσια της παρούσας διπλωματικής, εξετάζουμε την εφαρμογή διαφορετικών μεθόδων συσταδοποίησης, συγκεκριμένα των BIRCH και DBSCAN, στο στάδιο της συσταδοποίησης της δειγματοληπτικής διαδικασίας και συγκρίνουμε την επίδρασή τους στην ποιότητα του τελικού χαρακτηρισμού του χώρου παραμέτρων αλλά και στον αριθμό των απαιτούμενων προσομοιώσεων.

Ακόμη, μελετάμε την επίδοση μιας επιπλέον μεθόδου βελτιστοποίησης, της Αναζήτησης με προσομοιωμένη ανόπτηση, στο δεύτερο μέρος της προτεινόμενης μεθόδου που αφορά την ανακάλυψη των πιο αποτελεσματικών θεραπειών. Η διατήρηση πολλαπλών πιθανών λύσεων σε κάθε γενιά από τον Γενετικό Αλγόριθμο απαιτεί πληθώρα υπολογιστικών πόρων. Στο σενάριο στο οποίο ο αρχικός πληθυσμός του Γενετικού Αλγόριθμου περιέχει γνώση από το πρώτο μέρος της μεθόδου, ο αρχικός χώρος αναζήτησης περιορίζεται σε περιοχές με αποτελεσματικές θεραπείες. Σε έναν περιορισμένο χώρο αναζήτησης η Αναζήτηση με προσομοιωμένη ανόπτηση μπορεί να οδηγήσει σε παρόμοια αποτελέσματα με εκείνα του Γενετικού Αλγόριθμου, αφού και οι δύο μέθοδοι επιδεικνύουν καλές επιδόσεις στην βελτίωση ενός σετ παραμέτρων γύρω από μια αρχική λύση [7]. Ωστόσο, η Αναζήτηση με προσομοιωμένη ανόπτηση απαιτεί λιγότερους υπο-

λογιστικούς πόρους, αφού εστιάζει στην βελτίωση μιας μοναδικής λύσης, ενώ ο Γενετικός Αλγόριθμος στην εξέλιξη ενός πλήθους πιθανών λύσεων. Στην παρούσα διπλωματική, εξετάζουμε την επίδοση των δύο μεθόδων ως προς την αποτελεσματικότητα των ανακαλυφθεισών θεραπειών και ως προς τον απαιτούμενο αριθμό προσομοιώσεων.

Τα πειράματα που διενεργήθηκαν υποδεικνύουν ότι υπάρχει αισθητή διαφορά στις επιδόσεις των διάφορων εξεταζόμενων μεθόδων. Επιπλέον, παρατηρούμε ότι υπάρχει ένα ισοζύγιο μεταξύ του αριθμού των απαιτούμενων προσομοιώσεων και της ποιότητας των παραγόμενων αποτελεσμάτων, τόσο στον χαρακτηρισμό των περιοχών που περιέχουν υποσχόμενες θεραπείες, όσο και στην ανακάλυψη των πιο αποτελεσματικών θεραπειών.

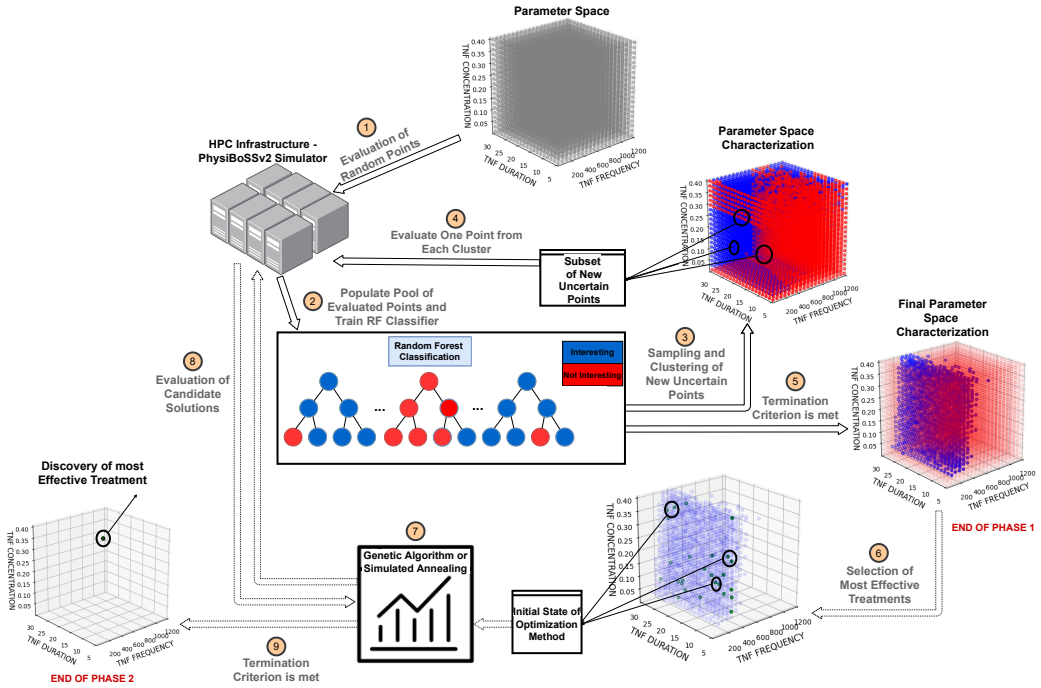
Μέθοδος για την in silico εξερεύνηση θεραπειών ενάντια στον καρκίνο.

Επισκόπηση

Η μέθοδος που εξετάζουμε αποτελείται από δύο μέρη. Το πρώτο μέρος αποτελείται από έναν αλγόριθμο Ενεργητικής Μάθησης που αφορά τον διαχωρισμό του χώρου των παραμέτρων των συνθέσεων των θεραπειών σε περιοχές που εμπεριέχουν υποσχόμενες και μη θεραπείες, ενώ το δεύτερο μέρος σκοπεύει μέσω της εφαρμογής μιας μεθόδους βελτιστοποίησης στον προσδιορισμό των πιο αποτελεσματικών θεραπειών. Το Σχήμα 1 παρέχει μια συνολική επισκόπηση της εξεταζόμενης μεθόδου.

Αρχικά, ένας ταξινομητής Random Forest χρησιμοποιείται ώστε να κατατάξει τις διάφορες περιοχές του χώρου σε ενδιαφέρουσες και μη. Μια περιοχή χαρακτηρίζεται ως ενδιαφέρουσα αν θεωρείται ότι περιέχει αποδοτικές θεραπείες, δηλαδή θεραπείες που μειώνουν τον αριθμό των καρκινικών κυττάρων κάτω από ένα ορισμένο όριο. Ο τελικός στόχος του πρώτου μέρους είναι να κατατάξουμε τις υπο-περιοχές του χώρου παραμέτρων, χωρίς να χρειαστεί να αξιολογήσουμε όλες τις δυνατές θεραπείες. Ο αλγόριθμος εκτελείται κατά επανάληψη, έως ότου εκτελεστεί ένας προκαθορισμένος αριθμός επαναλήψεων. Σε κάθε επανάληψη επιλέγονται θεραπείες προς αξιολόγηση βάσει μιας δειγματοληπτικής διαδικασίας αποτελούμενη από δύο στάδια. Στο πρώτο στάδιο, επιλέγονται τα πιο αβέβαια σημεία. Τα συγκεκριμένα σημεία χαρακτηρίζουν τις περιοχές για τον χαρακτηρισμό των οποίων ο ταξινομητής έχει την μεγαλύτερη αβεβαιότητα. Ένα παράδειγμα του συνόλου των επιλεγμένων αβέβαιων σημείων σε κάποια επανάληψη του αλγορίθμου απεικονίζονται στο Σχήμα 2

Στο δεύτερο στάδιο της δειγματοληψίας, τα αβέβαια σημεία ταξινομούνται σε συστάδες, όπως φαίνεται στο Σχήμα 3, βάσει της χωρικής κατανομής τους



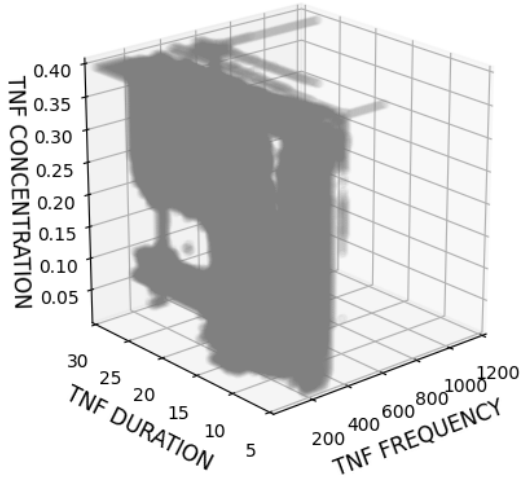
Σχήμα 1: Επισκόπηση της εξεταζόμενης μεθόδου

και αντιπροσωπευτικά σημεία επιλέγονται από κάθε μία.

Το πλήθος των αντιπροσωπευτικών σημείων είναι αρκετά μικρότερο από εκείνο των αβέβαιων σημείων, όπως φαίνεται στο Σχήμα 4.

Τυπικά, τα σημεία που επιλέγονται από κάθε συστάδα παρουσιάζουν μεγάλη ομοιότητα με τα υπόλοιπα σημεία της συστάδας. Έτσι, η αξιολόγηση τους μας προσφέρει σημαντική πληροφορία όχι μόνο για αυτά καθεαυτά, αλλά και για τα υπόλοιπα σημεία της. Επιπλέον, η επιλογή σημείων από διαφορετικές συστάδες εγγυάται την επιλογή σημείων που ορίζουν ανόμοιες θεραπείες.

Αφού λάβει χώρα η αξιολόγηση μέσω προσομοιώσεων των επιλεγμένων θεραπειών, ο ταξινομητής εκπαιδεύεται εκ νέου, προσθέτοντας στα εκπαιδευτικά δεδομένα τις νέες αξιολογήσεις. Ο ταξινομητής, πλέον, μας παρέχει έναν νέο, πιο λεπτομερή χαρακτηρισμό του χώρου παραμέτρων. Η παραπάνω περιγραφείσα διαδικασία επαναλαμβάνεται για ορισμένο πλήθος επαναλήψεων. Όπως είναι ευνόητο, η δειγματοληπτική διαδικασία καθορίζει τον αριθμό των απαιτούμενων προσομοιώσεων. Επομένως, η επιλογή των σημείων με την μέγιστη πληροφοριακή αξία αποτελεί καίριο βήμα στην μείωση των κοστοβόρων προσομοιώσεων. Μέσω της συσταδοποίησης των αβέβαιων σημείων και της επιλογής αντιπροσωπευτικών σημείων πραγματοποιούμε στοχευμένες προσομοιώσεις, των οποίων τα αποτελέσματα προσφέρουν πολύτιμη πληροφορία για ένα μεγάλο πλήθος

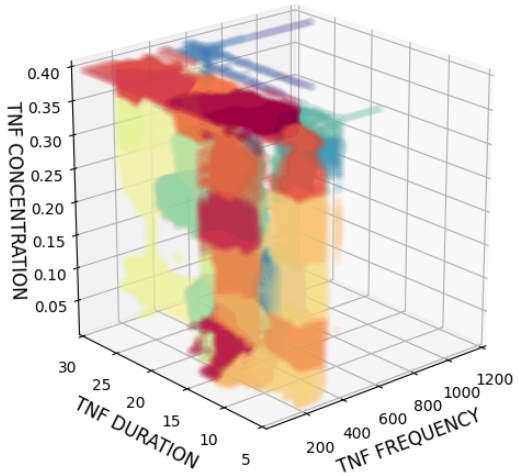


Σχήμα 2: Αβέβαια Σημεία

σημείων της συστάδας. Με αυτό τον τρόπο, μπορούμε με μικρότερο αριθμό προσομοιώσεων να αποκτήσουμε πληροφορία για μεγαλύτερο πλήθος σημείων. Μετά το πέρας των επαναλήψεων του αλγορίθμου, ο χώρος είναι χωρισμένος σε χώρους με υποσχόμενες και μη θεραπείες, όπως φαίνεται στο Σχήμα 5, στο οποίο η περιοχή που έχει χρωματιστεί με μπλε αποτελεί την περιοχή που θεωρείται από τον ταξινομητή ως η περιοχή των υποσχόμενων θεραπειών. Τα μη ενδιαφέροντα σημεία είναι όσα δεν χρωματίζονται στον χώρο του γραφήματος.

Στο δεύτερο μέρος της μεθόδου, στόχος είναι η ανακάλυψη των πιο αποτελεσματικών θεραπειών. Για να επιτευχθεί ο στόχος γίνεται χρήση μιας μεθόδου βελτιστοποίησης για να αποκτήσουμε την θεραπεία που οδηγεί μετά την εφαρμογή της στον μικρότερο αριθμό επιζώντων καρκινικών κύτταρων. Συγκεκριμένα, εξετάζουμε την εφαρμογή δύο μεθόδων βελτιστοποίησης που ονομάζονται Αναζήτηση με προσομοιωμένη ανόπτηση και Γενετικός Αλγόριθμος.

Όσον αφορά τον Γενετικό Αλγόριθμο, τα καλύτερα σημεία κάθε πληθυσμού επιλέγονται βάσει tournament selection. Τα επιλεγμένα σημεία διασταυρώνονται βάσει μιας ορισμένης πιθανότητας διασταύρωσης προτού υποστούν τυχαία μετάλλαξη. Ο πληθυσμός σημείων που προκύπτει από τα παραπάνω βήματα αξιολογείται βάσει μιας συνάρτησης καταλληλότητας. Στην περίπτωσή μας, η συνάρτηση καταλληλότητας που χρησιμοποιείται είναι ο αριθμός των ζωντανών καρκινικών κυττάρων στο τέλος της θεραπείας. Η παραπάνω διαδικασία επαναλαμβάνεται για ορισμένο αριθμό επαναλήψεων. Στην περίπτωση της Αναζήτησης με προσομοιωμένη ανόπτηση, ο αλγόριθμος προσπαθεί να βελτιώσει την αρχική λύση εξετάζοντας ως νέες πιθανές λύσεις θεραπείες που βρίσκονται στην γειτονιά της λύσης. Εξετάζονται δύο σενάρια σχετικά με τον αρχικό πληθυ-



Σχήμα 3: Συσταδοποίηση των πιο αβέβαιων σημείων

σμό των μεθόδων βελτιστοποίησης. Στο πρώτο σενάριο, ο αρχικός πληθυσμός αποτελείται από τυχαίες θεραπείες. Στο δεύτερο σενάριο μέρος του αρχικού πληθυσμού του Γενετικού Αλγόριθμου αποτελείται από θεραπείες που έχουν χαρακτηριστεί ως οι πιο αποτελεσματικές από το πρώτο μέρος της μεθόδου, ενώ ως σημείο εκκίνησης της Αναζήτησης με προσομοιωμένη ανόπτηση τίθενται η θεραπεία που αξιολογήθηκε ως η πιο αποτελεσματική από τον αλγόριθμο Ενεργητικής Μάθησης.

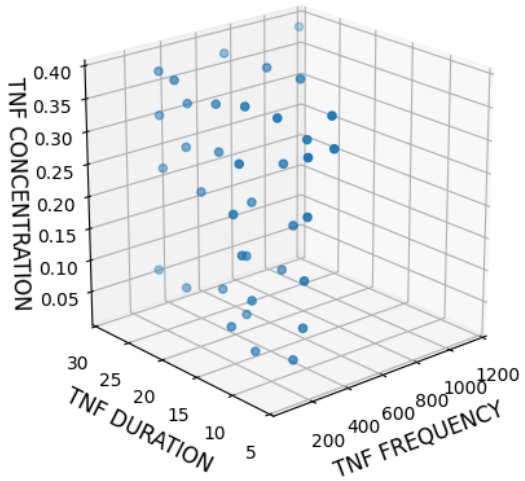
Ο ρόλος του δεύτερου μέρους είναι διττός, καθώς στοχεύει στην ανακάλυψη των πιο αποτελεσματικών θεραπειών και λειτουργεί και ως ένα μέσο επαλήθευσης των αποτελεσμάτων του πρώτου μέρους, αφού οι πιο αποτελεσματικές θεραπείες θα πρέπει να περιέχονται στις περιοχές που χαρακτήρισε ως ενδιαφέρουσες ο ταξινομητής του πρώτου μέρους.

Πειραματικά Αποτελέσματα

Ο κώδικας της υλοποίησής μας αλλά και οδηγίες για την αναπαραγωγή των πειραμάτων βρίσκονται σε ένα online repository.[3] Τα πειράματα έλαβαν χώρα κάνοντας χρήση του Mare Nostrum 4 (MN4) HPC infrastructure του Barcelona Supercomputing Centre.[4] Ο προσδιορισμός των υπερ-παραμέτρων των μεθόδων έγινε μελετώντας την επίδοση τους για διάφορες τιμές των υπερπαραμέτρων σε πειράματα μικρότερης κλίμακας.

[3] `https://github.com/xarakas/spheroid-tnf-v2-emews`

[4] `https://www.bsc.es/marenostrum/marenostrum`

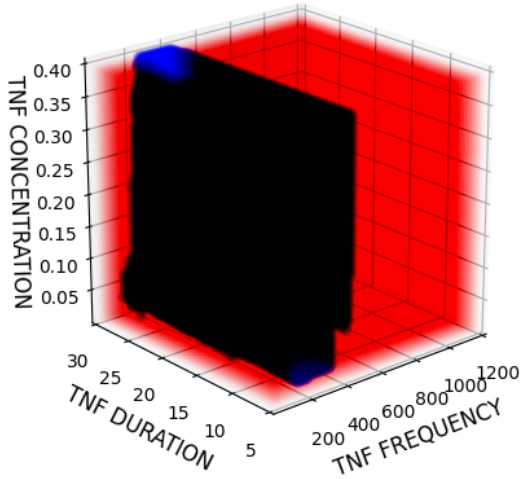


Σχήμα 4: Αντιπροσωπευτικά σημεία προς αξιολόγηση

Αξιολόγηση του χαρακτηρισμού του χώρου παραμέτρων

Στα πειράματά μας εξετάζουμε την επίδοση διάφορων εκδοχών της πρώτης φάσης της μεθόδου μας. Οι εκδοχές τις μεθόδου διαφοροποιούνται ως προς τον αλγόριθμο συσταδοποίησης που χρησιμοποιήθηκε κατά το στάδιο της δειγματοληψίας του αλγορίθμου ενισχυτικής μάθησης. Στην πρωτότυπη μέθοδο εφαρμόστηκε ο αλγόριθμος συσταδοποίησης K-Means. Για να αποκτήσουμε ένα επίπεδο αναφοράς, εξετάσαμε την επίδοση του αλγόριθμου K-Means για K ίσο με 20 και 50. Θα αναφερόμαστε στις παραπάνω μεθόδους ως KMEANS_20 και KMEANS_50, αντίστοιχα. Επιπλέον, εφαρμόζουμε τον αλγόριθμο K-Means με K ίσο με 500 (KMEANS_500) ως μια εξαντλητική μέθοδο δειγματοληψίας. Ο αλγόριθμος DBSCAN ρυθμίστηκε ως $Eps = 0.025$ και $MinPts = 20$, ενώ ο αλγόριθμος BIRCH με branching factor ίσο με $B = 100$ και distance threshold ίσο με $T = 0.1$. Επιπλέον, εκτελέσαμε και μια εξαντλητική αναζήτηση του χώρου παραμέτρων, την οποία αποκαλούμε sweep search. Το sweep search αποτελείται από μια επαναληπτική αξιολόγηση θεραπειών που ανήκουν σε ένα πλέγμα θεραπειών ομοιόμορφης κατανομής και λειτουργεί ως benchmark.

Για να μπορέσουμε να διαχωρίσουμε τις θεραπείες σε αποτελεσματικές και μη θα πρέπει να ορίσουμε, πρώτα, μια μετρική η οποία να υπολογίζει την αποτελεσματικότητα μιας θεραπείας. Ορίζουμε ως βαθμό επιζώντων καρκινικών κυττάρων μιας θεραπείας τον λόγο του πλήθους των επιζώντων καρκινικών κυττάρων μετά την θεραπεία ως προς τον αριθμό των ζωντανών κυττάρων πριν την θεραπεία, δηλαδή ως:

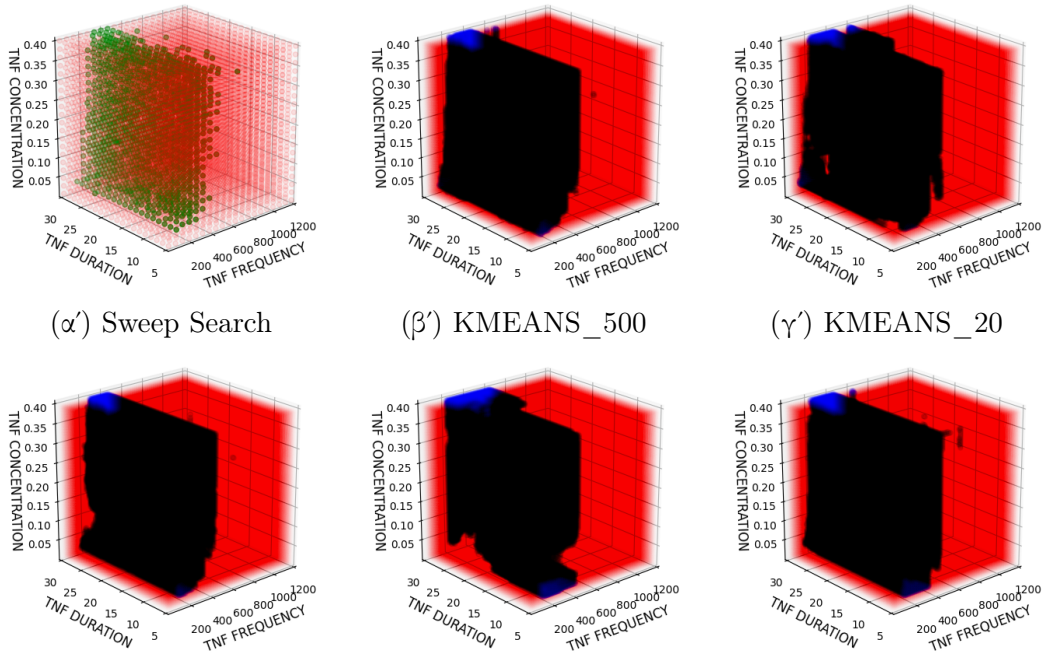


Σχήμα 5: Χαρακτηρισμός υποσχόμενων περιοχών από τον ταξινομητή Random Forest.

$$\begin{matrix}\text{Βαθμός Επιζώντων} \\ \text{Καρκινικών Κυττάρων}\end{matrix} = \frac{\text{Πλήθος τελικών καρκινικών κυττάρων}}{\text{Πλήθος αρχικών καρκινικών κυττάρων}}$$

Στα πειράματά μας, θεωρούμε ως αποτελεσματική μια θεραπεία που έχει βαθμό επιζώντων καρκινικών κυττάρων μικρότερο του 0.3, δηλαδή που ο αριθμός των τελικών καρκινικών κυττάρων είναι μικρότερος ή ίσος με το 30% των αρχικών καρκινικών κυττάρων. Για κάθε πείραμα, ο αλγόριθμος Ενεργητικής Μάθησης έτρεξε για 20 επαναλήψεις.

Για να αξιολογήσουμε τα αποτελέσματα της πρώτης φάσης συγκρίνουμε τα αποτελέσματα της μεθόδου μας με εκείνα του benchmark Sweep Search. Συγκεκριμένα, η μελέτη της γραφικής αναπαράστασης του χαρακτηρισμού του χώρου παραμέτρων μάς παρέχει ένα ποιοτικό μέσο για την αξιολόγηση της επίδοσης της πρώτης φάσης της μεθόδου. Τα αποτελέσματα της μεθόδου για τις διάφορες εξεταζόμενες εκδοχές της προέκυψαν από πολλαπλά πειράματα με διαφορετικά random seeds. Το γεγονός αυτό μας επιτρέπει να συμπεριλάβουμε στα αποτελέσματα την τυχαιότητα της διαδικασίας, αλλά και να εξετάσουμε τις εκδοχές της μεθόδου κάτω από τις ίδιες συνθήκες. Στα πειράματα κάθε seed το αρχικό σύνολο δεδομένων για την αρχική εκπαίδευση του ταξινομητή ήταν ίδιο για όλες τις μεθόδους. Επιπλέον, εξετάζουμε το πλήθος των αβέβαιων σημείων του ταξινομητή και το συνολικό αριθμό των απαιτούμενων προσομοιώσεων.



(α′) Sweep Search (β′) KMEANS_500 (γ′) KMEANS_20

(δ′) KMEANS_50 (ε′) DBSCAN (ϛ′) BIRCH

Σχήμα 6: Τελικός χαρακτηρισμός του χώρου παραμέτρων. Το υποσχήμα (α) απεικονίζει τα αποτελέσματα του benchmark sweep search. Τα υποσχήματα (β),(γ),(δ),(ε),(ζ) απεικονίζουν τα αποτελέσματα των εξεταζόμενων εκδοχών της μεθόδου.

Χαρακτηρισμός του χώρου παραμέτρων

Το Σχήμα 6 παρέχει μια γραφική απεικόνιση του χαρακτηρισμού του χώρου παραμέτρων από τον ταξινομητή για τις διάφορες εξεταζόμενες εκδοχές της μεθόδου. Οι περιοχές που έχουν χρωματιστεί με πράσινο και μπλε έχουν χαρακτηριστεί ως περιοχές αποτελεσματικών θεραπειών από το benchmark Sweep Search και από τον ταξινομητή της πρώτης φάσης της μεθόδου, αντίστοιχα. Με κόκκινο έχουν χρωματιστεί οι περιοχές μη-αποτελεσματικών θεραπειών. Παρατηρούμε ότι όλες οι εκδοχές της μεθόδου καταφέρνουν να εντοπίσουν την γενικότερη περιοχή των αποτελεσματικών θεραπειών. Παρατηρούμε ότι σε όλες τις περιπτώσεις η μέθοδός μας οδηγεί στον χαρακτηρισμό μιας ενδιαφέρουσας περιοχής παρόμοιας και άμεσα συγκρίσιμης με εκείνη του Sweep Search. Η μόνη περίπτωση στην οποία η μέθοδός μας επιδεικνύει μειωμένη απόδοση είναι όταν ο αλγόριθμος ενισχυτικής μάθησης χρησιμοποιεί τον αλγόριθμο DBSCAN για την συσταδοποίηση των αβέβαιων σημείων, στην οποία περίπτωση οδηγούμαστε στην αναγνώριση μιας μειωμένης εκδοχής της ενδιαφέρουσας περιοχής.

Επιπλέον, κάνοντας χρήση των αλγορίθμων BIRCH και KMEANS_500, η μέθοδος οδηγεί στον πιο λεπτομερή χαρακτηρισμό της ενδιαφέρουσας περιοχής. Επιπρόσθετα, η πρώτη φάση της μεθόδου που χρησιμοποιεί τους αλγορίθμους και KMEANS_20 και KMEANS_50 αντιλαμβάνεται τα σύνορα της ενδιαφέρουσας περιοχής σε αρκετά μεγάλο βαθμό, ωστόσο αποτυγχάνει στο να εντοπίσει περιοχές με αποτελεσματικές μεθόδους που δεν ανήκουν στο κύριο σώμα που εντοπίζεται.

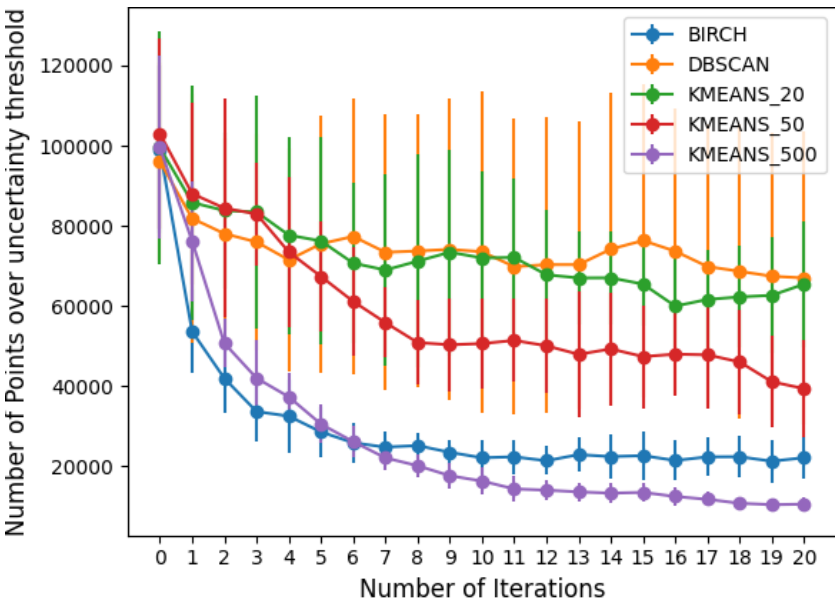
Πλήθος αβέβαιων σημείων

Ο αριθμός των αβέβαιων σημείων του ταξινομήτη αποτελεί ένα πολύτιμο μέσο αξιολόγησης της επίδοσης του αλγόριθμου, αφού μας παρέχει μια εικόνα για την σιγουριά του ταξινομητή για το τελικό αποτέλεσμα. Έστω, το σύνολο όλων των πιθανών θεραπευτικών συνθέσεων X, οι κλάσεις $\{0, 1\}$, που αντιπροσωπεύουν τις μη-αποτελεσματικές και αποτελεσματικές θεραπείες, αντίστοιχα, και η πιθανότητα $Pr(i, x)$ με την οποία θεωρεί ο ταξινομητής ότι η θεραπεία x ανήκει στην κλάση i. Τότε ο αριθμός των αβέβαιων σημείων υπολογίζεται ως:

$$N_U = \left|\left\{x \in X \mid \min(Pr(0, x), Pr(1, x)) \geq \text{κατώφλι αβεβαιότητας}\right\}\right|$$

δηλαδή ως το πλήθος των σημείων των οποίων η αβεβαιότητα είναι πάνω από ένα προκαθορισμένο κατώφλι αβεβαιότητας. Στα πειράματά μας, θέτουμε το κατώφλι αβεβαιότητας ίσο με 0.4. Το πλήθος των αβέβαιων σημείων υπολογίζεται στην αρχή κάθε επανάληψης του αλγορίθμου. Έτσι, μπορούμε να υπολογίσουμε την επίδραση της αξιολόγησης των επιλεγμένων σημείων στην ποιότητα του χαρακτηρισμού του χώρου παραμέτρων.

Το Σχήμα 7 απεικονίζει το πλήθος των αβέβαιων σημείων κατά την διάρκεια των πειραμάτων. Παρατηρούμε ότι οι εκδοχές της μεθόδου που χρησιμοποιούν τους KMEANS_500 και BIRCH στην δειγματοληψία οδηγούν στο μικρότερο αριθμό αβέβαιων σημείων. Συγκεκριμένα, οι εκδοχές που χρησιμοποιούν τους BIRCH και KMEANS_500 επιτυγχάνουν αισθητά καλύτερες επιδόσεις από τις υπόλοιπες εκδοχές. Επιπλέον, οι δύο εκδοχές τις μεθόδου επιδεικνύουν και την πιο σταθερή επίδοση στα διάφορα πειράματα που έλαβαν χώρα και δεν παρουσιάζουν μεγάλες διακυμάνσεις. Όπως παρατηρούμε από το Σχήμα 7, οι εκδοχές του αλγόριθμου ενισχυτικής μάθησης που χρησιμοποιούν τους DBSCAN και KMEANS_20 οδηγούν σε παρόμοιο πλήθος αβέβαιων σημείων στο τέλος του αλγόριθμου, με την εφαρμογή του DBSCAN, ωστόσο, να οδηγεί σε μεγάλες διακυμάνσεις. Τέλος, η πρώτη φάση της μεθόδου που χρησιμοποεί τον KMEANS_50 παρουσιάζει καλύτερη επίδοση από τις εκδοχές που χρησιμοποιούν τους DBSCAN και KMEANS_20, αλλά χειρότερη από εκείνες που

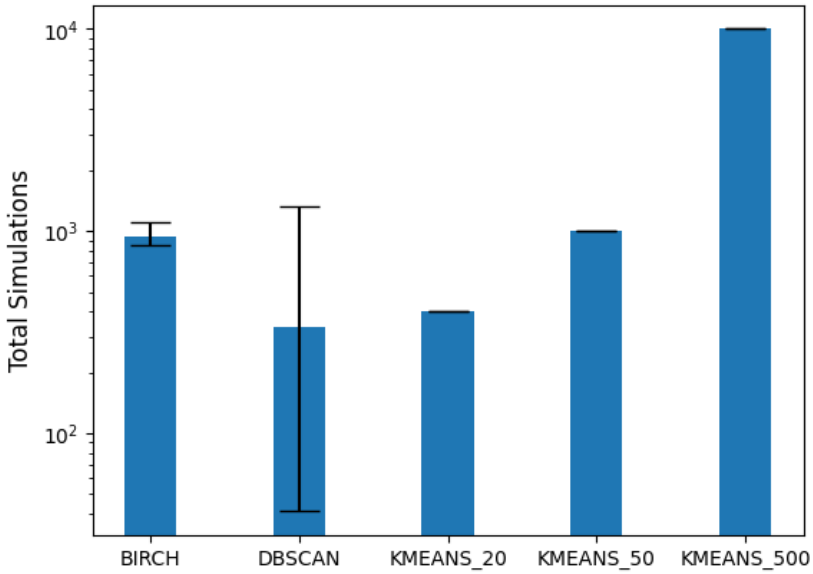


Σχήμα 7: Πλήθος αβέβαιων σημείων ανά επανάληψη.

εφαρμόζουν τους BIRCH και KMEANS_500. Είναι αξιοσημείωτο ότι ο αριθμός των αβέβαιων σημείων μειώνεται απότομα στις πρώτες επαναλήψεις, ενώ μειώνεται πιο συντηρητικά κατά τις τελευταίες. Αυτό συμβαίνει καθώς ο ταξινομήτης οριοθετεί την περιοχή των αποτελεσματικών θεραπειών στα αρχικά βήματα του αλγόριθμου και εστιάζει στις λεπτομέρειες του χώρου στα τελευταία. Επιπλέον, αξίζει να σημειώσουμε ότι ο αριθμός των αβέβαιων σημείων δεν αγγίζει ποτέ το μηδέν, ακόμη και στην περίπτωση της εξαντλητικής εκδοχής της μεθόδου που χρησιμοποιεί τον KMEANS_500 στο στάδιο της δειγματοληψίας. Αυτό οφείλεται στην στοχαστικότητα των βιολογικών μοντέλων που χρησιμοποιούνται για να προσομοιώσουν την συμπεριφορά των ανθρωπίνων κυττάρων. Λόγω της στοχαστικότητας των μοντέλων, η αποτελεσματικότητα των θεραπειών ενδέχεται να διαφέρει ελαφρώς μεταξύ προσομοιώσεων. Επομένως, για τις θεραπείες των οποίων η αποτελεσματικότητα είναι αρκετά κοντά στο κατώφλι που έχουμε ορίσει, η αβεβαιότητα δεν θα μειωθεί ποτέ κάτω από το κατώφλι αβεβαιότητας.

Αριθμός απαιτούμενων προσομοιώσεων

Οι προσομοιώσεις που διενεργούνται για να αποφανθούμε για την αποτελεσματικότητα των θεραπειών είναι αρκετά χρονοβόρες και απαιτούν πληθώρα υπολογιστικών πόρων. Επομένως, ο αριθμός των συνολικών απαιτούμενων προσομοιώσεων αποτελεί μια καλή μετρική για την αποδοτικότητα της πρώτης



Σχήμα 8: Συνολικός αριθμός απαιτούμενων προσομοιώσεων (ο άξονας y είναι σε λογαριθμική κλίμακα.

φάσης της μεθόδου εξερεύνησης θεραπειών, όσον αφορά τόσο τους υπολογιστικούς πόρους, όσο και τον απαιτούμενο χρόνο. Παρατηρούμε από το Σχήμα 8 ότι η εκδοχή του αλγορίθμου ενισχυτικής μάθησης που χρησιμοποιεί τον BIRCH απαιτεί αισθητά λιγότερες προσομοιώσεις από εκείνη που χρησιμοποιεί τον KMEANS_500, αν και η μέθοδος μας οδηγεί σε παρόμοια αποτελέσματα στις συγκεκριμένες περιπτώσεις, όπως είδαμε προηγουμένως. Επιπλέον, παρατηρούμε ότι οι εκδοχές που εφαρμόζουν τους BIRCH και KMEANS_50 για την συσταδοποίηση των αβέβαιων σημείων απαιτούν παρόμοιο αριθμό προσομοιώσεων. Ωστόσο, η επίδοση τους, σύμφωνα με τα Σχήματα 6 7, διαφέρει αισθητά. Αυτή η διαφορά οφείλεται στο γεγονός ότι ο αλγόριθμος συσταδοποίησης BIRCH απομονώνει τα θορυβώδη σημεία σε συστάδες 'θορύβου', ενώ στον KMEANS_50 ο θόρυβος παίζει ενεργό ρόλο στον καθορισμό των συστάδων. Επομένως, εφαρμόζοντας τον αλγόριθμο BIRCH, καταφέρνει η μέθοδός μας να αντλήσει πληροφορία τόσο από συστάδες χωρίς θόρυβο όσο και από τα απομακρυσμένα σημεία των αβέβαιων σημείων. Επιπλέον, παρατηρούμε ότι όταν η μέθοδός μας χρησιμοποιεί τον DBSCAN απαιτεί τις λιγότερες προσομοιώσεις, ωστόσο και σε αυτή την περίπτωση δεν επιδεικνύει σταθερότητα. Αυτή η αστάθεια οφείλεται στο γεγονός ότι στις περισσότερες περιπτώσεις τα αβέβαια σημεία ορίζουν μια επιφάνεια που δεν χωρίζεται σε περιοχές με υψηλή και χαμηλή πυκνότητα, επομένως ο DBSCAN καθορίζει μεγάλες συστάδες που αποτελούν μεγάλες επιφάνειες με μεγάλη πυκνότητα σημείων. Επιπλέον, η α-

στάθεια οφείλεται και στην μεγάλη ευαισθησία που έχει ο DBSCAN στις τιμές των υπερ-παραμέτρων του.

Αξιολόγηση αναζήτησης αποτελεσματικότερης θεραπείας

Στην πρωτότυπη μέθοδο εφαρμόζεται ένας Γενετικός Αλγόριθμος για την ανακάλυψη των πιο αποτελεσματικών θεραπειών, όπως αναφέραμε στην Ενότητα . Ο Γενετικός Αλγόριθμος επαναλαμβάνεται για 30 γενιές, με πληθυσμό ίσο με 50, με tournament selection με μέγεθος ίσο με 3, ομοιόμορφη διασταύρωση με πιθανότητα διασταύρωσης ίση με 0.75 και πιθανότητα μετάλλαξης ίση με 0.5. Η Αναζήτηση με προσομοιωμένη ανόπτηση ρυθμίστηκε με T_o=100, T_{min}=15 και N=10. Το χρονοδιάγραμμα για την μείωση της θερμοκρασίας ορίστηκε ως το γεωμετρικό χρονοδιάγραμμα, με συντελεστή ίσο με 0.8. Στα πειράματά μας εξετάζουμε το σενάριο κατά το οποίο οι δύο μέθοδοι αρχικοποιούνται βάσει των αποτελεσμάτων του πρώτου μέρους της μεθόδου. Συγκεκριμένα, ο αρχικός πληθυσμός του Γενετικού Αλγορίθμου αποτελείται από 12 θεραπείες (25%) που αξιολογήθηκαν ως οι πιο αποτελεσματικές από τον αλγόριθμο Ενεργητικής Μάθησης, ενώ τα υπόλοιπα άτομα του πληθυσμού (75%) αποτελούν τυχαίες θεραπείες. Το σημείο εκκίνησης της Αναζήτησης με προσομοιωμένη ανόπτηση τίθεται ως η θεραπεία που αξιολογήθηκε ως η πιο αποτελεσματική από το πρώτο μέρος. Εξετάζουμε την αρχικοποίηση των μεθόδων βελτιστοποίησης βάσει των αποτελεσμάτων του εκδοχών του αλγόριθμου Ενεργητικής Μάθησης που εφαρμόζουν τις μεθόδους BIRCH, DBSCAN, KMEANS_20 KMEANS_50 και KMEANS_500 στο στάδιο της συσταδοποίησης της δειγματοληπτικής διαδικασίας.

Οι πίνακες 1, 2 παρουσιάζουν την επίδοση του Γενετικού Αλγόριθμου και της Αναζήτησης με προσομοιωμένη ανόπτηση, αντίστοιχα. Παρατηρούμε ότι σε όλες τις εξεταζόμενες περιπτώσεις, ο Γενετικός Αλγόριθμος επιδεικνύει την καλύτερη επίδοση, αφού ανακαλύπτει αποτελεσματικότερες θεραπείες. Επιπλέον, παρατηρούμε ότι η επίδοση του Γενετικού Αλγόριθμου είναι πανομοιότυπη σε όλα τα σενάρια, ενώ δεν παρατηρούμε το ίδιο για τα αποτελέσματα της Αναζήτησης με προσομοιωμένη ανόπτηση. Συγκεκριμένα, παρατηρούμε ότι η αρχικοποίηση της μεθόδου με βάση τα αποτελέσματα του αλγόριθμου Ενεργητικής Μάθησης με εφαρμογή του KMEANS_500 οδηγεί σε αισθητά καλύτερα αποτελέσματα.

Πίνακας 1: Αποτελέσματα αναζήτησης της πιο αποτελεσματικής θεραπείας με χρήση Γενετικού Αλγόριθμου.

Μέθοδος	Βαθμός Επιζώντων Καρκινικών Κυττάρων	Αριθμός προσομοιώσεων της δεύτερης φάσης	Συνολικός αριθμός απαιτούμενων προσομοιώσεων
BIRCH	0.166	811	1853
DBSCAN	0.164	810	1007
KMEANS_20	0.166	742	1242
KMEANS_50	0.174	814	1914
KMEANS_500	0.169	661	10761

Πίνακας 2: Αποτελέσματα αναζήτησης της πιο αποτελεσματικής θεραπείας με χρήση Αναζήτησης με προσομοιωμένη ανόπτηση.

Μέθοδος	Βαθμός Επιζώντων Καρκινικών Κυττάρων	Αριθμός προσομοιώσεων της δεύτερης φάσης	Συνολικός αριθμός απαιτούμενων προσομοιώσεων
BIRCH	0.196	91	1133
DBSCAN	0.209	91	288
KMEANS_20	0.209	91	591
KMEANS_50	0.183	91	1191
KMEANS_500	0.177	91	10191

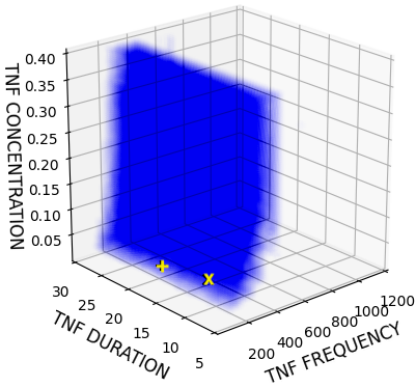
Επιπλέον, παρατηρούμε ότι η εκδοχή της μεθόδου που χρησιμοποιεί την Αναζήτηση με προσομοιωμένη ανόπτηση απαιτεί αισθητά λιγότερες προσομοιώσεις από όταν η αναζήτηση για την αποτελεσματικότερη θεραπεία πραγματοποιείται με εφαρμογή του Γενετικού Αλγόριθμου, γεγονός που την καθιστά αισθητά πιο αποδοτική όσον αφορά τις απαιτούμενες προσομοιώσεις.

Όπως αναφέρθηκε παραπάνω, το δεύτερο μέρος μπορεί να λειτουργήσει και σαν μέσο επιβεβαίωσης του χαρακτηρισμού του χώρου παραμέτρων των θεραπειών. Όπως φαίνεται από το Σχήμα 9, οι πιο αποτελεσματικές θεραπείες που εντοπίζονται από τις δύο μεθόδους βελτιστοποίησης ανήκουν είτε στα σύνορα είτε στο εσωτερικό του χώρου που χαρακτηρίστηκε ως ενδιαφέρον από τον ταξινομητή του πρώτου μέρους, γεγονός που επαληθεύει τα αποτελέσματα του πρώτου μέρους της μεθόδου.

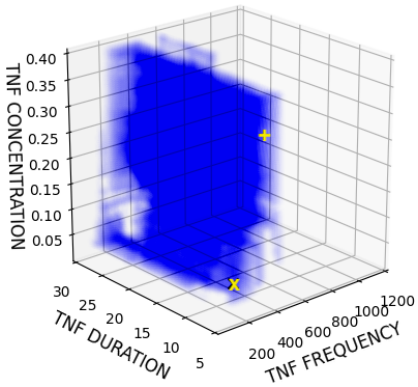
Σύνοψη

Στα πλαίσια της παρούσας διπλωματικής, επεκτείναμε την μέθοδο για την εξερεύνηση που παρουσιάστηκε από τους Ozik et.al [3], δίνοντας την δυνατότητα στον εκάστοτε χρήστη να επιλέξει ανάμεσα σε διαφορετικές μεθόδους συσταδοποίησης και βελτιστοποίησης. Επιπλέον, προσπαθήσαμε να χαρακτηρίσουμε τον χώρο παραμέτρων καρκινικών θεραπειών, κάνοντας χρήση ενός νέου υπολογιστικού μοντέλου, το οποίο περιγράφει την αλληλεπίδραση της πρωτεΐνης Tumor Necrosis Factor με τα καρκινικά κύτταρα. Επιπλέον, μελετήσαμε την επίδραση

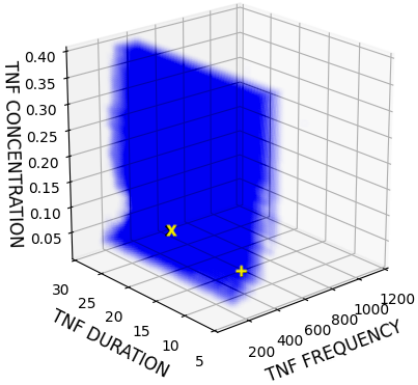
διάφορων ευρέως γνωστών αλγορίθμων συσταδοποίησης στην δειγματοληπτική διαδικασία της μεθόδου χαρακτηρισμού του χώρου παραμέτρων, αλλά και την επίδοση μεθόδων βελτιστοποίησης στην ανακάλυψη των πιο αποτελεσματικών θεραπειών, με στόχο την μείωση των απαιτούμενων προσομοιώσεων. Τα πειραματικά αποτελέσματα υποδεικνύουν την αποτελεσματικότητα της μεθόδου εξερεύνησης θεραπειών κατά του καρκίνου και την ύπαρξη ενός ισοζυγίου μεταξύ της ποιότητας των αποτελεσμάτων και των απαιτούμενων πόρων.



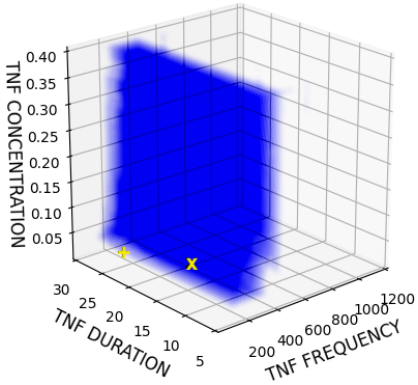
(α′) KMEANS_500



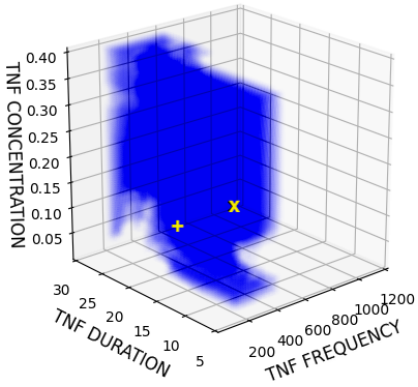
(β′) KMEANS_20



(γ′) KMEANS_50



(δ′) BIRCH



(ε′) DBSCAN

Σχήμα 9: Γραφική απεικόνιση των αποτελεσμάτων των μεθόδων βελτιστοποίησης. Οι θεραπείες που ανακαλύπτονται από τον Γενετικό Αλγόριθμο και την Αναζήτηση με προσομοιωμένη ανόπτηση σημειώνονται με 'x' και '+' , αντίστοιχα.

Chapter 1

Introduction

1.1 Motivation

Cancer disease is one of the leading causes of death globally, being responsible for approximately 10 million deaths in 2020 according to the World Health Organization.[1] The alarming statistics have made the discovery of promising treatments a top priority for the medical and biological research community [8]. The development of a new treatment can be divided into five discrete phases. Phase 0 consists the pre-clinical stage of the development in which basic research around the body processing and the efficiency of the drug candidate is conducted [9]. The three subsequent phases consist the clinical stage of the drug development. In particular, Phase I is designed to determine the dose-toxicity in humans, the short term side effects associated with increasing doses and get a first indication regarding the efficiency of the examined drug [10]. Phase II is designed to evaluate the performance of the treatment under examination, while Phase III aims at the estimation of the balance between safety and effectiveness [11]. Lastly, Phase IV aims at the monitoring of long-term side effects and the interaction of the drug with other treatments. The first four aforementioned phases consist the main pipeline in drug discovery and take on average from 9 to 12 years [12] to be completed.

It is apparent that the process described above is both costly and time-consuming. Moreover, in the development of cancer treatments, the dynamic nature and the mortality of the disease make conducting extensive clinical trials difficult in many cases, since the recovery time window is often narrow and any possible error or miscalculation may have devastating results to the patient's health. To tackle this, computational biology has assisted

[1] `https://www.who.int/news-room/fact-sheets/detail/cancer`

the researchers in drug discovery by providing models that attempt to describe the behavior of tumor cells in the human organism. These models allow practitioners to study via simulations the effectiveness of various tumor treatments *in-silico*, without putting the human lives on the line. Experiments conducted using such models can lead to the reduction of the time required for the pre-crinical phase of drug development. These experiments can, also, reduce the percentage of treatments failing to meet the required performance standards in later stages of the treatment development, as it allows researchers to proceed to clinical trials focused only on promising treatments [13].

Although biological computational models are detailed and insightful, in most cases they are computationally expensive and time-consuming, hindering the researchers from exhaustively examining the effect of every possible configuration of the examined treatments. To that end, Machine Learning (ML) methods can be used to reduce the required number of simulations in an informed manner, and to provide accurate predictions about the effectiveness of treatments. In particular, Ozik et al. [3] presented a workflow that aims to discover the characteristics of promising drug treatments, by identifying the value ranges of the drug-related parameters that are expected to induce desirable effects. Their proposed workflow (mentioned as "original workflow" in the rest of this thesis) consists of two parts. In both parts of the workflow, subsets of the treatment parameter space are sampled and evaluated. The first part of the workflow comprises of an Active Learning (AL) algorithm [4] that aims to identify the interesting regions of the treatments parameter space, by iteratively evaluating sampled points and training a classifier. In particular, the evaluated samples are used to train a classifier, which then indicates the most uncertain regions that in turn determine the samples to be evaluated next. These regions, however, presumably contain a large number of points. In order for the exploration in the subsequent iterations to progress, the uncertain points are clustered according to their spatial similarity and only a few representative points are finally evaluated. The second part consists of a Genetic Algorithm (GA) for discovering optimized treatment configurations. Particularly, the GA is applied in order to discover the most effective treatment configuration. In the random scenario of the second part, the initial state of the GA is random. However, the application of the GA may, also, incorporate knowledge obtained by the first part of the workflow (seeded scenario). In the seeded scenario, part of the initial population consists of treatments found to be the most effective by the AL algorithm. This workflow provides a valuable tool in the researchers' arsenal and enables them to perform targeted and promising real-world trials.

1.2 Contribution

In this thesis, we extend the original workflow, and apply it in a different experimental setting. Specifically, we examine the effect of a signal molecule that can induce death in tumor cells, in a similar manner to tumor treatments, called *Tumor Necrosis Factor* (TNF) [5]. Moreover, we conduct experiments utilizing PhysiBoSSv2,[2] an add-on that expands upon the PhysiCell cell simulator [6] used in the original workflow. PhysiBoSSv2 introduces an agent-based multi-scale model for tumor cell growth used to examine the impacts of given drug configurations. Both parts of the workflow sample subsets of the available parameter values, to evaluate them on a high-performance computing infrastructure.

The requirements for computational resources to sufficiently explore alternative treatments are quite high. Thus, it is important to devise faster and less resource-demanding alternatives for such approaches, without compromising performance. Such modifications can be proven really valuable in even more complex settings, such as when exploring the synergistic effects of multiple drugs being administered simultaneously to the patient's organism [14].

In the first part of the workflow, the clustering component of the sampling process plays an important role in the number of the required simulations and the quality of the results produced. However, based on their design principles, clustering methods of different categories may lead to results with quite noticeable differences in a variety of settings. Moreover, they require user-defined hyper-parameters in order to operate that may be difficult for non-experts to determine in advance. In the original workflow, the K-Means clustering algorithm was used in the sampling process. However, K-Means fails to identify clusters of arbitrary shape and to eliminate noise in the candidate points set. It is also unable to dynamically adjust the number of clusters to the spatial distribution of each different candidate set. These restrictions lead to a distortion of the sampling process, as outlying points are taken into account in the formation of the clusters and, in many cases, larger clusters are broken down to smaller ones (or vice versa) in order to reach a user-defined number of clusters, which is a required configuration parameter. Such restrictions might degrade the sampling process and often result in the execution of redundant or less informative simulations. To this end, we examine the effect of additional clustering methods, i.e., DBSCAN and BIRCH, to the effectiveness of the sampling process in the workflow, and compare their performance, in regards to the number of simulations required

[2] https://github.com/bsc-life/PhysiBoSSv2

and the quality of the parameter space characterization achieved by the AL algorithm of the workflow.

Furthermore, we also examine the performance of an additional optimization method, that is Simulated Annealing (SA), for the discovery of the most efficient treatments. The maintenance of a big pool of candidate solutions by the GA requires multiple computational resources in each generation. In the seeded scenario of the optimal treatment discovery part of the workflow, the initial search space of the optimization method is restricted to the areas found by the AL part to contain viable treatments. If the search space is small, SA and GA may yield similar results, as both methods perform well in the improvement of the set of parameters around an initial solution [7]. However, SA focuses on a single candidate solution, while GA maintains a population of possible solutions. The different approach of the methods makes SA less demanding in terms of computational resources when facing such problems. We take advantage of this characteristic and apply SA in the optimized drug treatments discovery by incorporating the knowledge we have deprived from the AL component, focusing only in regions already classified as viable. In particular, we set the initial point of SA to be one of the most promising treatments evaluated by AL. In this way, the algorithm examines an area that has already been characterized as viable and mainly consists of local optima that are of similar, or better magnitude. In this thesis, we compare the performance of the two aforementioned optimization methods, in regards to the resources required by each optimization method, namely the number of simulations, and the effectiveness of the treatments discovered by each method.

Our experiments demonstrate that our workflow achieves a treatment space characterization of high quality. Also, results indicate the existence of a trade-off between the amount of simulations performed and the stability of the produced solutions, both in the characterization of the treatment configuration space and in the discovery of the most effective treatments.

1.3 Thesis Outline

The remainder of this thesis is structured as follows. In Chapter 2 we present the related work. First, we present various computational models designed to describe the tumor microenvironment and then present various works applying machine learning methods in the Bioinformatics field. Then, in Chapter 3 we present the necessary theoretical background. In particular, we introduce the examined clustering, optimization and classification methods. In Chapter 4 we introduce the multi-scale simulator used for the experimental purposes

and describe in detail the workflow for tumor treatment exploration used. In Chapter 5 we present the results of our experiments, while in Chapter 6 we summarise the work and discuss further research directions. In appendices, results of various experimental runs are provided in order to supplement the presentation of the experimental results, while avoiding overcrowding the main body of the thesis.

Chapter 2

Related Work

2.1 Computational Modeling methodologies of tumor microenvironment

Computational modeling provides a resource-effective tool in the examination of the interactions taking part in various natural systems, such as the initiation and growth stages of cancer. These mathematical models can be divided on categories based on their attributes in several ways. One possible division is the categorization based on the stochastisity of the model. In particular, we call discrete models the models that lead to the same end state for certain initial conditions, while we call stochastic those that may lead to different end states for the same initial conditions due to the inclusion of randomness of the cellular interactions. Another popular categorization divides the model into discrete and continuous ones. Discrete models examine the behavior of discrete cells, e.g. the interactions between individual cells, whereas continuous models take into consideration groups of individual cells. A multi-scale setting may incorporate approaches belonging to different categories. Models following this approach are called hybrid models. Computational models of the tumor environment lying into different categories have been used widely in the examination of the response of the immune system to cancer progression and immunotherapy. In this thesis, we use a multi-scale agent-based model that simulates the growth of tumor cells and can take into consideration the binding of a signaling molecule that binds to cell receptors and may induce death to tumor cells.

Dreau et al. [15] developed an agent-based model that studies the association between tumor intrinsic properties, the responsiveness of the immune system and the vascularization and the progression of solid tumor treatment. Their model mimics tumor growth based on the nutrition needs of tu-

mor cells and consists of components representing tumor cells, immune cells (macrophages and lymphocytes) and vascular regions with with low, moderate and high vascularization. The model not only sets principles that describe the relationships among the aforementioned components, but also uses time-dependant interactions. The researchers concluded that an increased initial immune response leads to a slow tumor growth and to decreased surviving tumor cells. Moreover, experimental results presented that repeated increases in the number of immune cells throughput the experimental runs lead to a substantial decrease in tumor burden.

During cancer development, surviving cancer cells develop features that allow them to avoid detection from the immune system. One of these features is the inhibitory signalling from molecules that reduce the functionality of the immune cells. The programmed cell death protein (PD-1) and its ligand PD-L1 act as an immune checkpoint pathway, i.e. as a regulator of the immune system that prevents excessive immune activities. Cancer cells can stimulate immunity checkpoint targets, such as PD-1/PD-L1, and suppress the host's anti-tumor immunity. Immune checkpoint therapy [16] targets at the control of such regulatory factors of the immune system in order to release the total power of the anti-tumor immune response. Gong et al. [17] presented an agent-based model to study the spatial dynamics of tumor cells and T-cells, a type of lymphocyte cells that play an important role in the adaptive immune response. In the presented model, the effect of the the anti-PDL1 treatment is modeled as a factor that decreases the probability that a T-cell is suppressed by a $PDL1^+$ cancer cell. Gong et al. found that the effectiveness of the anti-PDL1 treatment is affected by the level of the mutational burden of the cancer and by the neoantigen characteristics of each patient. Moreover, experiments showed that, in the setting in which tumor vasculature is responsible solely for the transportation of tumour antigen specific T-cells, there is no clear correlation between the vasculature density distribution and tumor growth. Based on the aforementioned results, the authors proposed a scoring system to assess potential predictive biomarkers for anti-PDL1 treatments.

Rejniak et al. [18] introduced a model that describes the effect of the structure and distribution of tumor cells on the delivery of chemical compounds, such as those of cancer treatments. Simulations demonstrated that the tumor cell distribution, which is identified by the cellular porosity and density of the tumor tissue, play an important role in the depth of a drug's advective penetration. In particular, experiments showed that low density tissues lead to longer times of drug penetration and to slower interstitial fluid flow. Moreover, they showed that irregularities in the cells spatial configuration may reduce the drug concentrations, as they lead to tissue zones with

low exposure to molecules of the drug. Hence, the experimental findings suggested that the tissue architecture influences greatly the depth of the tissue penetration achieved by drug molecules.

2.2 Machine Learning in Bioinformatics

The amount of biological data that requires analysis by experts has risen exponentially in recent years. In order to interpret the increasing available data and discover behavioral patterns of biological systems, ML methods have been widely used in Computational Biology and Bioinformatics. The predictive models generated by ML give insights to the functional relationships of the systems and provide accurate statistical predictions in a range of biological applications [19, 20]. For instance, Błażewicz et al. [21] proposed a time-effective method to discover low energy protein structures using a heuristic optimization method. The authors used simplified protein structure prediction models and the Tabu Search algorithm in order to discover the native structure of the protein. An essential feature of the Tabu Search algorithm is the exclusion of possible candidate solutions that are characterized as forbidden. In a similar manner, the AL algorithm of our workflow aims at the mapping of uninterested regions. These regions are then eliminated from the treatment search by the GA during the optimization.

Another common application area of machine learning methods in Bioinformatics is the biomedical image processing. The aim of the biomedical imaging is the analysis of medical images for diagnostic and treatment purposes. An important step of this analysis is the classification of the biomedical images. In many instances, GAs have been combined with clustering techniques in AL workflows for the classification of biomedical images. [22] adopts such an approach combined with self-organizing maps, to reduce the amount of manual labor required for annotating and analyzing cancer patient screening images. Even though the framework presented reduces the required human labor, it achieves an accuracy level of equal or greater than that achieved solely by human annotators. Interactivity allows human supervision and intervention, but this is only required in smaller scale. Unsupervised learning helps to detect uncertain regions and ask for more targeted input by experts, while automatically expanding learned classification rules to known cases.

Moreover, Evolutionary Algorithms have been widely used in *de novo* drug design [23]. *De novo* drug design involves an incremental construction of new molecules based on the structure of a receptor [24]. Wang et al. [25] presented a tool called LigBuilder, which uses a GA approach to build

new ligand molecules. In this approach, a step by step construction of new molecules was applied resulting to an immerse possible solution space. In order to render the construction process more efficient and to obtain a reduced solution space, a GA was used in order to control the building process of the new molecules, with the fittest individuals of the last generation of the GA being selected as the final results.

Throughout the years, Active Learning algorithms have been used in various fields of Bioinformatics, such as the detection of potentially promising regions of drug configurations [26], due to their ability of reducing the required resources based on their selective sampling strategy. Warmuth and Putta [27] presented an AL algorithm in order to assist drug discovery. In particular, they examine the application of a Support Vector Machine (SVM) classifier in order to find efficient (active) drug compounds from a large collection of compounds. Their goal is to divide the compounds dataset to active and non-active compounds with the minimum possible amount of screening trials. Moreover, they examined the performance of various selection strategies for the sampling stage of the AL algorithm. Experiments showed that the sampling of points that have the maximum positive distance from the separation boundary (i.e. the ones supposed to be most active) is most suitable for exploitation purposes, i.e. when the final goal is to find a high number of active compounds. On the other hand, the sampling of points that are near the separation boundary appeared to yield better results in the exploration of the entire data set, i.e. it provided a better understanding of the distribution of the active and non active compounds in the whole collection. In a similar manner, the AL algorithm of our workflow, which aims at the exploration of the treatment parameter space, considers as most informative the points lying on the classification boundary, i.e. the most uncertain points as mentioned in Section 4.2.

Microarray is a tool for studying the molecular basis of interactions used widely in cancer research [28], allowing researchers to study a vast number of genes simultaneously [29]. One objective of medical researchers is to identify small sets of genes that have strong predictive performance regarding the examined disease. The identification of this small set of genes and the elimination of redundant genes allows researchers to focus their diagnostics examinations. Liu [30] applied an AL algorithm with a SVM classifier in the classification of cancer based expression data from DNA microarray hybridization experiments. In particular, Liu used the genes profiles of samples of three common type of cancer, i.e. colon, lung and prostate cancer. The SVM classifier was trained on a training dataset consisting of samples with the largest predictive value (Active Learning) and its performance was compared to a SVM classifier trained on random samples (Passive Learning).

Experiments showed that although AL required the evaluation of significantly fewer samples, it performed evenly or even better in many cases as compared to passive learning. In a similar setting, Diaz et.al [31] presented the application of a random forest classifier for the classification of microarray data and the identification of small sets of genes that lead to good predictive performance and can be used for diagnostic purposes, such as the classification of cancer. Moreover, they compared the performance of the Random Forest to other Machine Learning methods used by researchers. The results show that Random Forest yields similar results in the classification task and leads to smaller sets of genes in many cases compared to the other methods.

AL algorithms have assisted, also, in the understanding of protein patterns and interactions. Proteins control the biological systems of a cell, with the majority of proteins controlling biological activity via interaction with other proteins. Hence, the understanding and prediction of the protein-protein interactions can lead to a deeper understanding of the biology of the human cells. Mohamed et al. [32] compared AL methods for guiding the selection of protein pairs for future experimentation in order to accelerate accurate prediction of the human protein interactions. The results suggested that AL manages to accelerate the discovery of interacting protein pairs, even in datasets where the ratio of interacting pairs is very low.

In our approach, we examine the application of different Machine Learning submodules in the exploration of tumor treatments and focus on the impact of these submodules on the quality of the results. Moreover, we examine their effectiveness with respect to the computational resources required. In comparison with the aforementioned approaches, our method allows for the selection of the different submodules to be used in each experiment. Thus, it allows us not only to examine the perfomance of the submodules in the given experimental setting, but also gives us the ability to select submodules accordingly to the setting under examination. Hence, it makes the expansion of the workflow to different experimental settings easier.

Chapter 3

Theoretical Background

In this section, we cover the theoretical background behind the various methods examined in the treatment discovery. In particular, we introduce the clustering algorithms used in the the sampling process of the workflow, the optimization methods used for the optimal treatment discovery and the classification method applied in the separation of the parameter space to viable and non viable treatments.

3.1 Clustering Algorithms

Clustering algorithms are used for the identification of similarities between different data instances in a plethora of fields, such as finance [33] or document analysis [34]. The main goal of the clustering algorithms is to divide a given set of data instances into groups, called clusters. Instances in the same cluster must be similar as much as possible. The traditional clustering algorithms can be divided into 9 categories depending on the approach they use for the clustering of the data [35]. The three main categories of clustering algorithms are partitioning, hierarchical and density-based [36].

The partitioning clustering algorithms are based on the idea that a center data point can represent a cluster [37]. Partitioning algorithms' goal is to divide the data instances space into k clusters. Initially, k random partitions are created and the partitioning is, then, iteratively improved using a relocation method. Finding the optimal k parameter is crucial and requires prior knowledge regarding the true distribution of the instances. This knowledge is usually not available in most applications, especially in problems in which the distribution of the data instances changes dynamically. Hierarchical algorithms aim in the hierarchical decomposition of the space, i.e. in the understanding of the hierarchical relationship among the data in-

stances. The decomposition is represented as a tree that splits the space into smaller sub-spaces (clusters) until a certain termination criterion is met, e.g., a distance threshold between clusters [38]. Finally, density-based algorithms cluster the data based on the density and connectivity of the points. Clusters are considered to be sets of points of high density separated by low density regions [39]. In this thesis, we examine the effect that well-known clustering algorithms belonging to the three aforementioned main categories have on the performance of the treatment discovery workflow. For our experimentation purposes we selected a representative algorithm from each category, in particular the K-Means, BIRCH, and DBSCAN algorithms [35].

3.1.1 K-Means

One of the most widely used clustering methods is the *K-Means* algorithm [40], in which each cluster is represented by a point called the centroid of the cluster and all points in the dataset are assigned to the closest centroid.

More formally, consider a set of points $X = \{x_1, x_2, \ldots, x_n\}$ a set of clusters $C = \{c_1, c_2, \ldots, c_k\}$, a function $d(x, y)$ that measures the distance of two points in the dataset and a function $\phi c(x_i) = argmin_{c \in C} d(x_i, c)$ that finds the closest centroid to point x_i. The goal is to find the optimal set C of clusters that minimize the inner cluster distance, defined as follows:

$$min \sum_{x \in X} d\big(\phi c(x), x\big)$$

In brief, K-Means consists of three steps. First, k points are selected from the dataset to form the initial centroids. Then, each point is assigned to the nearest centroid, and each centroid is redefined as the center of mass of all the points assigned to that cluster. The two last steps are repeated until convergence is reached.

The K-Means algorithm is one of the most widely used algorithms due to its simplicity and speed, and was utilized by [3]. However, it entails some limitations that may lead to poor accuracy. As mentioned above, the final number of clusters of the dataset must be decided by the user prior to the application of the algorithm. However, the number of clusters may be difficult to determine beforehand, as in most cases there is no prior knowledge regarding the spatial distribution of the data instances. In addition, the clustering process can be affected by noise, as outliers participate in the calculation of the centroids. The participation of noise in the determination of the centroids may lead to clusters with shifted centers, leading to a distorted result. Moreover, K-Means identifies spherical clusters of similar sizes with

Algorithm 1 K-Means Clustering

Input: $D = \{D_1, D_2, \ldots, D_n\}$: Data instances, k number of desired clusters
Output: Data instances with cluster memberships

```
1: procedure K-MEANS(D,k)
2:     Randomly initialize k centroids.
3:     repeat
4:         for each D_i data instance do
5:             Assign D_i to closest centroid
6:         Update the centroids of the clusters
7:     until converge is met
8:     return data instances with cluster memberships
```

radius equal to the distance of the centroid to the boundary points of the cluster. Hence, K-Means fails to find clusters of arbitrary sizes and densities.

3.1.2 DBSCAN

Ester et al. [41] introduced the *DBSCAN* algorithm, a density-based clustering approach designed to identify clusters of arbitrary shapes. DBSCAN relies on the Eps and $MinPts$ parameters. Eps defines the maximum distance between two points to be considered as neighbors. Thus, the Eps neighborhood of a point p is defined as follows:

$$N_{Eps}(p) = \{q \in D \mid d(p,q) \leq Eps\}$$

$MinsPts$ is the minimum number of points required to be in the neighborhood of a point p in order for the point p to be considered a core point of a cluster.

The concept of DBSCAN is based on the notions of *density-reachability* and *density-connectivity*. A point p is density reachable from a point q if there is a chain of points $p_1 = p, p_2, \ldots, p_n = q$ such that for each pair of points p_i, p_{i+1}: $d(p_i, p_{i+1}) \leq Eps$ and $Neps(p_{i+1}) > MinPts$. A point p is density connected to a point q if there is a point o, such that p and q are density-reachable from o.

According to DBSCAN, points can be divided into three categories, the core, border, and noisy points. Core points refer to representative points of the cluster, while border points are the ones on the edges of the cluster. Every other is considered as an outlying point.

The algorithm arbitrarily picks a point p and calculates the number of

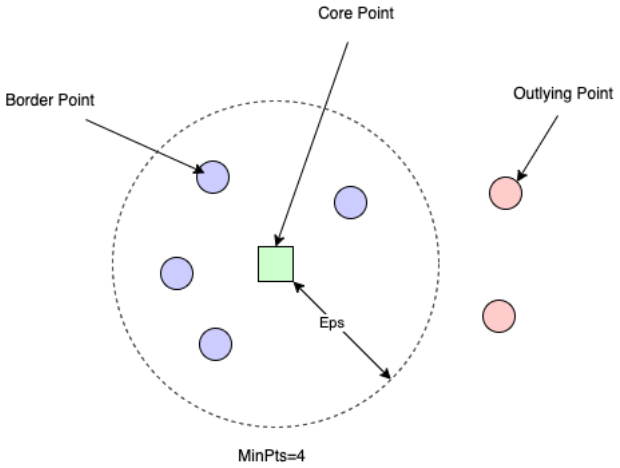


Figure 3.1: Points categories according to DBSCAN

points in its *Eps* neighborhood. If $|N_{Eps}(p)|$ is greater than $MinPts$, then p is the core point and a cluster is formed. If p is a border point, then DBSCAN visits the next point in the dataset. The process is repeated until all points are examined.

DBSCAN is a commonly used clustering algorithm due to its ability to identify clusters of arbitrary shapes and sizes and its robustness to noise in the dataset. Moreover, it is not required to specify the number of clusters in the dataset beforehand. However, performance is sensitive to the user-defined parameters *Eps* and $MinPts$. Also, DBSCAN does not perform well with clusters of different densities, since it relies on a universal combination of Eps-MinPts combination. Lastly, DBSCAN performs poorly in flat geometries, due to the lack of density changes between clusters, as it considers clusters to be sets of points of high density separated by low density regions.

3.1.3 BIRCH

BIRCH is a hierarchical clustering algorithm introduced by Zhang et al. [42] to handle large datasets efficiently. Its efficiency is achieved by creating a summary of the dataset and then processing this summary, instead of clustering the original dataset as a whole. BIRCH bases on the concepts of the *Clustering Feature* (CF) vector and the *CF tree*.

Given N d-dimensional data points in a cluster x_i, where $i = 1, 2, \ldots, N$, the clustering feature vector (CF) of the cluster is defined as a triple $CF = \{N, LS, SS\}$, where N is the number of data points in the cluster, LS is the linear sum of the N data points and SS is the square sum of them. A CF

Algorithm 2 DBSCAN

Input: $D = \{D_1, D_2, \ldots, D_n\}$: Data instances, Eps, Min_Pts
Output:

```
1: procedure DBSCAN(D,Eps,Min_Pts)
2:     for each unvisited point D_i in D do
3:         mark D_i as visited
4:         neighPts ← calculatetheEps_NeighborsofD_i
5:         if size(neighPts) < Min_Pts then
6:             mark D_i as outlying point
7:         else
8:             C ← New Cluster
9:             expandCluster(D_i, neighPts, C, Eps, Min_Pts)
10: procedure EXPANDCLUSTER(P, neighPts, C, Eps, Min_Pts)
11:     add P to cluster C
12:     for each point Neigh in neighPts do
13:         if Neigh is not visited then
14:             mark Neigh as visited
15:             neighPts' ← calculatetheEps_NeighborsofNeigh
16:             if size(neighPts') < Min_Pts then
17:                 neighPts ← neighPts ∪ neighPts'
18:             if Neigh not part of a cluster then
19:                 C ← C ∪ Neigh
```

Tree is a height-balanced tree, which acts as a compact representation of the original dataset. A leaf node in the tree contains at most L entries of CFs and two pointers linking the node to the previous and the next leaf node. Internal nodes contain entries of the form $[p, CFp]$, where p is a pointer to a child node and CFp is the sum of all the CFs in the child node.

Each CF Tree requires two parameters, the branching factor B and the threshold T. Each internal node of the tree can contains at most B entries and the diameter of each leaf entry has to be less than T. The algorithm scans the data and creates a CF Tree by iteratively selecting data samples. At each step, a new data sample is selected and the nearest leaf node sub-cluster in the existing CF tree is obtained. If the distance between the centroid of the closest sub-cluster and the new sample is less than the threshold T, then the sample is added to the sub-cluster and the properties of the leaf node and its parent nodes are updated. Otherwise, a new sub-cluster is created and added to the CF Tree. In case the addition of the new sub-cluster breaks the

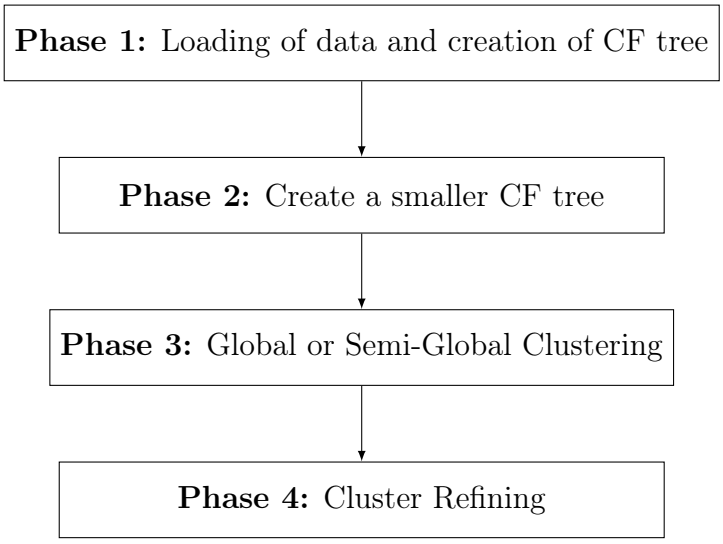


Figure 3.2: Overview of BIRCH clustering algorithm

branch factor condition, then the parent node is split. In this way, outlying points do not distort the existing clusters and can be isolated into smaller clusters consisting only from close noisy points. An overview of the BIRCH algorithm is illustrated in Figure 3.2.

In detail, the four phases of BIRCH algorithm are:

Phase 1 The algorithm scans the data and creates a CF tree with an initial threshold value by inserting points into the tree. If it runs out of memory before the completion of the scan, it increases the threshold value and rebuilds a new CF tree by re-inserting the leaf entries of the old tree and resuming the scanning from the interruption point.

Phase 2 (Optional) In this phase, the algorithm scans the CF tree and tries to merge the clusters of the leaf nodes in order to create a smaller CF tree. This phase is optional and serves as a preparatory step for the optimal performance of the clustering algorithm used in the third phase.

Phase 3 A global or semi-global algorithm is used to cluster all leaf entries and a set of clusters is obtained. This set of clusters captures in a good manner the correlations of the data instances. The algorithm can be terminated after this phase.

Method	Characteristics	Limitations
K-Means	Clusters of similar shape and size	Difficult definition of optimal k Clustering data of varying sizes and density
DBSCAN	Clusters of varying size and shape Noise detection	Parameter sensitive Varying density clusters
BIRCH	Time and memory efficient Noise detection	Not scalable with high dimensional data

Table 3.1: Examined Clustering Methods Overview

Phase 4 (Optional) Another scan of the original data is performed and each point scanned is assigned to closest centroid of the clusters found in the third phase of the algorithm. This phase is optional and corrects any possible inaccuracies. This phase can also act as a noise detection step. Each point that lies too far from its closest centroid is considered to be an outlier and is eliminated.

In our approach, we consider the leaf nodes of the CF Tree as the final clusters. These clusters can be categorized to ones containing informative points and those containing outlying points.

BIRCH is a fast algorithm, which efficiently clusters large datasets, does not require specifying the number of clusters and can also detect outlying points.

Table 3.1 summarizes the characteristics and limitations of each clustering method under examination.

3.2 Search Procedure for Optimized Treatments

Heuristic search methods, such as the GAs, have been developed in order to solve optimization problems that are difficult or even impossible to be reduced into an analytical form and thus solved by exact numerical algorithms. Such methods require little or no knowledge of the problem's domain and aim in the discovery of the global minimum (or maximum) of an objective function. Although they cannot provide guarantees for finding the true global optimal solution, they can discover many "good" solutions that are locally optimal. Such methods have been applied in a wide variety of fields, e.g. finance [43], or power systems [44]. We examine the performance of two well-know

and widely used optimization methods in the discovery of optimized treatments. In particular, we examine the optimization methods called Genetic Algorithms and Simulated Annealing. The Genetic Algorithms consist one of the most widely used optimization methods, as they search for the optimal solution based on a population of candidate solutions, thus avoiding getting stuck in local optima. Moreover, they have demonstrated good performance in noisy environments in numerous applications. On the other hand, SA aims at the discovery of the optimal solution based on a single candidate solution. Hence, SA may require less resources in comparison to the optimization methods that are based on groups of solutions. Moreover, the simplicity of the algorithm of SA allows the easy adaptation of the method to the nature of each examined problem.

3.2.1 Genetic Algorithms

The Genetic Algorithms (GA) have been used widely in optimization applications in various fields [45, 46, 47]. GAs are a family of optimization models inspired by evolution, and especially by natural selection [48]. These algorithms encode each possible solution of the examined problem on a data structure resembling a chromosome structure. The encoding selected depends on the nature of the examined problem and may vary deeply from application to application [49]. The aim of the algorithms is to derive the best attainable solutions (fittest individuals) after applying some recombination operators.

A Genetic Algorithm consists of five phases. In particular, a set of initial solutions is created. Each solution is called an *individual* and a set of individuals is called a *population*. Each individual is characterized by its *genes*, i.e. the parameters specifying the represented solution. The genes of each individual are joined into a data structure, which mimics the structure of a chromosome. In order to evaluate each individual an evaluation function is defined. This evaluation function is called *fitness function* and measures the fitness of an individual. The encoding of the possible solutions and the fitness function are the two problem dependent parts of GA and the selection of the most suitable ones play a crucial role in the performance of the algorithm. In the second phase, the population is evaluated and each individual is assigned a fitness score. Sequentially, the individuals with the highest fitness scores are selected. These individuals are the fittest of the population and will pass on their characteristics to the next population, via the crossover phase of the algorithm. In particular, the fittest individuals are mated and new individuals are derived maintaining desirable genes from the original individuals. In order to maintain a certain level of diversity in the newly generated population, some individuals undergo a mutation pro-

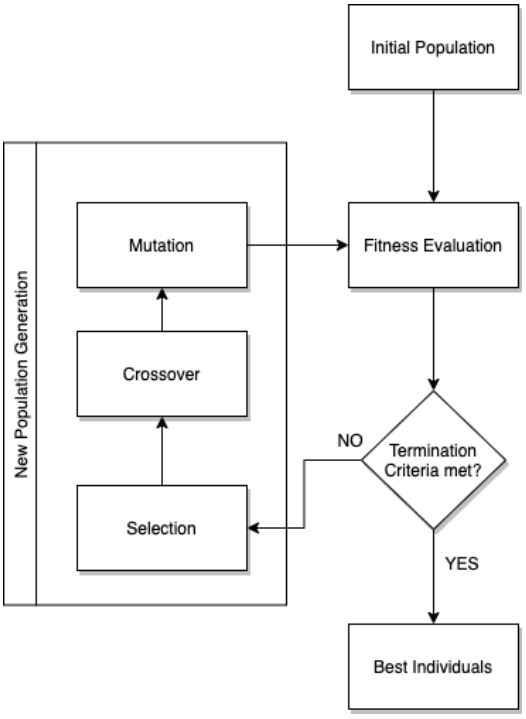


Figure 3.3: Overview of genetic algorithm.

cess. The procedure, i.e. the evaluation, selection, crossover and mutation of the individuals is repeated until some set termination criteria are met. The repetition of this procedure leads to evolved-better solutions (individuals) to the examined problem. Figure 3.3 illustrates an overview of the phases of the genetic algorithms.

GAs consist a valuable tool when the search space is very large and when each solution is defined by a large number of parameters. Moreover, they provide a pool of good solutions, instead of a single best solution, in contrast with many other optimization methods. Furthermore, the evaluation of the population of each generation can be done efficiently using parallelization techniques. However, if the fitness function is computationally expensive the required resources for the maintenance of the populations may out weight the benefits of parallelization. Lastly, the stohastic nature of the algorithm, and especially of the recombination operators used, does not guarantee the quality of the provided solutions, as different runs of the same Genetic Algorithm may lead to different results.

3.2.2 Simulated Annealing

In addition to the GA, we examine the application of the Simulated Annealing (SA) optimization technique [50, 51] for discovering the best treatment configurations. SA has been applied for many years in problems that aim at the minimization or maximization of an objective function, such as portfolio problems in finance [52] or the physical design of Very Large Scale Integration (VLSI) circuits [53]. SA is a probabilistic optimization method that mimics the process of metal annealing, in which a metal is heated and cooled slowly in order to solidify its crystals and reach an optimal state of minimum energy. The basic elements of the SA method are the set of possible points S, an energy function $E : S \mapsto \mathbb{R}$ (objective function), an initial temperature (T_o), a minimum temperature (T_{min}), the temperature at k_{th} level T_k, the number of iterations in each temperature level N and the cooling schedule. The SA algorithm consists of two nested loops. The algorithm starts from an initial point (current solution s) and evaluates its "energy". In the inner loop, the set of neighbors of the current point is generated, a random neighbor n is selected and in turn, its "energy" is evaluated as well. The selected neighbor is accepted as the new solution s with probability:

$$P_{accept}(n) = \begin{cases} 1, & E(n) \leq E(s) \\ exp\big(-\frac{\Delta E}{kT_k}\big) & E(n) > E(s) \end{cases}$$

i.e. the new candidate solution is always accepted if it performs better than the current solution. Otherwise, the candidate solution is accepted with an acceptance probability. The acceptance probability decreases exponentially with the inferiority of the candidate solution. The inner loop is repeated until N iterations are completed (equilibrium condition). The acceptance of inferior candidate points gives the ability to escape from local minima and continue the search for the globally optimal solution. In the outer loop, the temperature level is decreased according to the cooling schedule, until reaching T_{min} (cooling condition).

Algorithm 3 Simulated Annealing

Input: T_0: Initial Temperature, T_{min}: Minimum Temperature, Cooling Schedule

Output: S_{best}: Best Solution

procedure SIMULATED ANNEALING(T_0,T_{min}, COOLING SCHEDULE
    $s \leftarrow$ initial solution
    $s_{best} \leftarrow s$
    $T \leftarrow T_0$
    while $T > T_{min}$ **do**
        repeat
            $s_n \leftarrow$ neighbor solution of s
            $\Delta E \leftarrow$ difference between current solution and neighbor solution
            if $\Delta E > 0$ **then**
                $s \leftarrow s_n$
                if $s_n < s_{best}$ **then**
                    $s_{best} \leftarrow s_n$
            else
                $P_{accept} \leftarrow exp\big(-\frac{\Delta E}{kT_k}\big)$
                $v \leftarrow$ random number in $[0, 1]$
                if $P_{accept} > v$ **then**
                    $s \leftarrow s_n$
        until max iterations per temperature level reached
        $T \leftarrow$ Decrased temperature based on cooling schedule
    return s_{best}

At a high temperature level, SA is more tolerant towards moving to inferior solutions and aims in the discovery of the neighborhood of the optimal solutions. As T decreases, the algorithm allows smaller deterioration in energy and focuses into the discovery of the globally optimal solution.

The simplicity of the SA algorithm allows for the easy implementation of the algorithm and its adaptation to different problems and energy functions. However, the quality of the solutions yielded relies heavily on the cooling schedule selected, with most commonly used cooling schedules being very slow, especially in cases where the energy fucntion is expensive to compute.

3.3 Classification

In Supervised Learning, an agent observes some training input and output pairs and aims at learning a function that maps from input instances to output. A learning problem in which the output is one of a finite set of values is called a *classification* problem. Various learning models, such as linear models, nonlinear models or SVMs, have been used in practise to tackle classification problems in many fields. With the rise of classification problems, the enhancement of the performance of the learning models is crucial. Ensemble Learning [54] is a popular technique used to improve the performance of learning algorithms. Ensemble learning methods combine multiple base-models in order to reach a decision. The main concept behind ensemble learning is that the combination of multiple base models will lead to a prediction of better overall quality, based on the fact that errors in the prediction of a single model may be compensated by the other models. For example, suppose that we use M=9 base models in order to assign one of two possible categories to a data instance. In order to classify falsely the instance, at least 5 out of 9 base models should assign a wrong category to the data instance. This example illustrates at best the idea that lies behind ensemble learning. Taking into consideration the prediction made by multiple base models leads to a reduction of the expected error. Suppose that the errors between the predictions of each model are independent. Then, it is evident that by increasing the number of base models participating in the final prediction, we can significantly reduce the expected error [7]. Ensemble learning methods have been used widely in many fields due to their application versatility and their effectiveness [55]. In the original workflow, a well known ensemble learning method called Random Forest Classifier is used. Random Forest is a simple and effective classification method using a large number of Classifications Trees as its base models.

3.3.1 Classification Tree

A Classification Tree [56] aims at the assignment of a class to a data instance. The root and internal nodes of the tree represent decisions and the edges represent attributes, while leaf nodes represent the possible classes. Each non-leaf node splits into two descendant nodes according to the value of one of the categorical predictor variables. A categorical predictor variable X_i takes values from a set of categories $S_i = \{s_{i,1}, \ldots, s_{i,n}\}$. The split of the internal node sends a subset S of Si to one of the child nodes and the remaining categories to the other child, as illustrated in Figure 3.4.

The split used to partition an internal node to its child nodes is chosen as

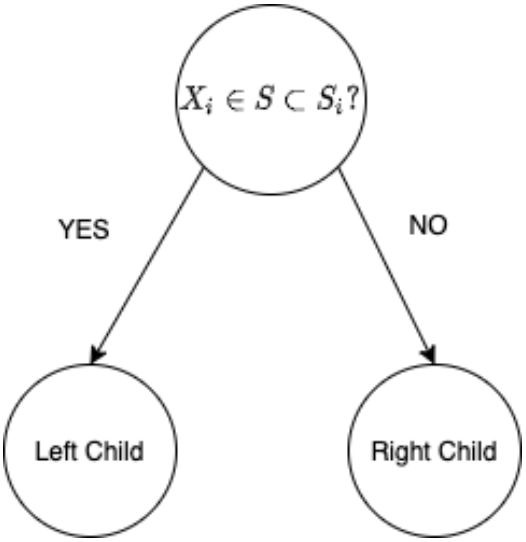


Figure 3.4: Internal node splitting in classification tree

the best on some criterion, after examining all possible splits. A commonly used criterion is the Gini index, also known as Gini impurity. The Gini index is defined as [57]:

$$i(t) = \sum_{i \neq j} p(i|t)p(j|t) = 1 - \sum_{i} p^2(i|t),$$

where $p(i|t)$ is the probability that a random sample belongs to class i, given we are at node t. The Gini index calculates the probability of a specific categorical variable being classified incorrectly, thus gives a measure of the purity of a node. The classification tree is created using a training dataset by iteratively splitting the internal nodes until a stopping criterion is met. In order to assign a class to a data instance d, the instance d is dropped down the tree until we meet a leaf node. The instance d belongs to the class assigned to this leaf node. Each observation falls into one of the leaf nodes.

In many cases, the classification trees are grown as large as possible and then pruned by cutting off branches leading to a subtree of the original classification tree [58]. Pruning plays an important role in the avoidance of over-fitting to the training dataset.

3.3.2 Random Forest Classifier

The Random Forest Classifier [56] is one of the most popular ensemble methods [59, 60], used widely in numerous fields [61, 62]. In recent years, various extensions such as Random Survival Forests [63] and Enriched Random

Forests [64] have been introduced by researchers. The individual classification trees used as base learners are constructed using a training dataset as mentioned in subsection 3.3.1 with some modifications to inject randomness in the trees. Firstly, each individual tree is fitted to an independent sample from the original training set. Secondly, rather than choosing the best split for each internal node, a subset of the attributes is sampled and the best split for them is selected. The aim of the Random Forest classifier is to combine the decisions of each individual tree regarding the data instances under assignment. This is achieved by an unweighted voting of the individual trees, with the final final result being the result found by the majority of the trees. In the original approach, the classification trees used are unprunned, i.e. the trees are grown until the leaf nodes are pure. However, recent suggestions have been presented in which the number of leaf nodes is set by the user. In this setting, the user defines the maximum number of classes in the dataset under examination.

Chapter 4

Framework for model exploration

4.1 Multi-Scale Model Simulations

In the examined method, we incorporate a multi-scale model of a 3D tumor spheroid using the PhysiCell framework [6] and the PhysiBoSSv2 add-on.[1] PhysiCell is a physics-based cell simulator designed to study the interaction between cells in 3-D microenviroments. The microenviroment is modelled using a partial differential equations solver desinged for biological problems called BioFVM [65]. The solver provides the necessary tools required for the mathematical modelling of the responses of the tumor cells in changes in the microenviroment, such as changes in the spatial distribution of cells. Moreover, we use the PhysiBoSSv2 extension of PhysiCell, which simulates the intracellular signal transduction models within each individual cell-agent. Tumor spheroids are composed from different cells, with each tumor cell being modelled as an individual agent. Each individual cell has a Tumor Necrosis Factor (TNF) receptor connected to the signal transduction network. The simulated 3-D domain includes two diffusive molecules, one corresponding to oxygen and another corresponding to TNF. Oxygen is responsible for the cell growth, while TNF is a molecule capable of inducing death to tumor cells. More details regarding the calibration of the simulator and the discovery of treatments using the simulator can be found in [66]. TNF is a critical cytokine that binds to cell receptors and activates signalling pathways, restraining in this way the growth and spread of tumor cells. Thus, the observation of the effects of TNF can be used in order to estimate the effectiveness of various tumor treatment configurations. In our setting, each treatment configuration is defined by the duration, the frequency of administration and the concentration of the TNF. Hence, each treatment can be defined by the tu-

[1] https://github.com/bsc-life/PhysiBoSSv2

ple (TNF_FREQUENCY, TNF_DURATION, TNF_CONCENTRATION) and the set of all possible treatment configurations defines a 3D parameter space of our experiments.

4.2 Workflow for in silico tumor treatment exploration

The workflow for tumor treatment exploration consists of two phases. In the first one, an AL algorithm is used in order to divide the treatment parameter space into promising and non-promising regions, while in the second part, an optimization method is used in order to find the most promising treatment configurations and to validate the results of the parameter space characterization. Figure 4.1 illustrates an overview of the workflow.

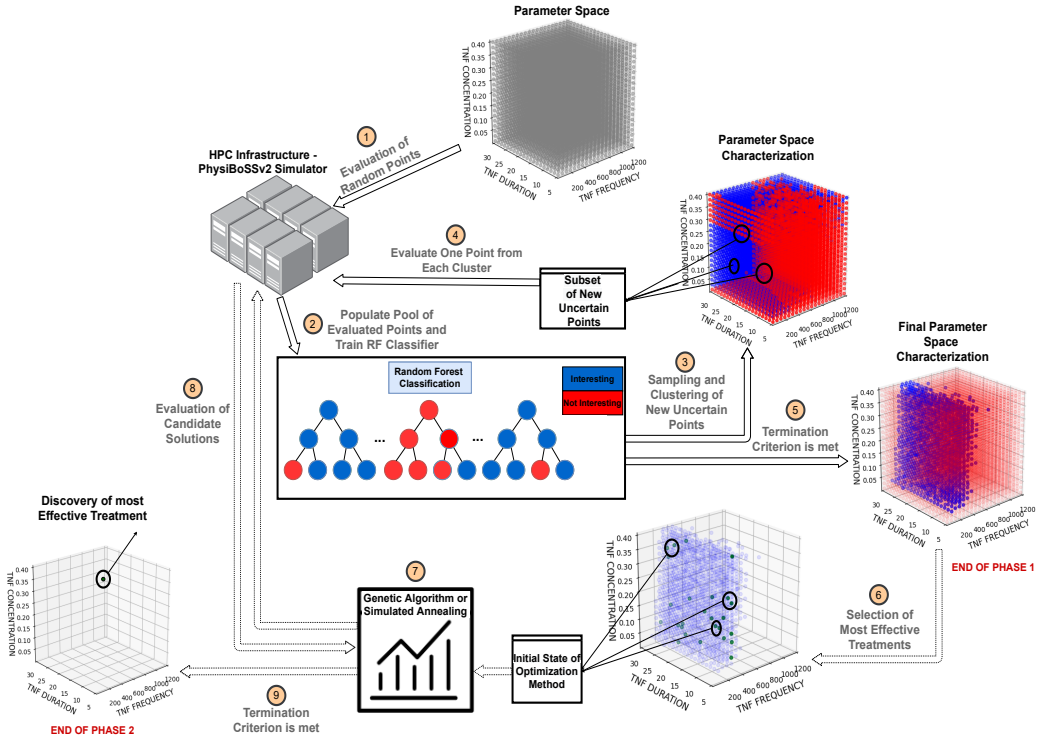


Figure 4.1: Overview of the examined workflow.

4.2.1 Characterization of the treatment parameter space

The aim of the first phase of our workflow is to identify the regions of the treatment parameter space which contain effective treatments. In the AL algorithm applied in the first phase, a Random Forest classifier is used to divide the parameter space into promising and non-promising areas. An area is considered to be promising if it includes treatments that reduce the count of the alive tumor cells below a set threshold. The final goal of the AL workflow is to obtain an understanding of the parameter space and its sub-regions, without exhaustively evaluating the effectiveness of each possible treatment configuration. Points are iteratively selected based on a two-step sampling process.

First, the most uncertain points are selected in order to exploit the results of previous iterations. These points compose the classification boundary and indicate the regions with the most uncertainty, thus regions with points of high informative value. Hence, if we understand the effectiveness of the treatments represented by this points in the treatment parameter space, we will reach a superior characterization of the parameter space. Figure 4.2 illustrate the uncertain points an example of the most uncertain points in an iteration of the AL algorithm.

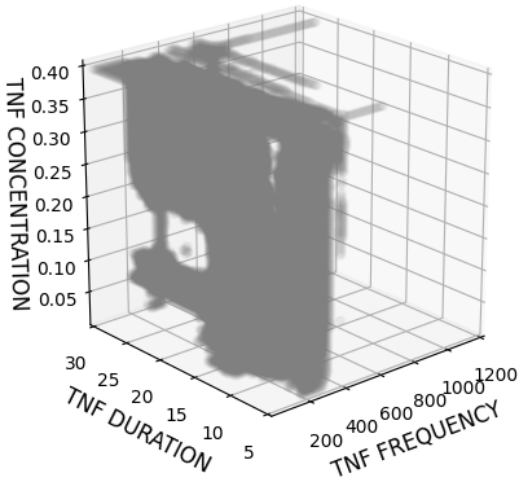


Figure 4.2: Points with highest uncertainty

The most uncertain points are, then, clustered in order to exploit their spatial distribution. The clustering of the selected points is the stepping stone for sampling only the most informative points and is depicted in Figure 4.3.

Points representing each cluster are selected, ensuring the diversity of the sample. Then, evaluation via simulations takes place and the results are

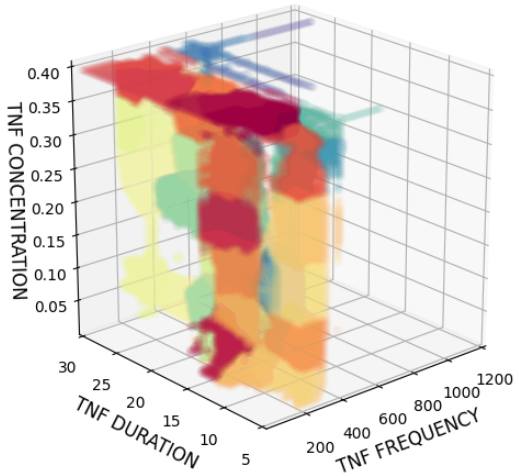


Figure 4.3: Clusters of the most uncertain points

included in the training set. We perform clustering on the candidate points in order to exploit their spatial distribution, determine the most influential points, and reduce the required number of simulations, by selecting only a few representatives. Figure 4.4 illustrates the points selected as representatives of the clusters of the uncertain points.

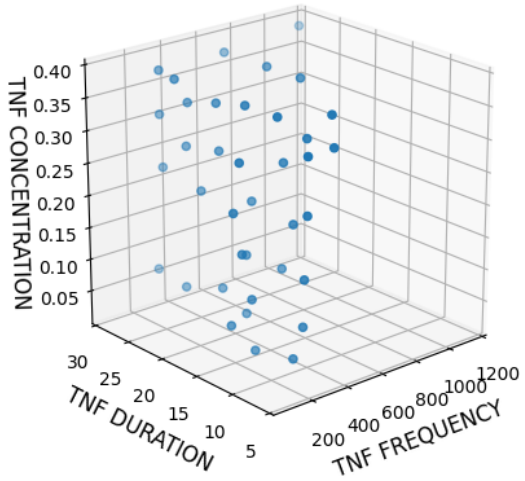


Figure 4.4: Selected representative points for simulation

It is evident by comparing the number of points over the uncertainty threshold and the representative points selected for simulation depicted in Figure 4.2 and Figure 4.4, respectively, that the required number of simu-

lations is reduced dramatically. After the simulations of the selected points are completed, the classifier is refitted, leading to a revised vision of the parameter space. This process is repeated for a set number of iterations. The sampling process of the AL controls the number of simulations required for a successful parameter space characterization. Since simulations are computationally expensive, the selection of the most informative instances is of crucial importance. In our approach, we examine the effect of three well-known clustering algorithms, namely the K-Means, BIRCH and DBSCAN algorithms, on the quality of the sampling process. Figure 4.5 depicts the separation of the treatment parameter space into areas containing promising and non promising treatments after the termination of the AL algorithm. The regions of the parameter space shaded with blue and red color are considered by the classifier to contain promising and non-promising treatments, respectively.

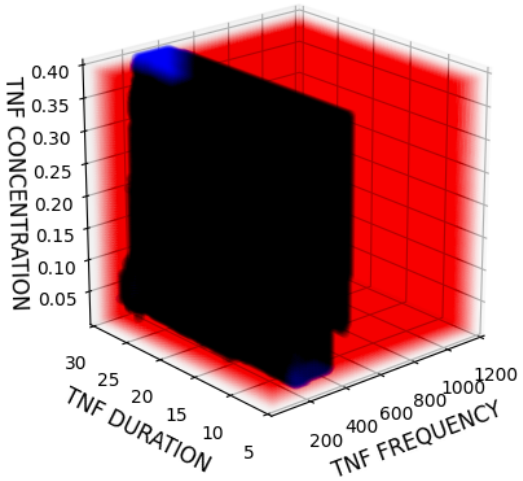


Figure 4.5: Characterization of the treatment parameter space using a random forest classifier.

4.2.2 Optimal treatment discovery

In the second phase of the workflow, an optimization technique is used in order to obtain the most promising treatments, i.e. the treatments that lead to the lowest final tumor cell counts. In particular, two optimization methods, namely Genetic Algorithm and Simulated Annealing, are examined.

Regarding the GA, a tournament selection is used in order to sample the best points from the current population. The sampled points are combined

according to a crossover probability, and then mutated according to a mutating probability, before the new population is evaluated. The fitness function we use for the evaluation of the treatments is the number of tumor cells after the application of the treatment. The process is repeated until a predefined number of iterations is reached. SA attempts to improve on the initial solution by iteratively examining candidate points in the neighborhood of the solution. The temperature of the algorithm is reduced based on a cooling schedule set by the user until a minimum temperature is reached. The energy function used by SA in order to evaluate each candidate solution is set as the final count of tumor cells, as in the case of the fitness function of GA.

Two scenarios regarding the initial state of the optimization methods are available. In the first scenario, the initial state is random, while in the second one, the initialization of the methods takes into consideration the results of the first part of the workflow (seeded scenario). In the seeded scenario, part of the population of the GA consists of treatments evaluated as the most effective by the AL algorithm. In the case of SA, the initial solution is set to be the treatment found to be the most effective by the first part of the workflow. The incorporation of the results of the AL part into the initial state of the optimization methods guarantees that the initial solutions consist of treatments considered as viable, and thus leads to a more focused search for the most promising treatments.

The aim of this part of the workflow is twofold. Firstly, it aims at the discovery of the most promising treatment. Secondly, it acts as a validation check for the results of the AL part. The treatment found to be most promising by the optimization part of the workflow should lie inside the regions found to contain promising treatments by the AL part of the workflow.

Chapter 5

Empirical Analysis

Our code implementation can be found in an online repository,[1] along with instructions on how to reproduce the experiments presented in this section. The scikit-learn Python library [2] was used for the clustering methods and the DEAP Python library was used for the Genetic Algorithm.[3] All experiments were performed using the Mare Nostrum 4 (MN4) HPC infrastructure provided by the Barcelona Supercomputing Centre.[4] First, we present the results of the first phase of our workflow. In particular, we examine the performance of different versions of the AL algorithm. The versions of the AL algorithm use different clustering algorithms for the sampling process of the AL part of the workflow. Furthermore, we evaluate the effectiveness of the optimal treatments discovered by the second phase of the workflow. To be exact, we compare the performance of the GA against the SA in the discovery of the most promising treatment configurations. The hyperparameter configuration chosen for each method in both parts of the workflow was obtained after performing smaller scale experimental runs and monitoring the performance of the different configurations.

5.1 Evaluation of treatment parameter space characterization

For each experiment conducted, the AL algorithm of the workflow was run for 20 iterations. We examine the performance of our various versions of the AL algorithm of the first phase of our workflow. The versions of the AL

[1] `https://github.com/xarakas/spheroid-tnf-v2-emews`
[2] `https://github.com/scikit-learn/scikit-learn`
[3] `https://github.com/DEAP/deap`
[4] `https://www.bsc.es/marenostrum/marenostrum`

algorithm under examination apply different clustering algorithms for the sampling process of the first phase. In the original workflow, K-Means clustering was applied in the step of the sampling process. To obtain a baseline, we examine the application of K-Means variants with k equal to 20 and 50 in the sampling step of the AL algorithm. We refer to the aforementioned configurations as KMEANS_20 and KMEANS_50, respectively. Moreover, we examine a version of the first part of the workflow in which a K-Means clustering with k equal to 500, referred to as KMEANS_500, was applied in the sampling of the uncertain points, in order to obtain a reference of an exhaustive application of KMEANS clustering in the sampling process of our workflow. For the evaluation purposes, the parameters of DBSCAN were configured with $Eps = 0.025$ and $MinPts = 20$ and the parameters of BIRCH with branching factor $B = 100$ and distance threshold $T = 0.1$. An exhaustive sweep search of the parameter space was conducted as a benchmark for comparing the performance of the first phase of the workflow. The sweep search iteratively evaluates a grid of individuals that are uniformly distributed in the search space. The grid of the sweep search is predetermined and contains a finite set of points, thus representing a discretised version of the treatment configuration parameter space. The sweep search consists a valuable benchmark in the evaluation of the performance of the different methods. However, the discretization of the parameter space and the trade-off between the required simulations and number of points evaluated (size of the uniform grid) leads to points lying between consecutive individuals in the uniform grid not being evaluated, and thus important areas therein might be ignored.

In order to characterize a treatment as viable or not, we need to define a metric that measures the effectiveness of each treatment. We define the tumor cell survival rate of each treatment configuration as follows:

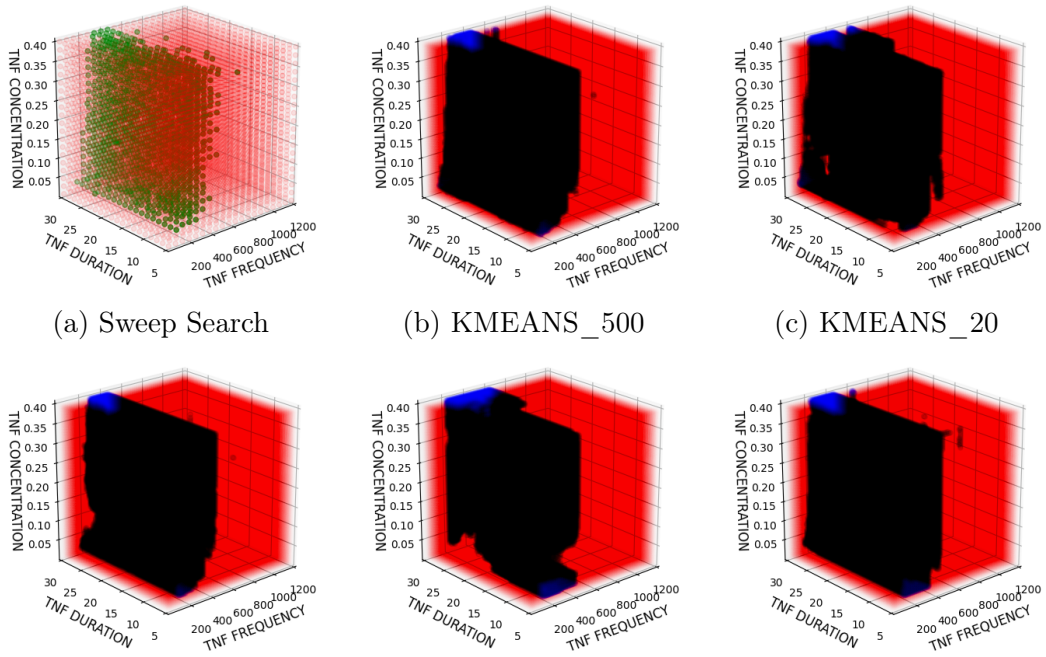
$$\text{Tumor Cell Survival Rate} = \frac{\text{Final Tumor Cell Count}}{\text{Initial Tumor Cell Count}}$$

i.e., as the ratio of the count of alive tumor cells after the application of the treatment to the count of the initial alive tumor cells, for a simulation duration of 24 hours. This metric reflects the number of the final alive cells as a percentage of the alive cells before the application of the treatment. We consider as viable the treatments that achieve a treatment effectiveness score of less than 0.3.

To measure the performance of our workflow, we compare the quality of the treatment space characterization achieved by the first phase of our workflow to the one achieved by the benchmark sweep search. In particular, the graphical representation of the characterization of the treatment parameter

space achieved acts a qualitative measure of the performance of our workflow. Moreover, we examine the number of uncertain points at the end of the AL algorithm and the total simulations performed until the execution terminates. Experiments using various random seeds were conducted for each examined version of the AL algorithm, allowing us to simulate the randomness of the process, while still being able to compare the methods in similar experimental conditions. For the initialization of the Random Forest classifier, 100 points were selected at random and evaluated. These points compose the initial training set of the classifier. In each experimental seed, the initial set of points is identical for each version of the workflow and their selection is independent of the clustering method used in the sampling process of the AL algorithm.

5.1.1 Characterization of treatment parameter space



(a) Sweep Search (b) KMEANS_500 (c) KMEANS_20

(d) KMEANS_50 (e) DBSCAN (f) BIRCH

Figure 5.1: Final characterization of the viable regions of the parameter space. Subfigure (a) depicts the results of the sweep search of the parameter space. Subfigures (b), (c), (d), (e), (f) illustrate the results of the first phase of the workflow using different clustering algorithms in the sampling process of the AL algorithm.

Figure 5.1 provides a visual representation of the parameter space characterization achieved by the first part of the workflow, using different clustering algorithms in the sampling process of the AL algorithm. The results presented in this figure were produced by conducting experiments initiated with the same random seed for each scenario examined. The figure also includes results from the uninformed sweep search, that predetermines a relatively sparse number of points, and evaluates them exhaustively. The areas shaded in blue and green represent regions that are considered to contain viable treatments by the classifier of our method and the benchmark sweep search, respectively, while the regions shaded in red are considered to contain non-viable treatment configurations. We observe that in all examined cases, our workflow succeeds in characterising the general region of the viable treatments and identifying a viable region comparable to the one identified by the benchmark sweep search. In particular, the versions of the workflow using BIRCH, KMEANS_20, KMEANS_50 and KMEANS_500 in the sampling process achieve a space characterization similar and comparable to the one obtained by the benchmark sweep search. In particular, the versions of the AL algorithm using the KMEANS_20 and KMEANS_50 clustering methods in the sampling step "understand" in a good manner the overall borders of the interesting region, but fail in the identification of viable treatments that are not part of the main interesting region. In the cases in which our workflow uses the BIRCH and KMEANS_500 algorithms, the first part of the workflow identifies an interesting region with finer details and smoother borders. The smoothness of the borders indicates the certainty of the classifier regarding the boundaries of the region. On the contrary, the AL algorithm using DBSCAN for the clustering of the uncertain points identifies a reduced version of the interesting region, failing to capture the details of the edges of the area that includes the viable treatments.

5.1.2 Number of uncertain Points

As already noted, the set of the most uncertain points in each iteration of the AL workflow defines the classification boundary of the classifier. Thus, the number of uncertain points serves as a valuable indicator for monitoring the performance of the first phase of the workflow, as it provides an estimate of the certainty of the classifier regarding the characterization of the treatment parameter space. Consider the set of all treatment configurations X, the classes $\{0, 1\}$ representing the non-viable and viable treatments, respectively and $Pr(i, x)$ the probability, assigned by the classifier, that a treatment configuration x belongs to class i. Then, the number of uncertain points is

computed based on the following rule:

$$N_U = \left|\left\{x \in X \mid \min(Pr(0,x), Pr(1,x)) \geq \text{threshold}\right\}\right|$$

i.e. the number of points whose uncertainty is above a predefined threshold. In our experiments, we set the uncertainty threshold to be equal to 0.4. The number of uncertain points is measured at the beginning of each iteration. In this way, we measure the effect of the evaluation of the points selected after the sampling process on the quality of the space characterization. The clustering method used in the second part of the sampling process aims at the selection of the most representative points in the set N_U and, thus, in the reduction of the required simulations, as mentioned in Section 4.2.1. These points are the most informative ones and their evaluation leads to a better classification of a wide region around them. A large number of uncertain points at the end of an experiment reveals a weaker performance of the AL algorithm, as it signals the existence of large ambiguous regions that cannot be classified with relative certainty as viable or non-viable.

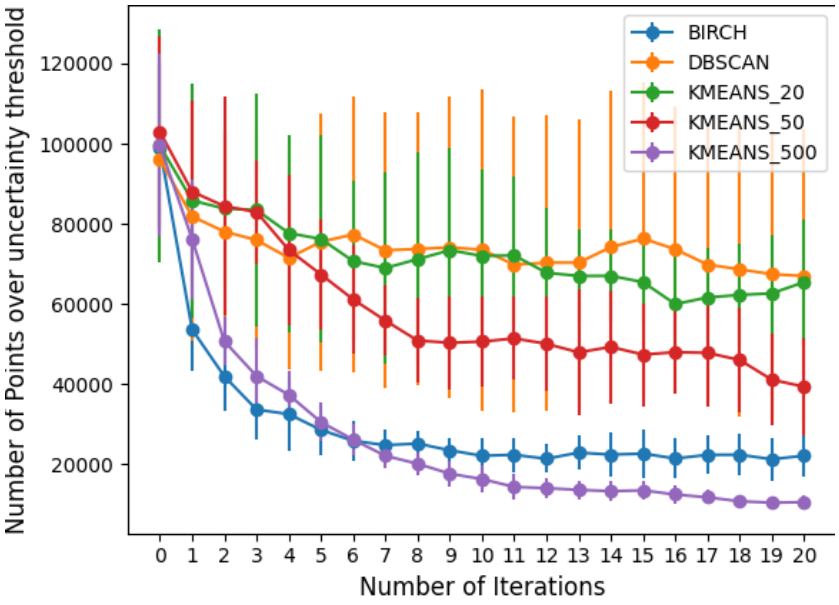
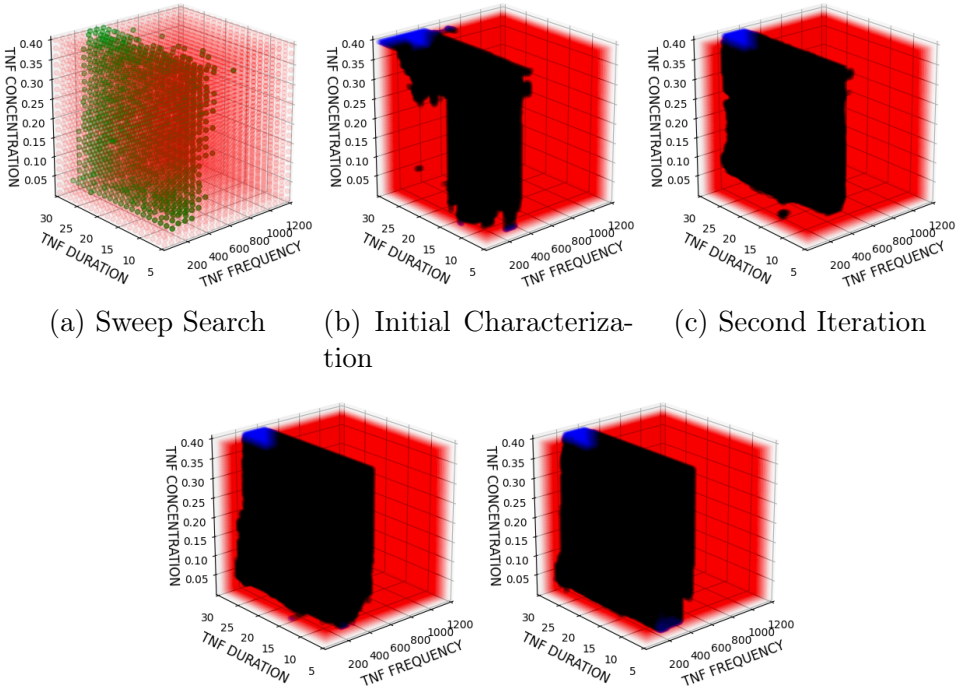


Figure 5.2: Number of uncertain points per iteration.

Figure 5.2 depicts the number of uncertain points per iteration of the AL algorithm. The results suggest that, the versions of the first phase of the workflow that applied BIRCH and KMEANS_500 in the sampling process achieve the lowest number of uncertain points in our experiments. Furthermore, we note that the performance of both versions is stable across

the different experimental runs without any large variations as iterations progress. Moreover, the versions using the DBSCAN and KMEANS_20 in the second stage of the sampling process converge to similar numbers of uncertain points within the experimental time frame, while the version using the KMEANS_50 method exhibits the "median" performance of the examined versions of the AL algorithm. It is worth noting that the number of uncertain points is reduced relatively more in the early iterations of the algorithm, while it stabilizes at the later iterations.



(a) Sweep Search (b) Initial Characterization (c) Second Iteration

(d) Tenth Iteration (e) Last Characterization

Figure 5.3: Characterization of the viable regions of the parameter space throughout the experimental run. Subfigure (a) depicts the results of the sweep search of the parameter space. Subfigure (b) demonstrates the initial parameter space characterization. Subfigures (c), (d), (e) illustrate the characterization in the first, second and tenth iteration of the algorithm, respectively. Subfigure (f) represents the final parameter space characterization.

Figure 5.3 illustrates an example of the characterization of the parameter space in various stages of the AL part using the BIRCH clustering algorithm in the sampling process and provides a general picture of the aforementioned

observation. The areas shaded in blue and green represent regions that are considered to contain viable treatment configurations by the classifier of our method and the benchmark sweep search, respectively. The regions of the treatment parameter space shaded in red are considered to contain non-viable treatments. As can be clearly seen from Figure 5.3b, the considered viable region at the initial stage of the algorithm is a reduced version of the area considered as promising by the sweep search benchmark. In particular, a great part of the area considered as viable by the benchmark is classified as non-viable. In the initial iteration, the classifier solidifies the area considered as viable, i.e. the uncertainty regarding various parts of the reduced viable region is decreased. In the second iteration, the classifier captures a good image of the viable region. In these stages, the classifier re-considers the limits of the viable region as illustrated in Figure 5.3c. Hence, the subsequent iterations allow the classifier to determine the details of the viable region, as seen in Figure 5.3d. In particular, the classifier has determined, in this stage, in great detail the boundaries of the viable region. Comparing the results of the tenth and the last iteration, we observe that only fine details regarding the shape of the area containing promising treatments are determined in the last iterations. These results agree with our conclusion regarding the number of uncertain points presented above, as the classifier determines the overall shape and position of the viable region in the initial stages of the algorithm and more refined details in the later stages. Thus, the uncertainty regarding the parameter space characterization is reduced drastically in the initial iterations, followed by a moderate reduction in the uncertainty points as the classifier refines its consideration regarding the viable region.

As can be clearly seen in Figure 5.2, the number of uncertain points is not reduced to zero in all examined scenarios. The existence of uncertain points is due to the stochasticity of the biological models used in order to simulate the behaviour of the human cells. Hence, the observed effectiveness of a treatment may vary slightly from simulation to simulation. This may lead to a point considered always as *uncertain*, if it represents a treatment configuration whose effectiveness is in the neighborhood of the viable treatment effectiveness threshold.

5.1.3 Number of total simulations

As we stressed earlier, PhysiBoSSv2 simulations used to estimate the effectiveness of tumor treatments are time-consuming and require multiple CPUs for their execution. The resource-effectiveness of the examined workflow is analyzed via the total number of simulations performed. This number can also be considered a good indicator of the time performance of our approach.

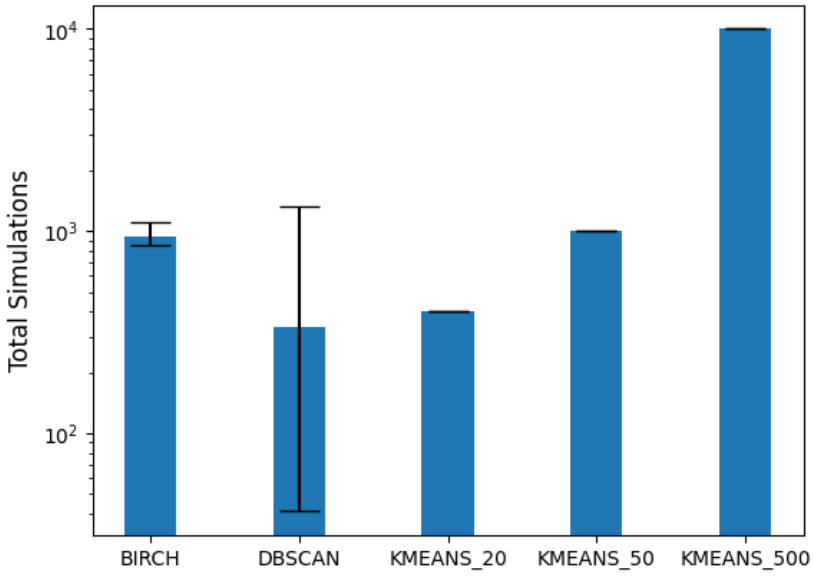


Figure 5.4: Total simulations performed by each version of the method (y-axis in log-scale).

Results are depicted in Figure 5.4. In our experiments, the initial training set of the Random Forest Classifier consisted of 100 random treatments and their effectiveness evaluations. We observe that although the version of the AL algorithm using BIRCH in the sampling process yields comparable results to the version using KMEANS_500 with respect to the quality of the treatment space characterization and the minimization of the number of uncertain points, it requires significantly less simulations. In particular, the latter version required $10,100$ simulations, while the former required only $1,082$ on average, yielding approximately a 90% decrease. It is worth noting that the version of our workflow using DBSCAN required the fewest simulations from the examined versions, however the number of simulations required is not stable enough across experiment repetitions. In particular, it required only 323 simulations, while the versions using KMEANS_50 and KMEANS_20 required $1,100$ and 500 runs, respectively.

Figure 5.5 presents the number of simulations performed in each iteration of the AL algorithm. In each iteration, a point from each resulting cluster of the uncertain points is selected, and simulations are conducted for all selected points. Hence, the number of simulations required in each iteration is equal to the number of clusters identified in the second stage of the sampling process of the AL algorithm.

Although the versions applying KMEANS_50 and BIRCH in the sam-

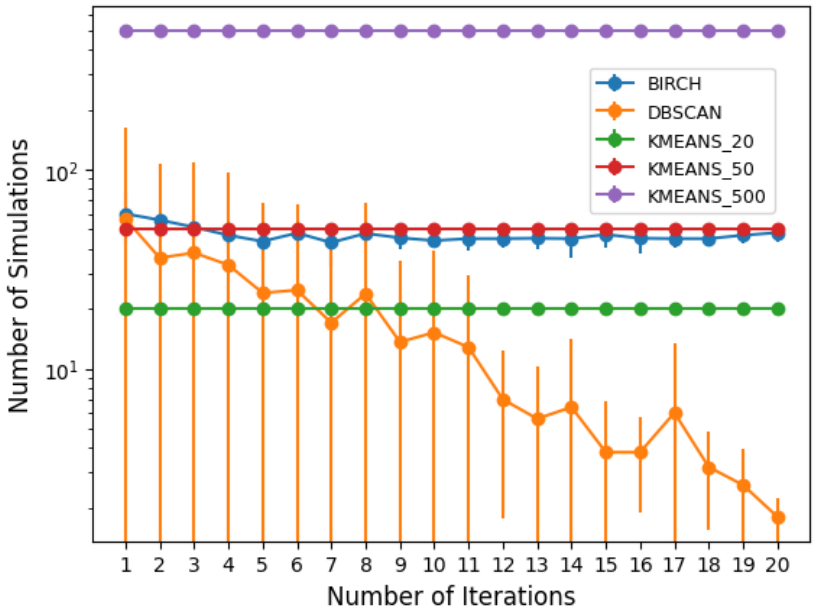
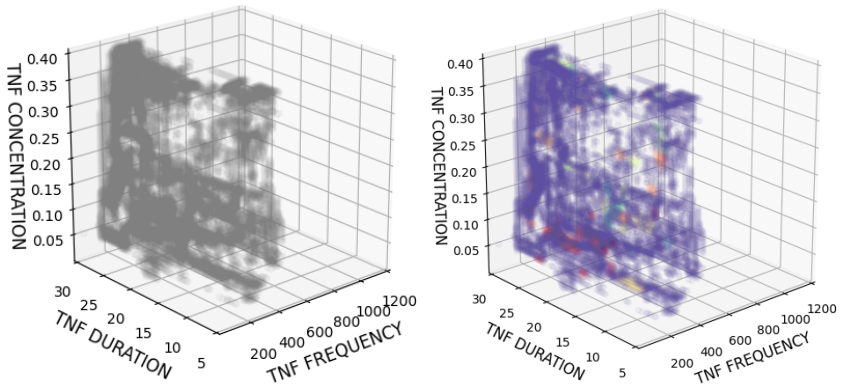


Figure 5.5: Simulations performed per iteration (y-axis in log-scale).

pling process identify a similar number of clusters per iteration as seen in Figure 5.5, their performance differs substantially as illustrated in subsections 5.1.1 and 5.1.2. An explanation to this is that the incorporation of outlying points into the clusters of the uncertain points degrades the results of the clustering process using KMEANS_50, leading to the identification of distorted clusters for the candidate points. On the contrary, the BIRCH clustering algorithm identifies outlying points and isolates them into smaller sub-clusters, thus managing to identify clusters that consist either of informative points that play an important role in the definition of the classification boundary, or of outlying points. The separation of the outlying points allow the retrieval of information from both types of points.

Also, we observe an increased volatility in the number of the clusters identified by DBSCAN, which reveals the high sensitivity of the method to its hyperparameters and the density of the distribution of the candidate points. In particular, we observe from both Figures 5.4, 5.5 that the version of the AL algorithm applying DBSCAN requires the fewest simulations in the majority of the conducted experiments. This is due to the distribution of the uncertain points. In most cases, the uncertain points lie on the border of the region considered as viable from the classifier at each stage of the algorithm.

As mentioned in Section 3.1.2 DBSCAN performs poorly in flat geometries such the one depicted in Figure 5.6, which illustrates an example of the



(a) Uncertain Points (b) Clusters

Figure 5.6: Example of uncertain points clustering by DBSCAN. Subfigure (a) depicts the set of uncertain points, while Subfigure (b) illustrates the result of the clustering.

clustering of the uncertain points during an iteration of the algorithm. As can been seen, the uncertain points define a flat area, which is not divided in regions of high and low densities. Thus, DBSCAN fails to identify multiple clusters and identifies only big clusters and some small ones. Although this may lead to fewer simulations, the informative value of the selected points is plummeted, as they are representatives of bigger clusters, with lower similarity between their members.

5.2 Evaluation of optimal treatment discovery

As presented in Section 4.2.2, our workflow uses a GA in order to discover the most promising treatment configuration. For the experimental purposes, the GA was configured to run for 30 generations, with a population of 50 individuals, a tournament selection with tournament size equal to 3, a uniform crossover with crossover probability 0.75 and a mutation probability 0.5. The alternative method examined, SA, was configured with T_o=100, T_{min}=15 and N=10. A geometrical cooling schedule with a cooling factor equal to $a = 0.8$ was applied, in which at each temperature level i the new temperature T_i is calculated as: $T_i = a^i \cdot T_o$. We examine the scenario in which both methods are initialized using information from the active learning part of the workflow. The initial population of the GA consists of 12 individuals (25% of the total population that the GA evaluates) found to be the most promising by the active learning, as well as of 38 random treatments (75% of the GA

population). The initial point of the SA is set to be the most promising one discovered by the AL. The incorporation of the findings of the first phase of our workflow to the initialization step of the optimization methods allows us to perform a more focused search for the optimal treatments. In our experiments, we examine the initialization of both methods using information from the first part of the workflow using the BIRCH, DBSCAN, KMEANS_20 and KMEANS_500 in the clustering step of the sampling process. In this way, we also examine the effect of the quality of the space characterization achieved by the first phase on the discovery of the optimal treatment.

Tables 5.1, 5.2 present the results of the Genetic Algorithm and Simulated Annealing, respectively, for the various examined clustering methods. We observe that in all cases the GA discovers a more effective treatment than the one discovered by SA. However, the effectiveness of the treatments discovered by both methods presents no big difference. Moreover, we observe that the treatments discovered by GA in the various scenarios yield a similar effectiveness. However, SA discovers a noticeable more effective treatment in the scenario in which the initial point is the treatment found to be most effective by the version of the AL algorithm using KMEANS_500 in the sampling process. Since SA focuses on the improvement of only one candidate solution, the initial state plays an important role in the performance of the method. The exhaustive approach of the AL algorithm using KMEANS_500 leads to a high number of simulated treatments, thus results to a bigger pool of candidate treatments for the initial solution. It is also worth noting that the most effective treatment is discovered by the GA in the scenario in which the initial population is generated based on the results of the first phase in which the DBSCAN algorithm was used for the clustering of the uncertain points. Although this version of the first phase of our workflow results to the identification of a reduced version of the viable region, as presented in the previous section, the selection of the treatments found to be the most effective by the AL workflow using the DBSCAN clustering method leads to the discovery of the most effective treatment by the GA. Moreover, the optimization part in this scenario yields results of similar (better) quality to the other versions of the first phase of our method that led to a more fined detailed characterization. Thus, we can conclude that the core part of the viable region discovered by the version of the AL part using DBSCAN contains some of the most effective treatments, which are a good starting point for the optimization part.

Although GA yields better results, the treatment discovery by GA required noticeably more resources than SA, as presented in Tables 5.1 and 5.2. In particular, the second phase of the workflow using SA for the discovery of the optimal treatment required noticeably less simulations in all

scenarios compared to the version applying GA. The maintenance of only one candidate solution makes SA less demanding in regards to the number of required simulations, in comparison to the examination of a pool of 50 candidate treatments in each generation by the GA.

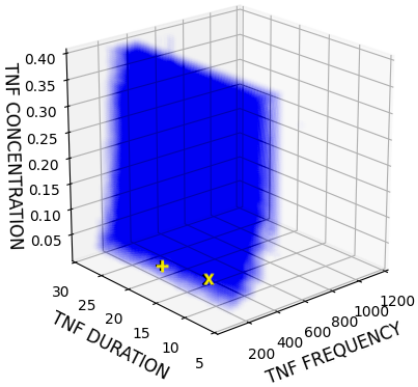
Table 5.1: Optimized drug treatment configuration exploration results for the Genetic Algorithm.

Method	Tumor Cell Survival Rate	Number of Simulations for the optimization phase	Total number of Simulations of the optimal treatment discovery
BIRCH	0.166	811	1853
DBSCAN	0.164	810	1007
KMEANS_20	0.166	742	1242
KMEANS_50	0.174	814	1914
KMEANS_500	0.169	661	10761

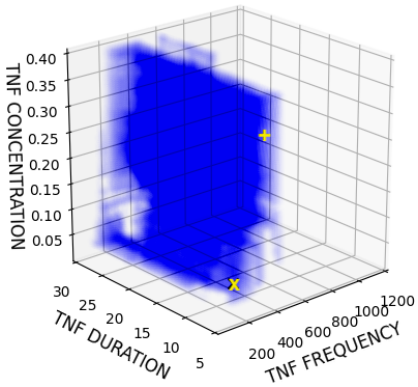
Table 5.2: Optimized drug treatment configuration exploration results for the Simulated Annealing method.

Method	Tumor Cell Survival Rate	Number of Simulations for the optimization phase	Total number of Simulations of the optimal treatment discovery
BIRCH	0.196	91	1133
DBSCAN	0.209	91	288
KMEANS_20	0.209	91	591
KMEANS_50	0.183	91	1191
KMEANS_500	0.177	91	10191

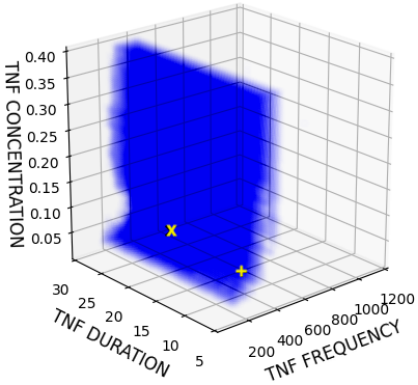
As stressed in Section 4.2, the optimization part of the workflow may also act as a validation tool for the characterization of the treatment parameter space. As seen in Figure 5.7, the treatments found to be the most effective by both optimization methods lie either on the borders or inside the regions considered to contain viable treatments by the Random Forest classifier in each examined scenario. These results, allow us to validate in another way the results of the first phase of our workflow. Hence, we observe, as already stated in Section 5.1, that all versions of the AL algorithm of the first phase of our workflow lead to good results regarding the overall boundaries of the viable regions of the treatment parameter space.



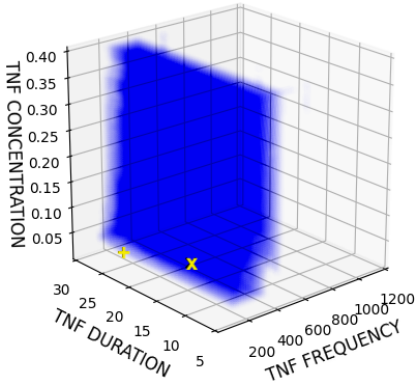
(a) KMEANS_500



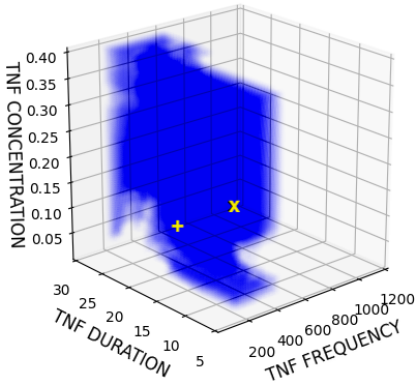
(b) KMEANS_20



(c) KMEANS_50



(d) BIRCH



(e) DBSCAN

Figure 5.7: Visual representation of the results of the optimization methods for the various examined clustering methods. Treatments marked with 'x' and '+' represent the optimal treatments found by GA and SA, respectively. The area shaded blue represents the area characterized as interesting by the version of the first phase of the workflow under examination.

Chapter 6

Summary & Future Directions

6.1 Summary

Multi-scale simulations provide a valuable tool to researchers for various fields and applications, including the discovery of promising tumor treatments. We applied a multi-scale simulator in order to characterize the treatment configuration space and identify the most effective treatments using an active learning approach. In particular, we extended the workflow presented by Ozik et.al [3], allowing users to select between various combinations of clustering and optimization algorithms for their experiments, and applied the workflow in a different experimental setting. In our approach, we incorporated a new simulation model imitating the application of the protein called Tumor Necrosis Factor (TNF) in order to estimate the effect of tumor treatments. Moreover, we examined the application of various well-known clustering algorithms in the sampling process of the parameter space characterization algorithm, as well as the application of the Genetic Algorithm and Simulated Annealing optimization methods in the discovery of promising drug treatment configurations aiming at the achievement of high quality results and the reduction of required simulations. Simulation trials conducted in an HPC environment show that our workflow achieves a fine parameter space characterization comparable to the one performed by the benchmark sweep search. Moreover, the versions of the first phase of the workflow using BIRCH and KMEANS_500 in the sampling process lead to the least uncertain points and identify the region of viable treatments with the highest certainty and detail. On the other hand, the version of the AL algorithm applying KMEANS_20 for the clustering of the uncertain points requires fewer simulations and achieves a fine characterization of the parameter space with less detail and higher uncertainty. Moreover, the application of the GA leads

to the discovery of slightly more effective treatments than SA. However, GA requires substantially more resources and simulations than SA. Thus, the experimental results indicate that a trade-off between the required resources and the quality of the results is evident in both parts of the workflow.

6.2 Future Directions

In future work, the examination of additional classification algorithms in the active learning workflow, such as the Gradient Boosting Trees [67, 68] might lead to a more detailed characterization of the treatment parameter space, as compared to the characterization achieved using the Random Forest classifier. Moreover, future studies could investigate the performance of different optimization methods, such as Bayesian Optimization [69, 70] or Particle Swarm Optimization [71, 72]. Furthermore, the workflow can also be extended to explore the synergistic effects of multiple drugs administration [14, 73] in even more complex simulations. Lastly, forecasting techniques can be applied for the early termination of simulation instances that cannot be used to extract useful information.

Bibliography

[1] A. Ambesi-Impiombato and D. Bernardo. Computational biology and drug discovery: From single-target to network drugs. *Current Bioinformatics*, 1(1):3–13, 2006.

[2] J. I. Griffiths, A. L. Cohen, V. Jones, R. Salgia, J. T. Chang, and A. H. Bild. Opportunities for improving cancer treatment using systems biology. *Current Opinion in Systems Biology*, 17:41–50, 2019.

[3] J. Ozik, N. Collier, R. Heiland, G. An, and P. Macklin. Learning-accelerated discovery of immune-tumour interactions. *Mol. Syst. Des. Eng.*, 4:747–760, 2019.

[4] B. Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

[5] W.-M. Chu. Tumor necrosis factor. *Cancer Letters*, 328(2):222–225, 2013.

[6] A. Ghaffarizadeh, R. Heiland, S. H. Friedman, S. M. Mumenthaler, and P. Macklin. Physicell: An open source physics-based cell simulator for 3-d multicellular systems. *PLOS Comp. Biology*, 14(2):1–31, 2018.

[7] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3 edition, 2010.

[8] A. K. Mitra, V. Agrahari, A. Mandal, K. Cholkar, C. Natarajan, S. Shah, M. Joseph, H. M. Trinh, R. Vaishya, X. Yang, Y. Hao, V. Khurana, and D. Pal. Novel delivery approaches for cancer therapeutics. *Journal of Controlled Release*, 219:248–268, 2015.

[9] James H Doroshow and Shivaani Kummar. Role of phase 0 trials in drug development. *Future Medicinal Chemistry*, 1(8):1375–1380, November 2009.

[10] Rosa M. Abrantes-Metz, Christopher Adams, and Albert D. Metz. Pharmaceutical development phases: A duration analysis. *SSRN Electronic Journal*, 2004.

[11] Lawrence Friedman. *Fundamentals of clinical trials.* Springer, New York, 2010.

[12] M. Dickson and J. P. Gagnon. The cost of new drug discovery and development. *Discov Med*, 4(22):172–179, Jun 2004.

[13] Clémence Réda, Emilie Kaufmann, and Andrée Delahaye-Duriez. Machine learning applications in drug development. *Computational and Structural Biotechnology Journal*, 18:241–252, 2020.

[14] R. B. Mokhtari, T. S. Homayouni, N. Baluch, E. Morgatskaya, S. Kumar, B. Das, and H. Yeger. Combination therapy in combating cancer. *Oncotarget*, 8(23):38022–38043, 2017.

[15] D. Dréau, Dimitre Stanimirov, Ted Carmichael, and M. Hadzikadic. An agent-based model of solid tumor progression. In *BICoB*, 2009.

[16] Padmanee Sharma and James P. Allison. The future of immune checkpoint therapy. *Science*, 348(6230):56–61, April 2015.

[17] Chang Gong, Oleg Milberg, Bing Wang, Paolo Vicini, Rajesh Narwal, Lorin Roskos, and Aleksander S. Popel. A computational multiscale agent-based model for simulating spatio-temporal tumour immune response to PD1 and PDL1 inhibition. *Journal of The Royal Society Interface*, 14(134):20170320, September 2017.

[18] Katarzyna Rejniak, Veronica Estrella, Tiangan Chen, Allison Cohen, Mark Lloyd, and David Morse. The role of tumor tissue architecture in treatment penetration and efficacy: An integrative study. *Frontiers in Oncology*, 3:111, 2013.

[19] P. Baldi, S. Brunak, and F. Bach. *Bioinformatics: the machine learning approach.* MIT press, 2001.

[20] R. Dias and A. Torkamani. Artificial intelligence in clinical and genomic diagnostics. *Genome Medicine*, 11(1), 2019.

[21] J. Błażewicz, P. Łukasiak, and M. Miłostan. Application of tabu search strategy for finding low energy structure of protein. *Artificial Intelligence in Medicine*, 35(1):135–145, 2005.

[22] N. Kutsuna, T. Higaki, S. Matsunaga, T. Otsuki, M. Yamaguchi, H. Fujii, and S. Hasezawa. Active learning framework with iterative clustering for bioimage classification. *Nature communications*, 3(1):1–10, 2012.

[23] R. Vasundhara Devi, S. Siva Sathya, and Mohane Selvaraj Coumar. Evolutionary algorithms for de novo drug design – a survey. *Applied Soft Computing*, 27:543–552, 2015.

[24] S.K. Jain and A. Agrawal. De novo drug design: An overview. *Indian Journal of Pharmaceutical Sciences*, 66:721–728, 01 2004.

[25] Renxiao Wang, Ying Gao, and Luhua Lai. Ligbuilder: A multi-purpose program for structure-based drug design. *Journal of Molecular Modeling*, 6:498–516, 01 2000.

[26] D. Reker and G. Schneider. Active-learning strategies in computer-assisted drug discovery. *Drug discovery today*, 20(4):458–465, 2015.

[27] Manfred Warmuth, Jun Liao, Gunnar Rätsch, Michael Mathieson, Santosh Putta, and Christian Lemmen. Support vector machines for active learning in the drug discovery process. 02 2003.

[28] Karunakaran Kaliyappan, Murugesan Palanisamy, Rajeshwar Govindarajan, and Jeyapradha Duraiyan. Microarray and its applications. *Journal of Pharmacy and Bioallied Sciences*, 4(6):310, 2012.

[29] Giuseppe Russo, Charles Zegar, and Antonio Giordano. Advantages and limitations of microarray technology in human cancer. *Oncogene*, 22(42):6497–6507, September 2003.

[30] Ying Liu. Active learning with support vector machine applied to gene expression data for cancer classification. *Journal of Chemical Information and Computer Sciences*, 44(6):1936–1941, November 2004.

[31] Ramón Díaz-Uriarte and Sara Alvarez de Andrés. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7(1):3, 2006.

[32] T. P. Mohamed, J. G. Carbonell, and M. K. Ganapathiraju. Active learning for human protein-protein interaction prediction. *BMC bioinformatics*, 11(1):1–9, 2010.

[33] Diego León, Arbey Aragón, Javier Sandoval, Germán Hernández, Andrés Arévalo, and Jaime Niño. Clustering algorithms for risk-adjusted

portfolio construction. *Procedia Computer Science*, 108:1334–1343, 2017.

[34] Venkata Srikanth Reddy, Patrick Kinnicutt, and Roger Lee. Text document clustering: The application of cluster analysis to textual document. In *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 1174–1179, 2016.

[35] D. Xu and Y. Tian. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193, 2015.

[36] C. C. Aggarwal. *Data Mining - The Textbook*. Springer, 2015.

[37] T. Velmurugan and T. Santhanam. A survey of partition based clustering algorithms in data mining: An experimental approach. *Information Technology Journal*, 10(3):478–484, February 2011.

[38] F. Murtagh. A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, 26(4):354–359, November 1983.

[39] Hans-Peter Kriegel, Peer Kröger, Jörg Sander, and Arthur Zimek. Density-based clustering. *WIREs Data Mining and Knowledge Discovery*, 1(3):231–240, April 2011.

[40] S. P. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inf. Theory*, 28(2):129–136, 1982.

[41] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231. AAAI Press, 1996.

[42] T. Zhang, R. Ramakrishnan, and M. Livny. Birch: An efficient data clustering method for very large databases. *SIGMOD Rec.*, 25(2):103–114, 1996.

[43] Krzysztof Drachal and Michał Pawłowski. A review of the applications of genetic algorithms to forecasting prices of commodities. *Economies*, 9(1), 2021.

[44] M. Niu, C. Wan, and Z. Xu. A review on applications of heuristic optimization algorithms for optimal power flow in modern power systems. *J. of Modern Power Sys. and Clean Energy*, 2(4):289–297, 2014.

[45] Dilip Datta, André R.S. Amaral, and José Rui Figueira. Single row facility layout problem using a permutation-based genetic algorithm. *European Journal of Operational Research*, 213(2):388–394, September 2011.

[46] C.K.H. Lee. A review of applications of genetic algorithms in operations management. *Engineering Applications of Artificial Intelligence*, 76:1–12, November 2018.

[47] Eun Yi Kim and Se Hyun Park. Automatic video segmentation using genetic algorithms. *Pattern Recognition Letters*, 27(11):1252–1265, August 2006.

[48] Melanie Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA, 1996.

[49] Darrell Whitley. A genetic algorithm tutorial. *Statistics and Computing*, 4(2), June 1994.

[50] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[51] V. Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, 1985.

[52] Y. Crama and M. Schyns. Simulated annealing for complex portfolio selection problems. *European Journal of Operational Research*, 150(3):546–571, November 2003.

[53] D. F. Wong. *Simulated annealing for VLSI design*. Kluwer Academic, Boston, 1988.

[54] Omer Sagi and Lior Rokach. Ensemble learning: A survey. *WIREs Data Mining and Knowledge Discovery*, 8(4), February 2018.

[55] Cha Zhang and Yunqian Ma, editors. *Ensemble Machine Learning*. Springer US, 2012.

[56] Leo Breiman. *Machine Learning*, 45(1):5–32, 2001.

[57] Leo Breiman, Jerome Friedman, Charles J. Stone, and R.A. Olshen. *Classification and regression trees*. Chapman & Hall, New York, 1993.

[58] Alan J. Izenman. *Modern Multivariate Statistical Techniques*. Springer New York, 2008.

[59] Adele Cutler, D. Richard Cutler, and John R. Stevens. Random forests. In *Ensemble Machine Learning*, pages 157–175. Springer US, 2012.

[60] Omer Sagi and Lior Rokach. Ensemble learning: A survey. *WIREs Data Mining and Knowledge Discovery*, 8(4):e1249, 2018.

[61] V.F. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J.P. Rigol-Sanchez. An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 67:93–104, January 2012.

[62] Ahmad Taher Azar, Hanaa Ismail Elshazly, Aboul Ella Hassanien, and Abeer Mohamed Elkorany. A random forest classifier for lymph diseases. *Computer Methods and Programs in Biomedicine*, 113(2):465–473, February 2014.

[63] D. Amaratunga, J. Cabrera, and Y.-S. Lee. Enriched random forests. *Bioinformatics*, 24(18):2010–2014, July 2008.

[64] Hemant Ishwaran, Udaya B. Kogalur, Eugene H. Blackstone, and Michael S. Lauer. Random survival forests. *The Annals of Applied Statistics*, 2(3), September 2008.

[65] Ahmadreza Ghaffarizadeh, Samuel H. Friedman, and Paul Macklin. BioFVM: an efficient, parallelized diffusive transport solver for 3-d biological simulations. *Bioinformatics*, 32(8):1256–1258, December 2015.

[66] Charilaos Akasiadis, Miguel Ponce de Leon, Arnau Montagud, Evangelos Michelioudakis, Alexia Atsidakou, Elias Alevizos, Alexander Artikis, Alfonso Valencia, and Georgios Paliouras. Parallel model exploration for tumor treatment simulations, 2021.

[67] R. E. Schapire. *The Boosting Approach to Machine Learning: An Overview*, pages 149–171. Springer, 2003.

[68] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189 – 1232, 2001.

[69] Jonas Mockus. *Bayesian Approach to Global Optimization*. Springer Netherlands, 1989.

[70] Peter I. Frazier. A tutorial on bayesian optimization, 2018.

[71] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4, 1995.

[72] Y. Shi and R. Eberhart. A modified particle swarm optimizer. In *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, pages 69–73, 1998.

[73] Rupal Ramakrishnan and Dmitry I. Gabrilovich. Novel mechanism of synergistic effects of conventional chemotherapy and immune therapy of cancer. *Cancer Immunology, Immunotherapy*, 62(3):405–410, February 2013.

Appendix A

Experimental results of the parameter space characterization

In this section we present the results of the various experiments conducted using different random seeds. The initial training dataset of the Random Forest classifier for each experimental seed was the same for each examined clustering method.

A.1 Experimental Seed 1

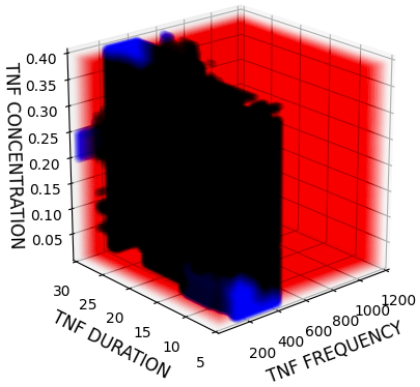
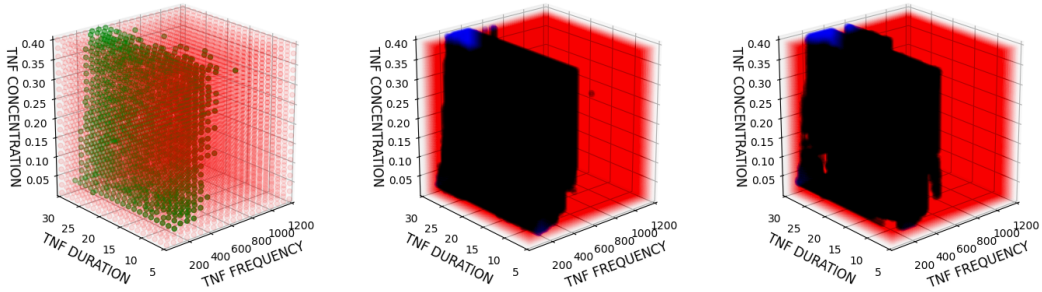
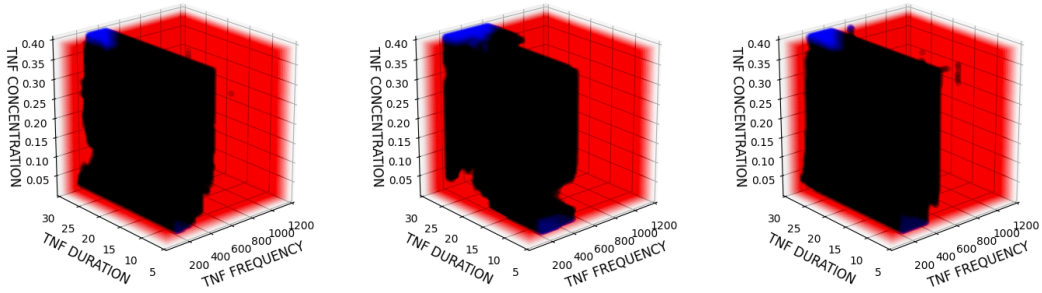


Figure A.1: Initial characterization of treatment parameter space (Experimental Seed 1).



(a) Sweep Search (b) KMEANS_500 (c) KMEANS_20



(d) KMEANS_50 (e) DBSCAN (f) BIRCH

Figure A.2: Final characterization of the viable regions of the parameter space (Experimental Seed 1)

A.2 Experimental Seed 2

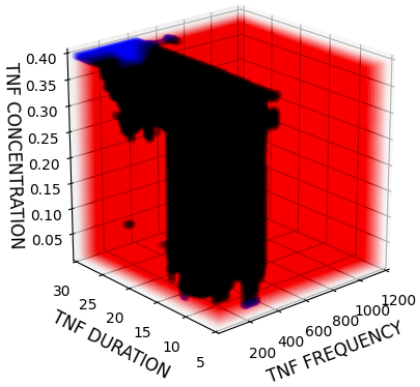
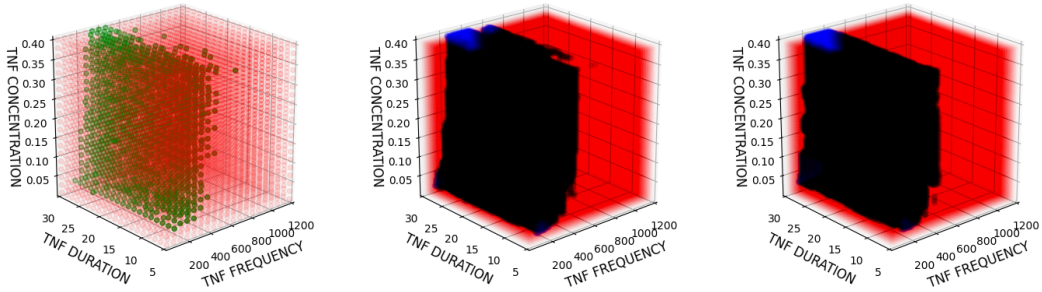
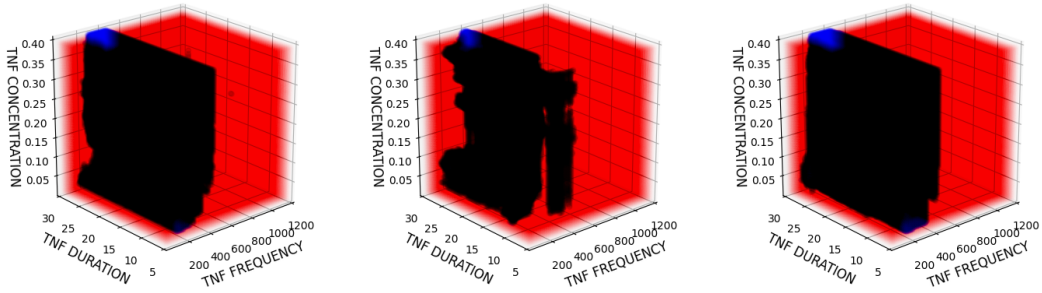


Figure A.3: Initial characterization of treatment parameter space (Experimental Seed 2).



(a) Sweep Search (b) KMEANS_500 (c) KMEANS_20



(d) KMEANS_50 (e) DBSCAN (f) BIRCH

Figure A.4: Final characterization of the viable regions of the parameter space (Experimental Seed 2)

A.3 Experimental Seed 3

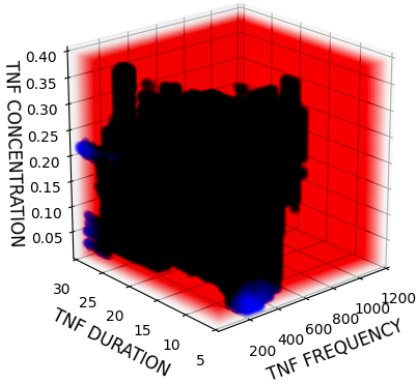
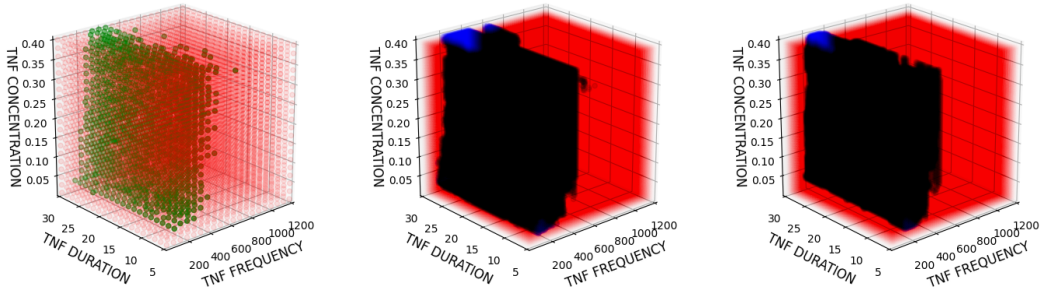
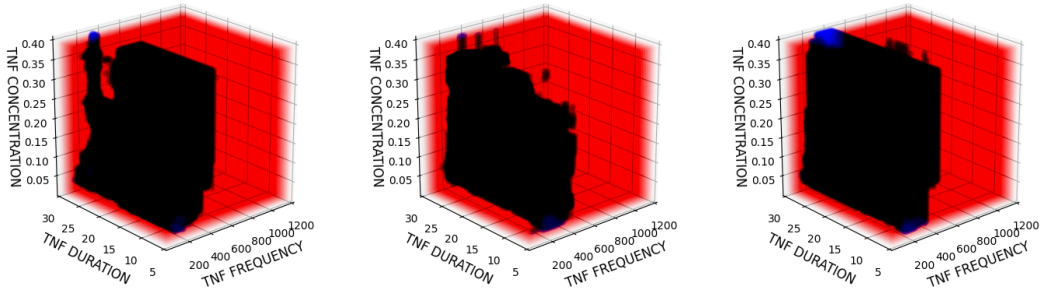


Figure A.5: Initial characterization of treatment parameter space (Experimental Seed 3).



(a) Sweep Search (b) KMEANS_500 (c) KMEANS_20



(d) KMEANS_50 (e) DBSCAN (f) BIRCH

Figure A.6: Final characterization of the viable regions of the parameter space (Experimental Seed 3)

A.4 Experimental Seed 4

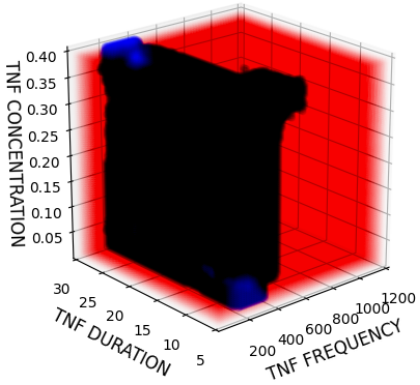
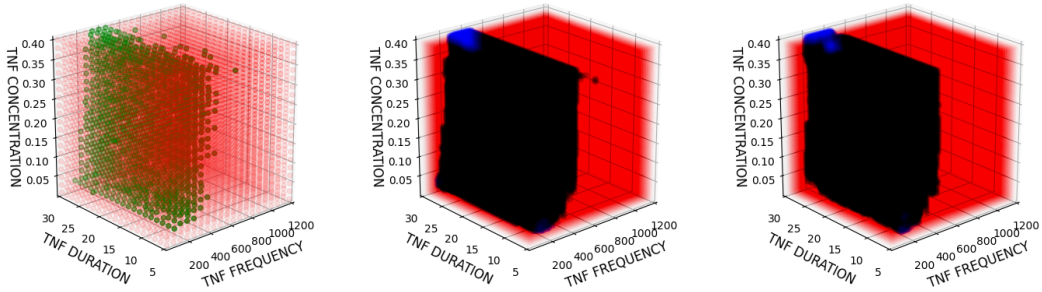
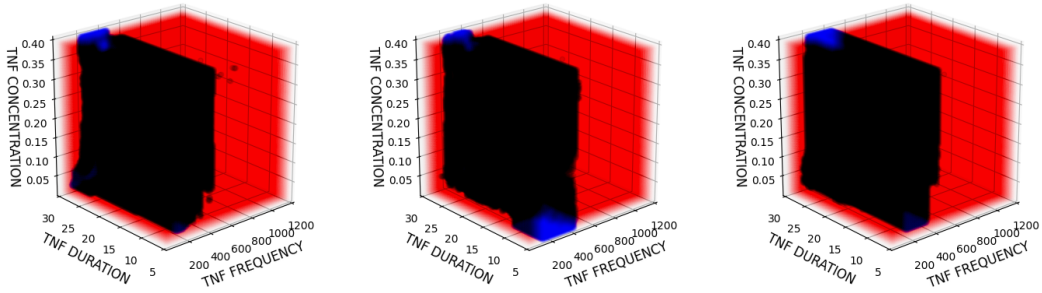


Figure A.7: Initial characterization of treatment parameter space (Experimental Seed 4).



(a) Sweep Search (b) KMEANS_500 (c) KMEANS_20



(d) KMEANS_50 (e) DBSCAN (f) BIRCH

Figure A.8: Final characterization of the viable regions of the parameter space (Experimental Seed 4)

A.5 Experimental Seed 5

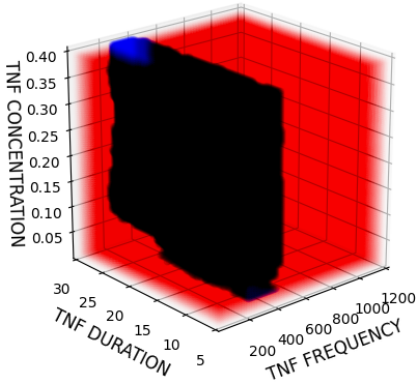
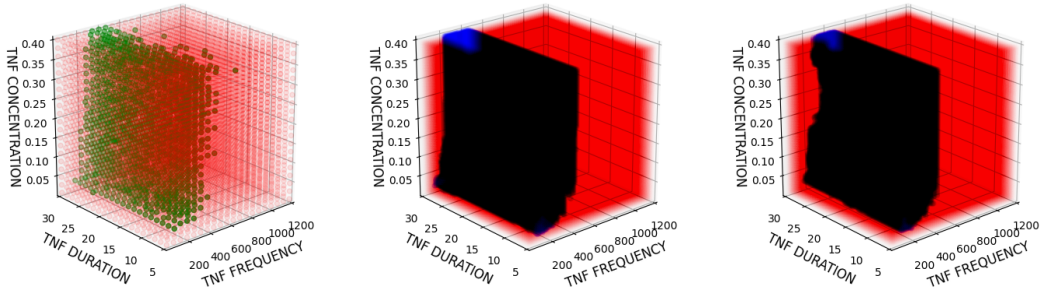
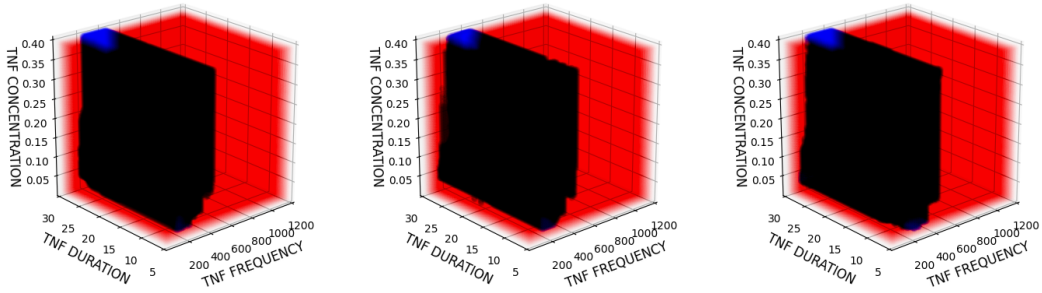


Figure A.9: Initial characterization of treatment parameter space (Experimental Seed 5).



(a) Sweep Search (b) KMEANS_500 (c) KMEANS_20



(d) KMEANS_50 (e) DBSCAN (f) BIRCH

Figure A.10: Final characterization of the viable regions of the parameter space (Experimental Seed 5)